Excel 2013

Formeln & Funktionen

Für Excel 97 bis 2013

- Vollständige Referenz aller Formeln und Funktionen
- Schnell die richtige Lösung finden
- Verständliche Anleitungen und praxisnahe Beispiele

Helmut Vonhoegen

Vierfarben

Excel 2013

Formeln und Funktionen

von
Helmut Vonhoegen

Vierfarben

Liebe Leserin, lieber Leser,

Microsoft Excel hat sich seit seinem Erscheinen im Jahr 1985 zu einem mächtigen und umfangreichen Werkzeug entwickelt. Die Möglichkeiten des Programms sind enorm. Auf fast alle Fragestellungen bzw. Rechenanwendungen gibt es mit Excel eine Antwort bzw. Lösung – allerdings ist für den Nutzer nicht immer gleich ersichtlich, welche Funktion bei der Lösungssuche eingesetzt werden muss.

Es ist aber auch gar nicht notwendig, alle Excel-Funktionen zu kennen. Wenn Sie auf der Suche nach einer Lösung sind, hilft Ihnen dieses Buch weiter. In diesem umfassenden Nachschlagewerk hat unser Autor Helmut Vonhoegen alle Funktionen übersichtlich und ausführlich für Sie zusammengestellt. Jede Funktion wir anhand eines Praxisbeispiels erklärt, das leicht auf die eigene Fragestellung übertragbar ist. Zahlreiche Abbildungen, Hinweise, Tipps und Fehlerwarnungen helfen Ihnen dabei, schnell die richtige Lösung für Ihre Aufgabenstellung zu finden.

Zu diesem Zweck sind die Funktionen in diesem Buch analog zu Excel in Kategorien eingeteilt. Innerhalb der Kategorien sind die einzelnen Funktionen alphabetisch geordnet. Wenn Sie bereits wissen, welche Funktion Sie benötigen werden, schlagen Sie am besten in der Funktionenübersicht im Anhang nach. Dort finden Sie außerdem eine alphabetische Liste der Funktionen mitsamt kurzer Beschreibung sowie eine Funktionenliste Deutsch–Englisch/Englisch–Deutsch.

Dieses Buch ist auf dem aktuellsten Stand, sowohl, was die beinhalteten Funktionen betrifft, als auch die Beschreibung derselben. Nichtsdestoweniger werden Sie auch mit älteren Excel-Versionen von diesem Buch profitieren, denn alle Abweichungen und Veränderungen seit Excel 97 sind selbstverständlich vermerkt und erklärt.

Dieses Buch wurde mit größter Sorgfalt geschrieben und hergestellt. Sollten Sie dennoch einmal einen Fehler finden oder inhaltliche Anregungen haben, freue ich mich, wenn Sie mit mir in Kontakt treten. Für konstruktive Kritik bin ich dabei ebenso offen wie für lobende Worte. Doch zunächst einmal wünsche ich Ihnen viel Freude beim Lesen!

Ihr Vaclav Demling
Lektorat Vierfarben

vaclav.demling@vierfarben.de

Auf einen Blick

1	Einstieg in Berechnungen mit Excel	33
2	Berechnungstools	147
3	Finanzmathematische Funktionen	169
4	Datums- und Zeitfunktionen	255
5	Mathematische und trigonometrische Funktionen	301
6	Technische Funktionen	393
7	Statistische Funktionen	443
8	Kompatible statistische Funktionen	615
9	Nachschlage- und Verweisfunktionen	663
10	Datenbankfunktionen	699
11	Cube-Funktionen	717
12	Textfunktionen	745
13	Logische Funktionen	781
14	Informationsfunktionen	799
15	Webfunktionen	821
16	Analyse mit Pivot-Tabellen und -Diagrammen	829
17	Zusätzliche Tools für die Datenanalyse	843
18	Entwicklung eigener Funktionen	871
A	Alphabetische Liste der Tabellenfunktionen	893
B	Funktionenliste Deutsch–Englisch/Englisch–Deutsch	923
C	Funktionen im Überblick	955
D	Funktionstasten, Tasten und Tastenkombinationen	963

Sie haben Fragen, Wünsche oder Anregungen zum Buch?
Gerne sind wir für Sie da:

Anmerkungen zum Inhalt des Buches: jan.watermann@vierfarben.de
Bestellungen und Reklamationen: service@vierfarben.de
Rezensions- und Schulungsexemplare: thomas.losch@vierfarben.de

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischem oder anderen Wegen und der Speicherung in elektronischen Medien.

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

An diesem Buch haben viele mitgewirkt, insbesondere:

Lektorat Jan Watermann, Vaclav Demling
Korrektorat Marita Böhm
Herstellung Norbert Englert
Einbandgestaltung Daniel Kratzke
Coverentwurf Daniel Kratzke
Typographie und Layout Vera Brauner
Satz III-satz, Husby
Druck und Bindung Beltz, Bad Langensalza

Gesetzt wurde dieses Buch aus der ITC Charter (10,25 pt/14 pt) in Adobe FrameMaker.
Und gedruckt wurde es auf chlorfrei gebleichtem Offsetpapier (90 g/m^2).
Hergestellt in Deutschland.

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN 978-3-8421-0114-2

1. Auflage 2014
© Vierfarben, Bonn 2014
Vierfarben ist ein Verlag der Galileo Press GmbH
Rheinwerkallee 4, D–53227 Bonn
www.vierfarben.de

Der Verlagsname Vierfarben spielt an auf den Vierfarbdruck, eine Technik zur Erstellung farbiger Bücher. Der Name steht für die Kunst, die Dinge einfach zu machen, um aus dem Einfachen das Ganze lebendig zur Anschauung zu bringen.

Inhalt

Vorwort 29

1 Einstieg in Berechnungen mit Excel 33

1.1 Neuerungen unter Windows 8 33
1.2 Hinweise zum Programmstart 35
1.3 Excel 2013 auf dem Touchscreen 36
1.4 Hinweise zur Dateneingabe 40
1.4.1 Zellen oder Zellbereiche auswählen 40
1.4.2 Dateneingabe: Text oder Zahl? 42
1.4.3 Eingabe von Zahlen 42
1.4.4 Eingabe- und Ausgabeformat 43
1.4.5 Zahlengröße und Spaltenbreite 44
1.4.6 Eingabe von führenden Nullen 45
1.4.7 Eingabe von Brüchen 45
1.4.8 Eingabe von Datum und Uhrzeit 45
1.4.9 Erzeugen von Datenreihen 46
1.4.10 Umwandlungen per Mustererkennung 50
1.4.11 Bearbeiten von Zellinhalten 51
1.4.12 Löschmethoden 52
1.5 Die Rolle der Zahlenformate 54
1.5.1 Formatsymbole und Tastenkombinationen 54
1.5.2 Zuweisen eines Zahlenformats per Dialog 55
1.5.3 Währungsformate 57
1.5.4 Datums- und Zeitformate 58
1.5.5 Textformate und Sonderformate 58
1.5.6 Selbstdefinierte Formate 59
1.5.7 Formatcodes 61
1.5.8 Darstellung von Nullwerten 63
1.5.9 Interpretation unvollständiger Jahreszahlen 64
1.5.10 Ein Format für Zeitberechnungen 65
1.6 Formellose Berechnungen 65

1.7 Einsatz von Formeln ... 67
1.7.1 Formelsyntax ... 67
1.7.2 Formeltypen ... 69
1.7.3 Datentypen ... 70
1.8 Arbeit mit Operatoren ... 70
1.9 Hinweise zu den Grundrechenarten ... 72
1.9.1 Addition und Subtraktion ... 72
1.9.2 Multiplikation und Division ... 72
1.9.3 Division durch null abfangen ... 73
1.10 Texte verketten ... 74
1.11 Hinweise zu logischen Formeln ... 74
1.12 Formeln mit Bezugsoperatoren ... 75
1.13 Tabellenfunktionen ... 76
1.13.1 Funktionsergebnisse und Datentypen ... 76
1.13.2 Aufbau und Einsatz von Funktionen ... 77
1.14 Eingabe von Formeln und Funktionen ... 79
1.14.1 Konstanten in Formeln ... 79
1.14.2 Eingabe von Bezügen ... 80
1.14.3 Operatoren über Symbole eingeben ... 81
1.14.4 Bereichsangaben ... 82
1.14.5 3D-Bezüge ... 84
1.14.6 Externe Bezüge ... 84
1.14.7 Hilfe bei der Eingabe von Funktionen ... 87
1.14.8 Manuelle Eingabe ... 87
1.14.9 Der Dialog »Funktion einfügen« ... 90
1.14.10 Funktionen bearbeiten ... 92
1.14.11 Verschachtelte Funktionen ... 93
1.14.12 Formeln kopieren ... 94
1.14.13 Formeln dokumentieren ... 96
1.15 Einsatz von relativen und absoluten Bezügen ... 97
1.15.1 Arbeit mit relativen Bezügen ... 97
1.15.2 Absolute Bezüge ... 98
1.15.3 Gemischte Bezüge ... 101
1.16 Einsatz von strukturierten Bezügen in Tabellen ... 101
1.17 Arbeit mit benannten Bereichen ... 103
1.17.1 Vorteile von Bereichsnamen ... 103

1.17.2 Regeln für Namen ... 104
1.17.3 Methoden der Namensgebung ... 104
1.17.4 Benannte Formeln und Konstanten definieren ... 107
1.17.5 Anwenden von Namen in Formeln ... 108

1.18 Matrixberechnungen ... 110
1.18.1 Matrixbereiche in Excel ... 110
1.18.2 Arbeit mit Matrixformeln ... 111
1.18.3 Vereinfachung von Berechnungen ... 112
1.18.4 Matrizenrechnung und lineare Gleichungssysteme ... 113
1.18.5 Matrixformeln bearbeiten ... 114

1.19 Verfahren der Fehlervermeidung ... 115
1.19.1 Prüfung der Dateneingabe durch Gültigkeitsregeln ... 115
1.19.2 Fehler in Formeln vermeiden ... 118
1.19.3 Syntaxprüfung ... 119
1.19.4 Fehler durch Werte ... 119
1.19.5 Fehlerüberprüfung im Hintergrund ... 120
1.19.6 Formelüberwachung ... 122
1.19.7 Spuren verfolgen ... 122
1.19.8 Werteprüfung im Überwachungsfenster ... 123
1.19.9 Zirkuläre Formeln ... 124
1.19.10 Formeln schrittweise prüfen ... 124

1.20 Berechnungseinstellung und -optimierung ... 125
1.20.1 Berechnungsoptionen ... 125
1.20.2 Kontrolle iterativer Berechnungen ... 126
1.20.3 Berechnungsoptionen für die Arbeitsmappe ... 127

1.21 Arbeit mit bedingten Formaten ... 128

1.22 Exkurs über die Visualisierung von Daten: Sparklines und Diagramme ... 130
1.22.1 Sparklines ... 131
1.22.2 Darstellungsvarianten ... 134
1.22.3 Neuerungen für die Diagrammgestaltung ... 134
1.22.4 Von der Tabelle zum Diagramm ... 136
1.22.5 Diagrammtypen ... 137
1.22.6 Ein Diagramm erstellen ... 138
1.22.7 Achsenskalierung ... 139
1.22.8 Ändern der Diagrammdaten und des Diagrammtyps ... 141
1.22.9 Bessere Lesbarkeit mit Gitternetzlinien ... 143

2 Berechnungstools 147

2.1 Zielwertsuche ... 147

2.2 Lösungen mit dem Solver suchen ... 149

2.2.1 Zur Arbeitsweise des Solvers ... 150

2.2.2 Beispiel Materialkostenoptimierung ... 150

2.2.3 Lösungsmethoden und Optionen ... 154

2.2.4 Hinweise zu den Lösungsmethoden ... 155

2.2.5 Allgemeine Optionen ... 155

2.2.6 Spezielle Optionen ... 156

2.2.7 Auswertung der Ergebnisse und Berichte ... 156

2.3 Was wäre, wenn ... mit Datentabellen ... 157

2.3.1 Datentabelle mit einer Variablen ... 158

2.3.2 Auswertung mehrerer Formeln ... 160

2.3.3 Mehrfachoperation mit zwei Variablen ... 161

2.3.4 Beispiel Ratenberechnung ... 161

2.4 Arbeit mit Szenarios ... 162

2.4.1 Erst vergleichen, dann entscheiden ... 163

2.4.2 Aufbau des Modells ... 163

2.4.3 Welche Werte sind veränderbar? ... 164

2.4.4 Besser mit Namen ... 165

2.4.5 Einrichten verschiedener Szenarios ... 165

2.4.6 Werte für die anderen Angebote eingeben ... 166

2.4.7 Zusammenfassende Berichte ... 167

3 Finanzmathematische Funktionen 169

3.1 Einsatzbereiche der finanzmathematischen Funktionen ... 170

3.1.1 Einfache Zinsrechnung ... 170

3.1.2 Zinseszinsrechnung ... 170

3.1.3 Rentenrechnung ... 170

3.1.4 Tilgungsrechnung ... 171

3.1.5 Investitionsrechnung ... 171

3.1.6 Abschreibungsrechnung ... 171

3.1.7 Kursrechnung ... 172

3.2 Zur Berechnung von Zins und Zinseszins ... 172

3.2.1 Die Formeln für die einfache Verzinsung ... 172

3.2.2 Die Formel für den Zinseszinseffekt ... 174
3.3 Häufig benötigte Argumente und ihr Zusammenhang ... 175
3.4 Berechnungen zu Darlehen ... 177
3.5 Die Berechnung von Abschreibungen ... 178
3.6 Funktionen für Wertpapierberechnungen ... 180
3.6.1 Termine ... 181
3.6.2 Zeitbasis ... 182
3.6.3 Verkauf eines festverzinslichen Wertpapiers ... 183
3.7 Referenz der finanzmathematischen Funktionen ... 184
AMORDEGRK() ... 184
AMORLINEARK() ... 186
AUFGELZINS() ... 187
AUFGELZINSF() ... 189
AUSZAHLUNG() ... 190
BW() ... 191
3.7.1 Barwert regelmäßiger Zahlungen ... 192
3.7.2 Tilgungsrechnung ... 193
3.7.3 Barwert für einen zukünftigen Wert berechnen ... 193
DIA() ... 194
DISAGIO() ... 195
DURATION() ... 196
EFFEKTIV() ... 198
GDA() ... 199
GDA2() ... 200
IKV() ... 202
ISPMT() ... 203
KAPZ() ... 204
KUMKAPITAL() ... 205
KUMZINSZ() ... 206
KURS() ... 207
KURSDISAGIO() ... 208
KURSFÄLLIG() ... 210
LIA() ... 211
MDURATION() ... 212
NBW() ... 213
NOMINAL() ... 215
NOTIERUNGBRU() ... 215

NOTIERUNGDEZ() ... 216
PDURATION() ... 217
QIKV() ... 218
RENDITE() ... 220
RENDITEDIS() ... 221
RENDITEFÄLL() ... 222
RMZ() ... 223
TBILLÄQUIV() ... 225
TBILLKURS() ... 226
TBILLRENDITE() ... 227
UNREGER.KURS() ... 228
UNREGER.REND() ... 229
UNREGLE.KURS() ... 230
UNREGLE.REND() ... 232
VDB() ... 233
XINTZINSFUSS() ... 234
XKAPITALWERT() ... 235
ZINS() ... 236
3.7.4 Anpassung an monatliche Zahlungen ... 238
ZINSSATZ() ... 239
ZINSTERMNZ() ... 240
ZINSTERMTAGE() ... 241
ZINSTERMTAGNZ() ... 242
ZINSTERMTAGVA() ... 243
ZINSTERMVZ() ... 244
ZINSTERMZAHL() ... 245
ZINSZ() ... 246
ZSATZINVEST() ... 247
ZW() ... 248
3.7.5 Zukünftiger Wert regelmäßiger Zahlungen ... 248
3.7.6 Zukünftiger Wert einer einmaligen Einzahlung ... 249
3.7.7 Zukunftswert bei regelmäßigen Zahlungen und Einmalzahlung ... 249
ZW2() ... 250
ZZR() ... 251
3.7.8 Zinsperioden bei Einmalzahlung ... 252
3.7.9 Zahlungsperioden bei regelmäßigen Zahlungen ... 252
3.7.10 Geld liegen lassen ... 253

4 Datums- und Zeitfunktionen 255

4.1 Einsatzbereich der Datums- und Zeitfunktionen ... 255
4.1.1 Serielle Datums- und Zeitwerte ... 255
4.1.2 Die Rolle der Datums- und Zeitformate ... 256

4.2 Periodische Datumsreihen berechnen ... 257

4.3 Periodische Zeitreihen berechnen ... 260

4.4 Uhrzeit und Dauer ... 261

4.5 Tabellen für die Erfassung der Arbeitszeit ... 262
4.5.1 Erfassen der Stammdaten ... 263
4.5.2 Monatskalender erstellen ... 264
4.5.3 Arbeitstage und Arbeitszeiten berechnen ... 266
4.5.4 Nettoarbeitstage ermitteln ... 266
4.5.5 Ermitteln der Sollarbeitsstunden ... 266
4.5.6 Stundenermittlung ... 267
4.5.7 Formeln für den Gesamtsaldo ... 268
4.5.8 Name, Personal-Nr. und Abteilung anzeigen ... 268
4.5.9 Anlegen der anderen Monatsblätter ... 268
4.5.10 Arbeiten im fertigen Arbeitsblatt ... 269

4.6 Referenz der Datums- und Zeitfunktionen ... 269
ARBEITSTAG() ... 269
ARBEITSTAG.INTL() ... 271
BRTEILJAHRE() ... 273
DATEDIF() ... 274
DATUM() ... 275
4.6.1 Besonderheiten der Funktion DATUM() ... 276
4.6.2 Datumsberechnungen in Makros ... 276
DATWERT() ... 277
EDATUM() ... 278
HEUTE() ... 279
ISOKALENDERWOCHE() ... 280
JAHR() ... 281
JETZT() ... 282
KALENDERWOCHE() ... 283
MINUTE() ... 286
MONAT() ... 286
4.6.3 Nach dem Monat sortieren und summieren ... 287

MONATSENDE() ... 288
NETTOARBEITSTAGE() ... 289
NETTOARBEITSTAGE.INTL() ... 290
SEKUNDE() ... 292
STUNDE() ... 293
TAG() ... 294
TAGE() ... 294
TAGE360() ... 295
4.6.4 Berechnung von Tageszinsen ... 296
WOCHENTAG() ... 297
ZEIT() ... 299
ZEITWERT() ... 300

5 Mathematische und trigonometrische Funktionen 301

5.1 Einsatzbereiche ... 302
5.1.1 Mathematische Basisoperationen ... 302
5.1.2 Werte runden ... 302
5.1.3 Rechnen mit Matrizen ... 303
5.1.4 Trigonometrische und hyperbolische Funktionen ... 303
5.1.5 Neue Funktionen in Excel 2010 und Excel 2013 ... 304
5.2 Zu den trigonometrischen Funktionen ... 304
5.2.1 Darstellung am Einheitskreis ... 305
5.2.2 Umkehrfunktionen ... 308
5.3 Zu den hyperbolischen Funktionen ... 309
5.4 Referenz der mathematischen Funktionen ... 311
ABRUNDEN() ... 311
ABS() ... 312
AGGREGAT() ... 313
ARABISCH() ... 316
ARCCOS() ... 317
ARCCOSHYP() ... 319
ARCCOT() ... 320
ARCCOTHYP() ... 320
ARCSIN() ... 321
ARCSINHYP() ... 323
ARCTAN() ... 324

ARCTAN2() 325
ARCTANHYP() 326
AUFRUNDEN() 327
BASIS() 328
BOGENMASS() 329
COS() 330
COSEC() 332
COSECHYP() 333
COSHYP() 334
COT() 335
COTHYP() 337
DEZIMAL() 337
EXP() 339
FAKULTÄT() 340
GANZZAHL() 341
GERADE() 341
GGT() 342
GRAD() 343
KGV() 344
KOMBINATIONEN() 345
KOMBINATIONEN2() 346
KÜRZEN() 347
LN() 348
LOG() 348
LOG10() 350
MDET() 351
MEINHEIT() 352
MINV() 353
MMULT() 354
OBERGRENZE.MATHEMATIK() 355
PI() 356
POLYNOMIAL() 357
POTENZ() 358
POTENZREIHE() 359
PRODUKT() 360
QUADRATESUMME() 361
QUOTIENT() 362
REST() 363

5.4.1 Zyklische Wertreihen ... 364
RÖMISCH() ... 365
RUNDEN() ... 366
SEC() ... 367
SECHYP() ... 368
SIN() ... 369
SINHYP() ... 371
SUMME() ... 372
5.4.2 Aufsummierungen ... 373
SUMMENPRODUKT() ... 374
SUMMEWENN() ... 374
SUMMEWENNS() ... 376
SUMMEX2MY2() ... 377
SUMMEX2PY2() ... 378
SUMMEXMY2() ... 379
TAN() ... 380
TANHYP() ... 382
TEILERGEBNIS() ... 383
UNGERADE() ... 384
UNTERGRENZE.MATHEMATIK() ... 385
VORZEICHEN() ... 386
VRUNDEN() ... 387
WURZEL() ... 388
WURZELPI() ... 389
ZUFALLSBEREICH() ... 390
ZUFALLSZAHL() ... 391
ZWEIFAKULTÄT() ... 392

6 Technische Funktionen 393

6.1 Einsatzbereiche der technischen Funktionen ... 394
6.2 Besselfunktionen ... 394
6.3 Umwandlungen zwischen Zahlensystemen ... 396
6.4 Umwandeln von Maßeinheiten ... 397
6.5 Rechenoperationen mit komplexen Zahlen ... 398
6.6 Referenz der technischen Funktionen ... 401

BESSELI() 401
BESSELJ() 402
BESSELK() 403
BESSELY() 404
BININDEZ() 404
BININHEX() 405
BININOKT() 406
BITLVERSCHIEB() 407
BITODER() 408
BITRVERSCHIEB() 408
BITUND() 409
BITXODER() 410
DELTA() 410
DEZINBIN() 411
DEZINHEX() 412
DEZINOKT() 413
GAUSSF.GENAU() 414
GAUSSFEHLER() 414
GAUSSFKOMPL() 415
GAUSSFKOMPL.GENAU() 416
GGANZZAHL() 416
HEXINBIN() 417
HEXINDEZ() 419
HEXINOKT() 419
IMABS() 420
IMAGINÄRTEIL() 421
IMAPOTENZ() 421
IMARGUMENT() 422
IMCOS() 422
IMCOSEC() 423
IMCOSECHYP() 423
IMCOSHYP() 424
IMCOT() 424
IMDIV() 425
IMEXP() 425
IMKONJUGIERTE() 426
IMLN() 427
IMLOG10() 427

IMLOG2() ... 428
IMPRODUKT() ... 428
IMREALTEIL() ... 429
IMSEC() ... 429
IMSECHYP() ... 430
IMSIN() ... 430
IMSINHYP() ... 431
IMSUB() ... 431
IMSUMME() ... 431
IMTAN() ... 432
IMWURZEL() ... 432
KOMPLEXE() ... 433
OKTINBIN() ... 434
OKTINDEZ() ... 435
OKTINHEX() ... 435
UMWANDELN() ... 436

7 Statistische Funktionen 443

7.1 Einsatzbereiche für statistische Funktionen ... 444
7.1.1 Deskriptive und induktive statistische Methoden ... 444
7.1.2 Unterschiedliche Skalen ... 445
7.1.3 Urliste und Merkmalsverteilung ... 446
7.2 Stichproben und Grundgesamtheiten ... 447
7.3 Zufallsvariable und Wahrscheinlichkeit ... 447
7.3.1 Theoretische Wahrscheinlichkeit ... 448
7.3.2 Empirische Wahrscheinlichkeit ... 448
7.3.3 Untersuchung von Stichproben ... 449
7.3.4 Berechnung der Standardabweichung bei Testergebnissen ... 449
7.4 Korrelation ... 451
7.5 Regressionsanalyse ... 451
7.5.1 Ein Beispiel für lineare Regression ... 452
7.5.2 Vergleich der realen und der generierten Werte ... 455
7.5.3 Andere Methoden der Regression ... 455
7.6 Statistische Tests ... 456

7.7 Verteilungsfunktionen ... 457
7.7.1 Diskrete Verteilungen ... 457
7.7.2 Stetige Verteilungen ... 458
7.7.3 Dichtefunktion und Verteilungsfunktion ... 459
7.7.4 Umkehrfunktionen ... 461
7.8 Neue und geänderte Funktionen ... 461
7.8.1 Benennungsschema ... 461
7.8.2 Funktionen für Kompatibilität ... 464
7.8.3 Neue statistische Funktionen in Excel 2013 ... 465
7.9 Referenz der statistischen Funktionen ... 465
ACHSENABSCHNITT() ... 465
ANZAHL() ... 466
ANZAHL2() ... 468
7.9.1 Beispiel Rückstandsberechnung ... 470
ANZAHLLEEREZELLEN() ... 471
BESTIMMTHEITSMASS() ... 473
BETA.INV() ... 474
BETA.VERT() ... 475
BINOM.INV() ... 477
BINOM.VERT() ... 478
7.9.2 Anteile von Merkmalen ermitteln ... 479
BINOM.VERT.BEREICH() ... 480
CHIQU.INV() ... 481
CHIQU.INV.RE() ... 482
CHIQU.TEST() ... 482
CHIQU.VERT() ... 484
CHIQU.VERT.RE() ... 487
EXPON.VERT() ... 488
F.INV() ... 490
F.INV.RE() ... 491
F.TEST() ... 492
F.VERT() ... 493
F.VERT.RE() ... 495
FISHER() ... 496
7.9.3 Prüfen der Korrelation zweier Faktoren ... 496
FISHERINV() ... 497
G.TEST() ... 498

GAMMA() ... 499
GAMMA.INV() ... 501
GAMMA.VERT() ... 502
GAMMALN() ... 504
GAMMALN.GENAU() ... 504
GAUSS() ... 505
GEOMITTEL() ... 506
GESTUTZTMITTEL() ... 508
HÄUFIGKEIT() ... 509
7.9.4 Daten klassifizieren ... 510
HARMITTEL() ... 511
HYPGEOM.VERT() ... 512
KGRÖSSTE() ... 514
KKLEINSTE() ... 515
KONFIDENZ.NORM() ... 517
KONFIDENZ.T() ... 519
KORREL() ... 521
KOVARIANZ.P() ... 523
KOVARIANZ.S() ... 524
KURT() ... 524
LOGNORM.INV() ... 526
LOGNORM.VERT() ... 527
MAX() ... 528
MAXA() ... 529
MEDIAN() ... 529
MIN() ... 531
MINA() ... 532
MITTELABW() ... 532
MITTELWERT() ... 534
7.9.5 Mittelwert bei klassifizierten Daten ... 535
MITTELWERTA() ... 536
MITTELWERTWENN() ... 538
MITTELWERTWENNS() ... 540
MODUS.EINF() ... 542
MODUS.VIELF() ... 543
NEGBINOM.VERT() ... 545
NORM.INV() ... 546
NORM.S.INV() ... 547

NORM.S.VERT() ... 548
NORM.VERT() ... 549
7.9.6 Allgemeine Merkmale der Normalverteilung ... 550
7.9.7 Wahrscheinlichkeit einer Größe ... 552
PEARSON() ... 553
PHI() ... 554
POISSON.VERT() ... 555
QUANTIL.EXKL() ... 557
QUANTIL.INKL() ... 558
QUANTILSRANG.EXKL() ... 559
QUANTILSRANG.INKL() ... 561
QUARTILE.EXKL() ... 562
QUARTILE.INKL() ... 563
RANG.GLEICH() ... 564
RANG.MITTELW() ... 565
RGP() ... 566
RKP() ... 569
SCHÄTZER() ... 572
SCHIEFE() ... 574
SCHIEFE.P() ... 575
STABW.N() ... 577
STABW.S() ... 578
STABWA() ... 579
STABWNA() ... 581
STANDARDISIERUNG() ... 581
STEIGUNG() ... 583
STFEHLERYX() ... 584
SUMQUADABW() ... 585
T.INV() ... 587
T.INV.2S() ... 587
7.9.8 Einsatz der Funktion in Testverfahren ... 588
7.9.9 Vergleich zwischen Stichprobe und Grundgesamtheit ... 588
7.9.10 Vergleich zweier Stichproben ... 589
T.TEST() ... 590
T.VERT() ... 592
T.VERT.2S() ... 594
T.VERT.RE() ... 595
TREND() ... 595

7.9.11 Trendberechnung mit mehreren unabhängigen Variablen ... 597
VAR.P() ... 598
VAR.S() ... 600
VARIANZA() ... 601
VARIANZENA() ... 602
VARIATION() ... 603
VARIATIONEN() ... 605
VARIATIONEN2() ... 606
WAHRSCHBEREICH() ... 607
WEIBULL.VERT() ... 609
ZÄHLENWENN() ... 610
ZÄHLENWENNS() ... 612

8 Kompatible statistische Funktionen 615

8.1 Referenz der kompatiblen statistischen Funktionen ... 618
BETAINV() ... 618
BETAVERT() ... 619
BINOMVERT() ... 620
CHIINV() ... 621
CHITEST() ... 622
CHIVERT() ... 624
EXPONVERT() ... 625
FINV() ... 626
FTEST() ... 627
FVERT() ... 628
GAMMAINV() ... 629
GAMMAVERT() ... 629
GTEST() ... 631
HYPGEOMVERT() ... 632
KONFIDENZ() ... 633
KOVAR() ... 635
KRITBINOM() ... 636
LOGINV() ... 637
LOGNORMVERT() ... 638
MODALWERT() ... 639
NEGBINOMVERT() ... 640
NORMINV() ... 641

NORMVERT() ... 642
OBERGRENZE() ... 644
POISSON() ... 645
QUANTIL() ... 646
QUANTILSRANG() ... 647
QUARTILE() ... 649
RANG() ... 650
STABW() ... 651
STABWN() ... 652
STANDNORMINV() ... 652
STANDNORMVERT() ... 653
TINV() ... 654
8.1.1 Vergleich der Mittelwerte von Stichprobe und Grundgesamtheit 655
8.1.2 Vergleich der Mittelwerte zweier Stichproben ... 655
TTEST() ... 656
TVERT() ... 657
UNTERGRENZE() ... 658
VARIANZ() ... 659
VARIANZEN() ... 660
WEIBULL() ... 661

9 Nachschlage- und Verweisfunktionen 663

9.1 Einsatzbereiche für Nachschlage- und Verweisfunktionen ... 663
9.2 Arbeiten mit Verweisfunktionen ... 663
9.3 Einsatz der INDEX()-Funktion ... 665
9.4 Einsatz der WAHL()-Funktion ... 667
9.5 Referenz der Nachschlage- und Verweisfunktionen ... 668
ADRESSE() ... 668
BEREICH.VERSCHIEBEN() ... 670
9.5.1 Dynamische Bereiche ... 671
BEREICHE() ... 673
FORMELTEXT() ... 675
HYPERLINK() ... 676
9.5.2 Dynamische Links ... 677
9.5.3 Bedingte Links ... 677
INDEX() ... 678

9.5.4 Matrizenwerte abfragen ... 678
9.5.5 Bereichsbezüge liefern ... 679
INDIREKT() ... 680
MTRANS() ... 681
PIVOTDATENZUORDNEN() ... 682
RTD() ... 684
SPALTE() ... 684
SPALTEN() ... 686
9.5.6 Kombination mit der Funktion INDEX() ... 686
SVERWEIS() ... 687
VERGLEICH() ... 688
9.5.7 Zweidimensionales Lookup ... 689
VERWEIS() ... 690
WAHL() ... 691
9.5.8 Monatsnamen oder Wochentage ausgeben ... 692
9.5.9 Bereichsbezüge, wahlweise ... 693
WVERWEIS() ... 694
ZEILE() ... 695
ZEILEN() ... 696

10 Datenbankfunktionen 699

10.1 Einsatzbereiche für Datenbankfunktionen ... 699
10.1.1 Datenbanken, Datenlisten und Tabellen ... 699
10.1.2 Tabellenstrukturen ... 700
10.1.3 Datentypen und Feldlängen ... 701
10.1.4 Gemeinsame Merkmale der Datenbankfunktionen ... 702
10.1.5 Kriterienbereiche ... 702
10.1.6 Syntax der Auswahlkriterien ... 703
10.1.7 Statistische Übersicht über eine Datentabelle ... 706
10.2 Referenz der Datenbankfunktionen ... 707
DBANZAHL() ... 707
DBANZAHL2() ... 708
DBAUSZUG() ... 708
DBMAX() ... 709
DBMIN() ... 710
DBMITTELWERT() ... 711

DBPRODUKT() ... 712
DBSTDABW() ... 713
DBSTDABWN() ... 714
DBSUMME() ... 714
DBVARIANZ() ... 715
DBVARIANZEN() ... 716

11 Cube-Funktionen 717

11.1 Einsatzbereiche für Cube-Funktionen ... 717
11.2 Voraussetzungen für den Einsatz von Cube-Funktionen ... 718
11.2.1 OLAP-Cubes ... 718
11.2.2 SQL Server Data Tools ... 719
11.2.3 Dimensionen ... 719
11.2.4 Measures ... 720
11.2.5 Ein Cube-Projekt anlegen ... 720
11.2.6 Bereitstellen des Cubes ... 724
11.2.7 Die Abfragesprache MDX ... 725
11.2.8 Attribute und Attributhierarchien ... 726
11.2.9 Tupel und Mengen ... 727
11.2.10 Key Performance Indicators ... 728
11.2.11 Definieren einer Verbindung zu einem Analysis Services Server ... 730
11.2.12 Offline-Cubes ... 732
11.2.13 Cube-Formeln automatisch erzeugen ... 735
11.2.14 Besonderheiten der Cube-Funktionen ... 736
11.2.15 Beispiel für eine Lösung mit Cube-Funktionen ... 737
11.3 Referenz der Cube-Funktionen ... 739
CUBEELEMENT() ... 739
CUBEELEMENTEIGENSCHAFT() ... 740
CUBEKPIELEMENT() ... 740
CUBEMENGE() ... 742
CUBEMENGENANZAHL() ... 743
CUBERANGELEMENT() ... 744
CUBEWERT() ... 744
11.3.1 Weblinks zum Thema ... 744

12 Textfunktionen 745

12.1 Einsatzbereiche der Textfunktionen 745
12.1.1 Zahlen in Text umwandeln 746
12.1.2 Sortiermöglichkeiten durch Textfunktionen 746
12.1.3 Logische Werte in Texte aufnehmen 747
12.1.4 Verknüpfung von Text mit einem Datum 747
12.1.5 Hinweis zu Umwandlungen mit der Blitzvorschau 747

12.2 Referenz der Textfunktionen 747
BAHTTEXT() 747
CODE() 748
12.2.1 Gruppenbildung über CODE() 749
DM() 750
ERSETZEN() 751
FEST() 752
FINDEN() 753
GLÄTTEN() 754
GROSS() 755
GROSS2() 756
IDENTISCH() 757
KLEIN() 758
LÄNGE() 759
LINKS() 759
RECHTS() 761
12.2.2 Abgreifen von Namen 762
12.2.3 Minuszeichen umstellen 763
SÄUBERN() 764
SUCHEN() 764
T() 765
TEIL() 766
12.2.4 Zerlegung einer ISBN-Nummer 767
12.2.5 Datumswerte aufbereiten 768
TEXT() 768
UNICODE() 769
UNIZEICHEN() 770
VERKETTEN() 771
WECHSELN() 772
12.2.6 Ersetzen von Trennzeichen 773

12.2.7 Ersetzen von nicht druckbaren Zeichen ... 774
WERT() ... 774
WIEDERHOLEN() ... 775
ZAHLENWERT() ... 776
ZEICHEN() ... 777
12.2.8 Fortlaufende Zeichenreihen ... 778

13 Logische Funktionen 781

13.1 Einsatzbereiche der logischen Funktionen ... 781
13.2 Prüfen mit der WENN()-Funktion ... 783
13.3 Bewertungen erzeugen ... 785
13.4 Bedingte Berechnungen ... 786
13.5 Bedingte Textanzeige ... 786
13.6 Prüfungen mit komplexen Bedingungen ... 787
13.7 Mehrfachverzweigungen ... 788
13.8 Bedingte Formate ... 789
13.9 Referenz der logischen Funktionen ... 790
FALSCH() ... 790
NICHT() ... 791
ODER() ... 791
13.9.1 Vorkommen prüfen ... 792
UND() ... 793
13.9.2 Mehrfachprüfung über einen Bereich ... 793
WAHR() ... 794
WENN() ... 795
WENNFEHLER() ... 796
WENNNV() ... 797
XODER() ... 797

14 Informationsfunktionen 799

14.1 Einsatzbereiche für Informationsfunktionen ... 799
14.1.1 Beispiel für bedingte Berechnungen ... 799
14.1.2 Funktionen zur Prüfung des Datentyps ... 800

14.2 Referenz der Informationsfunktionen ... 801
BLATT() ... 801
BLÄTTER() ... 802
FEHLER.TYP() ... 803
INFO() ... 804
ISTBEZUG() ... 806
ISTFEHL() ... 806
ISTFEHLER() ... 806
ISTFORMEL() ... 807
ISTGERADE() ... 807
ISTKTEXT() ... 808
ISTLEER() ... 808
ISTLOG() ... 809
ISTNV() ... 810
ISTTEXT() ... 810
ISTUNGERADE() ... 811
ISTZAHL() ... 812
14.2.1 Nicht numerische Werte zählen ... 813
N() ... 813
NV() ... 814
TYP() ... 815
ZELLE() ... 817
14.2.2 Einsatz mit bedingten Formaten ... 820

15 Webfunktionen 821

15.1 Einsatzbereich der Webfunktionen ... 821
URLCODIEREN() ... 822
15.1.1 Exkurs zur URL-Kodierung ... 823
WEBDIENST() ... 825
XMLFILTERN() ... 825
15.1.2 Exkurs zu XPath ... 826
15.1.3 Lokalisierungspfade ... 827

16 Analyse mit Pivot-Tabellen und -Diagrammen 829

16.1 Eine Pivot-Tabelle anlegen ... 831

16.2 Layout der Pivot-Tabelle ... 834
16.3 Elementfilter und Datenschnitte ... 836
16.4 Ändern der Berechnungsart ... 837
16.5 Auswertungen mit Pivot-Diagrammen ... 839
16.6 Anwenden der Cube-Funktionen bei Datenmodellen ... 841

17 Zusätzliche Tools für die Datenanalyse 843

17.1 Aktivieren der Analyse-Funktionen ... 843
17.2 Anova-Varianzanalyse mit einem Faktor ... 845
17.3 Anova: Zweifaktorielle Varianzanalyse mit Messwiederholung ... 848
17.4 Anova: Zweifaktorielle Varianzanalyse ohne Messwiederholung ... 850
17.5 Korrelation ... 851
17.6 Kovarianz ... 852
17.7 Populationskenngrößen ... 853
17.8 Exponentielles Glätten ... 853
17.9 Zwei-Stichproben F-Test ... 855
17.10 Fourieranalyse ... 855
17.11 Histogramme für die Darstellung von Häufigkeiten ... 856
17.12 Gleitender Durchschnitt ... 858
17.13 Zufallszahlengenerierung ... 860
17.14 Rang und Quantil ... 861
17.15 Regression ... 862
17.16 Stichprobenziehung ... 864
17.17 t-Test-Varianten ... 865
17.18 Gauß-Test ... 868

18 Entwicklung eigener Funktionen 871

18.1 Funktionen mit VBA erstellen ... 871
18.1.1 Eigene Zinseszinsfunktion ... 871
18.1.2 Eigene Funktionen in einem Add-In zur Verfügung stellen ... 874
18.2 Hinweise zu VBA ... 874

18.2.1 Objekte, Eigenschaften und Methoden ... 874
18.2.2 Variablen und Konstanten in VBA ... 877
18.2.3 Explizite Deklaration ... 878
18.2.4 Zum Einsatz von Operatoren ... 880
18.2.5 Anweisungen und VBA-Funktionen ... 881
18.2.6 Mit Verzweigungen und Schleifen Abläufe steuern ... 881
18.2.7 Einschränkungen bei Funktionen ... 884

18.3 Eine komfortablere Funktion für die Zinseszinsberechnung ... 884

18.4 Effektivzins und Nominalzins bei Ratenkrediten ... 886

18.5 Funktion für eine normgerechte Berechnung der Kalenderwoche ... 890

18.6 Anlegen einer Blattliste ... 890

18.7 Funktion zur Berechnung des Osterdatums ... 891

Anhang

A Alphabetische Liste der Tabellenfunktionen 893

B Funktionenliste Deutsch–Englisch/Englisch–Deutsch 923

B.1 Deutsch–Englisch ... 923

B.2 Englisch–Deutsch ... 939

C Funktionen im Überblick 955

D Funktionstasten, Tasten und Tastenkombinationen 963

D.1 Allgemeine Tastenfunktionen im Überblick ... 963

D.2 Editiertasten und -tastenkombinationen ... 968

D.3 Navigations- und Markierungstasten ... 969

Index ... 971

Vorwort

Anders als bei vielen Excel-Büchern, die sich hauptsächlich auf kommerzielle Anwendungen beziehen, wird in den folgenden Kapiteln auf die Tatsache Wert gelegt, dass es sich bei Excel um ein universelles Werkzeug handelt, dessen Anwendungsspektrum entsprechend weit gefächert ist. Um dem gerecht zu werden, enthält der Text eine komplette Beschreibung aller Excel-Funktionen.

Das Buch bezieht sich auf den mit der Version Excel 2013 erreichten Stand. Da aber die überwiegende Mehrzahl sowohl der Funktionen als auch der speziellen Berechnungstools wie Zielwertsuche und Analyse-Funktionen seit der Version Excel 97 weitgehend unverändert geblieben sind, kann das Buch auch helfen, wenn ältere Versionen eingesetzt werden.

Excel hat, was die Benutzeroberfläche betrifft, bekanntlich mit der Version 2007 ein beachtliches Facelifting erlebt, deshalb wird an einigen Stellen zusätzlich die Bedienung in der älteren Version kurz dargestellt.

Was speziell die Funktionen betrifft, gibt es seit Excel 2007 die neue Gruppe der Cube-Funktionen, die in Kapitel 11, »Cube-Funktionen«, beschrieben wird. Ansonsten brachte Excel 2007 nur einige wenige neue Funktionen – (`SUMMEWENNS()`, `MITTELWERTWENN()`, `MITTELWERTWENNS()`, `ZÄHLENWENNS()`, `WENNFEHLER()`) – und Erweiterungen bestehender Funktionen, was die Zahl der möglichen Argumente angeht.

Für die Version Excel 2010 nahmen sich die Entwickler von Microsoft in dieser Hinsicht mehr vor. Die große Mehrzahl der Änderungen und Neuerungen betrifft dabei die Gruppe der statistischen Funktionen.

Qualifizierte Funktionsnamen

Insbesondere bei den statistischen Funktionen wurden zahlreiche Funktionsnamen geändert, um besser den in diesem Bereich üblichen Bezeichnungen zu entsprechen und die verschiedenen Funktionstypen gleichzeitig deutlicher zu kennzeichnen. Dabei wurden zusammengesetzte Namen eingeführt, bei denen die einzelnen Bestandteile durch Punkte getrennt sind.

Die Funktion FVERT() beispielsweise heißt jetzt F.VERT(). Das Suffix VERT kennzeichnet die Funktion als Verteilungsfunktion. Ähnlich wird das Suffix INV für inverse Funktionen verwendet. F.INV() ersetzt FINV(). Soll die Funktion nicht linksseitig, sondern rechtsseitig verwendet werden, wird noch ein zweites Suffix, RE, angehängt: F.VERT.RE() und F.INV.RE(). Die alte Funktion FVERT() wurde also durch zwei Funktionen ersetzt.

Damit es keine Probleme mit älteren Arbeitsmappen gibt, bleiben aber die umbenannten Funktionen weiterhin auch unter ihrem alten Namen verfügbar. Im Dialog **Funktion einfügen** sind diese Funktionen in der Kategorie **Kompatibilität** zu finden.

Bei einer Reihe von Funktionen wurden neue Algorithmen mit dem Ziel entwickelt, eine höhere Genauigkeit der Ergebnisse zu gewährleisten oder das Berechnungsverfahren zu beschleunigen.

Neue Funktionen in Excel 2010

Eine Reihe von Funktionen wurde mit Excel 2010 ganz neu oder in neuen Varianten eingefügt. In der Kategorie der Datums- und Zeitfunktionen sind dies die Funktionen ARBEITSTAG.INTL() und NETTOARBEITSTAGE.INTL(). Bei den mathematischen Funktionen sind AGGREGAT(), OBERGRENZE.GENAU() und UNTERGRENZE.GENAU() hinzugekommen. Bei den technischen Funktionen – früher Konstruktionsfunktionen genannt – sind neu: GAUSSF.GENAU() und GAUSSKOMPLF.GENAU(). Alle anderen neuen Funktionen gehören zur Kategorie der statistischen Funktionen:

BINOM.INV()	KONFIDENZ.T()	QUARTILE.INKL()
CHIQU.INV.RE()	KOVARIANZ.S()	RANG.MITTELW()
CHIQU.VERT.RE()	MODUS.VIELF()	T.INV.2S
F.INV.RE()	QUANTIL.INKL()	T.VERT.2S
F.VERT.RE()	QUANTILSRANG.INKL()	T.VERT.RE

Neue Funktionen in Excel 2013

Excel 2013 bringt eine ganze Reihe von neuen Funktionen in der Kategorie der mathematischen Funktionen, hauptsächlich Funktionen, die eine Lücke schließen in Bezug auf Dokumente, die im *.ods*-Format gespeichert werden. Das erleichtert den Austausch dieser Arbeitsmappen mit Anwendern, die Office Software wie etwa *Apache OpenOffice* oder *LibreOffice* einsetzen. Dazu gehören eine Reihe von trigonometrischen Funktio-

nen: ARCCOT(), ARCCOTHYP(), COSEC(), COSECHYP(), COT(), COTHYP(). Hinzugekommen sind auch die Funktionen ARABISCH(), BASIS() und DEZIMAL(), die mit Umwandlungen zwischen Zeichenfolgen und Zahlen zu tun haben, außerdem eine Variante für die Kombinatorik, KOMBINATIONEN2(), sowie MEINHEIT() zur Berechnung der Einheitsmatrix.

Die beiden Funktionen OBERGRENZE.MATHEMATIK und UNTERGRENZE.MATHEMATIK ersetzen die in der Version 2010 eingeführten Funktionen OBERGRENZE.GENAU und UNTERGRENZE.GENAU. Die älteren Funktionen OBERGRENZE() und UNTERGRENZE() wurden in die Kategorie **Kompatibilität** verschoben, sollten also in Zukunft möglichst nicht mehr verwendet werden.

Neu unter den finanzmathematischen Funktionen ist PDURATION() zur Berechnung der Perioden, bis eine Investition einen bestimmten Zielwert erreicht hat. Außerdem gibt es jetzt auch ZSATZINVEST() zur Berechnung des effektiven Jahreszinses für den Wertzuwachs einer Investition.

Bei den Datums- und Zeitfunktionen ist nur die Funktion ISOKALENDERWOCHE() neu, die es jetzt endlich erlaubt, die korrekte Wochenzahl ohne Umstände normgerecht zu ermitteln.

Eine paar neue Funktionen finden Sie auch in der Kategorie **Statistik**, nämlich BINOM.VERT.BEREICH(), GAMMA(), GAUSS(x), PHI(x), SCHIEFE.P() und VARIATIONEN2().

Bei den Verweisfunktionen gibt es die neue Funktion FORMELTEXT(), mit der die Formel in einer Zelle als Text ausgegeben wird, was für den Zweck der Dokumentation manchmal hilfreich sein kann.

Überfällig war in der Kategorie **Logik** die Funktion XODER(), die die Arbeit mit einem ausschließenden Oder vereinfacht. Neu ist auch WENNNV(), eine Funktion, mit der auf fehlende Werte reagiert werden kann.

Bei den Textfunktionen sind hinzugekommen: UNICODE() und UNIZEICHEN() sowie mit ZAHLENWERT() eine Funktion zur Konvertierung von Text in Zahlen unabhängig von dem aktuellen Gebietsschema auf dem System.

Bei den Info-Funktionen gibt es jetzt die neuen Funktionen BLATT() und BLÄTTER(), die ohne Argument die Nummer des aktuellen Blattes und die Zahl der Blätter in einer Arbeitsmappe liefern. Wird bei BLÄTTER() dagegen ein Bezug angegeben, liefert die Funktion die Anzahl der Blätter für diesen Bezug; wird bei BLATT() ein Wort angegeben, prüft die Funktion, ob ein Blatt mit diesem Namen existiert und liefert, wenn ja, die Blattnummer, ansonsten den Fehlerwert #NV. Bei den IST...()-Funktionen ist noch die Funktion ISTFORMEL() hinzugekommen.

In der Kategorie **Technisch** sind fünf Funktionen hinzugekommen, die mit Bitoperationen zu tun haben: BITLVERSCHIEB(), BITODER(), BITRVERSCHIEB(), BITUND() und BITXODER().

In der Gruppe der Funktionen, die mit imaginären Zahlen arbeiten, sind einige trigonometrische Funktionen neu in dieser Version: IMCOSEC(), IMCOSECHYP(), IMCOSHYP(), IMCOT(), IMSEC(), IMSECHYP(), IMSINHYP(), IMTAN().

Ganz neu in Excel 2013 ist eine kleine Gruppe von Funktionen, die in der Kategorie **Web** zusammengefasst sind: URLCODIEREN() wandelt Zeichenfolgen in gültige Webadressen um, WEBDIENST() liefert Daten aus einem Webdienst zurück, und XMLFILTERN() übernimmt aus XML-Dokumenten bestimmte Daten, die über einen XPath-Ausdruck herausgefiltert werden.

Das Buch übernimmt die von Excel vorgegebene Gruppierung der Funktionen in Kategorien. Für jede Kategorie ist ein eigenes Kapitel eingerichtet. Um den Zugriff zu vereinfachen, wird innerhalb der Kapitel die alphabetische Reihenfolge der Funktionen beibehalten.

Das erste Kapitel gibt einen Einstieg in die Art und Weise, wie in Excel Formeln und Funktionen eingesetzt werden. Im zweiten Kapitel werden spezielle Berechnungswerkzeuge vorgestellt, die die Arbeit mit Formeln ergänzen: Zielwertsuche, Solver, Datentabellen.

Kapitel 3 bis 15 behandeln jeweils eine Kategorie der Tabellenfunktionen, wobei am Anfang immer einige allgemeine Hinweise zu den Funktionen dieser Gruppe gegeben werden. Alle abgebildeten Beispiele können Sie unter der Adresse *www.vierfarben.de/excel-2013-formeln-und-funktionen/3503/titel/* herunterladen, wobei jeweils alle Funktionen einer Kategorie auch in einer Arbeitsmappe zusammengefasst sind. Dabei wird in den meisten Fällen für jede Funktion ein eigenes Blatt mit dem Funktionsnamen verwendet, nur ausnahmsweise werden auf einem Blatt zwei oder drei zusammenhängende Funktionen behandelt.

Ganz neu ist das Kapitel 16, das sich mit der Datenauswertung mit Hilfe von Datenmodellen und Pivot-Tabellen befasst. Im Anschluss daran werden in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, die speziellen Analysetools an Beispielen vorgestellt, die hauptsächlich für statistische Auswertungen angeboten werden. Den Abschluss bildet ein Kapitel, in dem gezeigt wird, wie eigene Funktionen mit Hilfe von VBA-Makros programmiert und eingebunden werden. Der Anhang enthält eine alphabetische Kurzreferenz aller Funktionen und eine Gegenüberstellung der deutschen und englischen Funktionsnamen.

Helmut Vonhoegen
www.helmut-vonhoegen.de

1 Einstieg in Berechnungen mit Excel

Der Einsatz von Excel lohnt sich häufig schon, wenn es nur darum geht, Informationen in ordentlicher Form zusammenzustellen. Die tabellarische Anordnung ist für die übersichtliche Darstellung von Daten gut geeignet. So richtig zum Zuge kommen die Fähigkeiten von Excel aber erst, wenn Sie das Programm für sich rechnen lassen. Das geschieht in erster Linie durch die Eingabe von Formeln.

In den folgenden Abschnitten werden zunächst ein paar grundlegende Dinge beschrieben, die die Arbeit mit Formeln und Funktionen in Excel betreffen.

1.1 Neuerungen unter Windows 8

Wenn Sie schon mit älteren Versionen von Excel gearbeitet haben und nun auf Windows 8 umsteigen sollten, stoßen Sie sofort auf ein paar Neuerungen. Der erste Punkt ist, dass es das altbekannte Start-Menü nicht mehr gibt. Stattdessen finden Sie nach dem Start des Systems einen ganzen Bildschirm mit dem Namen **Start**, auf dem anstelle von frei stehenden Programmsymbolen mehr oder weniger große Kacheln angeboten werden. Ein Klick oder Tipp auf die Kachel für Excel startet das Programm in der ansonsten vertrauten Desktop-Umgebung.

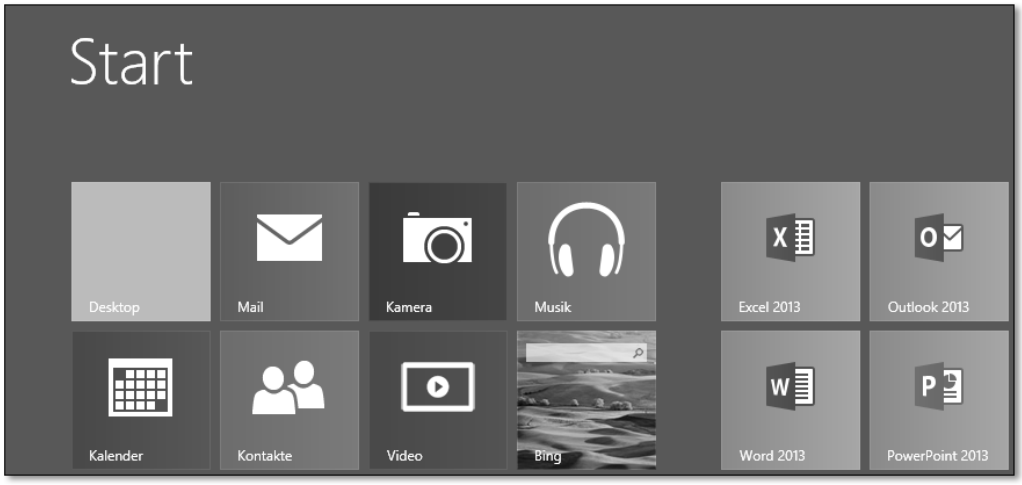


Abbildung 1.1 Die Kachel für Excel 2013 auf dem Startbildschirm von Windows 8

Wenn Sie die Kachel mit der rechten Maustaste anklicken oder sie mit Finger oder Stift etwas halten, erscheint im unteren Teil des Startfensters eine Leiste mit einigen Optionen, die für die meisten Kacheln zur Verfügung stehen.

Abbildung 1.2 Leiste mit Optionen zu einer Programmkachel

Von „Start“ lösen entfernt die Kachel von der Startseite, was sinnvoll erscheint, wenn Sie das Programm lieber über die Taskleiste auf dem Desktop starten und den Startbildschirm ausschließlich für die neuen Apps frei halten wollen. Die zweite Option verknüpft das Programm mit der Taskleiste auf dem Desktop oder löst es davon. Die dritte Option erlaubt die Deinstallation des ganzen Programms. Sie führt in den Dialog **Programme und Features**, den Sie vom Desktop aus auch über das Menü erreichen, das Ihnen die Tastenkombination ⊞ + X anbietet.

Neues Fenster öffnen bringt Ihnen das Programm auf dem Desktop mit einem neuen leeren Fenster. In speziellen Fällen ist es sinnvoll, das Programm mit Administratorrechten zu öffnen, was die fünfte Schaltfläche anbietet. Die Option **Speicherort öffnen** zeigt Ihnen auf dem Desktop im Windows-Explorer den Link auf das Programm Excel in dem dafür verwendeten Ordner. Vorgabe ist der Pfad *ProgramData* • *Microsoft* • *Windows* • *Start Menu* • *Programme* • *Microsoft Office 2013*. Das letzte Symbol erlaubt die Änderung der Kachelgröße. Der Pfeil in der unteren linken Ecke führt zur Auflistung aller installierten Apps. Ist eine Kachel einmal vom Startbildschirm entfernt worden, finden Sie diese Anwendung auf dieser Seite wieder. Über die Leiste zu diesem Anwendungssymbol finden Sie wieder die Option **An „Start“ anheften**, um die App auf die Startseite zu bringen.

Alternativ können Sie auch zunächst mit der entsprechenden Kachel auf den Desktop wechseln und für das Programm eine Verknüpfung in der Taskleiste nutzen, um es damit zu öffnen. Sie können dazu hier auch in der Leiste des Programmsymbols die Option **An Taskleiste anheften** nutzen. Ist das Symbol des Programms in die Taskleiste eingefügt, reicht ein Klick oder Tipp darauf.

An der beschriebenen Vorgehensweise wird sichtbar, dass Windows 8 Ihnen zwei verschiedene Welten von Anwendungen anbietet. Die eine, die neue Welt ist die der Apps in dem, was zunächst als *Metro Style* bezeichnet wurde und jetzt *Windows-8-style UI* heißt.

Die andere Welt ist die bisher vorherrschende Desktop-Umgebung, die noch einigermaßen der Umgebung entspricht, unter der Excel 2013 auch unter Windows 7 läuft. Eine Sonderstellung hat hier die Version von Excel 2013, die bei den Tablets auf ARM-Basis vorinstalliert ist und unter *Windows RT* läuft. Die Abkürzung RT steht für *Windows Runtime*, Geräte dieser Art stellen keine 32-Bit-API mehr zur Verfügung. Andere Desktop-Anwendungen lassen sich hier gar nicht installieren.

1.2 Hinweise zum Programmstart

Vorgabe für Excel 2013 ist unter allen Betriebssystemen, dass Ihnen beim Programmstart zunächst ein Startbildschirm angeboten wird, der neben einem Muster für eine neue Arbeitsmappe auch eine ganze Reihe von Vorlagen bereitstellt. Solange Sie Excel mit diesem Startbildschirm starten, werden Ihnen im linken Bereich immer auch die zuletzt bearbeiteten Mappen unter **Zuletzt verwendet** angeboten. Rechts von den Dateinamen finden Sie jeweils ein kleines Stiftsymbol, mit dem Sie eine Datei an die Spitze dieser Liste rücken können, um zu gewährleisten, dass sie in jedem Fall angezeigt wird.

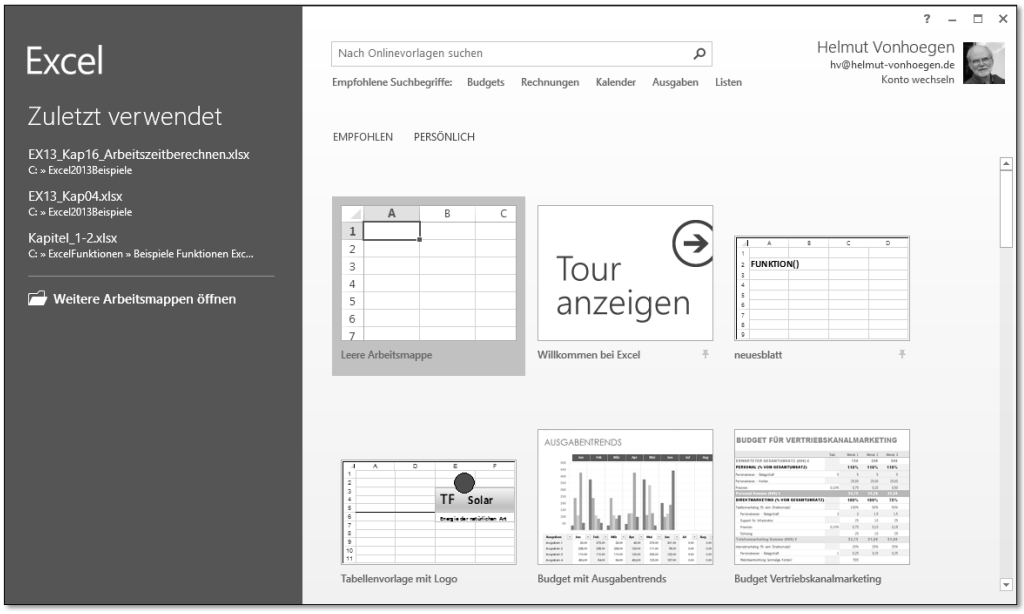


Abbildung 1.3 Der Startbildschirm von Excel kann den Zugang zu Ihren Dokumenten beschleunigen.

Ist Ihnen dieser Startbildschirm eher lästig, deaktivieren Sie über **Datei • Optionen** auf der Seite **Allgemein** die Option **Startbildschirm beim Start dieser Anwendung anzeigen**.

Dann wird das Programm gleich mit einer leeren Arbeitsmappe geöffnet. Vorgegeben wird der Name »Mappe1«.

Abbildung 1.4 zeigt die Bedienungsinstrumente, die Excel 2013 zur Verfügung stellt:

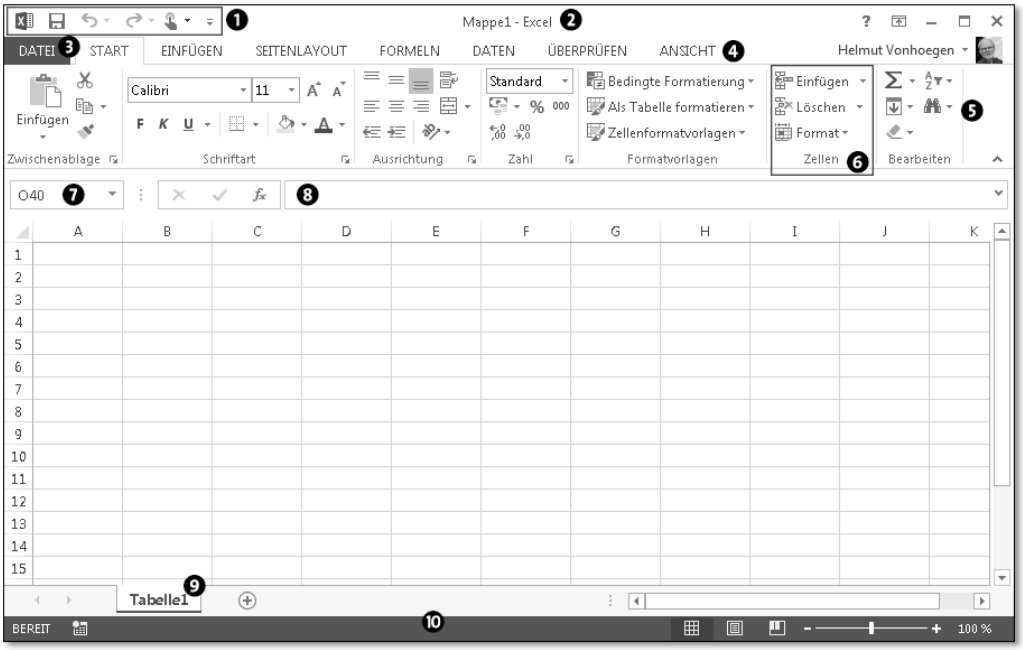


Abbildung 1.4 Die Bedieninstrumente in Excel 2013: ❶ Schnellzugriffsleiste, ❷ Titelleiste, ❸ Register Datei, ❹ Register, ❺ Menüband, ❻ Gruppe, ❼ Namenfeld, ❽ Bearbeitungsleiste, ❾ Blattregister, ❿ Statusseite

1.3 Excel 2013 auf dem Touchscreen

Die neue Version von Excel ist auch für die Arbeit auf einem Tablet oder einem anderen System mit einem Touchscreen gerüstet. Excel 2013 bietet jetzt neben der klassischen Maus- und Tastaturbedienung einen weiteren Bedienungsmodus für Finger oder Stift. Die Umschaltung in diesen Modus erfolgt auf Tablets normalerweise automatisch, Sie finden dafür in der Schnellzugriffsleiste aber auch die Schaltfläche **Fingereingabe-/Mausmodus**, um auch manuell zwischen den Modi zu wechseln.

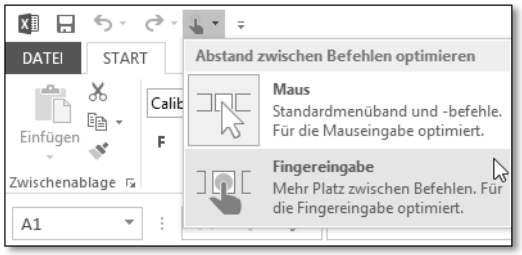


Abbildung 1.5 Das Menü zum Wechsel des Eingabemodus

Im Modus **Fingereingabe** werden die Schnellzugriffsleiste und das Menüband mit Symbolen dargestellt, die einen etwas größeren Abstand voneinander haben, damit die Finger beim Antippen das gewünschte Symbol sicherer treffen können.

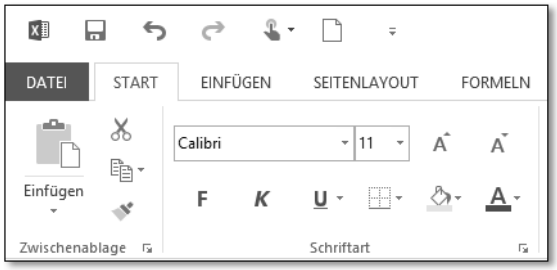


Abbildung 1.6 Das gespreizte Menüband für die Fingereingabe

Auch bei Kontextmenüs wie etwa dem für die Statusleiste ist der Abstand der Optionen vergrößert. Bei einigen Kontextmenüs ist die Aufteilung der Optionen so geändert, dass die am häufigsten verwendeten nebeneinander und die weniger häufig verwendeten untereinander angeordnet sind. Paletten wie die für die Farbauswahl werden vergrößert, damit Sie die einzelnen Muster leichter treffen.

Um eine Zelle zu markieren, reicht ein Tipp darauf. Die markierte Zelle bietet gleich an zwei Ecken Anfasser an.

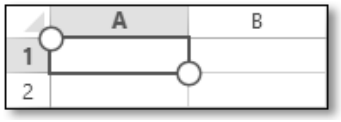


Abbildung 1.7 Markierung der Einfügestelle bei der Fingereingabe

Durch Ziehen dieser Anfasser – mit Finger oder Stift – können Sie Zellbereiche markieren.

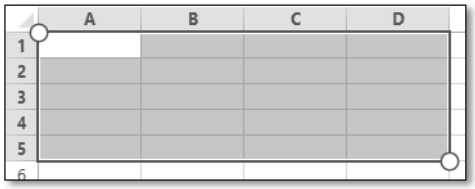


Abbildung 1.8 Markieren eines Zellbereichs bei der Fingereingabe

Soll innerhalb einer Zelle ein Teil des Textes markiert werden, tippen Sie den Text an eine Stelle an und ziehen den dort dann angezeigten Kreis, um die Auswahl zu erweitern. Das geht in beide Richtungen. Am Ende ist an jeder Seite der Auswahl ein Kreis zu sehen, mit dem die Auswahl erneut korrigiert werden kann.

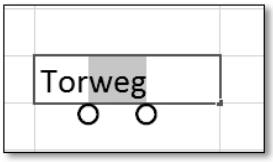


Abbildung 1.9 Markierung innerhalb einer Zelle bei der Fingereingabe

Um den Inhalt einer Zelle zu löschen, halten Sie den Finger oder Stift kurz auf der Zelle und benutzen in dem dann eingeblendeten horizontalen Kontextmenü die Schaltfläche **Markierung löschen**.

Abbildung 1.10 Kontextmenü zu einem Zellbereich bei der Fingereingabe

Der Doppelklick wird durch ein doppeltes Antippen erledigt. So wird beispielsweise eine Zelle zum Bearbeiten geöffnet. Für den Klick mit der rechten Maustaste gibt es ebenfalls eine Fingerentsprechung. Wird beispielsweise ein Zellbereich, ein Spalten- oder Zeilenkopf oder ein anderes Element, dem ein Kontextmenü zugeordnet ist, mit dem Finger oder Stift etwas länger berührt, blendet Excel die entsprechenden Befehle ein, nur die Anordnung ist – wie oben schon gezeigt – in einigen Fällen etwas anders als bei der Bedienung mit der Maus.

Eine spezielle Geste ist das Wischen. In diesem Fall kommt es auf die Richtung an. Wenn Sie im Zellraster nach oben wischen, wird der Bildschirmausschnitt nach unten verschoben, wischen Sie nach links, werden die Spalten angezeigt, die weiter rechts liegen. Die

Arbeit mit zwei Fingern erlaubt auch noch einige zusätzliche Gesten. So können Sie mit dem Spreizen oder Zusammenziehen sehr einfach den Bildschirmausschnitt vergrößern oder verkleinern, was häufig hilfreich ist, beispielsweise um Formeln in einer Zelle zu bearbeiten.

Abbildung 1.11 Das Tastatursymbol auf dem Microsoft Surface

Bei Tablets, die nur mit einer virtuellen Tastatur arbeiten, wird in der Taskleiste ein Tastatursymbol angeboten, das die Tastatur einblendet. Mit der untersten Schaltfläche der Tastatur können Sie verschiedene Tastaturlayouts wählen.

Abbildung 1.12 Die Bildschirmtastatur auf dem Microsoft Surface im Standardlayout

Für die Arbeit mit Excel sollten Sie in der Regel die Schaltfläche für die Standardtastatur verwenden, die ungefähr der normalen PC-Tastatur entspricht, nur die erweiterten Richtungstasten für den Bildschirmwechsel oder [Pos1] und [Ende] fehlen. Dieses Layout wird allerdings nur angeboten, wenn Sie unter den **PC-Einstellungen** in der Rubrik **Allgemein** unter **Bildschirmtastatur** die Option **Standardtastaturlayout verfügbar machen** eingeschaltet haben.

Ein Tipp auf die Schließen-Schaltfläche mit dem **X** blendet die Bildschirmtastatur wieder aus. Um die virtuelle Tastatur fortwährend eingeblendet zu halten, tippen Sie links daneben die Schaltfläche zum Andocken an. Ein erneutes Antippen löst diese Fixierung wieder auf; die Tastatur lässt sich dann auch nach unten verschieben. Ist die Tastatur dagegen angedockt, wird das Anwendungsfenster von Excel automatisch auf den noch freien Teil des Bildschirms reduziert. Wenn Sie in eine Zelle etwas eintippen, verschiebt Excel den Bildschirmausschnitt aber automatisch so, dass Sie immer die Eingabe sehen können. Damit möglichst viel von der Tabelle sichtbar bleibt, wenn die Tastatur verwendet wird, zeigt Excel als Vorgabe nur die Register des Menübands an, blendet also die Befehle aus.

1.4 Hinweise zur Dateneingabe

In der Regel besteht der Inhalt einer Excel-Tabelle aus drei Bestandteilen:

1. Beschriftungen
2. Werte
3. Formeln

Meist wird zunächst mit Beschriftungen die Struktur der Tabelle festgelegt, anschließend werden die Werte – hauptsächlich das Zahlenmaterial für die vorgesehenen Berechnungen – eingegeben. Schließlich werden die Formeln konstruiert.

1.4.1 Zellen oder Zellbereiche auswählen

Bei der Eingabe von Zellinhalten gilt in Excel immer: erst auswählen, dann eingeben. Die Auswahl kann mit der Maus, der Tastatur oder auf einem Touchscreen mit Finger oder Stift erfolgen. Es ist oft praktisch, die erste Zelle, in die etwas eingetragen werden soll, per Klick oder Tipp zu wählen und die Nachbarzellen dann mit den Richtungstasten anzusteuern. Solange eine Zelle im Fenster sichtbar ist, genügt ein Klick oder ein Tipp für die Auswahl. Bei der Auswahl von Zellen, die von der aktuellen Position des Zellzeigers weit entfernt sind, helfen die Bildlaufleisten, mit denen der Bildschirmausschnitt verschoben wird. Außerdem kann das Mausrad benutzt werden.

Um Zellbereiche auszuwählen, ziehen Sie mit gedrückter linker Maustaste über den gewünschten Bereich. Die Richtung spielt keine Rolle. Die Zelle, bei der Sie den Mauszeiger zum Ziehen aufsetzen, also die Markierung beginnen, ist jeweils die aktive Zelle des Bereichs. Auf einem Touchscreen ziehen Sie an den runden Anfassern, bis der Bereich die gewünschte Größe erreicht hat.

Wenn ein großer Bereich mit der Maus markiert werden soll, ist eine andere Technik effizienter: Sie klicken zunächst die erste Eckzelle des Bereichs an und halten dann beim Anklicken der gegenüberliegenden Eckzelle gleichzeitig die ⇧-Taste gedrückt.

Für die Zellauswahl mit der Tastatur sind die Richtungstasten ↑, ↓, → und ← zuständig. Bei größeren Bewegungen sind Tastenkombinationen günstiger. Um einen Zellbereich auszuwählen, können Sie die ⇧-Taste gedrückt halten und den Markierungsrahmen mit den Richtungstasten ausdehnen. Hier zunächst die Tastenfunktionen für Bewegen und Auswählen:

Taste/Tastenkombination	Beschreibung
→ oder ↹	Spalte nach rechts
← oder ⇧ + ↹	Spalte nach links
↑	Zeile nach oben
↓	Zeile nach unten
Pos1	Sprung zum Zeilenanfang
Strg + Pos1	Sprung zum Tabellenanfang
Strg + Ende	Sprung an das Ende des benutzten Tabellenbereichs
Bild ↓	ein Fenster nach unten
Bild ↑	ein Fenster nach oben
Strg + Bild ↑	Sprung zum nächsten Blatt
Strg + Bild ↓	Sprung zum vorherigen Blatt
Strg + ←	Macht die aktive Zelle wieder sichtbar, wenn der Ausschnitt vorher mit einer Bildlaufleiste oder mit Rollen verschoben worden ist.
Strg + ↑	Sprung in der Spalte nach oben zum Anfang eines Datenblocks bzw. zum Ende des vorherigen Datenblocks
Strg + ↓	Sprung in der Spalte nach unten zum Ende eines Datenblocks bzw. zum Anfang des nächsten Datenblocks
Strg + ←	Sprung in der Zeile nach links zum Anfang eines Datenblocks bzw. zum Ende des vorherigen Datenblocks
Strg + →	Sprung in der Zeile nach rechts zum Ende eines Datenblocks bzw. zum Anfang des nächsten Datenblocks

Tabelle 1.1 Die Tastenfunktionen für Bewegen und Auswählen

Für die Markierung von Bereichen stehen folgende Tastenkürzel zur Verfügung:

Tastenkombination	Beschreibung
⇧ + Richtungstaste	Ausweiten der Auswahl um eine Zeile oder Spalte
⇧ + Pos1	Ausweiten der Auswahl bis zum Zeilenanfang
⇧ + Leertaste	Auswählen der ganzen Zeile
Strg + Leertaste	Auswählen der ganzen Spalte

Tabelle 1.2 Die Tastenfunktionen für das Markieren von Bereichen

Tastenkombination	Beschreibung
Strg + ⇧ + Pos1	Ausweiten der Auswahl bis zum Tabellenanfang
Strg + ⇧ + Ende	Ausweiten der Auswahl bis an das Tabellenende
Strg + ⇧ + Leertaste	Auswählen der ganzen Tabelle
⇧ + Bild ↓	Ausweiten der Auswahl um ein Fenster nach unten
⇧ + Bild ↑	Ausweiten der Auswahl um ein Fenster nach oben
Strg + ⇧ + Richtungstaste	Ausweiten der Auswahl zum Ende/Anfang des Blocks in die Richtung der jeweiligen Taste bzw. zum Anfang/Ende des nächsten/vorherigen Datenblocks
Strg + *	Markiert den ganzen Datenblock, in dem sich die aktive Zelle befindet.
⇧ + ←	Reduzieren der Auswahl auf die aktive Zelle

Tabelle 1.2 Die Tastenfunktionen für das Markieren von Bereichen (Forts.)

Eine weitere Methode bei großen Bereichen ist, die erste Eckzelle auszuwählen, F8 zu drücken – in der Statuszeile erscheint: **Auswahl erweitern** – und anschließend die letzte Eckzelle zu wählen. Sie müssen noch einmal F8 tippen, um den Erweiterungsmodus wieder aufzuheben.

1.4.2 Dateneingabe: Text oder Zahl?

Wenn Sie in eine Zelle etwas eintragen, prüft Excel, was es damit anfangen kann. Das Programm entscheidet, ob die Eingabe als numerischer Wert, als Zeichenfolge oder als Formel verstanden werden kann. Alle Eingaben, die Excel nicht als numerischen Wert oder als Formel interpretieren kann, behandelt das Programm als Zeichenfolge. Soll dagegen eine Zahl als Zeichenfolge behandelt werden, muss Excel ausdrücklich dazu angewiesen werden. Das kann auf zweierlei Weise geschehen:

1. Sie formatieren den Zellbereich mit dem Format **Text**.
2. Sie setzen vor die Ziffern ein einfaches Anführungszeichen. Dieses Zeichen wird in der Zelle nicht angezeigt.

1.4.3 Eingabe von Zahlen

Während Texte in einem Kalkulationsprogramm insbesondere für Beschriftungen eingesetzt werden, liefern Zahlen das Futter für die Rechenarbeit. Für die Zahleneingabe sollten ein paar Regeln beachtet werden:

- Werte gleichen Typs sollten gleich formatiert werden.
- Wo es auf Dezimalstellen nicht ankommt, sollten sie weggelassen werden, damit die Tabelle übersichtlicher wird.
- Fehlen Werte zunächst, sollte festgelegt werden, ob die Zelle leer bleibt, ob eine Null eingetragen wird oder die Funktion `=NV()`, die dann mit dem Fehlerwert `#NV` anzeigt, dass der Wert noch nicht vorhanden ist.

Sie können Ziffern und bestimmte Sonderzeichen verwenden: Plus- oder Minuszeichen, Dezimalzeichen, Tausenderabtrennung, Schrägstriche, Klammern, Prozentzeichen, E bzw. e für die wissenschaftliche Schreibweise und Währungszeichen wie € oder $. Werden ausschließlich Ziffern eingegeben, wird die Eingabe als Vorgabe rechtsbündig ausgerichtet. Kommen Sonderzeichen vor, prüft Excel, ob die Eingabe einem der gültigen Zahlenformate entspricht. Ist das der Fall, wird die Eingabe als Zahl ausgegeben. Ansonsten behandelt Excel die Eingabe als Text und gibt sie linksbündig aus.

Eine Sonderrolle spielen die Wahrheitswerte `WAHR` und `FALSCH`, die allerdings selten direkt in eine Zelle eingegeben, sondern meist als Ergebnis einer Formel geliefert werden. Sie werden als Vorgabe zentriert und in Großbuchstaben angezeigt.

1.4.4 Eingabe- und Ausgabeformat

Als Vorgabe arbeitet Excel bei Zahlen zunächst mit dem sogenannten Standardformat. Dabei handelt es sich um ein Minimalformat, das automatisch vergeben wird, bevor speziellere Zahlenformate zugewiesen werden. Das Standardformat verwendet keine Tausendertrennzeichen und keine fest vorgegebene Anzahl von Nachkommastellen. Auf dieses Standardformat werden Zahlen auch wieder zurückgesetzt, wenn Sie in einem Zellbereich das Format löschen.

Sie haben aber schon bei der Dateneingabe die Möglichkeit, über dieses Minimalformat hinauszugehen. Entspricht die Eingabe einem gültigen Zahlenformat, ändert Excel für die Zelle die Formatierung vom Standardformat auf das entsprechende Zahlenformat der von Ihnen eingegebenen Zahl. Das gilt so lange, bis der Zelle ein anderes Format zugewiesen wird.

Wenn Sie also z. B. in Zelle B2 die Zahl `1.000.000` eingeben, wird diese Zahl auch genau so angezeigt, weil die Tausenderabtrennung zu einem gültigen Zahlenformat gehört. Die Zelle erhält durch die Dateneingabe ein fixes Zahlenformat. Das Format **Zahl** enthält u. a. die Tausenderabtrennung. Sie merken dies, wenn Sie den Wert in Zelle B2 mit

2000000 überschreiben, ohne Punkte für die Tausenderabtrennung zu benutzen. In der Zelle erscheint 2.000.000.

Auch Prozentwerte können direkt eingegeben werden. Wenn Sie in eine Zelle B10 19% als Mehrwertsteuersatz eingeben, wird dies auch so angezeigt. Eine Formel wie =1000*B10 rechnet dann die Höhe der entsprechenden Mehrwertsteuer aus.

Auf diese Weise können Sie mit dem ersten Eintrag in eine Zelle das Zahlenformat festlegen. Wenn Sie allerdings in eine Spalte beispielsweise eine Reihe von 400 Zahlen eingeben sollen, wäre es überflüssige Arbeit, die Zeichen für Tausenderabtrennung und Währungszeichen immer mit einzugeben. Da ist es praktischer, vor oder nach der Eingabe ein Format festzulegen, das diese Zeichen automatisch erzeugt, und manuell nur die nackten Zahlen einzutippen.

Excel bietet noch eine weitere Alternative. Wenn Sie in der ersten Zelle einer Spalte das Format durch die Eingabe vorgeben, können Sie mit der Maus das Ausfüllkästchen – das kleine Quadrat in der rechten unteren Ecke des Zellrahmens – bis ans Ende der Spalte ziehen. Klicken Sie dann auf die Schaltfläche **Auto-Ausfülloptionen**, und nehmen Sie aus dem Menü, das das kleine Dreieck öffnet, die Option **Nur Formate ausfüllen**. Die zunächst angezeigten Zahlenkopien verschwinden wieder, aber die Zellen übernehmen das Format der ersten Zelle.

1.4.5 Zahlengröße und Spaltenbreite

Ist eine eingegebene Zahl größer als die Spaltenbreite, lässt Excel, solange das Standardformat nicht durch ein anderes Format ersetzt ist, zunächst Dezimalstellen weg. Hilft auch dies nicht, wird die wissenschaftliche Schreibweise mit Exponent benutzt, z.B. 3,45E+12.

Bei allen anderen Zahlenformaten erweitert Excel automatisch die Spalte, wenn zu wenig Platz ist. Dieses freundliche Verhalten gilt aber nur, solange die Spaltenbreite nicht bereits manuell verändert worden ist. Ist die Spaltenbreite vorher verändert worden, gibt Excel anstelle der Zahl, die nicht in die Spalte passt, eine Reihe von #-Zeichen aus. Immerhin zeigt Excel den Wert an, wenn der Mauszeiger eine solche Zelle berührt. Dann sollte entweder die Spaltenbreite angepasst oder ein günstigeres Format gewählt werden.

Ist es nicht möglich, für die eingegebenen Zeichen ein gültiges Zahlenformat zu finden, wird die Eingabe als Zeichenfolge eingestuft und dementsprechend zunächst linksbündig in der Zelle ausgegeben. Wenn Sie beispielsweise 10.0 statt 10,0 in eine Zelle einge-

ben, wird dies einfach als Text gewertet. Bei Datumseingaben wird auch noch die logische Richtigkeit überprüft. Wenn Sie z.B. bei der Eingabe eines Datums irrtümlich `13.13.03` eingeben, wird dies ebenfalls als Text gewertet.

1.4.6 Eingabe von führenden Nullen

Alle von Excel direkt angebotenen Zahlenformate schneiden führende Nullen in der Anzeige weg. Werden führende Nullen aber benötigt, etwa bei Belegnummern oder Artikelnummern, hilft ein benutzerdefiniertes Zahlenformat, beispielsweise `000000` für eine sechsstellige Nummer. Mehr dazu in Abschnitt 1.5, »Die Rolle der Zahlenformate«.

1.4.7 Eingabe von Brüchen

Etwas eigentümlich ist die Art und Weise, wie Excel auf die Eingabe von Brüchen reagiert. Wenn Sie z.B. `1/3` eintippen, zeigt die Zelle `01.Mrz` an. `4/100` wird dagegen einfach als Text interpretiert, weil dieser Eintrag nicht in ein gültiges Datum umgesetzt werden kann, aber auch keine Formel ist, da das Gleichheitszeichen am Anfang fehlt.

Dagegen wird `2 3/4` anstandslos angenommen und als Wert von `2,75` verstanden, wie die Anzeige in der Bearbeitungsleiste zeigt.

Um zu erreichen, dass auch Brüche ohne vorhergehende Ganzzahl eingetippt werden können, muss eine Null vorangestellt werden. `0 1/3` ergibt das gewünschte Ergebnis, also `1/3`. Die andere Möglichkeit ist, den Bereich vorher mit dem einstelligen Bruchformat zu formatieren.

1.4.8 Eingabe von Datum und Uhrzeit

Excel betrachtet Datum und Uhrzeit als numerische Werte. Das hat den großen Vorteil, dass z.B. die Frage, welches Datum 20 Tage nach dem Tagesdatum liegt, durch eine einfache Addition geklärt werden kann. Datums- und Zeitwerte können entweder über Datums- und Zeitfunktionen, also über Formeln, in eine Zelle eingetragen werden oder direkt als Konstanten. Für die Datumseingabe als Konstante sind verschiedene Eingabeformate erlaubt, je nach den gewählten Ländereinstellungen:

```
12-10-13              12/10/13      12.10.13        12.10.2013
12. Dezember 2013     12/10         12-10
12-Jan-13             Jan-10        4-Februar
```

Alle diese Eingabeformate sind gültig. Das Standardausgabeformat hängt von den aktuellen Ländereinstellungen ab. Geändert werden können die Ländereinstellungen über die Systemsteuerung von Windows. Unvollständige Datumsangaben ergänzt Excel automatisch. Fehlt das Jahr, wird das aktuelle Jahr genommen. Fehlt der Tag, wird der Monatserste genommen. Das vollständige Datum wird jeweils in der Bearbeitungsleiste angezeigt und ist Grundlage eventueller Berechnungen.

Zeitangaben können in einem der folgenden Formate eingegeben werden:

```
10:30
10:30 PM
10:30 am
10:30:30
```

Wird PM/AM weggelassen, wird AM angenommen. Kleinbuchstaben werden in Großbuchstaben umgewandelt. In der Bearbeitungsleiste erscheint die Zeit immer in der 24-Stunden-Schreibweise. Das aktuelle Datum und die Uhrzeit können mit Strg + . bzw. Strg + : übernommen werden. Excel speichert Eingaben, die es als Datums- oder Zeitwerte erkennt, intern in Form einer seriellen Zahl, mit der die Anzahl der Tage seit dem 1.1.1900 (Tag 1) – oder alternativ dem 1.1.1904 (Tag 0) – gezählt werden.

Die Zeit wird über die Dezimalstellen der seriellen Zahl festgehalten. 12 Uhr entspricht dabei 0,50. Die serielle Zahl wird sichtbar, wenn Sie das Format löschen. Wie bei numerischen Werten üblich, richtet Excel Datums- und Zeitangaben als Vorgabe rechtsbündig aus. Wird ein »falsches« Datum eingegeben (etwa `31.04.2010`), behandelt Excel die Eingabe als Text und richtet sie entsprechend linksbündig aus.

1.4.9 Erzeugen von Datenreihen

Wo ein Zellbereich mit Daten gefüllt werden soll, die eine erkennbare Reihe bilden, lässt sich die Dateneingabe automatisieren. Reihen vereinfachen die Beschriftung von Zeilen und Spalten mit fortlaufenden Nummerierungen. Zwei Methoden stehen zur Verfügung: die Arbeit mit dem Ausfüllkästchen bzw. dem **AutoAusfüllen**-Symbol auf dem Touchscreen und der Befehl **Start • Bearbeiten • Füllbereich**. Die Schaltfläche öffnet ein Menü mit mehreren Optionen zur Reihenbildung. Bei kleinen Reihen kommen Sie mit der ersten Methode schnell zum Ziel, bei großen Reihen gibt der Befehl mehr Kontrolle über die Reihenbildung.

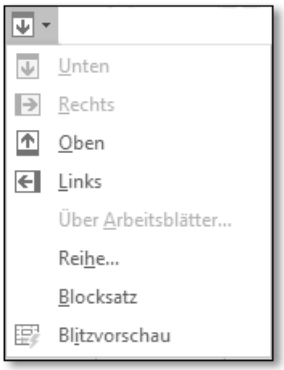


Abbildung 1.13 Menü der Schaltfläche **Füllbereich**

Eine häufig benötigte Datenreihe sind Quartalsbezeichnungen. Hier ist das Aufziehen ganz praktisch:

1. Sie geben die Bezeichnung für das Quartal ein, mit dem die Reihe beginnen soll.
2. Sie setzen den Mauszeiger genau auf das Ausfüllkästchen.
3. Mit gedrückter linker Maustaste wird der Hilfsrahmen, der dann erscheint, entweder innerhalb der Zeile nach rechts oder auch nach links oder innerhalb der Spalte nach unten oder oben ausgedehnt. Zur Kontrolle wird immer angezeigt, welches Quartal gerade erreicht wurde.
4. Wird die Maustaste wieder losgelassen, ist der mit dem Hilfsrahmen ausgewählte Bereich ausgefüllt.

Excel 2013 blendet normalerweise die Schaltfläche **Auto-Ausfülloptionen** mit einem Menü ein, in dem Sie beispielsweise wählen können, dass im Ausnahmefall die Formate der Ausgangszelle nicht mit übernommen werden. Im Normalfall überträgt Excel beim Ausfüllen auch die Formatierung der Ausgangszelle auf die neuen Zellen.

Ausgewählte Zellen auf einem Touchscreen bieten Ihnen zunächst kein Ausfüllkästchen an. Sie können aber ein etwas größeres Kästchen einblenden, mit dem sich ebenfalls Reihen bilden lassen.

Abbildung 1.14 Kontextmenü (Zellbereich) auf dem Touchscreen mit dem Symbol **AutoAusfüllen**

1. Wenn Sie die Zelle für das 1. Quartal kurz berühren, erscheint im Kontextmenü die Schaltfläche **AutoAusfüllen**. Diese Schaltfläche blendet anstelle des Ausfüllkästchens ein etwas größeres Symbol mit einem Pfeil ein.
2. Wenn Sie dieses Symbol mit Finger oder Stift nach rechts oder nach unten ziehen, fügt Excel eine Reihe mit den anderen Quartalen ein.

Was die Reihenbildung beim Ziehen mit Maus, Finger oder Stift liefert, hängt vom Inhalt der zunächst markierten Ausgangszelle(n) ab. Das Programm prüft, ob sich daraus eine fortlaufende Reihe erzeugen lässt oder nicht. Wenn nicht, werden die Zellinhalte nur kopiert.

Schreiben Sie in die erste Zelle `Haus 1`, unterstellt das Programm, dass Sie eine Reihe von Häusern durchnummerieren wollen, also Haus 1, Haus 2, Haus 3. Dabei spielt es keine Rolle, wo die Zählnummer steht, ob am Anfang des Zelleintrags oder am Ende. Am Anfang muss sie allerdings durch ein Leerzeichen vom Text getrennt sein. Dabei kann hinter der Nummer noch ein Punkt stehen. Ein Eintrag wie `1.Tag` ergibt allerdings immer nur `1.Tag`, während `1. Tag` (also mit Leerzeichen) eine durchnummerierte Reihe ergibt.

Soll eine fortlaufende Nummerierung erzeugt werden, reicht es aus, in die erste Zelle `1` oder eine andere Startzahl einzugeben. Allerdings muss dann über **Auto-Ausfülloptionen** die Option **Datenreihe ausfüllen** gewählt werden, weil Excel die Zahl sonst nur kopiert. Alternativ können Sie beim Ziehen mit der Maus auch die Strg-Taste gedrückt halten. Wenn die Aufeinanderfolge nicht in Einerschritten erfolgen soll, ist es notwendig, zunächst zwei Zellen auszufüllen und zu markieren. Die erste Zelle enthält den Ausgangswert und die zweite Zelle den nächsten Wert, so dass das Programm den Schrittwert erkennen kann.

Zeitreihen können linear wie beim Datum oder kreisförmig wie bei der Uhrzeit oder bei Wochentagen oder Monatsreihen sein. Die Abstände werden entweder nach dem ersten Wert automatisch in Einerschritten oder durch den Inhalt einer zweiten Zelle bestimmt. Hier einige Beispiele:

Anfangswerte	Fortsetzung
8:15	9:15 10:15 11:15
Di	Mi Do Fr Sa So Mo Di
Montag	Dienstag Mittwoch Donnerstag
1. Mai 15. Mai	29. Mai 12. Jun 26. Jun 10. Jul
Januar	Februar März April

Tabelle 1.3 Beispiele für Zeitreihen

Arithmetische Reihen sind Reihen mit konstantem Abstand. Um sie durch Ziehen zu erzeugen, müssen zwei Zellen markiert sein, die den ersten und den zweiten Wert der Reihe enthalten. Bei geometrischen Reihen wird jeweils der vorhergehende Wert mit einem konstanten Faktor multipliziert. Diese Reihen können jedoch nur über das Dialogfeld **Reihe** erzeugt werden, das mit **Start • Bearbeiten • Füllbereich • Reihe** geöffnet wird.

Benötigen Sie eine Liste von Wochentagen, finden Sie im Menü der Schaltfläche **Auto-Ausfülloptionen** die Option **Wochentage ausfüllen**, die automatisch nur die Werktage aufreiht.

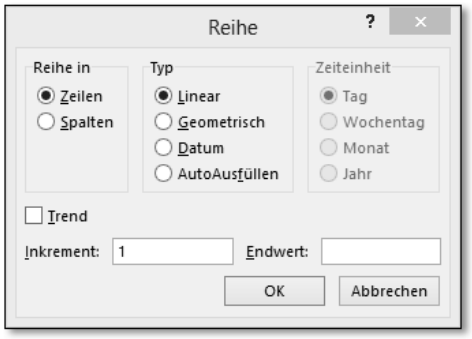


Abbildung 1.15 Der Dialog **Reihe**

Der Dialog **Reihe** bietet noch einige zusätzliche Möglichkeiten. Allerdings ist das Verfahren etwas anders. Zwei Varianten stehen zur Wahl:

- Sie markieren vor Aufruf des Befehls den kompletten Bereich, der gefüllt werden soll, einschließlich der Startzelle.
- Sie wählen nur die Ausgangszelle(n) und steuern die Ausdehnung der Datenreihe über den Endwert im Dialogfeld.

Wenn Sie nur die Ausgangszelle(n) ausgewählt haben, wählen Sie unter **Reihe in** aus, ob die Reihe in der Zeile, d.h. nach rechts, oder in der Spalte, also nach unten, gebildet werden soll. In der zweiten Optionsgruppe wird der **Typ** der Datenreihe ausgewählt. Excel gibt den Typ aufgrund der markierten Startzelle vor.

1.4.10 Umwandlungen per Mustererkennung

Eine in Excel 2013 ganz neue Methode, Spalten mit Inhalt zu füllen, wird unter dem irreführenden Namen **Blitzvorschau** angeboten, eine schlechte Übersetzung für **Flash Fill**. Dabei geht es um ein Verfahren, das mit Mustererkennung arbeitet. Die Methode kann beispielsweise als Hilfe beim Umgang mit Adressen eingesetzt werden. Angenommen, Sie übernehmen aus einer externen Datenquelle eine Spalte mit Zu- und Vornamen, getrennt durch ein Komma. Nun wollen Sie den Zunamen separat in der Nachbarspalte als Benutzernamen ablegen. Die Lösung geht so:

1. Sie legen in der ersten Zelle der neuen Spalte den ersten Benutzernamen ab.
2. Markieren Sie die Zelle unter dem ersten Namen. Rufen Sie den Befehl **Start • Bearbeiten • Füllbereich • Blitzvorschau** auf.
3. Das Programm geht davon aus, dass Sie diese Freistellung des Zunamens in der gesamten Spalte haben wollen, und erledigt dies auf der Stelle.
4. Bei der ersten Zelle erscheint noch die Schaltfläche **Blitzvorschauoptionen** mit Befehlen, um notfalls das Ausfüllen rückgängig zu machen oder um die gefüllten Zellen auszuwählen.

Abbildung 1.16 Spalten füllen durch Mustererkennung

Dieses Verfahren funktioniert auch mit zahlreichen anderen Mustern. Wenn Sie beispielsweise eine Spalte mit ISBN-Nummern haben und die zweite Zahl freistellen wollen, können Sie ebenso verfahren, falls die Zahlengruppen durch Striche getrennt sind.

Umgekehrt geht es auch: Sie können beispielsweise den Inhalt von zwei verschiedenen Spalten in einer neuen Spalte zusammenfügen, etwa die getrennten Vor- und Zunamen zu einem Kundennamen. Dabei lassen sich auch bestimmte erkennbare Bestandteile durch andere ersetzen. Es ist also möglich, an bestimmten Stellen Zeichenfolgen oder Ziffern hinzuzufügen. Wenn das erste Muster nicht eindeutig ist, arbeiten Sie mit zwei oder drei Mustern, so dass das Programm möglichst exakt verfahren kann.

1.4.11 Bearbeiten von Zellinhalten

Excel stellt es Ihnen frei, den Zellinhalt direkt in der Zelle zu bearbeiten oder in der Bearbeitungsleiste. Um den Zellinhalt direkt in der Zelle zu bearbeiten, klicken oder tippen Sie doppelt auf die entsprechende Zelle oder benutzen F2. Das Programm wechselt in den Bearbeitungsmodus. Bei Formeln wird statt des Ergebnisses die Formel angezeigt. Ein blinkender Strich zeigt die Einfügestelle an. Mit einem Mausklick, einem Tipp mit Finger oder Stift oder mit den Richtungstasten kann die Einfügestelle versetzt werden. Quittiert wird die Korrektur in der Zelle durch die Auswahl einer anderen Zelle per Klick oder Tipp oder mit ↵ oder mit ⇆.

Um den Zellinhalt in der Bearbeitungsleiste zu bearbeiten, wählen Sie die Zelle zunächst aus, damit ihr Inhalt in der Bearbeitungsleiste angezeigt wird. Wenn Sie eine bestimmte Stelle ändern wollen, klicken oder tippen Sie gleich auf diese Stelle in der Bearbeitungsleiste.

Mit der Maus können Sie Zeichen, die Sie durch andere ersetzen wollen, durch Ziehen markieren. Auf dem Touchscreen ziehen Sie den runden Anfasser. Ganze Worte oder Argumente in einer Formel können per Doppelklick oder -tipp markiert werden. Wenn Sie die ⇧-Taste festhalten, können Sie eine Markierung auch bis zu der Stelle erweitern, an die Sie den Mauszeiger setzen. Das nächste eingegebene Zeichen ersetzt dann jeweils die markierten Zeichen. Weitere Zeichen werden an der Stelle eingefügt. Das Programm arbeitet also normalerweise im Einfügemodus.

Sie können aber auch die Einfügestelle auf das erste Zeichen setzen, das überschrieben werden soll, und dann mit Einfg den Überschreibmodus einschalten. Sind die Zeichen überschrieben, schalten Sie mit Einfg wieder in den Einfügemodus zurück. Die Tabelle zeigt die Tastenfunktionen zur Zellbearbeitung:

Taste/Tastenkombination	Beschreibung
Pos1	Sprung zum Anfang des Eintrags
Ende	Sprung zum Ende des Eintrags
Strg + →	ein Wort oder Argument nach rechts
Strg + ←	ein Wort oder Argument nach links
⇧ + →	Markierung um ein Zeichen nach rechts erweitern
⇧ + ←	Markierung um ein Zeichen nach links erweitern
Strg + ⇧ + →	Markierung um ein Wort oder Argument nach rechts erweitern
Strg + ⇧ + ←	Markierung um ein Wort oder Argument nach links erweitern
Strg + ⇧ + Pos1	Markierung bis zum Anfang des Eintrags erweitern
Strg + ⇧ + Ende	Markierung bis zum Ende des Eintrags erweitern
Strg + ⇧ + A	Einfügen der Argumente bei einer Funktion. Kann benutzt werden, wenn die erste Klammer hinter dem Funktionsnamen eingegeben ist.
Entf	Löscht markierte Zeichen oder das Zeichen rechts von der Einfügemarke.
←	Löscht markierte Zeichen oder das Zeichen links von der Einfügemarke.
Strg + Entf	Löscht ab Einfügemarke oder Markierung bis zum Ende des Eintrags.
Strg + C	Kopiert die markierten Zeichen in die Zwischenablage.
Strg + X	Schneidet die markierten Zeichen aus und legt sie in die Zwischenablage.
Strg + V	Fügt die Daten aus der Zwischenablage an der Einfügestelle oder anstelle der markierten Zeichen ein.

Tabelle 1.4 Tasten und Tastenkombinationen zur Zellbearbeitung

1.4.12 Löschmethoden

Unter dem Begriff *Löschen* kann in Bezug auf die Eigenschaften einer Zelle oder eines Zellbereichs dreierlei verstanden werden:

1. Löschen des Zellinhalts
2. Löschen des Zellformats
3. Löschen von Kommentaren

Seit Excel 2007 wird für diese Maßnahmen unter **Start • Bearbeiten** die Schaltfläche **Löschen** angeboten, deren Menü die verschiedenen Optionen enthält.

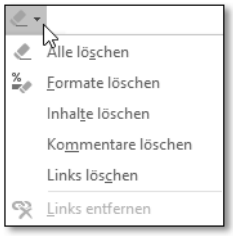


Abbildung 1.17 Optionen der Schaltfläche **Löschen** in der Gruppe **Bearbeiten**

In den älteren Versionen stehen diese Optionen über **Start • Bearbeiten • Löschen** zur Verfügung. Von diesen Löschoptionen muss das Entfernen ganzer Zellen unterschieden werden, das ebenfalls als Löschen bezeichnet wird. Die entsprechenden Befehle finden Sie über das Menü zu **Löschen** in der Gruppe **Start • Zellen**. Stattdessen können Sie im Kontextmenü einer Zellauswahl dafür auch den Befehl **Zellen löschen** verwenden.

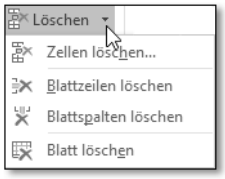


Abbildung 1.18 Optionen der Schaltfläche **Zellen löschen**

Wird der Inhalt einer Zelle gelöscht, ist die Zelle leer. Diese Banalität muss deshalb betont werden, weil eine Zelle manchmal zwar leer aussieht, aber nicht leer ist. Das kann vorkommen, wenn eine Zelle Leerzeichen enthält. Wenn Sie die Zelle doppelt anklicken, steht die Einfügestelle nicht ganz links, sondern rechts vom letzten Leerzeichen. Solche übersehenen Leerzeichen können z. B. beim Abfragen von Tabellen Verwirrung stiften.

Das Löschen des Zellformats wiederum bedeutet nicht, dass die Zelle anschließend überhaupt keine Formateigenschaften mehr hat. Die Zelle wird lediglich auf das Standardformat zurückgesetzt. Die Taste Entf löscht nur den Zellinhalt. Die Formatierung der Zelle bleibt davon unberührt.

Wenn Sie Zellinhalte in einer Zelle oder eines Zellbereichs mit der Maus löschen wollen, müssen Sie diese nur markieren und dann das Ausfüllkästchen so weit nach innen ziehen, bis der gesamte Bereich durch das Raster schraffiert ist. Wenn Sie loslassen, ist der markierte Bereich gelöscht. Auf dem Tablet können Sie den Bereich kurz antippen und dann im Kontextmenü den Befehl **Markierung löschen** verwenden.

1.5 Die Rolle der Zahlenformate

In Abschnitt 1.4 sind Zahlenformate schon kurz angesprochen worden. Zahlenformate bestimmen die Art und Weise, wie der in die Zelle eingetragene Wert dargestellt wird. Bei numerischen Werten betrifft das insbesondere die Frage, ob und mit wie vielen Dezimalstellen eine Zahl ausgegeben wird. Geregelt werden kann über das Format auch, wie negative Zahlen erscheinen. Neben dem Minuszeichen und den Klammern können solche Werte etwa auch durch eine rote Textfarbe hervorgehoben werden.

1.5.1 Formatsymbole und Tastenkombinationen

Für die gebräuchlichsten Zahlenformate stellt Excel 2013 in der Gruppe **Start • Zahl** eine Formatpalette und eine Reihe von Schaltflächen zur Verfügung. Das Listenfeld der Palette zeigt jeweils das Format der aktuellen Zelle an. Die Palette enthält zahlreiche Formate aus den verschiedenen Kategorien, die einfach per Klick oder Tipp einem vorher ausgewählten Zellbereich zugeordnet werden.

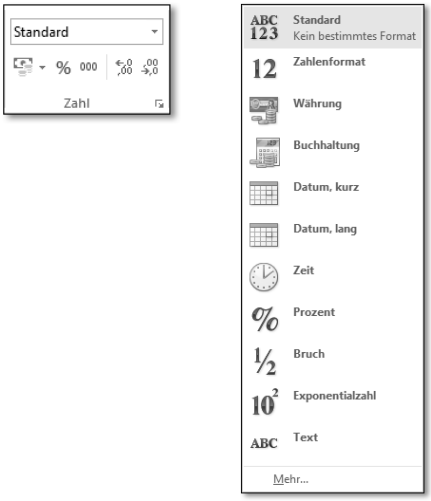


Abbildung 1.19 Symbole und Palette für gängige Zahlenformate

Die angesprochenen Schaltflächen werden ebenfalls in der Minisymbolleiste des Kontextmenüs zu einem Zellbereich angeboten. Auch einige Tastenkombinationen sind für Zahlenformate einsetzbar, wie die folgende Tabelle zeigt:

Symbol	Tastenkombination	Format
	Strg + $	Währungsformat
	Strg + %	Prozentformat
	Strg + !	1.000,00
	Strg + &	Standardformat
	Strg + §	Datumsformat TT.MMM JJ
		Dezimalstellen hinzufügen
		Dezimalstellen entfernen

Tabelle 1.5 Symbole und Tastenkombinationen für Zahlenformate

Das Währungssymbol bietet über die Pfeilschaltfläche ein Menü an, das den schnellen Wechsel zwischen verschiedenen Währungsformaten unterstützt.

Wenn Sie das Prozentformat einer Zelle zuweisen, wird nicht einfach nur das Prozentzeichen an den vorhandenen Wert angehängt, gleichzeitig wird mit 100 multipliziert. Da der Wert aber meist durch eine Division ermittelt wird – etwa nach dem Schema »Teilbetrag dividiert durch Gesamtbetrag« –, ist das beschriebene Verhalten ganz praktisch.

1.5.2 Zuweisen eines Zahlenformats per Dialog

Mehr Kontrolle über die Formatierung haben Sie über das auch in den älteren Excel-Versionen verwendete Dialogfeld **Zellen formatieren**, das in der Palette über die Option **Mehr** oder mit Strg + 1 geöffnet werden kann. Dabei spielt es keine Rolle, ob die Zellen vor oder nach der Eingabe von Daten formatiert werden.

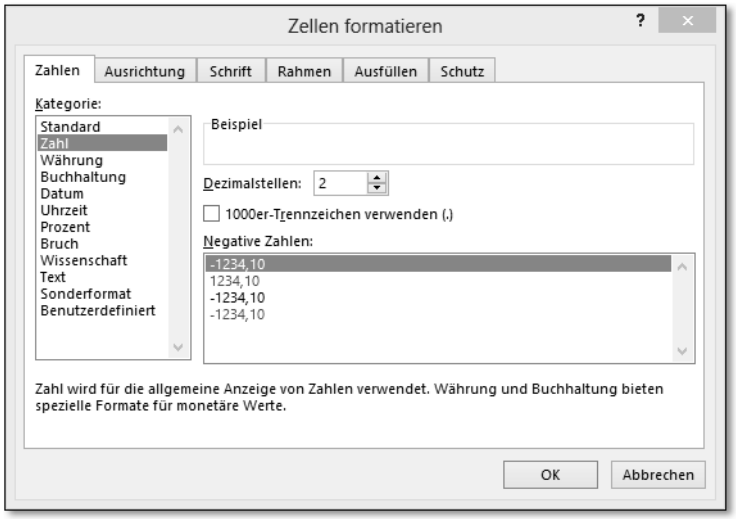


Abbildung 1.20 Dialogfeld **Zellen formatieren** mit dem Register **Zahlen**

Angenommen, Sie wollen in einer Tabelle mit statistischen Auswertungen Zahlen mit der Tausenderabtrennung, aber ohne Nachkommastellen anzeigen lassen.

1. Wählen Sie zunächst den Zellbereich aus.
2. Öffnen Sie das Dialogfeld **Zellen formatieren**.
3. Auf dem Register **Zahlen** gibt Excel zunächst das aktuelle Format der aktiven Zelle vor, es sei denn, Sie haben einen Bereich ausgewählt, in dem unterschiedliche Formate verwendet werden.
4. Wählen Sie unter **Kategorie** den Eintrag **Zahl**. Bei den Formaten der Kategorie **Zahl** können die verschiedenen Elemente des Formats einzeln ausgewählt werden.
5. Setzen Sie in diesem Fall die Anzahl der Dezimalstellen auf null.
6. Aktivieren Sie **1000er-Trennzeichen verwenden (.)**.
7. Was die Darstellung der negativen Zahlen angeht, wählen Sie aus der kleinen Liste die gewünschte Einstellung.
8. Wenn das unter **Beispiel** angezeigte Format Ihren Wünschen entspricht, bestätigen Sie mit **OK**.

1.5.3 Währungsformate

Bereits seit Excel 97 ist es kein Problem mehr, in einem Tabellenblatt mit unterschiedlichen Währungen zu operieren. Unter der Kategorie **Währung** und **Buchhaltung** werden Zahlenformate mit Währungszeichen angeboten. Diese Formate verwenden gleichzeitig das Tausendertrennzeichen. In der Liste für das Währungssymbol werden zahlreiche Währungszeichen zur Auswahl gestellt. Es ist also kein Umstand, in einer Spalte Eurobeträge anzuzeigen, in der nächsten Dollars und in der dritten britische Pfund.

Speziell für die buchhalterische Dateneingabe – z. B. für Bilanzen und Journale – werden Zahlenformate unter **Buchhaltung** angeboten. Bei diesen Formaten werden Nullwerte mit einem Bindestrich dargestellt. Die Zahlen werden so ausgerichtet, dass bei gleicher Anzahl von Dezimalstellen das Komma immer an derselben Stelle steht, egal ob Währungssymbole angezeigt werden oder nicht. Das Minuszeichen bei negativen Werten wird linksbündig dargestellt.

	A	B	C	D
1				
2	**Internationale Preisliste**			
3				
4		**Euro**	**Dollar**	**Pfund**
5	Artikel 1	50,00 €	$52,63	£22,73
6	Artikel 2	75,00 €	$78,95	£34,09
7	Artikel 3	100,00 €	$105,26	£45,45
8	Artikel 4	125,00 €	$131,58	£56,82
9	Artikel 5	150,00 €	$157,89	£68,18
10	Artikel 6	175,00 €	$184,21	£79,55

Abbildung 1.21 Tabelle mit unterschiedlichen Währungsformaten

Für Prozentzahlen kann in der Kategorie **Prozent** die Anzahl der Dezimalstellen eingestellt werden.

Unter **Bruch** gibt es zahlreiche Formatangebote. Wenn Sie das Format **Einstellig** einem Zellbereich schon vor der Dateneingabe zuordnen, können Sie beispielsweise `1/4` eingeben, ohne dass Excel daraus ein Datum macht. Es ist dann nicht notwendig, `0 1/4` einzugeben.

Unter **Wissenschaft** steht das Exponentialformat zur Verfügung, wobei die Anzahl der Dezimalstellen in der Mantisse frei gewählt werden kann. Bei diesem Format wird die Zahl jeweils als Produkt einer Zahl mit einer Zehnerpotenz dargestellt. Diese Darstellungsform ist insbesondere für sehr große oder sehr kleine Zahlen geeignet, wie sie im technisch-wissenschaftlichen Bereich häufig vorkommen.

1.5.4 Datums- und Zeitformate

Neben den Zahlenformaten bietet Excel zahlreiche Formate für die Anzeige von Datums- und Zeitangaben an. Auch auf internationale Formate aus vielen Ländern können Sie zugreifen, wenn Sie über das Listenfeld **Gebietsschema** eine entsprechende Einstellung wählen. Sehr praktisch ist oft das lange Datumsformat, bei dem gleich der Name des Tages vor dem Datum ausgegeben wird.

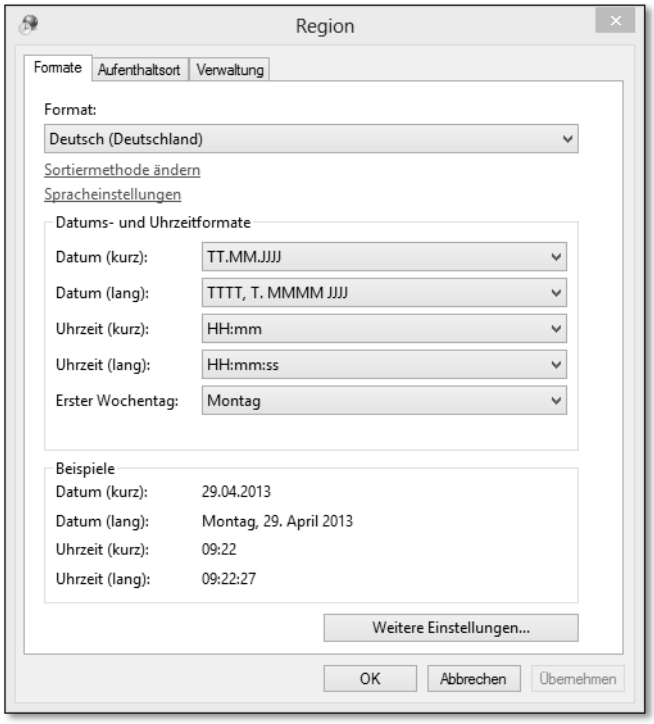


Abbildung 1.22 Die regionalen Einstellungen über die **Systemsteuerung** finden Sie unter **Region**.

Welches Ausgabeformat von Excel als Vorgabe benutzt wird, ist von den über **Systemsteuerung** gewählten Ländereinstellungen abhängig. Wenn Sie z.B. dort im Dialog **Region** auf der Registerkarte **Formate** ein kurzes Datumsformat wie `TT.MM.JJJJ` auswählen, wird eine zweistellige Jahreszahl automatisch vierstellig ausgegeben.

1.5.5 Textformate und Sonderformate

Ein spezielles Format ist das oben schon angesprochene Format **Text**. Wenn Sie einen leeren Zellbereich mit diesem Format belegen, werden alle Zahlen, die Sie anschließend

eingeben, von Excel als Zeichenfolgen behandelt und auch entsprechend linksbündig ausgerichtet. Diese Verfahrensweise ist praktisch, wenn Sie z.B. Codenummern, Kundennummern, Telefonnummern etc. eingeben wollen.

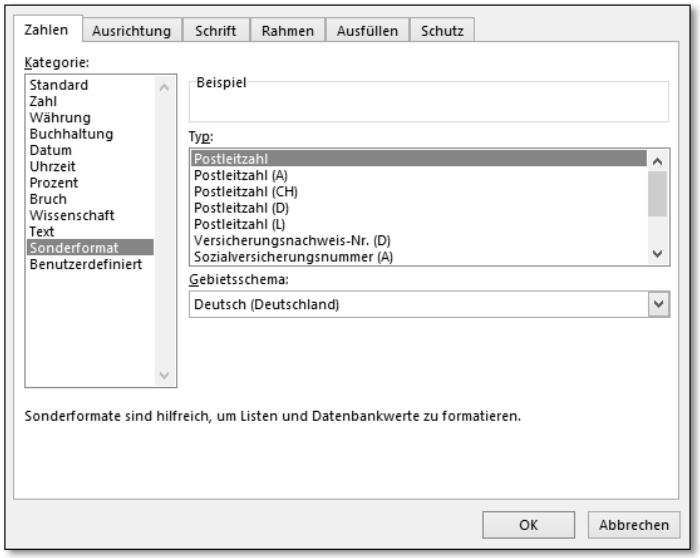


Abbildung 1.23 Sonderformate beziehen sich auf das ausgewählte Gebietsschema.

Ganz hilfreich ist eine Reihe von Sonderformaten, wie sie für Postleitzahlen, ISBN- oder Versicherungsnummern benötigt werden. Auch hier sind Formate für bestimmte Länder verfügbar, die Sie über **Gebietsschema** auswählen können.

1.5.6 Selbstdefinierte Formate

Wenn die integrierten Zahlenformate nicht ausreichen, können Sie das Dialogfeld **Zellen formatieren** nutzen, um eigene Formate zu definieren. Eine Formatbeschreibung kann aus bis zu vier Bestandteilen bestehen, die jeweils durch Semikola getrennt werden:

```
positives Format; negatives Format; Nullformat; Textformat
```

Enthält ein Zahlenformat nur drei Bestandteile, bedeutet dies, dass es kein spezielles Textformat gibt. Enthält die Zelle Text, wird er ganz normal als Text ausgegeben. Enthält ein Zahlenformat nur zwei Bestandteile, gilt das erste für positives Format und Nullformat, das zweite für negatives Format. Ist nur ein Bestandteil gegeben, gilt dies für positives Format, negatives Format und Nullformat.

Solche benutzerdefinierten Formate können anhand eines vorhandenen Formats oder von Hand gestaltet werden. Angenommen, Sie wollen sechsstellige Belegnummern verwenden, die führende Nullen anzeigen. Sie können folgendermaßen verfahren:

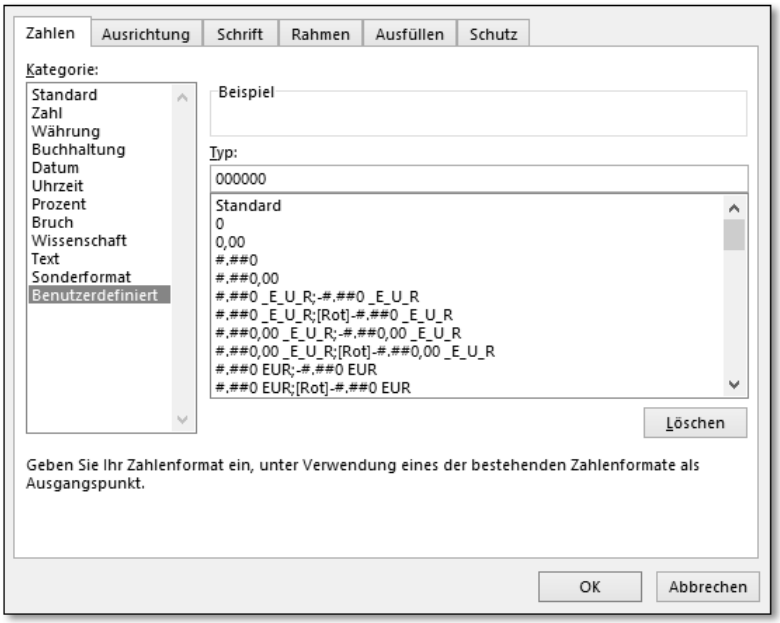


Abbildung 1.24 Definition eines eigenen Zahlenformats

1. Markieren Sie den Zellbereich.
2. Wählen Sie im Dialogfeld **Zellen formatieren** unter **Kategorie** den Eintrag **Benutzerdefiniert**.
3. In dem Listenfeld werden Ihnen Formatmuster angeboten, die Sie als Ausgangsmaterial für das eigene Format verwenden können. Markieren Sie den Eintrag mit der Null.
4. Ergänzen Sie in dem Eingabefeld **Typ** die restlichen fünf Nullen und bestätigen Sie mit **OK**.

Wenn Sie das einmal definierte Format erneut auf eine Zelle anwenden wollen, wiederholen Sie die Schritte 1 bis 2. Wählen Sie dann aus der Liste das neue Format. Excel zeigt es immer am Ende der Liste an. Benutzerdefinierte Formate können jederzeit wieder gelöscht werden. Markieren Sie das Format und benutzen Sie die Schaltfläche **Löschen**.

1.5.7 Formatcodes

Für die Darstellung der Zahlenformate verwendet Excel bestimmte Codezeichen, die in der folgenden Liste zusammengestellt sind. Dabei wird mit den verschiedenen Platzhaltern für Ziffern gesteuert, wie viele Dezimalstellen angezeigt werden. Excel erlaubt maximal 15 Dezimalstellen. Im Dialog **Zellen formatieren** können zwar für das Format **Zahl** bis zu 30 Dezimalstellen ausgewählt werden, ab der 16. Dezimalstelle erscheinen aber nur noch Nullen.

Auch die Behandlung führender oder nachfolgender Nullen wird mit Hilfe dieser Codezeichen geregelt. Vorzeichen können vor oder hinter die Zahl gesetzt werden. Excel zeigt normalerweise nur negative Vorzeichen an. Das Pluszeichen muss also ausdrücklich in das Formatmuster eingefügt werden, wenn es angezeigt werden soll.

Formatcode	Bedeutung
Standard	Anzeige im Standardformat
Kein Eintrag	Daten werden nicht angezeigt. Beispiel: ;;;.
#	Platzhalter für eine Ziffer. Führende Nullen vor und nachfolgende Nullen hinter dem Komma werden nicht angezeigt. Excel rundet auf die angegebenen Dezimalstellen. Sind vor dem Komma mehr Ziffern vorhanden als #-Zeichen, werden sie angezeigt.
0 (Null)	Platzhalter für eine Ziffer einschließlich führender Nullen vor und nachfolgender Nullen hinter dem Komma. Hat die Zahl weniger Ziffern als das Format, werden Nullen dafür angezeigt.
?	Platzhalter für eine Ziffer; führende Nullen vor und nachfolgende Nullen hinter dem Komma werden als Leerzeichen ausgegeben. Hat die Zahl weniger Ziffern als das Format, werden Leerzeichen dafür eingesetzt.
Komma	Dezimalzeichen
%	Prozentzeichen; Excel multipliziert den Wert mit 100 und fügt das Prozentzeichen hinzu.
Punkt	Tausendertrennzeichen; kann auch benutzt werden, um eine Zahl auf Tausender, Millionen etc. zu normieren. (#.. zeigt z. B. nur die Millionen an.)

Tabelle 1.6 Von Excel verwendete Codezeichen für die Darstellung von Zahlenformaten.

Formatcode	Bedeutung
E- E+ e- e+	Wissenschaftliches Zahlenformat; die Anzahl der Nullen bzw. #-Zeichen zur Rechten legt die Anzahl der Stellen im Exponenten fest. E- oder e- zeigt nur das Minuszeichen an, E+ oder e+ beide Vorzeichen des Exponenten.
€ $ – + / ( ) : Leerzeichen	Anzeige des betreffenden Zeichens. Sollen andere Zeichen angezeigt werden, setzen Sie sie in doppelte Anführungszeichen, oder Sie setzen \ davor. Der Bindestrich kann z. B. als Trennzeichen verwendet werden. Beispiel: ##-###-##.
*	Füllt die Zelle nach links mit dem unmittelbar folgenden Zeichen auf. ** kann z. B. als Sperrzeichen bei der Scheckausfüllung verwendet werden.
Unterstrich	Fügt eine Leerstelle von der Breite des nächsten Zeichens ein. _) kann z. B. benutzt werden, um positive Zahlen genau unter negative Zahlen zu setzen, die in Klammern angezeigt werden.
"Text"	Der Text innerhalb der Anführungszeichen wird angezeigt. Das erlaubt Ihnen z. B., die Maßeinheit direkt hinter eine Zahl zu setzen. Beispiel: #0,00 "qm".
@	Platzhalter für eine beliebig lange Zeichenfolge
M oder MM	Monatszahl ohne oder mit führenden Nullen (3 oder 03)
MMM	abgekürzter Monatsname (Jan, Feb etc.)
MMMM	ausgeschriebener Monatsname
MMMMM	Monatsname mit einem Buchstaben
T oder TT	Tag als Zahl ohne oder mit führenden Nullen (9 oder 09)
TTT	abgekürzter Tagesname (Sa, So)
TTTT	ausgeschriebener Tagesname
JJ oder JJJJ	zweistellige oder vierstellige Jahreszahl (93 oder 1993)
h oder hh	Stundenzahl ohne oder mit führenden Nullen (3 oder 03); wird im 12-Stunden-Format angezeigt, wenn das Format auch AM oder PM enthält.
m oder mm	Minutenzahl ohne oder mit führenden Nullen (3 oder 03); muss auf h oder hh folgen, wird sonst als Monatszahl interpretiert.

Tabelle 1.6 Von Excel verwendete Codezeichen für die Darstellung von Zahlenformaten. (Forts.)

Formatcode	Bedeutung
s oder ss	Sekundenzahl ohne oder mit führenden Nullen (3 oder 03)
[ ]	Erlaubt Zeitformate mit mehr als 24 Stunden bzw. mehr als 60 Minuten oder Sekunden. Dabei muss jeweils der äußerste linke Teil des Zeitformats in eckige Klammern gesetzt werden, z. B. [H]:mm:ss.
AM/am/A/a PM/pm/P/p	Bewirkt, dass die Uhrzeit im 12-Stunden-Format angezeigt wird: AM, am, A oder a für die Zeit bis Mittag (ante meridiem = vormittags), PM, pm, P, oder p für die Zeit bis Mitternacht (post meridiem = nachmittags). Ohne Angabe wird die Zeit im 24-Stunden-Format angezeigt.
[Farbe]	Die folgenden Zeichen werden in der angegebenen Farbe angezeigt. Möglich sind die Einträge SCHWARZ, BLAU, ZYAN, GRÜN, MAGENTA, ROT, WEISS, GELB oder FARBE n, wobei n für eine Zahl zwischen 0 und 56 steht und die Nummer der Farbe aus der Farbpalette angibt, die Excel verwendet.
[Bedingung Wert]	Kann für bedingte Formate genutzt werden. Als Bedingung kann einer der folgenden Operatoren eingegeben werden: <, >, =, >=, <=, <>. Wert steht für eine beliebige Zahl. Beispiel: >1000.

Tabelle 1.6 Von Excel verwendete Codezeichen für die Darstellung von Zahlenformaten. (Forts.)

1.5.8 Darstellung von Nullwerten

Bei Tabellen, die erst innerhalb größerer Zeiträume mit Daten gefüllt werden, stellt sich häufig das Problem der Nullwerte. Sie haben verschiedene Möglichkeiten:

Sie können für die gesamte Arbeitsmappe die Anzeige der Nullwerte unterdrücken. Dazu finden Sie über **Excel-Optionen** auf der Seite **Erweitert** die Arbeitsblattoption **In Zellen mit Nullwert eine Null anzeigen**. Wird das Häkchen gelöscht, werden Nullwerte nicht angezeigt. Die zweite Möglichkeit ist, ein Format zu definieren, bei dem die Nullwerte unterdrückt oder etwa durch einen Bindestrich ersetzt werden. Hier ein Beispiel: 0,00;-0,00;; oder 0,00;-0,00;"-". Das Format kann dann beispielsweise gezielt auf bestimmte Formeln beschränkt werden. Die dritte Möglichkeit ist, in die Zellen, für die noch die Werte fehlen, die Funktion =NV() einzutragen. In der Zelle erscheint #NV; auch die Formeln, die sich auf diese Zellen beziehen, liefern den Wert #NV.

Schließlich können Sie die Anzeige von Nullwerten natürlich auch über Formeln abfangen.

```
=WENN(ANZAHL(A3:E3) > 0;SUMME(A3:E3);"")
```

zeigt beispielsweise in Zelle F3 die Quersumme nicht an, wenn alle Zellen in A3:E3 leer sind.

1.5.9 Interpretation unvollständiger Jahreszahlen

Damit auch nach dem Jahr 2000 »unvollständige« Jahreszahlen vernünftig verarbeitet werden, bietet Ihnen Excel 2013 folgendes Standardverfahren an: Eine Eingabe wie `12.10.10` wird als `12.10.2010` verstanden. Dieses Verhalten gilt für alle Jahreszahlen von 00 bis 29. Geben Sie eine Zahl zwischen 30 und 99 an, vermutet Excel, dass das 20. Jahrhundert gemeint ist, und setzt die 19 als Jahrhundertzahl davor. `12.10.33` wird also zu `12.10.1933` ergänzt.

Excel zeigt die komplette Jahreszahl in der Bearbeitungsleiste automatisch an, wenn die Zelle ausgewählt wird. Wenn Sie eine Zelle nachträglich umformatieren, so dass die Jahreszahl auch tatsächlich vierstellig angezeigt wird, können Sie das Ergebnis ebenfalls prüfen. Intern speichert Excel also immer die komplette Jahreszahl. Benötigen Sie aus irgendeinem Grund ein Datum aus dem Jahr 1904, etwa das Geburtsdatum Ihrer Großmutter, können Sie das Datum vierstellig eingeben, auch wenn die Zelle nur für eine zweistellige Jahreszahl formatiert ist.

Eine Besonderheit in Bezug auf das Jahr 1900 soll an dieser Stelle noch erwähnt werden. Excel 2013 bietet neben dem vorgegebenen Dateiformat *Open XML* jetzt auch das Format *Strict Open XML* an, das den ISO-Standard für Open XML tatsächlich komplett umsetzt, während das bisherige Format Open XML hier aus Gründen der Rückwärtskompatibilität noch einige Lücken aufweist. In Strict Open XML wird nun auch ein Fehler bereinigt, der in allen Excel-Versionen mitgeschleppt worden ist. Das Jahr 1900, mit dem die Zählung der Jahre beginnt, wurde irrtümlich als Schaltjahr behandelt, es war also möglich, das Datum `29.02.1900` einzugeben. Wenn Sie das Strict-Format verwenden, wird die Eingabe `29.02.1900` nicht mehr als Datum akzeptiert und linksbündig als Zeichenfolge ausgegeben. Speichern Sie eine Open-XML-Datei mit dem falschen Datum erst später im Strict-Open-XML-Format, wird das Datum dagegen auf den 28.2.1900 korrigiert.

1.5.10 Ein Format für Zeitberechnungen

Das normalerweise von Excel verwendete Format für Zeitangaben liefert immer Werte, die eine bestimmte Uhrzeit darstellen. Die Zahl der Stunden bewegt sich folglich zwischen 0 und 24. Die Summe der Werte in Abbildung 1.25 würde normalerweise `4:30` ergeben, die Uhr läuft also weiter in den nächsten Tag, wenn 24 Stunden erreicht sind. Sollen Stundenwerte korrekt zusammenaddiert werden, müssen Sie dafür ein spezielles Format definieren: `[hh]:mm` statt `hh:mm`. Nun zeigt das Format auch Stundenzahlen an, die größer als 24 sind.

	A	B
1		
2	**Berechnung der Dauer**	
3		
4		**Übungsstunden**
5	**Mo**	2:30
6	**Di**	3:00
7	**Mi**	7:30
8	**Do**	8:00
9	**Fr**	7:30
10		28:30

Abbildung 1.25 Erst ein spezielles Zeitformat erlaubt das korrekte Addieren von Zeitwerten.

1.6 Formellose Berechnungen

Bevor auf die Konstruktion von Formeln im Detail eingegangen wird, soll an dieser Stelle zunächst noch auf eine Funktion von Excel hingewiesen werden, die Ihnen Rechenergebnisse ohne Formeln anbietet. Wenn Sie sofort wissen wollen, welchen Gesamtbetrag eine Gruppe von Werten ergibt, müssen Sie nicht eine vollständige Formel eingeben. Sobald Sie mehr als eine Zelle markieren, gibt Ihnen Excel unaufgefordert den Mittelwert, die Anzahl und die Summe dieser Werte in der Statusleiste aus. Das funktioniert nicht nur bei geschlossenen Zellblöcken, sondern auch, wenn Sie nicht zusammenhängende Zellen markieren. Das erspart in vielen Fällen den Zugriff auf den Taschenrechner.

Neben den drei vorgegebenen Auswertungen können über das Menü **Statusleiste anpassen** auch andere Funktionen aktiviert und berechnet werden. Dazu müssen Sie den Ergebnisbereich in der Statusleiste nur mit der rechten Maustaste anklicken und die gewünschte Funktion auswählen.

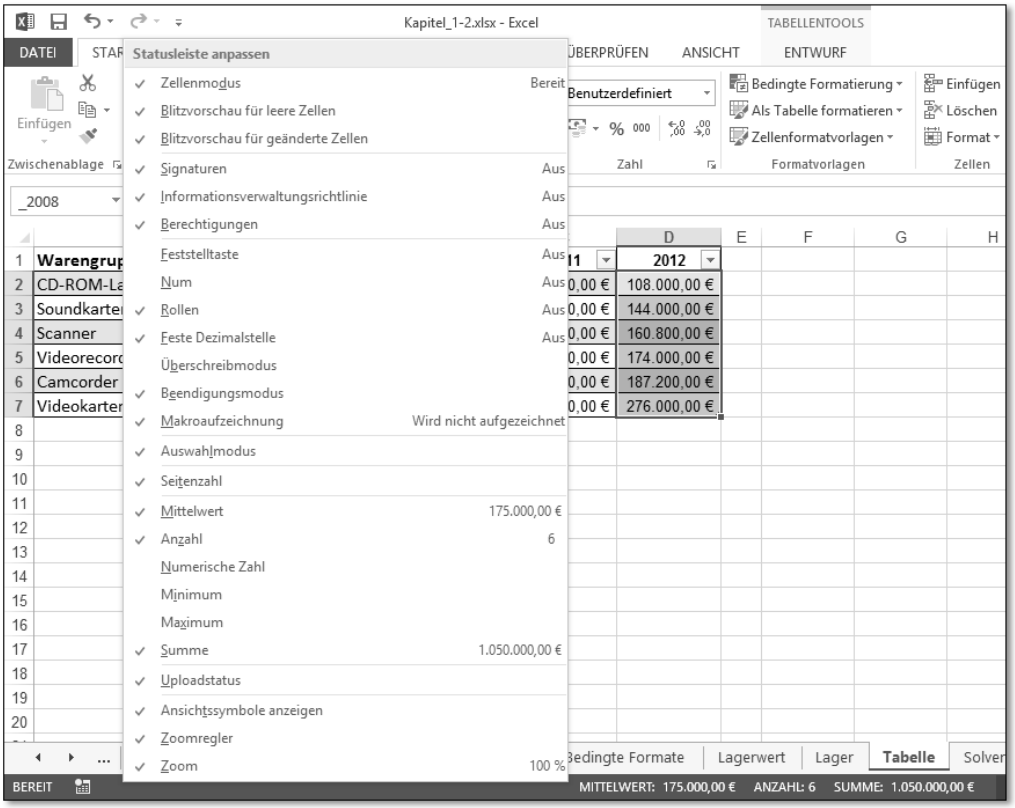


Abbildung 1.26 Mittelwert, Anzahl und Summe markierter Zellen und Wahl der Berechnungsart

Eine weitere Methode, Berechnungen ohne Formeln vorzunehmen, ist mit dem Kopieren verknüpft. Eine praktische Anwendung ist das Addieren von Werten in zwei Bereichen, um die Werte zusammenzufassen. Abbildung 1.27 zeigt ein Beispiel.

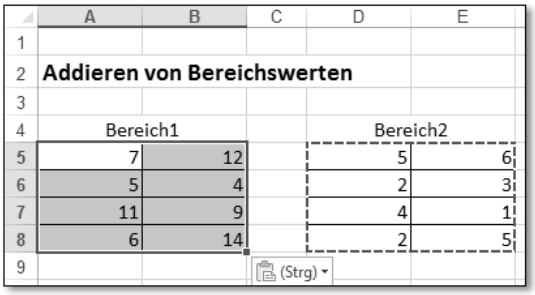


Abbildung 1.27 Aufaddieren von Bereichen beim Kopieren

Sie markieren die Zellen von Bereich2 und benutzen **Start • Zwischenablage • Kopieren.** Anschließend markieren Sie den Bereich A5:B8 und benutzen die untere Hälfte der Schaltfläche **Einfügen** in der Gruppe **Zwischenablage** und dort die Option **Inhalte einfügen.** Im Dialog wählen Sie die Option **Addieren**. Die Werte von Bereich2 werden auf die aktuellen Werte von Bereich1 aufaddiert.

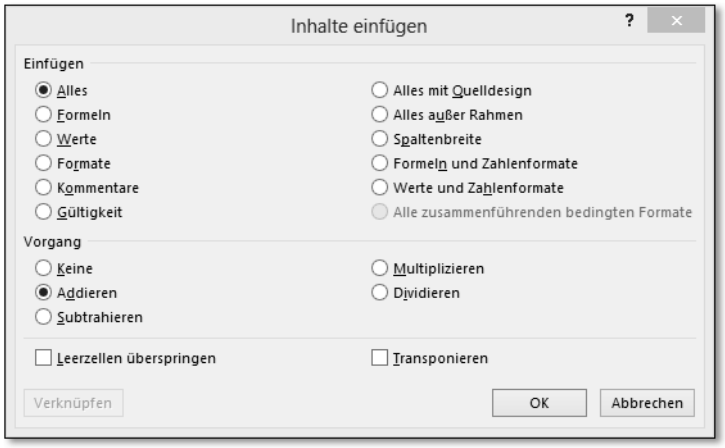


Abbildung 1.28 Aufaddieren von Bereichen beim Kopieren

Ähnlich können Sie vorgehen, wenn Sie einen Wertebereich mit einem bestimmten Faktor multiplizieren wollen. Sie kopieren den Faktor, wählen den Zielbereich und benutzen in dem Dialog **Inhalte einfügen** die Option **Multiplizieren**. Diese Methode ist bei der Änderung von Preislisten sehr handlich.

1.7 Einsatz von Formeln

Eine Formel ist in Excel so etwas wie ein Dauerauftrag an das Programm. Mit einer Formel bestimmen Sie, dass Excel immer wieder bestimmte Dinge erledigt. Das kann beispielsweise eine Rechenvorschrift sein wie »Bilde die Summe einer Zahlenkolonne« oder »Multipliziere den Betrag mit dem Faktor 1,19«, oder aber eine Anweisung wie »Setze eine Bezeichnung aus den aktuellen Werten von zwei Zellen zusammen«.

1.7.1 Formelsyntax

In Bezug auf die Syntax einer Formel ist zunächst nur festgelegt, dass sie immer mit einem Gleichheitszeichen beginnen muss. Das heißt umgekehrt auch, dass Excel jeden

Zelleintrag, der mit einem Gleichheitszeichen beginnt, als Formel interpretiert. Ist das nicht möglich, wird ein Fehlerwert ausgegeben. Allerdings erlaubt Excel, die Eingabe auch mit einem Plus- oder Minuszeichen zu beginnen; diese werden aber sofort in = oder =– umgewandelt, wenn es sich erkennbar um eine Formel handelt, also nicht nur ein einzelner Wert dem Vorzeichen folgt.

Ansonsten kann eine Formel aus verschiedenen Bestandteilen zusammengesetzt sein, die an dem folgenden Beispiel vorgestellt werden:

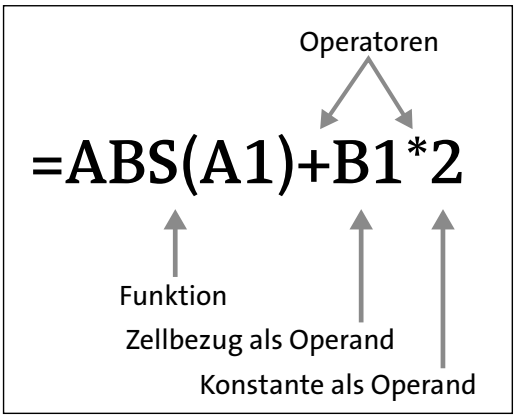


Abbildung 1.29 Beispiel für die Bestandteile einer Formel

Anders als bei der Arbeit mit einem Taschenrechner bestehen die Operanden in den Formeln in der Regel nicht aus fixen Werten, sondern aus Bezügen auf Zellen im Arbeitsblatt, sei dies nun über eine Zelladresse oder über einen entsprechenden Bereichsnamen. Wie solche Namen vergeben werden, wird in Abschnitt 1.17 im Detail beschrieben.

Ist die Formel einmal gebildet, steht sie anschließend gleichsam auf Abruf zur Verfügung. Ändern sich die Zahlen in der Kolonne oder der Betrag oder eine der Zeichenketten, liefern die Formeln sofort wieder das aktuelle Ergebnis.

Material für Formeln können nicht nur Zahlen und Zeichenfolgen, sondern auch selbst wiederum andere Formeln sein. Das heißt beispielsweise, mit dem Ergebnis einer Formel in der Zelle B12 wird in der Zelle F18 weitergerechnet. Die Zelle F18 ihrerseits gibt ihr Ergebnis an eine Formel in der Zelle H15 weiter. Auf diese Weise entstehen regelrechte Formelketten.

1.7.2 Formeltypen

Excel 2013 kennt verschiedene Typen von Formeln:

Arithmetische Formeln enthalten Konstanten, Zellbezüge und arithmetische Operatoren (+, –, *, /, %, ^). Diese Formeln errechnen das Ergebnis und zeigen es in der Zelle an, die die Formel enthält. Beispiele sind:

```
=B7+C7-D7
=8000 * 1,19
=C5/5
=B7 %
```

Das letzte Beispiel zeigt, dass der %-Operator auch direkt mit einem Zellbezug verknüpft werden kann. Der Wert in B7 wird dann durch 100 dividiert.

Zeichenfolgen-Formeln erlauben die Verknüpfung von zwei oder mehreren Zeichenfolgen mit dem &-Operator. Beispiele für solche Textverkettungen:

```
="Audio"&2006
="Farbe "&H4
="Margo "&"Lewin"
```

Logische Formeln enthalten Vergleiche zwischen Konstanten oder Zellbezügen mit Hilfe von Vergleichsoperatoren. Formeln wie `=Z8 > H5` ergeben den Wert `WAHR`, wenn die darin formulierte Beziehung der beiden Zellwerte tatsächlich wahr ist; wenn nicht, liefert eine solche Formel den Wert `FALSCH`. Der angezeigte Wert `WAHR` hat gleichzeitig den numerischen Wert `1`, der Wert `FALSCH` den numerischen Wert `0` (`=WAHR+1` ergibt also `2`).

Formeln, die Funktionen enthalten, z. B. die Summenfunktion:

```
=SUMME(F1:F30)
```

Es soll nicht unerwähnt bleiben, dass Excel algebraische Gleichungen mit Buchstaben für Unbekannte nicht direkt unterstützt. Es ist aber leicht möglich, einzelne Zellen mit den Namen dieser Unbekannten zu belegen und dann entsprechende Formeln aufzubauen, die diese Namen verwenden. Beispiele dazu finden Sie in Abschnitt 2.2, »Lösungen mit dem Solver suchen«.

Normalerweise liefert eine Formel nur einen Wert für die Zelle, in die sie eingetragen ist. Eine spezielle Variante sind Matrixformeln, die Werte für mehrere Zellen berechnen bzw. Wertegruppen gleichzeitig auswerten können. Mehr dazu in Abschnitt 1.18, »Matrixberechnungen«.

1.7.3 Datentypen

Die verschiedenen Formeltypen treten häufig gemischt auf. Dabei muss aber beachtet werden, dass die Ergebnisse der verschiedenen Teile einer Formel Daten liefern, die vom Typ her mit der weiteren Verwendung verträglich sind.

Zu dem Ergebnis einer Funktion, die beispielsweise eine Zeichenfolge liefert, kann nicht anschließend eine Zahl addiert werden. Das ist auch nicht sinnvoll. Eine Formel wie `=("Haus"&1)+1` ergibt nicht `Haus2`, sondern liefert den Fehlerwert `#WERT!` als Ergebnis.

Dagegen lässt der &-Operator auch Zahlen als Operanden zu, wandelt sie aber automatisch in Zeichenfolgen um. `=1111&2222` ergibt die Zeichenfolge `11112222`.

Umgekehrt wandelt Excel Funktionsargumente, die als Zahlen in Anführungszeichen eingegeben werden – etwa `=COS("1")` –, automatisch in Zahlen um, wenn ein numerisches Argument erwartet wird und sich die Zeichenfolge als gültige Zahl verstehen lässt. Die Ausnahme dieser Regel sind die IST-Funktionen, bei denen "1" als Text behandelt wird.

1.8 Arbeit mit Operatoren

Die einfachsten Formeln, die in einer Zelle abgelegt werden können, sind solche, in denen Zahlen mit den Operatoren der Grundrechenarten verknüpft werden. Die Schreibweise entspricht abgesehen von dem vorangestellten Gleichheitszeichen dem aus der Schulmathematik gewohnten Bild:

```
=13+25-5
=25*4/3
=3^2
```

Anstelle von Zahlen, also von Konstanten, können Variablen in die Formeln eingefügt werden. Wenn Sie mit Excel arbeiten, dienen die Zell- oder Bereichsadressen und die Bereichsnamen dazu, Variablen in eine Formel einzubeziehen.

Die Operatoren in einer Formel werden entsprechend ihrer Priorität behandelt. Die folgende Tabelle zeigt die Rangfolge der Priorität. Die höchste Priorität ist 1.

Operator	Beispiel	Bedeutung	Priorität
Bezugsoperatoren			
:	B3:B7	Bereich	1
Leer	B3:E8 C4:F12	Schnittmenge	2
;	B3:B12;C3:C12	Vereinigung	3
Arithmetische Operatoren			
-	-(5*2)	Vorzeichen	4
%	10%	Prozent	5
^	5^2	Potenzierung	6
*	5*6	Multiplikation	7
/	6/3	Division	7
+	7+4	Addition	8
-	4-2	Subtraktion	8
Verkettungsoperator			
&	B2&B3	Textverkettung	9
Vergleichsoperatoren			
=	B5=B7	gleich	10
>	B5>B7	größer	10
<	B5<B7	kleiner	10
<>	B5<>B7	ungleich	10
<=	B5<=B7	kleiner/gleich	10
>=	B5>=B7	größer/gleich	10

Tabelle 1.7 Priorität von Operatoren innerhalb einer Formel

Die Priorität bestimmt die Reihenfolge, in der Excel die Operatoren auswertet. Nehmen Sie das folgende Beispiel: Ein numerischer Ausdruck lautet `=5+6*3-7`. Würde das Programm schlicht von links nach rechts vorgehen, ergäbe sich `5+6=11`, `11*3=33` und schließlich `33-7=26`. Tatsächlich aber führt Excel 2013 korrekterweise zunächst die Multiplikation durch, die Priorität vor der Addition und Subtraktion hat: `6*3=18` und `5+18-7=16`.

Die Operation mit der höheren Priorität wird also zuerst ausgeführt. Bei Operatoren derselben Priorität wird von links nach rechts gearbeitet. Soll die Rangfolge der Prioritäten unterlaufen werden, kann wie üblich mit Klammern gearbeitet werden. Bei `=(5*6)*(3-7)` wird zunächst das Innere der beiden Klammern berechnet. Das ergibt `=30*-4`, also `=-120`.

1.9 Hinweise zu den Grundrechenarten

Im Folgenden werden zunächst noch einige Hinweise zu den Grundrechenarten gegeben, die in der Praxis vermutlich am häufigsten in den Formeln verwendet werden.

1.9.1 Addition und Subtraktion

Die folgende Liste zeigt einige Beispiele für korrekt eingegebene Formeln, um die Schreibweise zu demonstrieren. Wie Sie sehen, können auch Datumswerte als Operanden verwendet werden:

```
=12+27
=-15+33
=C17+F18-J21
=B14
=-(B13+B14+C17)*1,19
=SUMME(B12:F12)+1000
=SUMME(B12:B17)+SUMME(C12:C19)
=DATUM(12;12;01)+90
=SUMME11+SUMME12
```

Das letzte Beispiel verwendet Bereichsnamen. Es ist allerdings nur dann zulässig, wenn mit den beiden Namen jeweils nur eine Zelle benannt wird. Ist ein Bereich aus mehreren Zellen benannt, liefert diese Formel die Fehlermeldung `#WERT!`.

1.9.2 Multiplikation und Division

Bei der Multiplikation und Division muss beachtet werden, wie sich das Ergebnis in Bezug auf die Dezimalstellen verhält. Bei der Multiplikation addieren sich die Nachkommastellen: `=33,33*33,33` ergibt `1110,8889`. Wird die Darstellung des Ergebnisses durch die Zellformatierung auf zwei Dezimalstellen reduziert, erscheint zwar in der Zelle ein gerundeter Wert, im Beispiel also `1110,89`, intern bleibt aber das genauere Ergebnis gespeichert.

Wird auf diese Zelle in einer anderen Formel Bezug genommen, rechnet diese Formel also mit den vier Dezimalstellen weiter. Werden solche Werte dann erneut multipliziert, kann es zu deutlichen Differenzen kommen, je nachdem, ob mit den intern gespeicherten Werten weitergerechnet wird oder mit den gerundeten Werten. Abbildung 1.30 zeigt ein kleines Beispiel.

	A	B	C	D	E
1					
2	**Rundungsdifferenzen:**				
3					
4	**Formel:**	**Ergebnis:**	**gerundet:**		
5					
6	=12,13*12,6	152,838	152,84	=B11*17	**10598,71**
7	=13,87*32,77	454,5199	454,52	=C11*17	**10598,82**
8	=3,66*2,66	9,7356	9,74		
9	=2,12*3	6,36	6,36		
10	=SUMME(B6:B9)	623,4535	623,46		
11			▲		

Abbildung 1.30 Beispiel für Rundungsdifferenzen

In der Spalte C sind die Ergebnisse der benachbarten Spalte B mit der Funktion `RUNDEN()` bearbeitet worden. In E6 ist das nicht gerundete Gesamtergebnis mit 17 multipliziert, in E7 das gerundete Gesamtergebnis. Die Differenz ist deutlich.

1.9.3 Division durch null abfangen

Bei der Division kann die Anzahl der Dezimalstellen, die der Quotient erhält, ganz unterschiedlich sein. Hier kommt zu dem bei der Multiplikation angesprochenen Rundungsproblem ein weiteres hinzu, das unbedingt beachtet werden muss: Die Division durch null ist nicht erlaubt.

Solange der Divisor direkt eingegeben wird, lässt sich dieser Fehler relativ leicht vermeiden. Was aber, wenn der Divisor eine Zelladresse oder eine Funktion ist, deren Ergebnis unvorhersehbar ist, wenn also der Divisor eine Variable, eine Unbekannte ist? Der Fall, dass ein Divisor mit dem Wert 0 verwendet wird, sollte unterbunden werden. Das ist mit der in Kapitel 13, »Logische Funktionen«, ausführlich beschriebenen `WENN()`-Funktion möglich. Das kann etwa so aussehen:

```
=WENN(B3<>0;RUNDEN(A3/B3;2);"")
```

Wenn der Wert der Zelle B3 ungleich 0 ist, wird die Division `A3/B3` durchgeführt; ist B3 aber tatsächlich gleich 0, findet die Division erst gar nicht statt, die Zelle bleibt leer. Stattdessen könnte auch ein entsprechender Hinweis ausgegeben werden.

1.10 Texte verketten

Gelegentlich ist es sinnvoll, eine Zeichenfolge in einer Zelle durch eine Formel zu erzeugen, die verschiedene Zeichen oder Zeichenfolgen verknüpft. Angenommen, Sie wollen Artikelnummern um zwei Zeichen erweitern, die die Warengruppe beinhalten. Wenn Sie in die Zelle C9 `=C5&C8` eintragen, wird der Inhalt von Zelle C5 mit dem Inhalt von Zelle C8 verkettet. Wenn C5 das Warengruppenkennzeichen `PX` enthält und C8 die Artikelnummer `3370086`, ist das Ergebnis in Zelle C9 `PX3370086`. Es macht übrigens nichts aus, wenn die Artikelnummer in C8 als Zahl und nicht als Zeichenfolge eingetragen worden ist. Der Operator wandelt die Zahl automatisch in eine Zeichenfolge um.

Anschließend können Sie diese Formel noch in ihr Ergebnis verwandeln. Kopieren Sie die Zellen in die Zwischenablage, und fügen Sie sie an dieselbe Stelle wieder ein, indem Sie über das Kontextmenü **Inhalte einfügen** aufrufen und die Option **Werte** verwenden. Um zwischen die Textelemente ein Leerzeichen einzufügen, schreiben Sie `=C5&" "&C8`.

1.11 Hinweise zu logischen Formeln

Logische Formeln werden benutzt, um zu prüfen, ob bestimmte Tatsachen oder Bedingungen gegeben sind. Wenn Sie prüfen wollen, ob ein bestimmter Grenzwert in Zelle B4 überschritten ist, können Sie diese Bedingung als logische Formel formulieren: `=B4>80`. Excel prüft, ob dies der Fall ist. Wenn ja, zeigt die Zelle den Wert `WAHR`. Die Zelle hat zugleich den numerischen Wert `1`. Ist die Bedingung nicht erfüllt, erscheint in der Zelle `FALSCH`. Das entspricht dem numerischen Wert `0`.

Das ist ganz praktisch für Prüfsummen. Wenn Sie z. B. in drei Zellen untereinander Bedingungen in Form von logischen Formeln ablegen, können Sie leicht feststellen, ob alle Bedingungen erfüllt sind. Sie addieren einfach die Wahrheitswerte. Ist das Ergebnis `=3`, sind alle Bedingungen erfüllt.

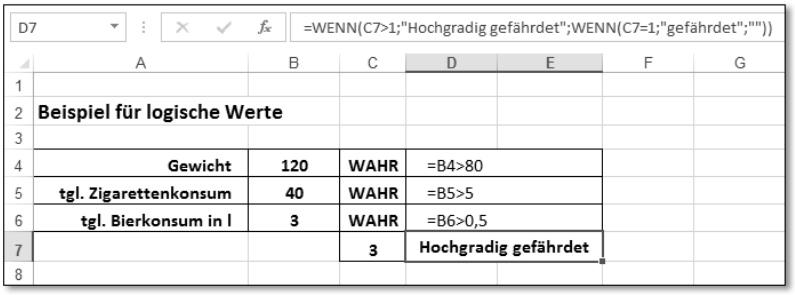


Abbildung 1.31 Logische Werte lassen sich auch addieren.

1.12 Formeln mit Bezugsoperatoren

Excel kann Bereichsbezüge in Formeln auch verknüpfen bzw. Schnittmengen von Bereichen bilden. Die Bereichsverknüpfung erfolgt mit dem Semikolon. In der Formel `=SUMME(B4:B12;F4:F12)` wird die Gesamtsumme der beiden getrennten Zellbereiche geliefert.

Um die Schnittmenge zweier Bereiche für eine Formel zu erhalten, kann mit dem Schnittmengenoperator, dem Leerzeichen, gearbeitet werden. Abbildung 1.32 zeigt ein Beispiel für die Verwendung einer Schnittmenge. Die Schnittmenge ist durch eine andere Farbe hervorgehoben.

Schnittmengenbezüge sind interessant, wenn auf Teilmengen in einer Tabelle zugegriffen werden soll. Die Arbeit mit Schnittmengen ist besonders effektiv, wenn die Bereiche selbst vorher benannt wurden. Darauf wird in Abschnitt 1.17 noch näher eingegangen.

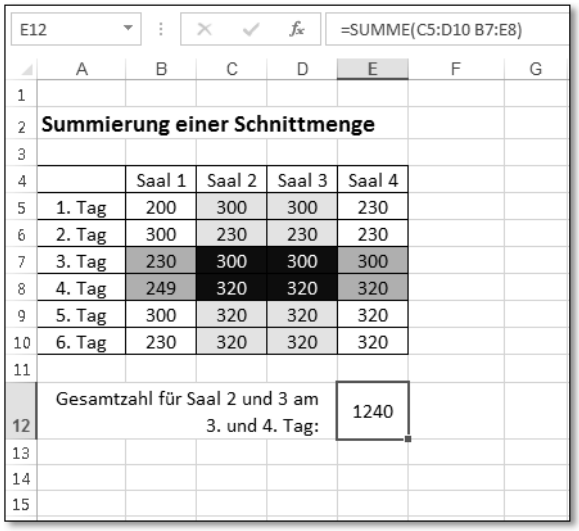


Abbildung 1.32 Beispiel für die Summierung einer Schnittmenge

Statt der in Abbildung 1.32 verwendeten Formel könnte auch eine Formel benutzt werden, die nur die Zeilennummern und die Spaltenbeschriftungen verwendet. Der Bereich wäre dann die Schnittmenge zweier kompletter Spalten und Zeilen. `=Summe(C:D 7:8)` liefert dasselbe Ergebnis wie die abgebildete Formel.

1.13 Tabellenfunktionen

Tabellenfunktionen sind spezielle Ausdrücke, die entweder direkt als Formel oder als Teil einer Formel im Tabellenblatt verwendet werden. Auch wenn eine Formel nur aus einer Funktion besteht, muss sie wie üblich mit einem Gleichheitszeichen beginnen. Im zweiten Fall werden die Funktionen mit Hilfe von Operatoren mit den anderen Teilen einer Formel verknüpft.

Die folgenden Formeln `=A3-SUMME(A10:A20)` oder `=B4/RUNDEN(C6;2)` sind gültige Ausdrücke, während in dem nächsten Beispiel ein Operator fehlt: `=MITTELWERT(kosten)G6`.

Funktionen ersetzen zum Teil komplexe Berechnungen, wobei Sie sich um die Art und Weise der Berechnung aber nicht zu kümmern brauchen. Sie füttern die Funktionen nur mit den notwendigen Argumenten, den Rest erledigt das Programm. Funktionen können geschachtelt werden, d.h., das Argument einer Funktion kann selbst wieder eine Funktion sein. Seit Excel 2007 ist eine Verschachtelungstiefe von bis zu 64 Ebenen erlaubt, in den älteren Versionen sind maximal 7 Ebenen möglich.

Tabellenfunktionen können auch von Makros verwendet werden, allerdings müssen dabei die englischen Funktionsnamen benutzt werden. Mehr zu diesem Thema finden Sie in Kapitel 18, »Entwicklung eigener Funktionen«.

Außerdem ist es möglich, Funktionen mit eigenen Namen zu belegen und sie wie Namen in Formeln einzufügen. Dazu wird im Dialog für die Definition eines Namens unter **Bezieht sich auf**: statt eines Zell- oder Bereichsbezugs die komplette Funktion eingetragen.

In den Kapiteln 3 bis 15 werden alle Tabellenfunktionen im Detail beschrieben und Beispiele dafür vorgestellt. In diesem Abschnitt soll es darum gehen, einige allgemeine Merkmale von Funktionen darzustellen und Hinweise zur Arbeit mit Funktionen zu geben.

1.13.1 Funktionsergebnisse und Datentypen

Excel 2013 wartet mit einer überwältigenden Fülle von Funktionen auf, die aus einem oder mehreren Ausgangswerten neue Werte ermitteln, und das teilweise auf eine recht komplexe Weise.

Eine Funktion ist eine Art von »Blackbox«, bei der auf der einen Seite etwas eingegeben wird und auf der anderen Seite etwas herauskommt. Ohne dass der Anwender das oft

sehr komplizierte Formelwerk für den Betrieb der Funktion kennen muss, wird ihm nach Eingabe der notwendigen Werte das Ergebnis der Operationen ausgegeben.

Jede Funktion liefert ein Ergebnis. Je nach Funktion kann dieses Ergebnis numerisch sein, also ein bestimmtes Rechenergebnis, eine Zeichenfolge, ein Text oder ein Wahrheitswert (`WAHR` oder `FALSCH`). Zuweilen ist das Ergebnis nicht nur ein Wert, sondern eine Matrix (Array) von Werten. Der Typ des Ergebnisses einer Funktion muss berücksichtigt werden, wenn in einer weiteren Formel auf diese Funktion Bezug genommen wird. Ansonsten kommt es eventuell zu einem Fehler, wenn die Datentypen nicht zusammenpassen. In der Regel erscheint dann der Fehlerwert `#WERT!`.

1.13.2 Aufbau und Einsatz von Funktionen

Eine Funktion benutzt in der Regel Werte, die als Argumente bereitgestellt werden, um aus ihnen andere Werte zu ermitteln. Dabei kann es sich um einfache oder komplexe Berechnungen handeln, aber auch um logische Analysen, die Zerlegung von Zeichenketten und dergleichen.

Den generellen Aufbau einer Funktion zeigt Abbildung 1.33 am Beispiel der `ZINS()`-Funktion:

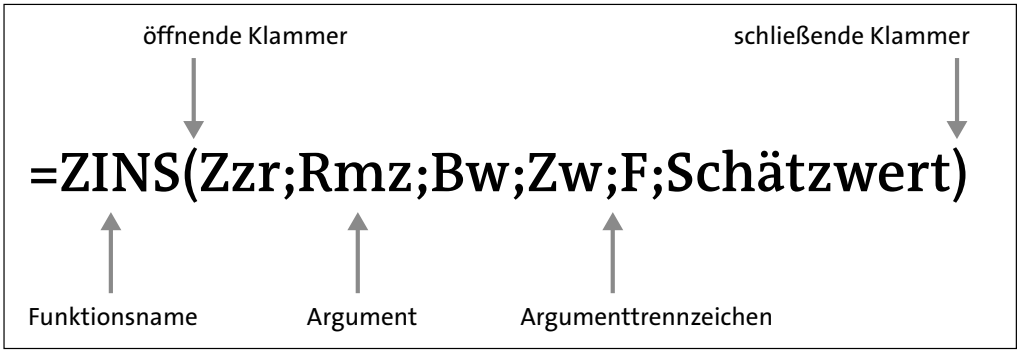


Abbildung 1.33 Die Komponenten einer Funktion

Das Gleichheitszeichen ist nötig, wenn die Funktion am Anfang einer Formel steht oder eine Formel nur aus einer Funktion besteht. Wird die Funktion innerhalb einer Formel an einer anderen Stelle verwendet, geht der Funktion ein Operator voraus.

Als Argumente einer Funktion kommen in Frage:

Argument	Beschreibung
Konstanten	Diese Werte werden direkt innerhalb der Klammer für die Argumente eingetragen.
Bezüge auf Zellen oder Bereiche	Die Werte, mit denen die Funktion arbeiten soll, sind bereits in Zellen oder Bereichen der Tabelle enthalten oder sollen dort eingetragen werden. Der Bezug darauf wird über Zell- oder Bereichsadressen hergestellt.
Bereichsnamen	Die Werte, mit denen die Funktion arbeiten soll, sind bereits in Zellen oder Bereichen der Tabelle enthalten oder sollen dort eingetragen werden. Der Bezug darauf wird über die Namen hergestellt, die für die entsprechenden Zellen oder Bereiche definiert wurden.
Funktionen	Als Argumente in einer Funktion können selbst wieder Funktionen verwendet werden, die ihrerseits diejenigen Werte liefern, mit denen die Funktion arbeiten soll. In diesem Fall werden Funktionen also verschachtelt.

Tabelle 1.8 Mögliche Argumente einer Funktion

Der Datentyp der Argumente muss dem entsprechen, was die Funktion an der entsprechenden Stelle erwartet. Die Argumente werden durch Semikola getrennt. Konstanten, die Texte enthalten, müssen in doppelte Anführungszeichen gesetzt werden.

Die Reihenfolge der Argumente muss eingehalten werden, es sei denn, es handelt sich um Argumente desselben Typs. Bei der Funktion `=SUMME()` können beispielsweise seit Excel 2007 bis zu 255 Argumente angegeben werden, die Konstanten, Zellbezüge oder Bereichsbezüge darstellen. Ihre Reihenfolge ist in diesem Fall beliebig.

Bei einer Anzahl von Funktionen werden die Argumente noch unterteilt in solche, die erforderlich sind, und solche, die optional sind, also nicht zwangsläufig angegeben werden müssen. Das heißt aber nicht, dass die Funktion ohne die optionalen Argumente in jedem Fall arbeiten würde, sondern dass sie, wenn diese Argumente nicht angegeben werden, vordefinierte Werte für diese Argumente verwendet.

Folgen auf ein nicht verwendetes optionales Argument weitere Argumente, so ist für jedes ausgelassene Argument ein Semikolon zu setzen. Im Dialog für die Eingabe der Funktionsargumente und in den QuickInfos werden die erforderlichen Argumente in fetter Schrift angezeigt. (In den Funktionsreferenzen in den Kapiteln 3 bis 15 werden die erforderlichen und die optionalen Argumente ebenfalls durch unterschiedliche Formatierung gekennzeichnet.)

Einige wenige Funktionen (z.B. `PI()`, `HEUTE()`, `JETZT()`) benötigen keine Argumente. Trotzdem müssen die Klammern immer mitgeschrieben werden, damit Excel die Funktion als solche erkennen kann.

1.14 Eingabe von Formeln und Funktionen

Um fehlerfreie Formeln zu produzieren, sind einige Regeln zu beachten:

- Das erste Zeichen einer Formel ist immer das Gleichheitszeichen. Es ist zwar erlaubt, stattdessen mit einem Plus- oder Minuszeichen zu beginnen, aber Excel setzt automatisch ein Gleichheitszeichen an die erste Stelle, wenn nicht nur ein einzelner Wert eingegeben wird. Das Pluszeichen wird weggelassen.
- Konstante Zeichenfolgen müssen in doppelte Anführungszeichen gesetzt werden.
- Formeln können zwar durch Leerzeichen lesbarer gemacht werden, das Leerzeichen darf aber nicht zwischen Funktionsnamen und erster Klammer auftauchen.

Excel 2013 erleichtert die Eingabe umfangreicher Formeln dadurch, dass die Bearbeitungsleiste automatisch um die dafür benötigten Zeilen erweitert wird. Mit Hilfe der Schaltfläche mit der Pfeilspitze am Ende der Leiste können die zusätzlichen Zeilen aus- und eingeblendet werden.

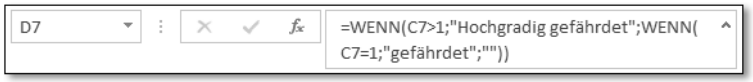


Abbildung 1.34 Erweiterte Bearbeitungsleiste mit umfangreicher Formel

Um sehr lange Formeln übersichtlicher zu gestalten, können Sie außerdem Zeilenschaltungen benutzen. Alt + ↵ erzeugt eine Zeilenschaltung.

1.14.1 Konstanten in Formeln

Wenn Zahlen als Konstanten in einer Formel vorkommen, müssen Sie beachten, dass, im Unterschied zur direkten Eingabe von Zahlen in eine Zelle, Klammern für negative Zahlen, Punkte für die Tausenderabtrennung und Währungszeichen nicht erlaubt sind.

Ein Datum oder eine Zeitangabe kann in einer Formel über eine entsprechende Funktion erzeugt oder als Konstante eingegeben werden. Im zweiten Fall müssen Sie die Werte aber in doppelte Anführungszeichen setzen. `="6/10/98"+100` liefert etwa ein Datum, 100 Tage nach dem 6.10.98.

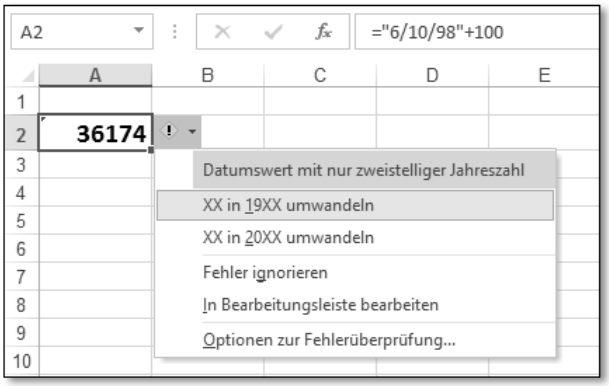


Abbildung 1.35 Menü zu einer zweistelligen Jahreszahl

Da die obige Eingabe eine nur zweistellige Jahreszahl enthält, bietet Excel 2013 in diesem Fall sofort eine Schaltfläche mit einem Menü an, das Ihnen erlaubt, die Jahreszahl in eine vierstellige Zahl umzuwandeln und dabei eindeutig einem Jahrhundert zuzuordnen.

Textkonstanten müssen in einer Formel zwar immer in doppelten Anführungszeichen eingegeben werden. Wenn Sie allerdings den Dialog **Funktion einfügen** benutzen, werden die Anführungszeichen automatisch gesetzt, falls Sie eine Zeichenfolge in ein Argumentfeld eintragen.

1.14.2 Eingabe von Bezügen

Bezüge in einer Formel können entweder manuell eingegeben werden oder durch Markierung der Zelle oder des Bereichs, sei es mit Maus, Finger oder Stift oder mit der Tastatur. Im zweiten Fall schaltet Excel automatisch in den **Zeigen**-Modus. In diesem Modus können Sie so lange auswählen, bis Sie den nächsten Operator eintragen.

Nehmen Sie als Beispiel folgende Formel für die Berechnung einer Relation: `=B2/B3`. Wenn Sie bei der Eingabe den **Zeigen**-Modus ausnutzen wollen, geben Sie zunächst das Gleichheitszeichen ein. Dann können Sie die Zelle B2 anklicken oder antippen. Die Zelladresse erscheint in der Zelle und in der Bearbeitungsleiste. Wenn Sie irrtümlich die falsche Zelle ausgewählt haben, wählen Sie noch einmal. Tippen Sie dann das Divisionszeichen. Wählen Sie anschließend die Zelle B3 aus. Bestätigen Sie die Formel mit dem Häkchen in der Bearbeitungsleiste. Ist die Formel misslungen, verwerfen Sie die Eingabe durch einen Klick oder Tipp auf das Storniersymbol mit dem schrägen Andreaskreuz.

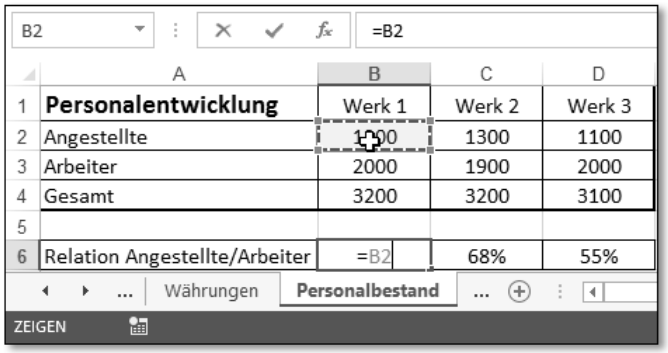


Abbildung 1.36 Zeigen auf einen Bereich bei der Formeleingabe

Die Technik mit der Tastatur ist ähnlich. Geben Sie = ein. Sobald Sie irgendeine Richtungstaste drücken, wechselt das Programm in den **Zeigen**-Modus. Markieren Sie mit den Richtungstasten die Zelle B2. Geben Sie das Divisionszeichen ein usw. Quittieren Sie die fertige Formel mit ↵. Ist die Formel nicht in Ordnung, brechen Sie die Eingabe mit Esc ab.

Beachten Sie dabei, dass Sie, solange Sie sich im **Zeigen**-Modus befinden, die Formeleingabe weder mit den Richtungstasten noch per Klick oder Tipp auf eine andere Zelle beenden können, wie es bei der Eingabe von Werten der Fall ist.

1.14.3 Operatoren über Symbole eingeben

Wenn es lästig ist, bei der Eingabe von Formeln häufig zwischen Tastatur und Maus, Finger oder Stift zu wechseln, lässt sich die Arbeitsumgebung auch so einrichten, dass die Operatoren als Symbole angeboten und per Klick oder Tipp ausgewählt werden können. In den älteren Excel-Versionen war es praktisch, über **Ansicht • Symbolleisten • Anpassen** dazu eine eigene Symbolleiste anzulegen und über das Register **Befehle** dann die benötigten Symbole aus der Kategorie **Einfügen** in diese Leiste zu ziehen.

In Excel 2013 lassen sich die Operatorsymbole in die Schnellzugriffsleiste ziehen, die dann am besten unterhalb des Menübands angeordnet wird. Verwenden Sie dazu im Dialog **Excel-Optionen** unter **Symbolleiste für den Schnellzugriff** und **Befehle auswählen** die Option **Nicht im Menüband enthaltene Befehle**, und übernehmen Sie aus der Liste die benötigten Symbole mit **Hinzufügen**.

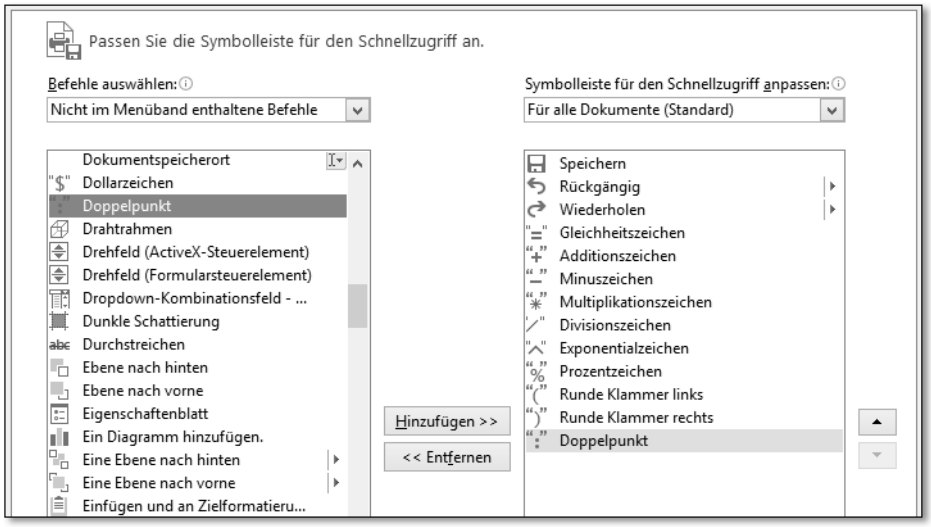


Abbildung 1.37 Operatoren lassen sich auch per Klick oder Tipp eingeben, wenn sie in die Schnellzugriffsleiste eingefügt werden.

Mit den Operatoren in der Symbolleiste lassen sich dann Formeln fast komplett per Klick oder Tipp editieren. In der deutschen Version fehlt nur ein Symbol für das Semikolon, das für die Trennung von Funktionsargumenten benötigt wird.

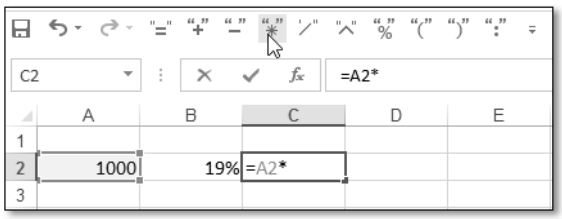


Abbildung 1.38 Die Schnellzugriffsleiste mit den Operatorsymbolen

1.14.4 Bereichsangaben

Wenn Sie einen Bereichsbezug als Argument in einer Formel benötigen, haben Sie zwei Möglichkeiten:

Anstatt mit der Maus nur eine Zelle anzuklicken, markieren Sie durch Ziehen der Maus den Bereich, den Sie benutzen wollen. Wenn Sie mit der Tastatur arbeiten, müssen Sie zunächst wieder eine Richtungstaste anschlagen, um in den **Zeigen**-Modus zu gelangen. Markieren Sie die Zelle, in der der Bereich beginnen soll. Halten Sie dann die ⇧-Taste

gedrückt, während Sie den Bereich mit den Richtungstasten markieren. Beenden Sie die Bereichsmarkierung mit der ↵-Taste. Soll die Formel dagegen noch fortgesetzt werden, beenden Sie die Bereichsmarkierung mit einer schließenden Klammer oder einem Operator.

Wenn ein großer Bereich mit der Maus markiert werden soll, ist eine andere Technik allerdings meist effektiver. Sie klicken zunächst die erste Eckzelle des Bereichs an und halten dann beim Anklicken der gegenüberliegenden Eckzelle gleichzeitig die ⇧-Taste gedrückt. Ist die Markierung nicht ganz korrekt, klicken Sie erneut mit gedrückter ⇧-Taste auf die richtige Stelle. Diese Technik ist insbesondere bei Bereichen sinnvoll, bei denen Sie, um die letzte Zelle sichtbar zu machen, zunächst mit den Bildlaufleisten den Bildausschnitt verschieben müssen.

Eine weitere Methode ist, die erste Eckzelle auszuwählen, F8 zu drücken – in der Statuszeile erscheint **Auswahl erweitern** – und anschließend die letzte Eckzelle zu wählen. Sie müssen noch einmal F8 tippen, um den Erweiterungsmodus wieder aufzuheben.

Um einen Bereich mit Finger oder Stift zu markieren, tippen Sie die linke obere Eckzelle an, und ziehen Sie anschließend die dann eingeblendeten runden Anfasser in die entsprechende Richtung. Auch im Modus **Fingereingabe** haben Sie – wie gerade beschrieben – die Alternative, mit F8 zu arbeiten.

Wenn ein Bereich benannt ist, tippen Sie zunächst beispielsweise =SUMME(ein. Nun können Sie den gewünschten Bereichsnamen mit F3 einfügen. Die F3-Taste öffnet das Dialogfeld **Namen einfügen,** in dem Sie den Namen doppelt anklicken oder antippen. Stattdessen können Sie auch mit **Formeln • Definierte Namen • In Formel verwenden** die Liste der vergebenen Namen einblenden und den benötigten Namen per Klick oder Tipp einfügen.

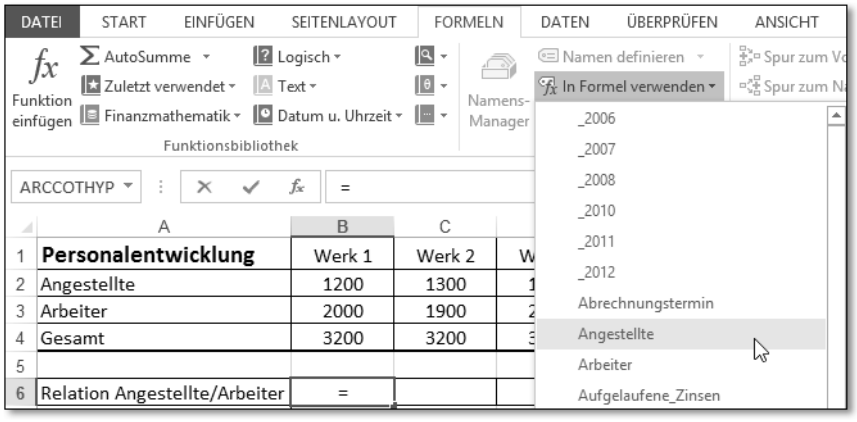


Abbildung 1.39 Übernahme von Namen in Formeln

Sollen sich Formeln auf andere Blätter innerhalb derselben Arbeitsmappe beziehen, muss jeweils vor die Bereichsadressen oder -namen der Blattname gesetzt werden, getrennt durch ein Ausrufezeichen. Wenn Sie im **Zeigen**-Modus zunächst das Blatt und dann den jeweiligen Bereich auswählen, wird der Blattname automatisch in die Formel übernommen.

1.14.5 3D-Bezüge

Eine spezielle Möglichkeit der Arbeitsmappen sind 3D-Formeln. Ein Beispiel wäre die Summierung der Werte, die in vier Tabellenblättern – Werk1 bis Werk4 – jeweils in der Zelle B7 stehen. Die Schreibweise lautet: `=SUMME(Werk1:Werk4!B7)`. Enthält der Name eines Blatts Leerzeichen, muss der Name in einzelnen Anführungszeichen eingefasst werden:

```
=SUMME('Werk Bonn:Werk Berlin'!B7)
```

Wird ein Bezug auf einen Bereich benötigt, ist die Syntax folgende:

```
Erstes_Blatt:Letztes_Blatt!Linke_obere_Bereichsecke:Rechte_untere_Bereichsecke
```

Also beispielsweise:

```
=SUMME(Werk1:Werk4!B7:F27)
```

Wenn Sie einen 3D-Bezug durch Zeigen mit der Maus eingeben wollen, müssen Sie zunächst die Blätter auswählen und dann die Zelle oder den Bereich. Klicken Sie auf das Register des ersten Blatts, das zu dem Bereich gehören soll, dann mit gedrückter ⇧-Taste auf das letzte Blatt des Bereichs. Danach markieren Sie die Zelle oder den Bereich.

Vergessen Sie nicht, anschließend die Blattgruppe durch Auswahl eines einzelnen Blatts wieder aufzulösen, weil sonst alle späteren Eingaben auf dem aktuellen Blatt auf allen anderen Blättern der Gruppe wiederholt werden.

1.14.6 Externe Bezüge

Eine Formel kann sich nicht nur auf Zellen im selben Tabellenblatt oder auf ein anderes Tabellenblatt in derselben Arbeitsmappe beziehen. Auch Bezüge auf Werte in anderen Arbeitsmappen sind möglich. Soll sich eine Formel auf Werte in einer anderen Arbeitsmappe beziehen, ist es in der Regel am besten, diese Mappe ebenfalls zu öffnen. Um in einer Formel dann auf Zellen dieser zweiten Tabelle Bezug zu nehmen, müssen Sie zwi-

schen den Fenstern wechseln. Dann können Sie während der Eingabe in der Bearbeitungsleiste das andere Fenster aktivieren und die Zellbezüge markieren.

Wenn Sie in einer Tabelle eine Formel entwickeln, die sich auf Daten aus einer anderen Arbeitsmappe bezieht, gilt folgende Schreibweise: `=[MAPPE1]Tabelle1!$E$2`. Der Mappenname muss also in eckigen Klammern eingeschlossen sein. Der Name des Tabellenblatts wird mit einem Ausrufezeichen von der Zelladresse oder einem Bereichsnamen abgetrennt.

Die Schreibweise bei Bereichen ist vereinfacht, der Dateiname und der Blattname müssen nur einmal eingegeben werden.

```
=SUMME([QUELLE.xlsx]Tabelle1!$G$10:$G$20)
```

ist eine gültige Summenformel. Liegt die Quellarbeitsmappe in einem anderen Ordner oder sogar auf einem anderen Laufwerk, müssen die Laufwerkbezeichnung und der Ordnername ebenfalls angegeben werden:

```
=SUMME('D:\EXTAB\[QUELLE.xlsx]Tabelle1'!$G$10:$G$20)
```

Einfacher ist die Formeleingabe auch im Fall der Verknüpfung von Dateien, wenn Sie mit benannten Bereichen arbeiten. Sobald Sie bei der Formeleingabe die andere Arbeitsmappe aktiviert haben, können Sie dort das Namenfeld oder F3 benutzen, um den entsprechenden Namen einzufügen.

Externe Bezüge können Sie auch durch Kopieren über die Zwischenablage erzeugen. Wechseln Sie dazu in die Quellarbeitsmappe. Markieren Sie die Zelle oder den Bereich der Zellen, die als Quelldaten verwendet werden sollen, und kopieren Sie diese in die Zwischenablage. Wechseln Sie dann in die abhängige Arbeitsmappe, und markieren Sie die Zelle oder den Bereich, in der oder in dem die Quelldaten erscheinen sollen. Verwenden Sie aus dem Menü der Schaltfläche **Einfügen** den Befehl **Verknüpfung einfügen**.

Durch solche Formeln entstehen paarweise Verknüpfungen zwischen zwei Arbeitsmappen, wobei die Arbeitsmappe, die den externen Bezug enthält, die Rolle der abhängigen Arbeitsmappe übernimmt, während die Arbeitsmappe, auf die der Bezug verweist, die Quellarbeitsmappe darstellt. Dabei können beliebig viele Arbeitsmappen Bezüge zu einer Quellarbeitsmappe enthalten.

Ein praktisches Beispiel für den Nutzen externer Bezüge ist eine Tabelle, in der täglich die Daten verschiedener Wechselkurse gepflegt werden. Auf diese Tabelle können sich dann mehrere Arbeitsmappen beziehen, die aufgrund dieser Kurse Preisberechnungen anstellen. Die Arbeitsmappe mit den Wechselkursen stellt in diesem Zusammenhang die Quellarbeitsmappe dar, die anderen Mappen, die sich über externe Bezüge auf die Quellarbeitsmappe beziehen, sind in diesem Sinne abhängige Arbeitsmappen.

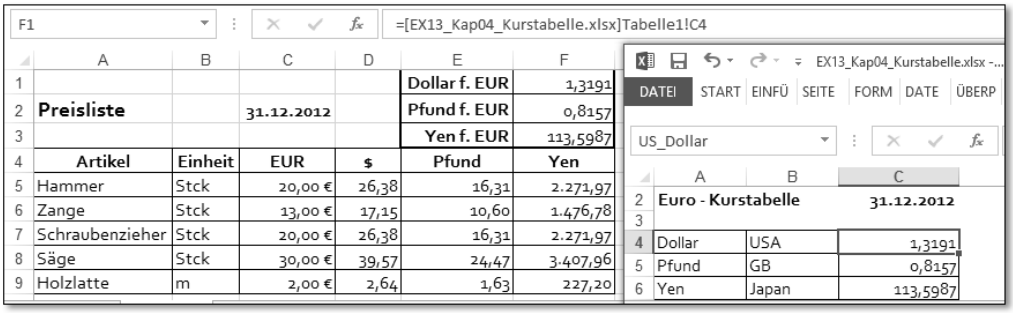


Abbildung 1.40 Beispiel für eine Verknüpfung zweier Tabellen

Verknüpfungen zwischen Dateien sind allerdings nicht unbedingt bruchsicher. Es kann beispielsweise geschehen, dass jemand den Namen der Quelldatei ändert oder sie in einen anderen Ordner verschiebt. Kann eine Formel in der abhängigen Datei keine Werte anzeigen, hilft der Dialog zu **Daten • Verbindungen • Verknüpfungen bearbeiten**. Er erlaubt, eine gebrochene Verbindung wieder in einen ordnungsgemäßen Zustand zu bringen, so dass die betreffenden Formeln auch wieder korrekte Bezüge verwenden und die Daten aus der Quelldatei übernehmen können.

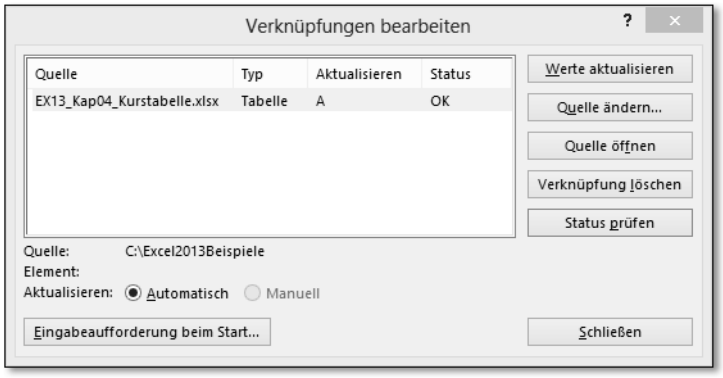


Abbildung 1.41 In diesem Dialog lassen sich Verknüpfungen zwischen Arbeitsmappen bearbeiten.

Sobald eine abhängige Arbeitsmappe mit den Verknüpfungsformeln abgespeichert ist, besteht eine feste Verbindung zur Quellarbeitsmappe. Wenn in der Quellarbeitsmappe eine Position geändert wird, auf die sich eine Formel in der abhängigen Arbeitsmappe bezieht, ändert sich normalerweise auch der Wert in der abhängigen Arbeitsmappe. Wenn die abhängige Arbeitsmappe nicht gleichzeitig geöffnet ist, geschieht dies allerdings erst in dem Augenblick, in dem sie erneut geöffnet wird. Wird nur die abhängige Tabelle geöffnet, erscheint unter Umständen die Abfrage, ob die Änderungen aktuali-

siert werden sollen. Diese Frage sollten Sie normalerweise mit **Aktualisieren** beantworten, denn dann übernimmt die abhängige Mappe Änderungen aus der Quellarbeitsmappe, die darin in der Zwischenzeit vorgenommen wurden.

1.14.7 Hilfe bei der Eingabe von Funktionen

Das Programm Excel verlangt bei der Benutzung von Funktionen die exakte Beachtung der vorgeschriebenen Schreibweise. Nicht von Belang ist, ob Groß- oder Kleinbuchstaben verwendet werden. Damit Sie nicht ständig nachsehen müssen, welche Argumente eine bestimmte Funktion braucht, hilft Ihnen Excel entweder mit direkt eingeblendeten QuickInfos oder mit dem Dialog **Funktion einfügen**.

1.14.8 Manuelle Eingabe

Wenn Sie sich gut mit den Funktionen auskennen, können Sie die Funktionen direkt in die Zelle eintragen. Excel 2013 unterstützt Sie dabei mit der Funktion **AutoVervollständigen für Formeln**, sofern diese über **Excel-Optionen • Formeln** nicht unter **Arbeiten mit Formeln** abgeschaltet ist. Bei Bedarf kann die Funktion aber auch im Bearbeitungsmodus mit Alt + ↓ ein- und ausgeschaltet werden.

Die manuelle Eingabe beginnt immer mit dem Gleichheitszeichen. Sobald Sie in eine Zelle ein Gleichheitszeichen und einen ersten Buchstaben eingeben, listet Excel die mit diesem Zeichen beginnenden Funktionen zur Auswahl auf. Zu jeder markierten Funktion wird eine Kurzinformation angezeigt. Doppelklick oder -tipp oder die ↹-Taste übernimmt eine ausgewählte Funktion und setzt die öffnende Klammer. Esc schließt die Liste ohne Übernahme eines Eintrags.

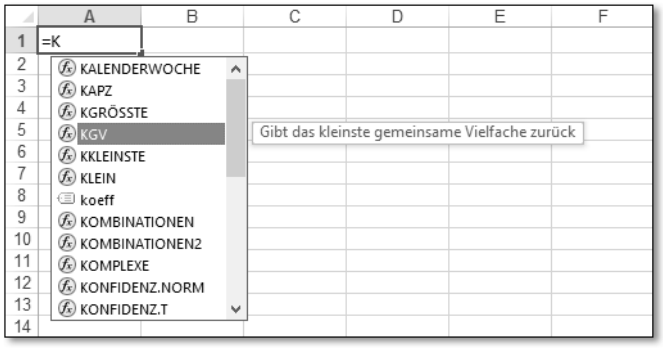


Abbildung 1.42 Auflistung möglicher Funktionen bei der Eingabe

Die Liste verfügt unter Umständen über eine Bildlaufleiste, so dass auch verdeckte Optionen erreicht werden können. In der Bearbeitungszeile kann mit der Taste nach links jeweils ein schon eingegebener Buchstabe gelöscht werden, damit wieder die größere Auswahl angezeigt wird. Die erste Übereinstimmung mit der bisherigen Eingabe wird jeweils hervorgehoben.

Symbole vor dem Eintrag geben Hinweise auf den Typ des jeweiligen Eintrags, so dass etwa Excel-Funktionsnamen und benutzerdefinierte Funktionsnamen gut zu unterscheiden sind. Bei Funktionen, die in Excel 2013 durch neuere Funktionen ersetzt worden sind, wird ein kleines Warnsignal auf dem Symbol angezeigt.

Sobald der komplette Name einer Funktion und die erste Klammer für die Argumente direkt in der Zelle oder in der Bearbeitungsleiste eingegeben sind, werden QuickInfos eingeblendet, die zeigen, welche Argumente die Funktion verlangt, jedenfalls solange die Option **QuickInfos für Funktionen anzeigen** nicht über **Excel-Optionen • Erweitert • Anzeige** abgeschaltet ist.

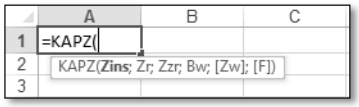


Abbildung 1.43 QuickInfos zur `KAPZ()`-Funktion

Die einzelnen Elemente in der QuickInfo arbeiten dabei wie Hyperlinks. Wenn Sie einen Platzhalter für ein Argument markieren, wird das Argument in der Formel ebenfalls ausgewählt und kann wenn nötig korrigiert werden. Klicken oder tippen Sie dagegen auf den Funktionsnamen, wird ohne alle Umstände sofort die Hilfe zu dieser Funktion eingeblendet.

Auch bei der Eingabe der Argumente kann die **AutoVervollständigen**-Funktion helfen. Das gilt beispielsweise, wenn eine Funktion für ein Argument bestimmte Konstanten zur Auswahl anbietet, wie beispielsweise die Funktion `ZELLE()`. Sobald die öffnende Klammer erscheint, wird eine Liste der möglichen Werte für das Argument **Infotyp** angeboten. Doppelklick oder -tipp übernimmt den angebotenen Wert. Entsprechendes gilt für die Funktion `TEILERGEBNIS()`, wie Abbildung 1.44 zeigt.

Sind im Tabellenblatt Namen definiert, werden sie ebenfalls über Auswahllisten angeboten. Sind Tabellen definiert, werden auch die Bezeichner für Tabellenspalten und spezielle Elemente wie `[#Alle]`, `[#Daten]`, `[#Kopfzeilen]` etc. angeboten, sobald hinter dem Tabellennamen – er kann über das Register **Entwurf** geändert werden – eine eckige

Klammer geöffnet wird. (Steht der Zellzeiger innerhalb der Tabelle, kann der Tabellenname weggelassen werden.) Abbildung 1.45 zeigt ein entsprechendes Beispiel. Die Formel `=SUMME(warenliste[#Daten])` liefert die Gesamtsumme aller Werte der Tabelle mit dem Namen »warenliste«.

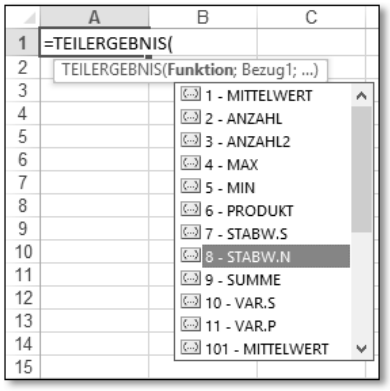


Abbildung 1.44 Liste der Werte für das Argument »Funktion« bei `TEILERGEBNIS()`

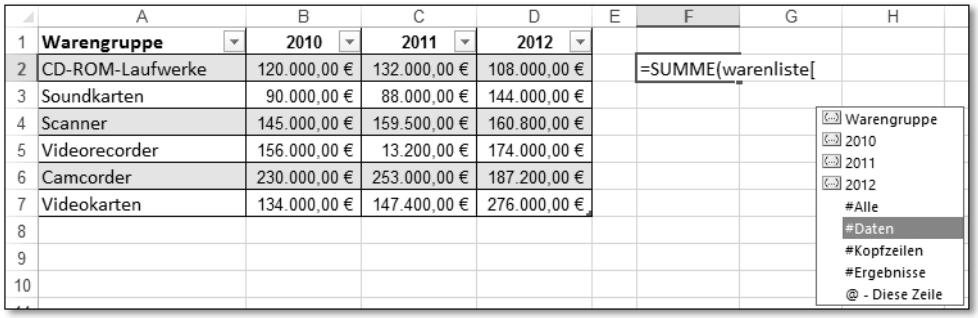


Abbildung 1.45 Übernahme von Bezeichnern von Tabellenelementen

Besonders praktisch ist die **AutoVervollständigen**-Funktion vor allem auch bei den in Excel 2007 neu eingeführten Cube-Funktionen. Mehr dazu in Kapitel 11, »Cube-Funktionen«.

Für den schnellen Zugriff auf bestimmte Funktionen stellt Excel 2013 außerdem in der Gruppe **Formeln • Funktionsbibliothek** für die am häufigsten genutzten Funktionskategorien eigene Symbole zur Verfügung, die entsprechende Funktionslisten zur Auswahl stellen. Die anderen Kategorien erreichen Sie über das Untermenü zu dem Symbol **Mehr Funktionen**, wie Abbildung 1.46 zeigt:

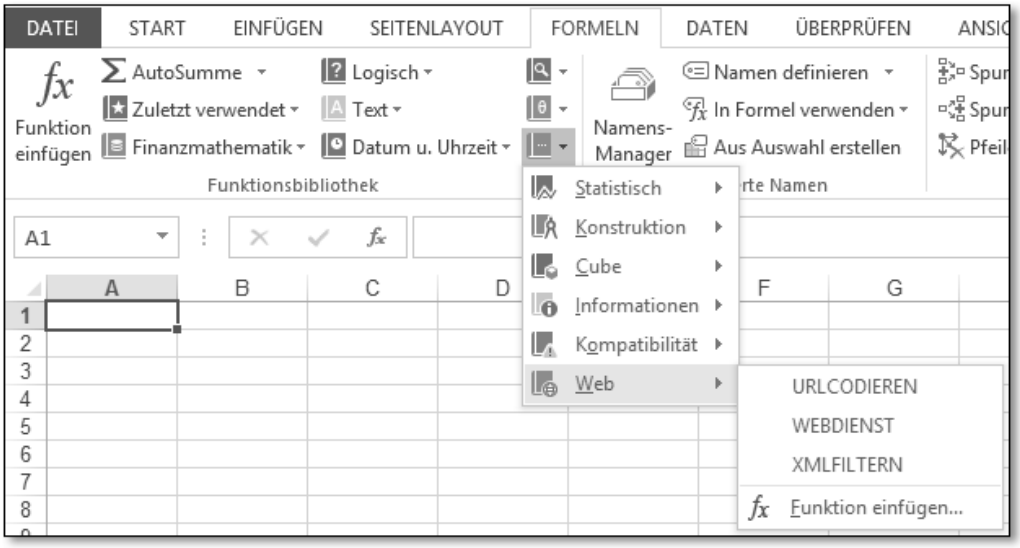


Abbildung 1.46 Das Angebot der Funktionsbibliothek

1.14.9 Der Dialog »Funktion einfügen«

Anstatt Funktionen direkt in eine Zelle einzutragen, können Sie auch mit dem Dialog **Funktion einfügen** arbeiten. Markieren Sie erst die Zelle, und klicken oder tippen Sie in der Bearbeitungsleiste auf das **fx**-Symbol. Das Gleichheitszeichen wird dann von Excel automatisch an den Anfang der Formel gesetzt. Oder wählen Sie seit Excel 2007 den Befehl **Formeln • Funktion einfügen** oder ⇧ + F3, in den älteren Versionen **Einfügen • Funktion**.

Das Dialogfeld erlaubt unter **Funktion suchen** die Eingabe von Fragen. Geben Sie etwa »Barwert berechnen« ein, und bestätigen Sie mit **OK**. Excel stellt unter der Kategorie **Empfohlen** die für Barwertberechnungen verwendbaren Funktionen zur Auswahl.

Wissen Sie, welche Funktion Sie benötigen, etwa die `ZINS()`-Funktion, wählen Sie erst die Kategorie **Finanzmathematik** aus, zu der diese Funktion gehört. Im nächsten Schritt wählen Sie dann die Funktion selbst. Um in der Liste der Funktionen zu blättern, können Sie mit den Bildlaufleisten arbeiten. Meist hilft es, den Anfangsbuchstaben der Funktion einzutippen, um das Auffinden zu beschleunigen, wenn die Liste länger ist. Wenn Sie unsicher sind, unter welcher Kategorie die Funktion eingeordnet ist, wählen Sie **Alle**.

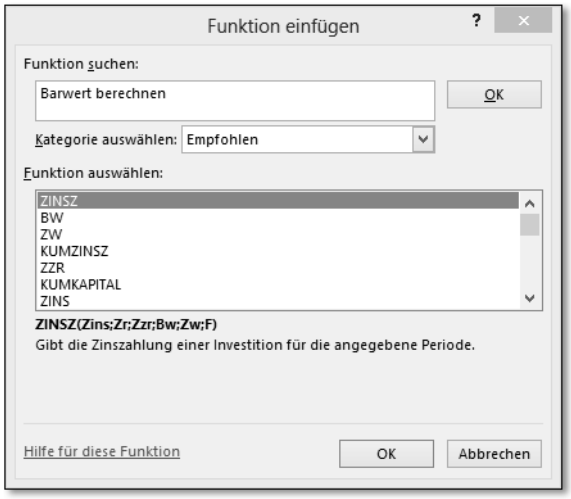


Abbildung 1.47 Suche nach Funktionen für einen speziellen Zweck

Wird eine Funktion mit einem Klick oder Tipp oder mit einer Richtungstaste markiert, erscheinen eine kurze Erklärung und die Syntax der Funktion. **OK** führt in den Dialog **Funktionsargumente**.

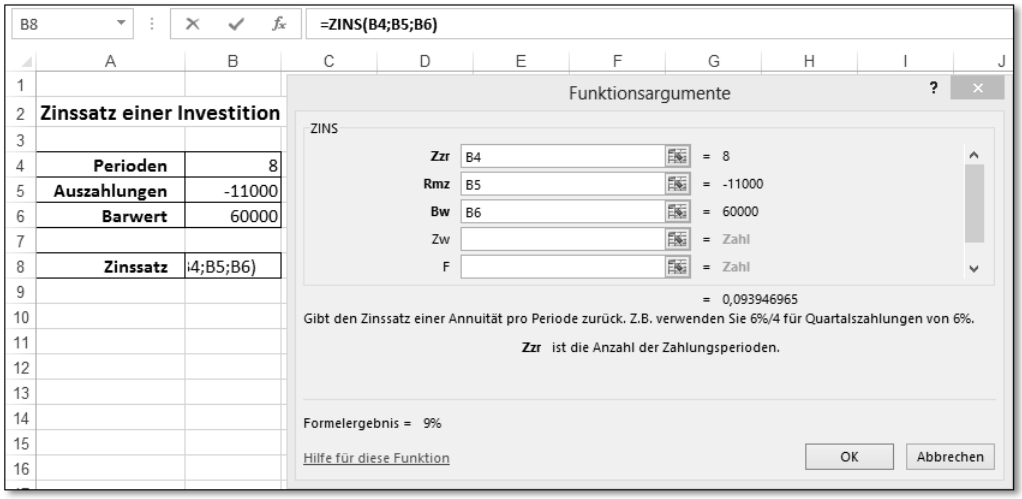


Abbildung 1.48 Dialog für die Argumente, die die Funktion benötigt

Für jedes Argument ist ein Eingabefeld vorgesehen. Wenn die Einfügestelle in einem Feld steht, erhalten Sie im unteren Bereich des Dialogs Hinweise, welche Art von Daten

benötigt wird. Obligatorische Argumente sind fett formatiert. Eventuell müssen Argumente mit der Bildlaufleiste sichtbar gemacht werden.

Um Bezüge auf Bereiche einzugeben, können Sie die kleine Schaltfläche am Ende eines Eingabefelds benutzen, die das Dialogfeld vorübergehend einschrumpft. Markieren Sie den Bereich, und klicken oder tippen Sie dann wieder auf das kleine Symbol am Ende des Felds. Noch bequemer ist es, wenn Sie das Dialogfeld etwas zur Seite schieben können und dann einfach jeweils die Zelle im Arbeitsblatt auswählen, die den benötigten Wert enthält. Während der Eingabe wird das erwartete Formelergebnis angezeigt, soweit das möglich ist, so dass Sie fehlerhafte Eingaben oft schon im Dialog erkennen können.

Die zuletzt verwendeten Funktionen werden übrigens immer auch über das Namenfeld angeboten, um den Zugriff zu beschleunigen:

Abbildung 1.49 Funktionsangebot über das Namenfeld

1.14.10 Funktionen bearbeiten

Wenn Sie das Funktionssymbol und dann eine Funktion in einer Formel markieren, können die Argumente im entsprechenden Dialog geändert werden. Enthält die Formel mehrere Funktionen, markieren Sie sie einfach nacheinander. Benötigen Sie eine Funktion an einer bestimmten Stelle innerhalb einer Formel, benutzen Sie die **Funktion einfügen**-Schaltfläche erneut.

Wenn Sie eine Formel haben, in der eine Funktion mit anderen Operatoren gemischt ist, erscheint ein verkürztes Dialogfeld **Funktionsargumente**, das nur das Gesamtergebnis der Formel ausgibt, wenn keine Funktion markiert ist.

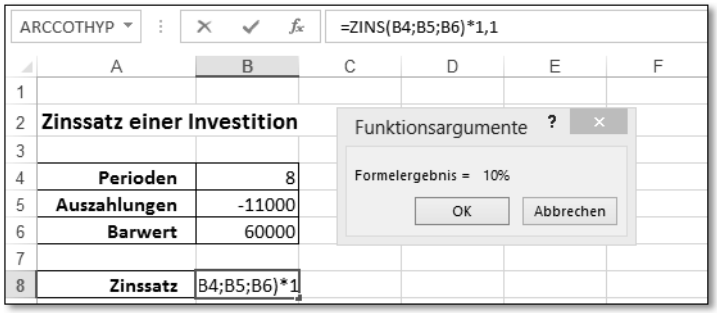


Abbildung 1.50 Das verkürzte Dialogfeld zeigt nur das Ergebnis.

1.14.11 Verschachtelte Funktionen

Der Dialog **Funktion einfügen** unterstützt Sie auch bei der Eingabe verschachtelter Funktionen, bei denen normalerweise die Fehleranfälligkeit am höchsten ist. Angenommen, Sie wollen Ihre Kunden in drei Gruppen einteilen, je nach der Höhe des Umsatzes. In der Spalte neben den Umsätzen soll jeweils ein Kennzeichen für die Gruppe abgelegt werden. Das kann mit einer `WENN()`-Funktion geschehen.

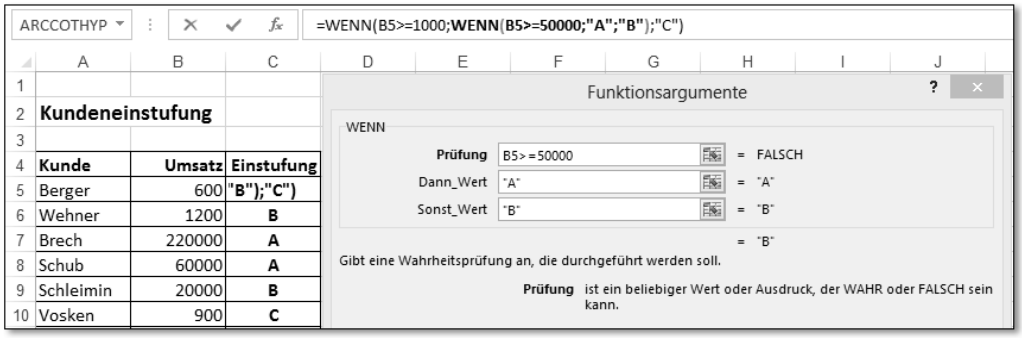
1. Benutzen Sie das Symbol **Funktion einfügen**, und wählen Sie aus der Kategorie **Logik** die Funktion `WENN()`. Bestätigen Sie mit **OK**. Im ersten Argumentfeld wählen Sie zunächst die Zelle B5 mit dem ersten Umsatzwert aus und geben dann `>1000` ein.

2. Wählen Sie das Feld **Dann_Wert** per Klick oder Tipp aus, oder drücken Sie die ⇆-Taste. Wenn die erste Bedingung erfüllt ist, soll eine zweite Bedingung geprüft werden. Also muss bei dem zweiten Argument wieder eine `WENN()`-Funktion eingefügt werden.

3. Benutzen Sie die **WENN**-Schaltfläche im Namenfeld, die nach der ersten Verwendung der `WENN()`-Funktion automatisch angeboten wird. Es erscheint wieder das Dialogfeld für die Eingabe der drei Argumente.

4. Tragen Sie die zweite Bedingung ein: `B5>50000`.

5. Geben Sie nun in das Feld **Dann_Wert** den Buchstaben `A` ein. Anführungszeichen sind nicht notwendig im Gegensatz zur manuellen Eingabe einer Funktion direkt in die Bearbeitungsleiste.



6 In das Feld **Sonst_Wert** tragen Sie den Buchstaben `B` ein.

7 Klicken oder tippen Sie auf das erste `WENN` in der Bearbeitungsleiste, erscheint wieder das Dialogfeld der übergeordneten Funktion. Geben Sie hier als **Sonst_Wert** den Buchstaben `C` ein, und bestätigen Sie mit **OK** oder ↵.

Damit ist die Formel fertig und muss nur noch auf die anderen Zellen kopiert werden.

Auch wenn Sie in einer fertigen Funktion nachträglich ein Argument durch eine Funktion ersetzen wollen, können Sie mit dem Funktionssymbol arbeiten, wenn dieses Argument ausgewählt ist.

1.14.12 Formeln kopieren

Ein großer Teil der Arbeitsersparnis, die Ihnen Excel ermöglicht, hängt davon ab, wie geschickt Sie das Kopieren von Formeln einsetzen. Es gibt dafür eine ganze Reihe von Techniken, und wir wollen Ihnen die wichtigsten kurz an einem Beispiel vorstellen.

Eine erste Technik besteht darin, Formeln für einen Zellbereich gleich in einem Zug anzulegen. Wenn Sie beispielsweise eine Tabelle mit Werten in dem Bereich A4:F24 haben und in G4:G24 die Quersummen anlegen wollen, markieren Sie den Bereich G4:G24 und geben die Formel für die erste Zeile ein: `=SUMME(A4:F4)`. Schließen Sie die Formel mit der Tastenkombination Strg + ↵ ab. Excel passt die Formeln automatisch an die Werte in den verschiedenen Zeilen an.

Solange Sie mit der Maus arbeiten, besteht in solchen Fällen immer die Möglichkeit, mit dem Ausfüllkästchen zu arbeiten. In dem genannten Beispiel geben Sie die erste Quersummenformel in die Zelle G4 ein und ziehen das Ausfüllkästchen anschließend bis zur Zelle G24.

	A	B	C	D	E	F	G
1							
2	**Messwerte**						
3	**Serie 1**	**Serie 2**	**Serie 3**	**Serie 4**	**Serie 5**	**Serie 6**	**Gesamt**
4	2,89	5,41	5,15	3,00	2,25	3,90	22,61
5	4,45	7,67	5,14	6,04	6,06	5,66	
6	7,75	3,60	5,67	6,25	7,68	4,26	
7	4,07	0,10	6,61	2,81	5,63	1,12	
8	3,86	3,61	4,10	8,82	2,97	3,74	
9	4,63	5,10	0,02	9,04	8,60	5,47	
10	5,11	2,96	0,62	6,68	9,60	9,09	
11	4,60	4,12	9,33	8,62	4,29	5,38	
12	7,94	6,72	9,30	2,70	5,59	2,10	
13	3,53	4,47	4,60	5,62	6,64	5,75	
14	6,16	9,58	5,55	5,47	3,97	9,18	
15	3,24	7,33	4,68	3,71	6,40	1,24	
16	5,12	9,20	7,55	1,41	0,62	4,76	
17	3,57	6,12	4,42	5,03	4,40	6,24	
18	5,17	9,07	8,74	4,94	7,02	1,40	
19	7,07	8,56	7,77	8,18	2,71	2,21	
20	1,09	7,18	0,48	2,28	9,85	7,81	
21	3,67	0,64	5,81	8,39	4,51	4,22	
22	9,45	1,99	6,20	2,83	0,65	8,16	
23	8,12	9,33	2,94	4,71	4,17	0,14	
24	3,55	4,51	5,67	9,46	6,10	6,54	

Abbildung 1.51 Typische Anordnung einer Tabelle, bei der das Kopieren der Formel für die Quersumme angesagt ist

Wenn die Spalte links neben der Spalte, die die Formeln aufnehmen soll, keine Lücken enthält, also keine leeren Zellen, hilft auch ein Doppelklick auf das Ausfüllkästchen, um die gesamte Spalte bis zu G24 mit den passenden Formeln zu füllen.

Eine ähnliche Methode können Sie auch auf einem Touchscreen im Modus **Fingereingabe** verwenden. Sie geben die erste Formel in G4 ein, tippen die Zelle anschließend an, bis das Kontextmenü erscheint, und blenden mit **AutoAusfüllen** das entsprechende Symbol ein. Dieses ziehen Sie dann mit dem Finger oder dem Stift bis zur Zelle G24.

D	E	F	G	H
Serie 4	**Serie 5**	**Serie 6**	**Gesamt**	
3,00	2,25	3,90	22,61	
6,04	6,06	5,66		
6,25	7,68	4,26		
2,81	5,63	1,12		
8,82	2,97	3,74		
9,04	8,60	5,47		

Abbildung 1.52 Auf dem Touchscreen übernimmt das Symbol **AutoAusfüllen** die Rolle des Ausfüllkästchens.

Die nächste Methode arbeitet mit den Symbolen, die auf dem Register **Start** in der Gruppe **Bearbeiten** in dem schon angesprochenen Menü der Schaltfläche **Füllbereich** an-

geboten werden. Sie geben in Zelle G4 die erste Formel ein und markieren anschließend mit Maus, Tastatur, Finger oder Stift den Bereich G4:G24. Um die Spalte zu füllen, verwenden Sie die Option **Unten**. Sollen Zellen in einer Zeile gefüllt werden, nehmen Sie das Symbol für **Rechts**.

Abbildung 1.53 Das Menü der Schaltfläche **Füllbereich** in der Gruppe **Start • Bearbeiten**

Schließlich haben Sie immer auch die Möglichkeit, Formeln über die Zwischenablage zu kopieren. Das ist die bevorzugte Methode, wenn die Zellen, in denen die Formeln eingefügt werden sollen, nicht direkt benachbart sind, sondern an ganz anderer Stelle, vielleicht sogar auf einem anderen Blatt oder in einer anderen Arbeitsmappe benötigt werden. Diese Methode hat außerdem den Vorteil, dass eine einmal in die Zwischenablage übernommene Formel nacheinander an verschiedene Stellen eingefügt werden kann. Die notwendigen Schritte sind die folgenden:

Markieren Sie die Formel oder den Formelbereich, der kopiert werden soll. Benutzen Sie die Befehle **Start • Zwischenablage • Kopieren** oder Strg + C. Markieren Sie den Zielbereich. Benutzen Sie **Start • Zwischenablage • Einfügen** oder Strg + V.

1.14.13 Formeln dokumentieren

Wenn Sie zur Prüfung eines Kalkulationsmodells im Tabellenblatt alle Formeln auf einmal sehen wollen, können Sie über **Excel-Optionen** auf dem Register **Erweitert** unter **Optionen für dieses Arbeitsblatt anzeigen:** das Kästchen bei **Anstelle der berechneten Werte Formeln in Zellen anzeigen** abhaken. Die Anzeige der Formeln kann insbesondere auch benutzt werden, um Tabellen zu dokumentieren.

Wollen Sie dagegen nur die Formeln einzelner Zellen dokumentieren, können Sie die in Excel 2013 neu eingeführte Funktion `FORMELTEXT()` verwenden. Sie zeigt die Formel der Zelle, deren Bezug Sie als Argument angeben, als Text an.

	A	B	C	D	E	F
7	Mittelwert	61%	=MITTELWERT(Relation_Angestellte_Arbeiter)			

Abbildung 1.54 Die Zelle C7 zeigt die in B7 verwendete Formel an.

1.15 Einsatz von relativen und absoluten Bezügen

Um die Art und Weise, wie Excel mit Formeln arbeitet, noch etwas näher kennenzulernen, soll ein einfaches Beispiel etwas genauer behandelt werden: In der Tabelle in Abbildung 1.55 sind die Umsätze und die Kosten einer Firma für verschiedene Artikelgruppen zusammengestellt.

	A	B	C	D
1				
2	**Betriebsergebnis**			**2013**
3				
4	**Artikelgruppe**	**Umsatz**	**Kosten**	**Gewinn**
5	Möbel	2000000	1240000	760000
6	Teppiche	3000000	1900000	1100000
7	Wohnungsbedarf	1500000	1050000	450000
8	Gartenmöbel	250000	150000	100000
9	**Gesamt**	6750000	4340000	2410000

Abbildung 1.55 Umsätze und Kosten für verschiedene Artikelgruppen

In dieser Tabelle sind zwei Berechnungen erforderlich: die Bildung von Gesamtsummen für die Spalten und die Berechnung des Rohgewinns. Anstatt nun die Zahlen direkt in die Formeln einzugeben, arbeiten Sie mit den Adressen der Zellen, d.h., auf den Inhalt der Zellen wird über ihre Adressen Bezug genommen.

1.15.1 Arbeit mit relativen Bezügen

Für die Bildung der Gesamtsummen bietet Excel die schon angesprochene Funktion `SUMME()`. Diese Funktion ist in der Lage, alle numerischen Werte in einem angegebenen Zellbereich zu addieren. Excel muss dafür nur wissen, wie groß der Bereich ist. Der Bereich muss der Funktion als Bereichsadresse übergeben werden. Eine Bereichsadresse ist eine Angabe in der Art `Von_Zelle:Bis_Zelle`. Der Bereich wird also definiert über die Adresse der Zelle, mit der der Bereich beginnt, und die Adresse der Zelle, mit der der Bereich endet, getrennt durch einen Doppelpunkt. Wenn Sie in der Beispieltabelle die Zelle B9 auswählen, könnten Sie `=SUMME(B5:B8)` eintragen.

Allerdings ist es in diesem Fall praktischer, das Summensymbol aus der Gruppe **Start • Bearbeiten** doppelt anzuklicken. In den älteren Versionen befindet sich das Symbol in der **Standard**-Symbolleiste. Statt die Summe für jede Spalte einzeln zu bilden, können Sie die Sache auch in einem Zug erledigen: Markieren Sie die Zellen B9 bis D9. Benutzen Sie anschließend das Summensymbol.

Um den Rohgewinn zu ermitteln, werden einfach vom Umsatz die Kosten abgezogen. Die Formel für die erste Zeile ist `=B5-C5`. Zelle D5 liefert das verlangte Ergebnis. Wenn die Zelle D5 noch markiert ist, müssen Sie nur das Ausfüllkästchen doppelt anklicken, um die Formeln für die anderen Artikelgruppen zu erzeugen. Statt des Doppelklicks kann das Ausfüllkästchen auch mit der Maus nach unten gezogen werden. Der Doppelklick ist insbesondere bei langen Spalten praktischer. Auf dem Touchscreen arbeiten Sie, wie oben beschrieben, am besten mit dem Symbol **AutoAusfüllen**.

Wenn Sie eine der kopierten Formeln auswählen, sehen Sie, dass Excel bei den verwendeten Zelladressen jedes Mal die Zeilennummer geändert hat. Beim Kopieren werden die Adressen der »Umsatz«- und »Kosten«-Zellen also nicht als fixe Adressen behandelt, sondern sinngemäß angepasst. Diese Anpassung ist meist erwünscht, aber nicht immer. Wenn Sie z. B. den Prozentsatz einzelner Artikelgruppen am Gesamtgewinn berechnen wollen, würde diese Anpassung zu Fehlern führen. Excel unterscheidet deshalb grundsätzlich zwei verschiedene Bezugsarten: relative und absolute Bezüge. Abgeleitet davon sind die Mischbezüge, bei denen ein Teil des Bezugs relativ – beispielsweise die Zeilennummer –, ein anderer – die Spaltenbezeichnung – absolut ist.

Bei einem relativen Bezug notiert Excel intern die Position der Zelle, auf die Bezug genommen wird, in der Form, dass die relative Entfernung von der Zelle gemessen wird, die den Bezug enthält. Excel interpretiert die Formel `=B5-C5` intern also wie die folgende Anweisung: Ziehe von dem Wert in der Zelle, die zwei Zellen weiter links liegt, den Wert ab, der eine Zelle weiter links liegt. Wenn Sie diese Formel nach unten kopieren, kann diese Beschreibung der Rechenaufgabe unverändert bleiben, nur bezieht sie sich nun auf die Zellen B6 und C6 etc. Wenn Sie einen relativen Bezug verwenden, weisen Sie also Excel an, diesen Bezug automatisch zu verändern, wenn die Formel an eine andere Stelle kopiert wird. Beim Verschieben einer Formel werden die Adressen dagegen nicht verändert.

1.15.2 Absolute Bezüge

Bei einem absoluten Bezug auf eine Zelle ist immer genau diese Zelle gemeint und keine andere. Das heißt auch, dass der Bezug beim Kopieren nicht verändert wird. Relative und absolute Bezüge können in einer Adresse auch gemischt werden. Der absolute Teil

des Bezugs bleibt beim Kopieren unverändert, der relative Teil des Bezugs wird angepasst. Die Tabelle zeigt die möglichen Bezugsarten:

Beispiel	Bezugsart
B2	Ist ein relativer Bezug auf die Zelle B2.
$B$2	Ist ein absoluter Bezug auf die Zelle B2.
$B2	Ist ein gemischter Bezug, bei dem der Bezug auf die Spalte absolut gesetzt ist, während der Bezug auf die Zeile relativ bleibt.
B$2	Ist ein gemischter Bezug, bei dem der Bezug auf die Zeile absolut gesetzt ist, während der Bezug auf die Spalte relativ bleibt.

Tabelle 1.9 Relative, absolute und gemischte Bezüge

Der relative Bezug ist für Excel der Normalfall. Jeder einfache Zellbezug ist ein relativer Bezug. Dagegen werden bei Bezügen auf benannte Bereiche, von denen in Abschnitt 1.17, »Arbeit mit benannten Bereichen«, noch die Rede sein wird, in der Regel absolute Bezüge verwendet. Wird der benannte Bereich durch Löschen von Zeilen oder Spalten oder durch Einfügen von Zeilen oder Spalten verkleinert oder vergrößert, werden die Bereichsadressen aber automatisch angepasst. Wird ein benannter Bereich verschoben, werden die Bezüge ebenfalls automatisch angepasst.

Wann ist es notwendig, mit absoluten oder gemischten Bezügen zu arbeiten? Der Fall der Prozentrechnung wurde schon angesprochen. Wenn Sie in unserem Beispiel in Zelle E5 die Formel `=B5/B9` eingeben, um den Anteil des Gruppenumsatzes am Gesamtumsatz zu berechnen, erhalten Sie ein richtiges Ergebnis. Klicken Sie in der Gruppe **Start • Zahl** auf das Symbol **Prozentformat**, um die Prozentdarstellung zu erreichen.

Ein Problem tritt erst auf, wenn Sie beispielsweise per Doppelklick auf das Ausfüllkästchen versuchen, die Formel nach unten zu kopieren. Excel gibt statt der Ergebnisse Fehlerwerte aus. Wenn Sie den Zellzeiger auf die Zelle E6 setzen, erscheint eine Schaltfläche, die zunächst einen Hinweis auf den Fehler gibt.

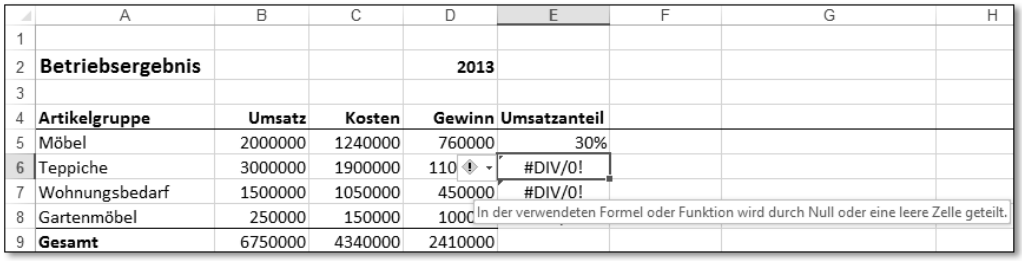


Abbildung 1.56 Die kopierten Formeln melden einen Fehler.

Wenn Sie die Schaltfläche benutzen, werden Ihnen verschiedene Optionen angeboten, auf den Fehler zu reagieren oder genauere Informationen dazu einzuholen.

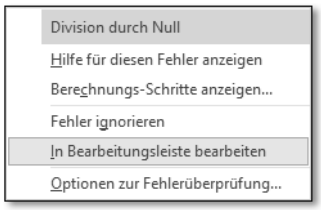
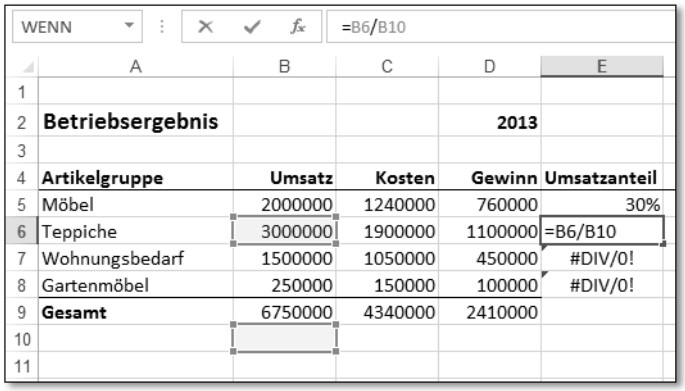


Abbildung 1.57 Das Menü zu dem Fehlerindikator

Wenn Sie die Option **In Bearbeitungsleiste bearbeiten** wählen, wird der Fehler sofort erkennbar.



	A	B	C	D	E
1					
2	**Betriebsergebnis**			**2013**	
3					
4	**Artikelgruppe**	**Umsatz**	**Kosten**	**Gewinn**	**Umsatzanteil**
5	Möbel	2000000	1240000	760000	30%
6	Teppiche	3000000	1900000	1100000	=B6/B10
7	Wohnungsbedarf	1500000	1050000	450000	#DIV/0!
8	Gartenmöbel	250000	150000	100000	#DIV/0!
9	**Gesamt**	6750000	4340000	2410000	
10					
11					

Abbildung 1.58 Offensichtlich ist ein Bezug der Formel falsch.

Die Formel bezieht sich zwar zunächst korrekt auf die zweite Artikelgruppe, dann aber nicht auf das Gesamtergebnis in B9, sondern auf die nächste Zelle B10, die leer ist. Leere Zellen wertet Excel in einer Formel aber als Zellen mit dem Wert Null. Dies führt zu dem Fehlerwert `#DIV/0!`, da ja eine Division durch null nach Adam Riese nicht erlaubt ist. Um den Fehler zu beheben, muss deshalb in allen Formeln der Bezug auf das Gesamtergebnis absolut gesetzt werden. Dann lässt sich die Formel auch kopieren. Klicken oder tippen Sie doppelt auf die Zelle E5 und dann doppelt auf die Adresse B9. Mit F4 können Sie den Bezug in einen absoluten Bezug verwandeln. `=B5/$B$9` lässt sich nun ohne Probleme bis zur Zelle E8 kopieren.

Natürlich können Sie die Dollarzeichen auch manuell eintippen, aber es ist bequemer, dafür die Taste F4 zu verwenden. Dabei ist es egal, ob die Einfügestelle direkt vor oder

direkt hinter dem Bezug steht oder ob der Bezug insgesamt markiert ist. Wenn Sie F4 mehrfach drücken, werden die möglichen Bezugsarten kreisförmig gewechselt.

1.15.3 Gemischte Bezüge

In diesem Beispiel ist der Bezug auf die Zelle, die das Gesamtergebnis enthält, zunächst absolut gesetzt worden. Es hätte auch ausgereicht, nur die Zeilennummer absolut zu setzen, da die Spaltenbezeichnung beim Kopieren gleich geblieben ist. Die Adresse `=B$9` ist ein teilabsoluter Bezug. Angenommen, Sie wollen nicht nur die Prozentanteile in Bezug auf den Umsatz, sondern auch in Bezug auf die Kosten und den Gewinn ermitteln. Geben Sie dazu in Zelle E5 die Formel `=B5/B$9` ein. Diese Formel lässt sich ohne weitere Änderungen auf den Bereich E5:G8 kopieren.

	A	B	C	D	E	F	G
1							
2	**Betriebsergebnis**			**2013**			
3							
4	**Artikelgruppe**	**Umsatz**	**Kosten**	**Gewinn**	**Umsatzanteil**	**Kostenanteil**	**Gewinnanteil**
5	Möbel	2000000	1240000	760000	30%	29%	32%
6	Teppiche	3000000	1900000	1100000	44%	44%	46%
7	Wohnungsbedarf	1500000	1050000	450000	22%	24%	19%
8	Gartenmöbel	250000	150000	100000	4%	3%	4%
9	**Gesamt**	6750000	4340000	2410000			

Abbildung 1.59 Anteil der Artikelgruppen an den Gesamtergebnissen

Das Ergebnis zeigt die Anteile der Artikelgruppen an den verschiedenen Gesamtergebnissen. Dadurch, dass in der Formel die Spaltenbuchstaben relativ benutzt werden, lässt sich die Formel in E6 gleich auf alle drei Spalten kopieren. An diesem Beispiel können Sie auch sehen, dass Sie ohne Weiteres eine Zelle auf einen ganzen Bereich kopieren können, wobei die Originalzelle auf sich selbst kopiert wird.

1.16 Einsatz von strukturierten Bezügen in Tabellen

Excel 2013 stellt für Formeln in **Tabellenbereichen**, die die Nachfolger der aus älteren Versionen bekannten Datenlisten darstellen, noch eine vereinfachte Bezugsform zur Verfügung, die als strukturierter Verweis bezeichnet wird. Soll beispielsweise in einer Lagerbestandstabelle für eine berechnete Spalte »Wert« der Preis der einzelnen Artikel mit dem Bestand, die beide in benachbarten Spalten stehen, multipliziert werden, wird bei der Markierung der Zellen während der Formeleingabe automatisch der jeweilige

Spaltenname in eckigen Klammern angezeigt, ohne dass die Bereiche explizit benannt werden müssen. Die Formel heißt in diesem Fall `=[@Bestand]*[@Preis]`, und sie wird in alle Tabellenzeilen unverändert übernommen, wertet aber jeweils die richtigen Werte in der jeweiligen Tabellenzeile aus. Bei der manuellen Eingabe einer solchen Formel kann das @-Zeichen auch weggelassen werden.

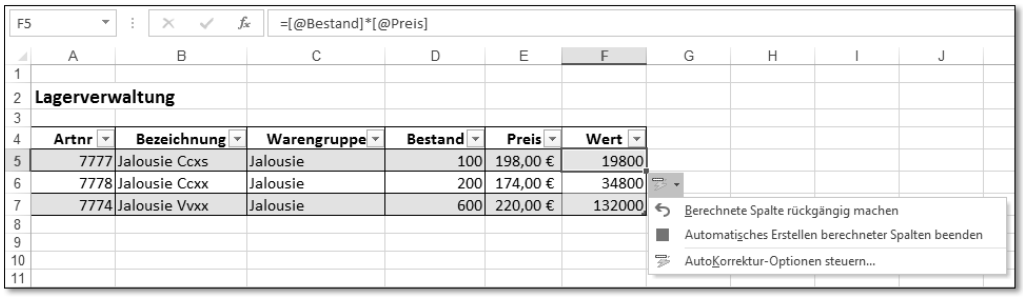


Abbildung 1.60 Hinzufügen einer berechneten Spalte mit strukturierten Bezügen

Die Verwendung von strukturierten Verweisen anstelle der in normalen Zellbereichen verwendeten Verweise erleichtert die Arbeit mit Tabellendaten. Solche Verweise können insbesondere auch Bezüge auf die verschiedenen Bereiche in einer Tabelle herstellen. Das gilt sowohl für Formeln in der Tabelle als auch für Formeln außerhalb der Tabelle. Im zweiten Fall muss allerdings immer der Tabellenname vorgesetzt werden, der bei Formeln innerhalb der Tabelle weggelassen werden kann.

Der in Excel 2013 verwendete Tabellenname verweist auf die gesamten Tabellendaten, aber ohne Überschriftenzeile und Ergebniszeile. Von den in der Tabelle vorhandenen Spaltennamen werden Spaltenbezeichner abgeleitet und in eckige Klammern gesetzt, sie verweisen auf die reinen Spaltendaten ohne Spaltenname und eventuelle Ergebniszeile. Werden Bezüge verschachtelt, sind doppelte Klammern erforderlich.

Außerdem sind folgende spezielle Bezeichner vorgegeben:

Bezeichner	Bedeutung
`#Alle`	Bezug auf die gesamte Tabelle einschließlich Spaltennamen, Daten und Ergebniszeilen
`#Daten`	Bezug auf die Daten ohne Ergebniszeile
`#Kopfzeilen`	Bezug auf die Kopfzeile mit den Spaltennamen
`#Ergebnisse`	Bezug auf die Ergebniszeile; ergibt 0, wenn diese nicht vorhanden ist.
`@`	Bezug auf die Daten in der aktuellen Zeile

Tabelle 1.10 Besondere Bezeichner in Excel

Ist eine Tabelle beispielsweise mit dem Namen `Lager` gekennzeichnet, liefert die folgende Formel `=ANZAHL2(Lager[#Alle])` die Anzahl aller Zellen der aktuellen Tabelle einschließlich der Beschriftungen. Die Bezugsoperatoren Doppelpunkt, Komma und Leerzeichen können auch hier verwendet werden. Damit strukturierte Verweise in dieser Form eingesetzt werden können, muss unter **Excel-Optionen** auf der Seite **Formeln** unter **Arbeiten mit Formeln** allerdings die Einstellung **Tabellennamen in Formeln verwenden** aktiviert bleiben.

	A	B	C	D	E	F	G	H	I
1									
2	**Lagerverwaltung**						**Stand: 30.04.2013**		
3									
4	**Artnr**	**Bezeichnung**	**Warengruppe**	**Bestand**	**Preis**	**Absat**	**Umsatz_lfd_Jah**		
5	7777	Jalousie Ccxs	Jalousie	100	198,00 €	120	23.760,00 €	7	=ANZAHL2(Lager[@])
6	7778	Jalousie Ccxx	Jalousie	200	174,00 €	330	57.420,00 €		
7	7774	Jalousie Vvxx	Jalousie	600	220,00 €	400	88.000,00 €		
8									
9			=ANZAHL(Lager[#Daten])		15				
10			=ANZAHL2(Lager[#Alle])		28				
11			=ANZAHL2(Lager[#Kopfzeilen])		7				

Abbildung 1.61 Beispiel für strukturierte Bezüge

Sollte die Tabelle nachträglich in einen normalen Zellbereich konvertiert werden, werden die strukturierten Bezüge automatisch wieder in die übliche Bezugsform zurückübersetzt. Beachtet werden muss noch, dass alle Spaltennamen als Textzeichenfolgen behandelt werden, auch wenn sie etwa ein Datum oder eine Jahreszahl enthalten. Enthält der Spaltenname Sonderzeichen, muss mit doppelten Klammern gearbeitet werden. Dies gilt für Leerzeichen, Komma, Strichpunkt, Punkt etc.

1.17 Arbeit mit benannten Bereichen

Der Bezug auf Zellen und Zellbereiche kann nicht nur über Adressen hergestellt werden. Zellen und Bereiche lassen sich auch benennen, so dass der Bezug über den Namen erfolgen kann.

1.17.1 Vorteile von Bereichsnamen

Die Verwendung solcher Namen ist zwar nicht unbedingt notwendig, aber sie macht die Arbeit mit Excel in vielen Fällen wesentlich komfortabler, zumal sich Namen auch besser merken lassen als abstrakte Adressen. Bereichsnamen machen insbesondere Formeln lesbarer. Solange Sie einfache Summenformeln verwenden, die direkt unter der addierten Kolonne stehen, ist leicht zu überblicken, was in einer Formel errechnet wird.

Das ändert sich aber schon dann, wenn an einer ganz anderen Stelle des Arbeitsblatts mit dem Ergebnis dieser Formel weitergerechnet werden soll. Dann steht dort vielleicht `=G20*1,05`. Wofür aber steht der Wert in der Zelle G20? Haben Sie die Zelle G20 aber mit dem Namen `Gewinn2012` benannt, lautet die Formel `=Gewinn2012*1,05`. Das sagt mehr.

Wird ein Bereichsname in verschiedenen Formeln verwendet, muss nicht jede Formel einzeln angefasst werden, wenn sich die Adressen des Bereichs ändern. In vielen Fällen wird die Bereichsänderung sogar automatisch vorgenommen, etwa wenn in einem definierten Bereich eine Zeile eingefügt oder gelöscht wird.

Eine ebenso praktische Sache ist, dass Sie benannte Zellen und Bereiche direkt über den Namen auswählen können. Am schnellsten gelingt dies über die Namenliste in der Bearbeitungsleiste. Auch mit F5 erhalten Sie sofort die Liste der definierten Namen. Diese Technik erspart gerade bei großen Arbeitsblättern oft mühseliges Herumsuchen.

1.17.2 Regeln für Namen

Bereichsnamen können bis zu 255 Zeichen enthalten. Groß- und Kleinschreibung ist bedeutungslos. Dennoch ist es nützlich, sie zu verwenden. Wird ein Name mit Kleinbuchstaben in einer Formel eingegeben, wandelt Excel den Namen entsprechend um, wenn der Name korrekt ist. Geschieht das nicht, hat sich vielleicht ein Tippfehler eingeschlichen.

Das erste Zeichen muss ein Buchstabe, ein Unterstrich oder ein umgekehrter Schrägstrich sein. Sonderzeichen wie !, {, }, [,], +, *, –, /, @, <, >, & und # dürfen in einem Namen nicht benutzt werden. Auch Leerzeichen, Doppelpunkt und Semikolon sind nicht erlaubt, weil diese Zeichen als Bezugsoperatoren benutzt werden. Verwenden Sie zur Trennung den Unterstrich _ oder den Punkt.

Bereichsnamen dürfen nicht wie mögliche Zelladressen lauten. `B10` oder `AA3` ist also nicht zulässig. Die Buchstaben `C`, `c`, `R`, `r` sind in Excel 2013 als Kurzform für Column – Spalte – und Row – Zeile – reserviert. Ein Name wie `c2r3` ist demnach ebenfalls nicht zulässig, da er für die Bezeichnung der Zelle in der zweiten Spalte und dritten Zeile steht.

1.17.3 Methoden der Namensgebung

In allen Versionen kann die aktuelle Markierung direkt im Namenfeld in der Bearbeitungsleiste benannt werden. Der Name muss nur mit der Eingabetaste quittiert werden.

Seit Excel 2007 können mit **Formeln • Definierte Namen • Namen definieren** Namen für Zellen und Zellbereiche, aber auch für konstante Werte oder auch häufig benötigte Formeln vergeben werden. In den älteren Versionen wird **Einfügen • Namen • Definieren** verwendet.

Normalerweise kann eine Liste von Namen pro Arbeitsmappe verwendet werden. Jeder Name muss dabei eindeutig sein. Im Namenfeld werden alle Namen angezeigt, die in der Mappe vorkommen. Es ist aber auch möglich, Namen zu vergeben, die nur für das einzelne Blatt gelten, für das sie definiert sind. Das ist dann vorteilhaft, wenn in mehreren Blättern derselbe Name für einen gleichartigen Wert verwendet werden soll, etwa `Mittelwert`. Dazu muss im Dialog **Neuer Name** unter **Bereich** die Vorgabe **Arbeitsmappe** durch den Namen des jeweiligen Blatts ersetzt werden, der über das Listenfeld ausgewählt werden kann.

Um solche Namen in Formeln anderer Blätter der Arbeitsmappe zu verwenden, muss der Blattname getrennt durch ein Ausrufezeichen vor den eigentlichen Namen gesetzt werden – etwa `Personalbestand!Mittelwert`. Im Namenfeld erscheint der blattspezifische Name nur, wenn das entsprechende Blatt aktiviert ist. Auf diesem Blatt kann der Blattname weggelassen werden. Ist der Name `Mittelwert` auch für die ganze Mappe vergeben, wird er in diesem Blatt ignoriert.

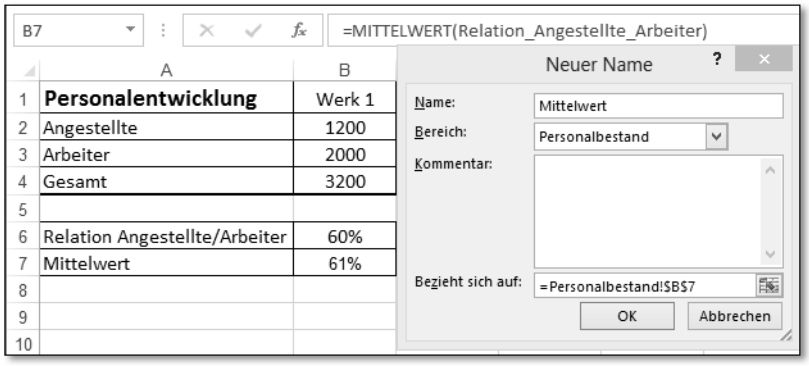


Abbildung 1.62 Definition eines blattspezifischen Namens

Mit dem Befehl **Formeln • Definierte Namen • Aus Auswahl erstellen** können Zellen oder Zellbereiche mit Namen verknüpft werden, die im Arbeitsblatt bereits als Beschriftung eingetragen sind. Voraussetzung ist, dass die Zelle mit dem Namen an die Zelle oder den Zellbereich angrenzt, der benannt werden soll. Dem entspricht in den älteren Versionen **Einfügen • Namen • Erstellen**.

Excel übernimmt einfach die vorhandenen Beschriftungen als Namen. Dabei werden die Namen an die oben beschriebenen Regeln angepasst. Ist das erste Zeichen eine Zahl, wird ein Unterstrich davorgesetzt. Auch Leerzeichen und Semikola werden durch Unterstriche ersetzt. Datumswerte werden in Zeichenfolgen verwandelt, wobei das gegebene Format verwendet wird.

Die Zellen, die benannt werden sollen, können über oder unter den Zellen mit der Beschriftung liegen oder rechts oder links neben diesen Zellen. Zellbereiche sind hier immer Zellgruppen in einer Zeile oder in einer Spalte.

Um in dem abgebildeten Beispiel zu einer korrekten Übernahme der Namen zu kommen, markieren Sie zunächst den Bereich A1 bis D3. Die Zeile mit den Gesamtsummen wird erst einmal ausgelassen, damit im nächsten Schritt die Namen für die Spaltenwerte in den Summenformeln verwendet werden können. Der Dialog **Namen aus Auswahl erstellen** bietet vier Kontrollkästchen, von denen die beiden ersten aufgrund der Situation in dem vorher ausgewählten Bereich abgehakt sind. Die oberste Zeile und die linke Spalte sollen die Namen liefern. Wenn Sie quittieren, erzeugt Excel benannte Bereiche, zwei Zeilenbereiche – »Angestellte«, »Arbeiter« – und drei Spaltenbereiche – »Werk_1«, »Werk_2« und »Werk_3«.

Wenn Sie die Liste der vergebenen Namen ansehen, finden Sie noch einen weiteren Bereich, den Excel in diesem Fall für die Werte in den Zellen B2:D3 vergeben hat. Dabei verwendet das Programm für die Namensgebung – falls vorhanden – den Inhalt der linken oberen Eckzelle des anfangs markierten Bereichs, in diesem Fall ist das A1, und der Bereichsname ist entsprechend »Personalentwicklung«.

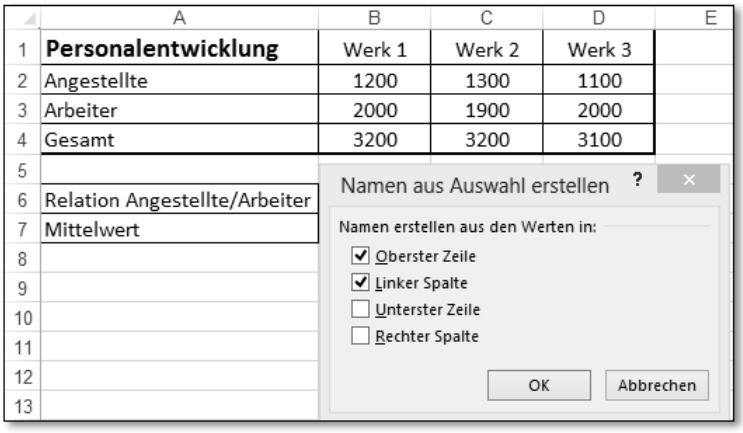


Abbildung 1.63 Übernahme bestehender Beschriftungen als Bereichsnamen

Sie können die neuen Bereiche nun sehr schnell über die Namenliste oder mit F5 über den Dialog **Gehe zu** auswählen.

Seit Excel 2007 gibt es zudem einen **Namens-Manager**, der über die gleichnamige Schaltfläche in der Gruppe **Formeln • Definierte Namen** aufgerufen wird. Der Dialog gibt eine übersichtliche Liste der vorhandenen Namen und ist mit dem oben verwendeten Dialogfeld verknüpft, das über die Schaltflächen **Neu** und **Bearbeiten** erreicht wird. Zum Löschen überflüssig gewordener Namen wird die Schaltfläche **Löschen** verwendet.

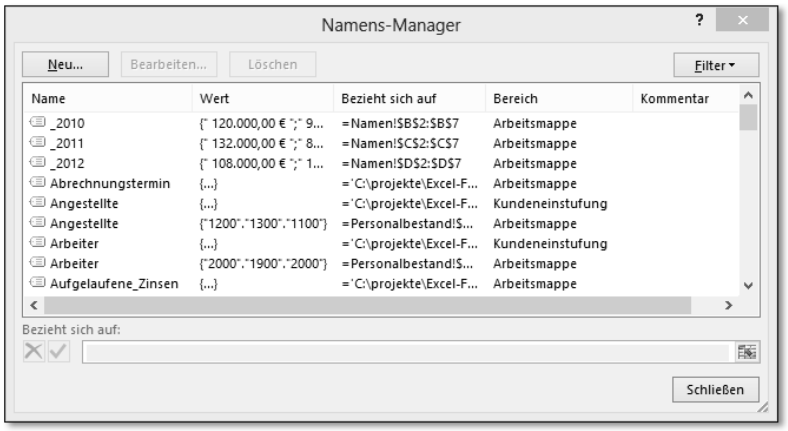


Abbildung 1.64 Der Namens-Manager erleichtert die Verwaltung umfangreicher Namensvergaben.

Interessant ist hier insbesondere die Filterfunktion, die verschiedene Auswahlmöglichkeiten anbietet, etwa eine Einschränkung auf blattbezogene oder definierte Namen.

1.17.4 Benannte Formeln und Konstanten definieren

Das Verfahren, eine Formel mit einem Namen zu verknüpfen, ähnelt der Vergabe von Zell- und Bereichsnamen. Excel speichert die Definition zwar zusammen mit der Arbeitsmappe, aber nicht in einer Zelle. Die Formel wird direkt im Dialogfeld eingetragen, in dem der Name definiert wird, und zwar anstelle eines Bereichsbezugs.

Sie können auch bestimmte Werte oder Textkonstanten mit einem Namen belegen. Um beispielsweise den Mehrwertsteuersatz abzulegen, geben Sie in das Bezugsfeld einfach `=19%` ein. Wenn der Wert mit `MWST` benannt ist, geben Sie in einer Formel zur Berechnung der Mehrwertsteuer für einen Betrag in B10 nur `=B10*MWST` ein.

Wenn Sie eine Textkonstante unter **Bezieht sich auf** eintragen, muss sie in Anführungszeichen eingeschlossen werden. Wenn Sie etwa unter dem Namen `Firma` den komplet-

ten Firmennamen ablegen, liefert Ihnen der Eintrag `=Firma` in einer Zelle den entsprechenden Namen.

Sowohl benannte Formeln als auch benannte Konstanten werden bei der Formeleingabe mit in der Funktionsliste angeboten, wie Abbildung 1.65 zeigt.

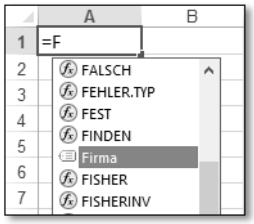


Abbildung 1.65 Eine benannte Konstante in der Funktionsliste

1.17.5 Anwenden von Namen in Formeln

Wenn Sie Bereiche benennen, werden in schon vorhandenen Formeln die Bezüge auf diese Bereiche nicht automatisch ersetzt. Dies muss vielmehr ausdrücklich mit dem Befehl **Namen übernehmen** geschehen, der seit Excel 2007 als zweite Option zu der Schaltfläche **Formeln • Definierte Namen • Namen definieren** angeboten wird. In den älteren Versionen gilt hier der Befehl **Einfügen • Namen • Übernehmen**. Bei der manuellen Eingabe von Formeln und im Funktions-Assistenten können Namen außerdem mit F3 angezeigt und abgerufen werden.

Abbildung 1.66 Das Dialogfeld **Namen übernehmen**

Welche Namen in den Formeln verwendet werden sollen, können Sie durch Anklicken oder Antippen auswählen. Die Auswahl hat auch Auswirkungen auf Formeln, die Sie später eingeben. Excel ersetzt eine passende Bereichsmarkierung dann automatisch durch den Bereichsnamen.

Beachtet werden muss, dass Bereichsnamen absolute Adressen verwenden. Beim Kopieren solcher Formeln werden die Bezüge also nicht angepasst. Wie sich Excel bei der Anwendung der Namen im Detail verhalten soll, kann über **Optionen** gesteuert werden. Die Einstellung **Relative/Absolute Bezugsart ignorieren** bewirkt, dass ein Bezug unabhängig von der in der Formel verwendeten Bezugsart ersetzt wird. Ist die Option abgewählt, werden nur absolute Zellbezüge durch die Bereichsnamen ersetzt.

Wird beim Anwenden der Namen das Kontrollkästchen **Zeilen- und Spaltennamen verwenden** abgehakt, kann festgelegt werden, ob in Formeln, die Schnittmengen von Bereichen auswerten, bei Bezügen innerhalb einer Spalte oder innerhalb einer Zeile die entsprechenden Namen entfallen können, um die Formulierung der Formel zu vereinfachen. Außerdem kann festgelegt werden, ob bei Bezügen, bei denen Spalte und Zeile angegeben werden müssen, erst die Zeile oder erst die Spalte angegeben wird.

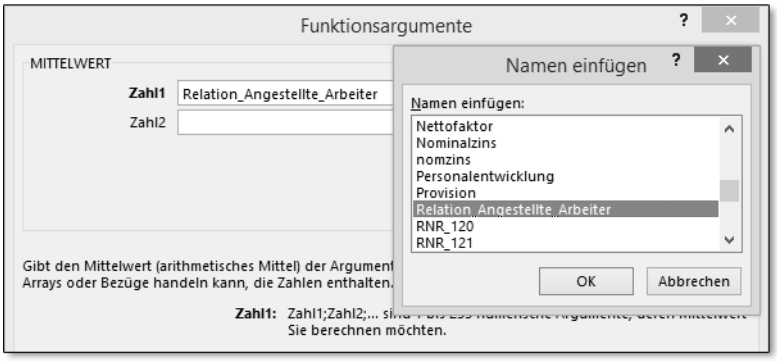


Abbildung 1.67 Übernahme eines Namens als Argument für eine Funktion

Soll ein Name bei der Eingabe einer Formel oder im Dialog **Funktionsargumente** eingefügt werden, können Sie den Namen entweder direkt an der Einfügestelle einfügen oder ein vorher markiertes Argument durch einen Namen ersetzen. Verwenden Sie F3 oder den Befehl **Formeln • In Formeln verwenden** und den gewünschten Namen.

Auch wenn Sie Formeln ohne Mithilfe des Dialogs **Funktion einfügen** eingeben, werden Ihnen die in der Arbeitsmappe definierten Namen angeboten, sobald Sie den ersten oder die ersten Buchstaben eines Namens eingeben. Abbildung 1.68 zeigt ein Beispiel.

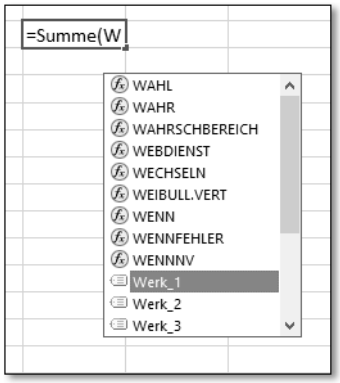


Abbildung 1.68 Übernahme eines Bereichsnamens als Argument in einer Formel

Voraussetzung ist allerdings, dass Sie die Funktion **AutoVervollständigen für Zellwerte aktivieren** nicht über den Dialog **Excel-Optionen** auf der Seite **Erweitert** abgeschaltet haben.

1.18 Matrixberechnungen

Durch eine Formel wie `=B2*B3` wird eine Operation ausgelöst, bei der ein einzelner Zellwert mit einem Wert aus einer anderen Zelle multipliziert wird und das Ergebnis in der Zelle erscheint, in der die Formel steht. Excel erlaubt aber auch, als Argument einer Formel gleich mehrere Werte zu benutzen. Auch als Ergebnis einer Formel können gleich mehrere Werte produziert werden. Dies ist z. B. bei bestimmten Funktionen der Fall. In beiden Fällen sind die Ausgangs- bzw. Ergebniswerte in einer bestimmten Form angeordnet, die *Matrix* genannt wird.

1.18.1 Matrixbereiche in Excel

Eine Matrix ist immer ein rechteckiger Zellbereich, in dem jede Zelle einen Wert enthält. Sie entspricht also, sofern es sich bei den Werten um numerische Werte handelt, dem, was auch in der Algebra eine Matrix genannt wird. Ein einfaches Beispiel ist das folgende Zahlenschema aus zwei Zeilen und drei Spalten.

23	26	65
16	10	45

Die Besonderheit, die zu beachten ist, besteht darin, dass Excel eine Matrix als eine Einheit behandelt, bei der die einzelnen Elemente in verschiedener Hinsicht auch nicht gesondert bearbeitet oder benutzt werden können. Soll eine Matrix als Argument in einer Formel verwendet werden, bestehen zwei Möglichkeiten:

Die erste Möglichkeit ist, in der Formel den Bezug auf einen rechteckigen Zellbereich einzugeben, der die Werte der Matrix enthalten soll. Ein Beispiel stellt die Funktion `INDEX()` dar, die als erstes Argument eine Matrix erwartet, etwa `=INDEX(B4:D6;2;2)`.

Die zweite Möglichkeit ist eine Matrixkonstante. Die Schreibweise innerhalb einer Formel sieht folgendermaßen aus: Die gesamte Matrix wird in geschweifte Klammern gesetzt. Die Werte einer Zeile werden jeweils durch einen Punkt getrennt, die Zeilen dagegen durch ein Semikolon. Das oben angeführte Zahlenschema sieht in der Bearbeitungsleiste so aus: `{23.26.65;16.10.45}`.

Matrixkonstanten können nicht nur Zahlen, sondern auch Texte, Wahrheitswerte oder Fehlerwerte enthalten. Dabei muss das Format der Eingabe beachtet werden. Texte müssen immer in Anführungszeichen gesetzt werden.

1.18.2 Arbeit mit Matrixformeln

Neben Matrixbereichen und Matrixkonstanten, die als Argumente oder Ergebnisse von Formeln erscheinen, können auch spezielle Matrix- oder Arrayformeln verwendet werden. Diese Formeln enthalten dann ihrerseits Matrixbereiche oder auch Matrixkonstanten als Argumente.

Die Vorgehensweise, um eine Formel als Matrixformel einzugeben, soll an dem folgenden Beispiel demonstriert werden. Der erste Schritt ist die Markierung der Matrix, die die Ergebnisse der Matrixformel aufnehmen soll. In Abbildung 1.69 ist das der Zellbereich D5:D7. Dieser Schritt ist ganz entscheidend, denn wenn Sie nur eine Zelle markieren, liefert die Matrixformel auch nur ein einziges Ergebnis. Die Größe dieser Ergebnismatrix sollte in der Regel exakt so groß sein, dass die Matrixformel alle Ergebnisse erzeugen kann, die durch die Wertegruppe(n), auf die sich die Formel bezieht, möglich sind. Ist die Ergebnismatrix kleiner, werden mögliche Ergebnisse »unterschlagen«, ist sie dagegen größer, erscheint in den überzähligen Zellen der Wert `#NV` als Hinweis darauf, dass für diesen Fall keine Werte vorliegen, mit denen die Formel arbeiten kann.

Der zweite Schritt ist die Eingabe der Formel in eine der Zellen der Ergebnismatrix. Dabei spielt es keine Rolle, in welche dieser Zellen die Formel eingetragen wird. Die Formel

kann wie jede Einzelwertformel Konstanten, Operatoren, Zellbezüge und Funktionen enthalten. Die Besonderheit besteht darin, dass in der Formel Bezug auf Wertegruppen genommen wird, die in der Matrixformel dann zeilen- oder spaltenweise ausgewertet werden.

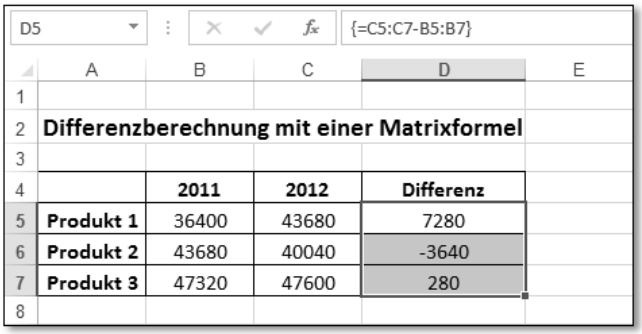


Abbildung 1.69 Differenzberechnung mit einer Matrixformel

Im Beispiel werden die Werte aus der Spalte für 2011 von der Spalte der Werte für 2012 abgezogen. Die Formel lautet also {=C5:C7–B5:B7}.

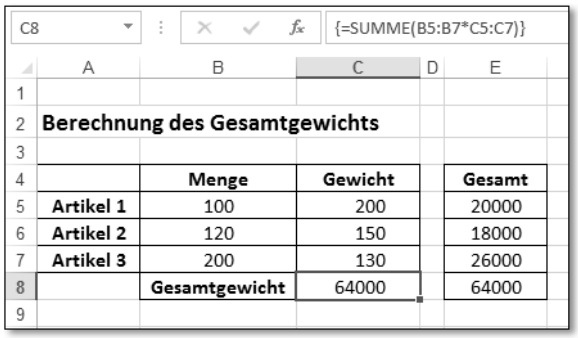
Die geschweiften Klammern, die die gesamte Formel einschließen und sie damit als Matrixformel kennzeichnen, dürfen allerdings nicht manuell eingegeben werden. Sie werden automatisch erzeugt, wenn die Formel korrekt quittiert wird. Dies geschieht mit der Tastenkombination Strg + ⇧ + ↵.

Wenn Sie sich die Ergebnisspalte anschauen, können Sie feststellen, dass Excel in jede Zelle dieselbe Formel gesetzt hat. Trotzdem entsprechen die Ergebnisse den jeweiligen Zeilenwerten. Das gewählte Beispiel setzt also zwei gleich große Zellbereiche zueinander in Beziehung und erzeugt eine entsprechend große Ergebnismatrix, in diesem Fall ist die Zahl der Zeilen gleich.

1.18.3 Vereinfachung von Berechnungen

Das folgende Beispiel ist etwas komplexer. Es zeigt, dass Matrixformeln dazu benutzt werden können, ein Tabellenmodell zu vereinfachen. Das ist möglich, indem in der Formel Rechenschritte zusammengefasst werden, die normalerweise über Zwischenergebnisse berechnet werden müssten. Soll z. B. das Gesamtgewicht einer Ladung berechnet werden, ist der übliche Weg, das jeweilige Gewicht der verschiedenen Artikel mit der

Stückzahl zu multiplizieren und dann die Ergebnisse zu addieren, wie in der Spalte E zu sehen ist.



	A	B	C	D	E
1					
2	**Berechnung des Gesamtgewichts**				
3					
4		**Menge**	**Gewicht**		**Gesamt**
5	**Artikel 1**	100	200		20000
6	**Artikel 2**	120	150		18000
7	**Artikel 3**	200	130		26000
8		**Gesamtgewicht**	64000		64000
9					

Abbildung 1.70 Gewichtsberechnung

Wird dagegen mit einer Matrixformel gearbeitet, ist es nicht erforderlich, die Ergebnisse der Multiplikation in der Tabelle darzustellen. Die Formel `{=SUMME(B5:B7*C5:C7)}` speichert die Zwischenergebnisse intern und gibt nur das Gesamtergebnis in der Zelle C8 aus.

1.18.4 Matrizenrechnung und lineare Gleichungssysteme

Eines der Hauptanwendungsgebiete von Matrizen ist die Lösung von linearen Gleichungssystemen mit mehreren Unbekannten. Dabei werden solche Gleichungssysteme zunächst in eine Koeffizientenmatrix übersetzt, auf die dann die Rechenverfahren angewandt werden, die in der Matrizenrechnung möglich sind. Dafür bietet Excel eine Reihe von speziellen Matrixfunktionen, die in Kapitel 5, »Mathematische und trigonometrische Funktionen«, beschrieben werden.

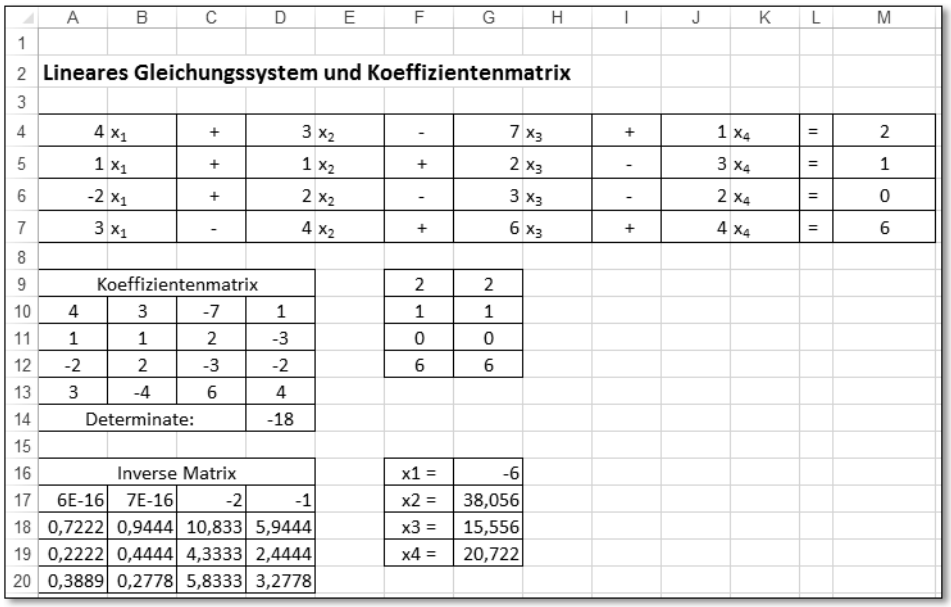
Abbildung 1.71 zeigt im Bereich A4 bis M7 ein kleines Beispiel für ein solches Gleichungssystem mit vier Unbekannten. Im Zellbereich A10 bis D13 wird die entsprechende Koeffizientenmatrix angezeigt.

Zunächst wird in der Zelle D14 eine Funktion verwendet, die die Determinante der Matrix ermittelt:

```
=MDET(A10:D13)
```

liefert einen Wert ungleich 0, woraus geschlossen werden kann, dass das Gleichungssystem lösbar ist. Zur Lösung des Gleichungssystems gibt es verschiedene Verfahren. Eine

Möglichkeit ist, die Inverse der Koeffizientenmatrix zu bilden und sie dann mit der Spaltenmatrix, die die Ergebnisse der vier Gleichungen enthält, zu multiplizieren.



Lineares Gleichungssystem und Koeffizientenmatrix

4	x_1	+	3	x_2	-	7	x_3	+	1	x_4	=	2
1	x_1	+	1	x_2	+	2	x_3	-	3	x_4	=	1
-2	x_1	+	2	x_2	-	3	x_3	-	2	x_4	=	0
3	x_1	-	4	x_2	+	6	x_3	+	4	x_4	=	6

Koeffizientenmatrix			
4	3	-7	1
1	1	2	-3
-2	2	-3	-2
3	-4	6	4
Determinate:			-18

2	2
1	1
0	0
6	6

Inverse Matrix			
6E-16	7E-16	-2	-1
0,7222	0,9444	10,833	5,9444
0,2222	0,4444	4,3333	2,4444
0,3889	0,2778	5,8333	3,2778

x1 =	-6
x2 =	38,056
x3 =	15,556
x4 =	20,722

Abbildung 1.71 Lösung eines linearen Gleichungssystems durch Matrizenrechnung

Für den ersten Schritt wird die Funktion `MINV()` verwendet. Die Formel muss als Matrixformel erzeugt werden. Dazu wird zunächst der Bereich A17 bis D20 markiert und die Formel `{=MINV(A10:D13)}` mit Strg + ⇧ + ↵ abgeschlossen.

Anschließend wird wieder eine Matrixformel für den Bereich G16 bis G19 erzeugt, die mit der `MMULT()`-Funktion arbeitet: `{=MMULT(A17:D20;F9:F12)}`.

Diese Formel liefert eine Matrix mit den Werten für die vier Unbekannten. Nur zur Kontrolle sind in den Zellen G9 bis G12 die errechneten Werte zeilenweise in die Gleichungen eingesetzt, um das Ergebnis zu bestätigen.

1.18.5 Matrixformeln bearbeiten

Wollen Sie eine Matrixformel ändern, können Sie eine beliebige Formelzelle in der Matrix auswählen. Sobald Sie die Bearbeitungsleiste aktivieren, verschwinden dort die geschweiften Klammern. Wenn die Änderungen an der Formel abgeschlossen sind, muss

die Formel erneut mit Strg + ⇧ + ↵ quittiert werden. Daraufhin werden alle zur Matrix gehörenden Formeln geändert.

Alle inhaltlichen Veränderungen betreffen also immer die gesamte Matrix. Deshalb können einzelne Zellen auch nicht gelöscht oder verschoben werden. Nur die komplette Matrix kann gelöscht oder verschoben werden. Auch das Einfügen von Zellen in eine Matrix, die mit Matrixformeln belegt ist, führt zu einer Fehlermeldung. Dagegen ist es durchaus möglich, den Inhalt einzelner Zellen an eine andere Stelle der Tabelle zu kopieren. Die Bezüge werden dabei automatisch angepasst. Um einen Matrixbereich vollständig zu markieren, genügt es, eine Zelle des Bereichs zu wählen und dann Strg + / zu drücken.

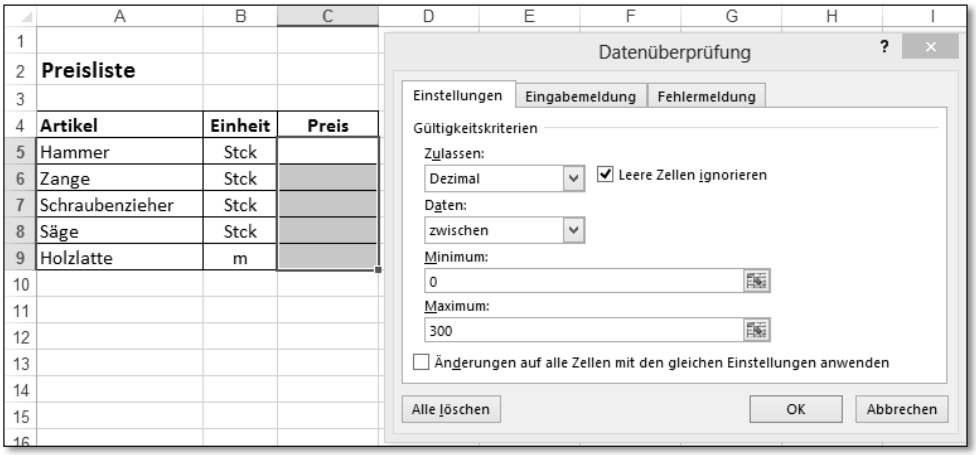
1.19 Verfahren der Fehlervermeidung

Formeln sollen nicht ein beliebiges Ergebnis produzieren, sondern das richtige. Deshalb bietet Excel verschiedene Verfahren für die Fehlervermeidung an, die im Folgenden kurz vorgestellt werden.

1.19.1 Prüfung der Dateneingabe durch Gültigkeitsregeln

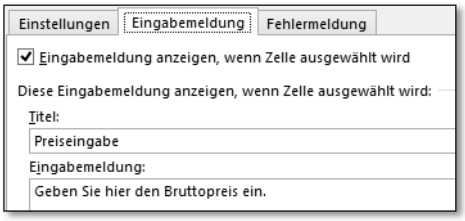
Die Dateneingabe in eine Zelle erfolgt zunächst ohne jede Kontrolle über den Datentyp, den Wert oder die Länge der Eingabe. Excel bietet aber die Möglichkeit, Regeln für Daten festzulegen, die in bestimmten Zellbereichen erlaubt sind. Um beispielsweise bei einer Preisliste auszuschließen, dass irrtümlich ein negativer Preis oder ein Preis, der über dem des teuersten Artikels liegt, eingegeben wird, können für die betreffende Spalte Regeln festgelegt werden.

1. Markieren Sie die Spalte für die Preise.
2. Wählen Sie in Excel 2013 (2010 oder 2007) über **Daten** • **Datentools** die Schaltfläche **Datenüberprüfung**. In den älteren Versionen ist das der Befehl **Daten** • **Gültigkeit**.
3. Auf dem Register **Einstellungen** wählen Sie unter **Zulassen** den Listeneintrag **Dezimal** und unter **Daten** den Eintrag **zwischen**.

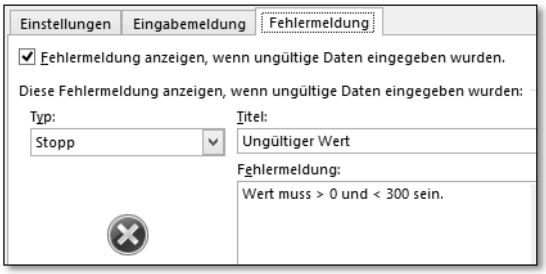


4 Unter **Minimum** kann 0 und unter **Maximum** beispielsweise 300 eingetragen werden.

5 Es ist sinnvoll, **Leere Zellen ignorieren** abzuhaken, damit der Fall, dass der Preis nicht eingegeben ist, nicht zur Fehlermeldung führt.



6 Auf dem Register **Eingabemeldung** kann zusätzlich ein Hinweis eingetragen werden, der angezeigt wird, wenn eine Zelle ausgewählt ist. Lassen Sie dazu **Eingabemeldung anzeigen, wenn Zelle ausgewählt wird** abgehakt.



7 Schließlich sollte auf dem Registerblatt **Fehlermeldung** ein Hinweis bei Fehleingaben eingetragen werden.

8 Unter **Typ** kann das Merkzeichen ausgewählt werden, mit dem die Fehlermeldung beginnt. Wenn Sie **Stopp** wählen, wird verhindert, dass die falsche Eingabe überhaupt in die Zelle eingetragen wird. Sie müssen die Eingabe wiederholen oder abbrechen. Bei den Optionen **Warnung** oder **Information** kann eine unzulässige Eingabe wahlweise zugelassen oder abgelehnt werden.

Wird eine solche Zelle ausgewählt, erscheint die festgelegte Eingabeaufforderung in Form eines Kommentars. Wird nun beispielsweise ein zu hoher Preis eingegeben, erscheint die vorher definierte Fehlermeldung.

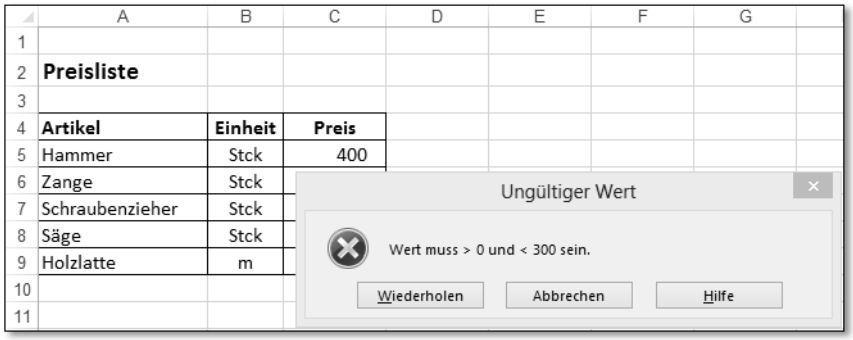


Abbildung 1.72 Fehlermeldung bei einem zu hohen Preis

Zur Datenüberprüfung lassen sich auch Listen verwenden, aus denen die möglichen Eingaben ohne Eintippen übernommen werden können. Wenn Sie in der Preisliste als Gebindeeinheit nur »m«, »cm« und »Stck« zulassen wollen, können Sie auf dem Register **Einstellungen** unter **Zulassen** den Eintrag **Liste** benutzen. Die Liste selbst kann unter **Quelle** eingetragen werden: `m; cm; Stck` – das genügt in diesem Fall. Statt des Eintrags der möglichen Werte kann auch ein Verweis auf einen Zellbereich verwendet werden, in dem sich eine entsprechende Liste befindet. Am besten verwenden Sie den Namen eines benannten Bereichs, damit der Bezug bei einer Vergrößerung der Liste nicht neu eingetragen werden muss. Wird eine Zelle ausgewählt, erscheint eine kleine Schaltfläche, mit der die Liste der möglichen Einträge geöffnet wird. Es reicht dann ein Klick oder Tipp auf den gewünschten Eintrag, um ihn zu übernehmen. Voraussetzung ist allerdings, dass auf der Registerkarte **Einstellungen** die Option **Zellendropdown** abgehakt bleibt.

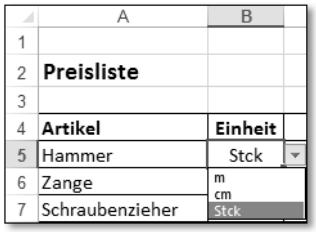


Abbildung 1.73 Auswahlliste zu den Werten für die Spalte »Einheit«

Gültigkeitsregeln können aber auch Formeln verwenden, um die Daten zu prüfen. Wählen Sie dazu unter **Zulassen** die Option **Benutzerdefiniert**, und tragen Sie die Formel komplett in das Feld **Formel** ein.

1.19.2 Fehler in Formeln vermeiden

Es gibt unterschiedliche Fehlertypen, die bei der Arbeit mit Formeln vorkommen. Leicht zu übersehen sind logische Fehler, zumal Excel solchen Fehlern nichts entgegenzusetzen hat. Wenn Sie irrtümlich eine Bedingung wie `=UND(PLZ<50000; PLZ>60000)` eintragen, wird Excel daran nichts auszusetzen haben, obwohl die Bedingung niemals erfüllt werden kann.

Günstiger ist die Situation, wenn Sie die vorgeschriebene Schreibweise von Formeln nicht beachten. Excel bietet in vielen Fällen die automatische Korrektur von Fehlern bei der Formeleingabe an. Wenn Sie z. B. einen Ausdruck mit einer öffnenden Klammer eingeben und dann die abschließende Klammer vergessen, schlägt Excel von selbst eine Korrektur mit einer schließenden Klammer vor.

Für ein gutes Dutzend von immer wieder vorkommenden Fehlern bietet Excel solche Korrekturvorschläge an. Das gilt z. B., wenn Sie bei Zelladressen die Zeilennummer vor den Spaltenbuchstaben setzen und beispielsweise statt `B6` `6B` eingeben. Auch Leerzeichen an der falschen Stelle werden erkannt, z. B. zwischen Funktionsnamen und erster Klammer, ebenso fehlende Anführungszeichen oder Doppelpunkte an der falschen Stelle.

Zwar kann Excel in einfachen Fällen für fehlende Klammern brauchbare Korrekturvorschläge machen, nicht aber da, wo es mehrere Möglichkeiten gibt, die Klammer zu setzen. Um Ihnen bei der Suche nach der Stelle, an der eine Klammer fehlt oder zu viel ist, zu helfen, zeigt Excel zusammengehörende Klammerpaare vorübergehend in Fettschrift

an, wenn Sie mit den Richtungstasten darüberstreichen. Wird etwa eine öffnende Klammer nicht fett angezeigt, heißt das, es fehlt die entsprechende schließende Klammer.

1.19.3 Syntaxprüfung

Wenn Sie eine Formel eingeben, prüft Excel automatisch, ob die Formel von der Syntax her korrekt ist, bevor das Ergebnis in der Zelle akzeptiert wird. Ist z. B. bei der Funktion `GDA()` ein Argument vergessen worden, erscheint die Fehlermeldung, dass Argumente fehlen. Quittieren Sie die Meldung, verbleibt das Programm in der Bearbeitungsleiste und markiert die Stelle, wo der Fehler liegt.

Durch die schon angesprochenen QuickInfos, die bei der Bearbeitung von Funktionen normalerweise eingeblendet werden, ist die Kontrolle über die Argumente der Funktion einfach. Wenn Sie ein Argument in einer Funktion markieren oder die Einfügestelle mit den Pfeiltasten bewegen, wird immer der Platzhalter in der QuickInfo angezeigt, dem das berührte Argument entspricht.

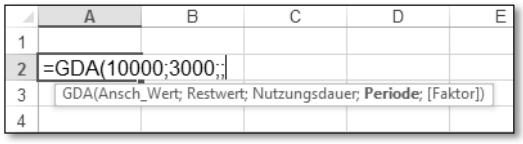


Abbildung 1.74 Kontrolle der Argumente mit Hilfe der QuickInfos

Geben Sie irrtümlich ein Argument zu viel ein, erhalten Sie eine entsprechende Meldung, und Sie werden nicht wieder aus der Bearbeitungsleiste losgelassen.

Wenn Sie einen Fehler nicht gleich erkennen, können Sie Excel überlisten und an den Anfang des Eintrags ein einfaches Anführungszeichen setzen. Dadurch wird die Formel zu einem Texteintrag. Nun haben Sie wenigstens die Chance, sich eine Denkpause zu gönnen.

1.19.4 Fehler durch Werte

Excel kann neben den Syntaxfehlern auch Fehler finden, die mit den Werten bzw. Argumenten zu tun haben, mit denen die Formel arbeiten soll. Sind diese für die Formel nicht brauchbar, wird in der Zelle ein Fehlerwert ausgegeben. Sie erhalten einen Hinweis, wenn Sie den Mauszeiger auf die eingeblendete Schaltfläche rücken. In der Formel in Abbildung 1.75 wird z. B. ein nicht definierter Name verwendet.

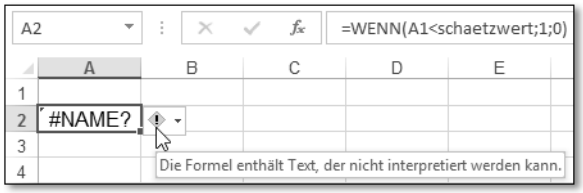


Abbildung 1.75 Hinweis zum Fehlerwert `#NAME?`

In der folgenden Liste sind die Fehlerwerte zusammengestellt:

Fehler	Bedeutung
`#BEZUG!`	Die Formel bezieht sich auf eine gelöschte oder verschobene Zelle. Beim Verschieben von relativen Adressen kommt dies beispielsweise vor, wenn eine Zelle nach links verschoben wird, so dass der relative Bezug auf eine Spalte links von A benötigt würde. Erscheint auch bei externen Bezügen, wenn die externe Datei nicht auffindbar ist.
`#DIV/0!`	In der Formel wird versucht, durch null zu teilen. Tritt auch auf, wenn die Zelle mit dem Divisor leer ist.
`#NV`	Der erwartete Wert ist nicht vorhanden. Tritt beispielsweise auf, wenn ein mit Hilfe von `SVERWEIS()` in einer Spalte gesuchter Wert nicht vorhanden ist.
`#NAME?`	Der in der Formel verwendete Name ist bisher nicht definiert worden. Tritt auch auf, wenn ein Funktionsname falsch geschrieben wird.
`#NULL!`	Wird angezeigt, wenn versucht wurde, mit dem Schnittmengenoperator (Leerzeichen) die Schnittmenge zweier Bereiche zu bestimmen, die sich nicht schneiden.
`#WERT!`	Ein Argument oder Operand hat den falschen Datentyp.
`#ZAHL!`	Tritt auf, wenn beispielsweise ein Schätzwert in einer Funktion, die einen solchen Wert erwartet, unbrauchbar ist oder wenn die Funktion als Argument einen negativen Wert erwartet, aber einen positiven Wert erhält.

Tabelle 1.11 Übersicht möglicher Fehlerwerte

1.19.5 Fehlerüberprüfung im Hintergrund

Seit Excel 2002 ist eine automatische Fehlerüberprüfung eingebaut, die im Hintergrund abläuft und in Excel 2013 über die Seite **Formeln** im **Excel-Optionen**-Dialog gesteuert wird. In den älteren Versionen benutzen Sie **Extras • Optionen • Fehlerüberprüfung**. Un-

ter **Regeln für die Fehlerüberprüfung** wird festgelegt, wann die Fehlerüberprüfung den Fehlerindikator einblenden soll.

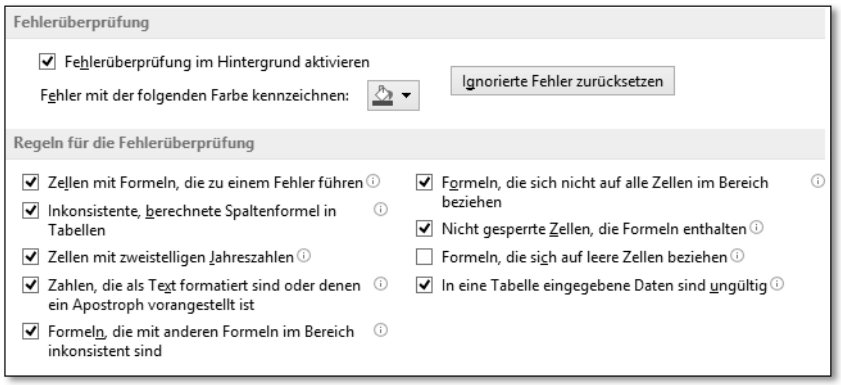


Abbildung 1.76 Einstellungen für die Fehlerüberprüfung

Solange unter **Fehlerüberprüfung** eben diese Fehlerüberprüfung im Hintergrund aktiviert ist, erscheinen im Fehlerfall farbige Fehlerindikatoren. Über den Pfeil der Schaltfläche werden mögliche Reaktionen angeboten.

Die erste Zeile enthält eine Problembeschreibung. Wenn möglich, wird eine Option angeboten, mit der der Fehler direkt korrigiert werden kann, etwa eine Bereichserweiterung. In diesem Fall handelt es sich allerdings um falschen Alarm. Die Warnung moniert, dass die Jahreszahl nicht mit in den Summenbereich aufgenommen wird, weil Excel die Bedeutung der Zahl als Jahreszahl nicht automatisch erkennen kann. Mit **Fehler ignorieren** kann der Hinweis deshalb abgeschaltet werden.

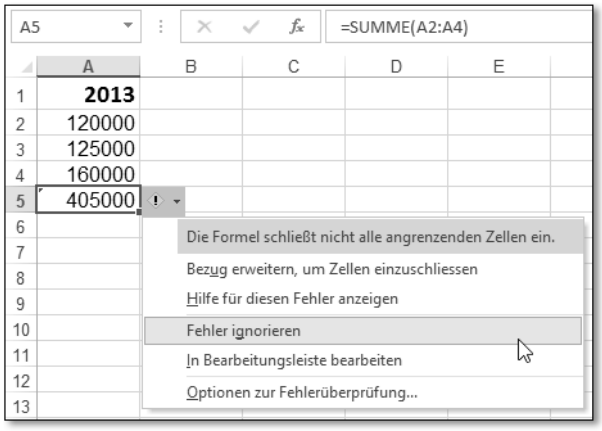


Abbildung 1.77 Optionen beim Auftauchen eines Regelverstoßes

1.19.6 Formelüberwachung

Für die Kontrolle von Formeln stellt Excel 2013 neben der gerade beschriebenen Hintergrundüberprüfung noch eine ganze Reihe von Werkzeugen zur Verfügung, die auf dem Register **Formeln** in der Gruppe **Formelüberwachung** zusammengestellt ist. Die entsprechenden Symbole finden Sie in den älteren Versionen in der Symbolleiste **Formelüberwachung**.

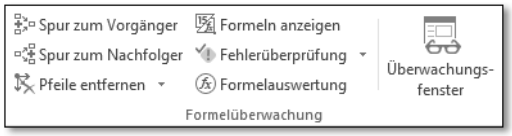


Abbildung 1.78 Werkzeuge für die Kontrolle von Formeln

Die Fehlerüberprüfung kann hier für jedes Arbeitsblatt auch manuell über die Schaltfläche **Fehlerüberprüfung** aufgerufen werden, unabhängig davon, ob die Fehlerüberprüfung im Hintergrund aktiviert ist oder nicht. Mit **Weiter** und **Zurück** lassen sich die Zellen mit Fehlerwerten oder Warnindikatoren einzeln ansteuern, der Dialog zeigt dann jeweils eine kurze Beschreibung des Fehlers und bietet Schaltflächen für das weitere Vorgehen an.

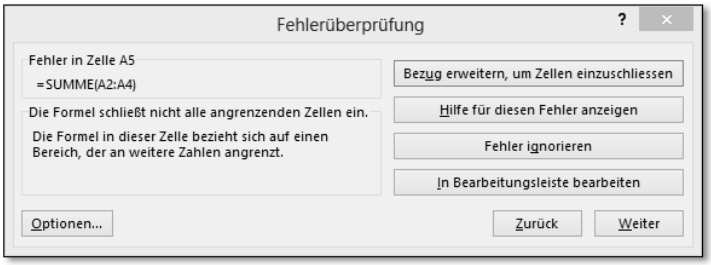
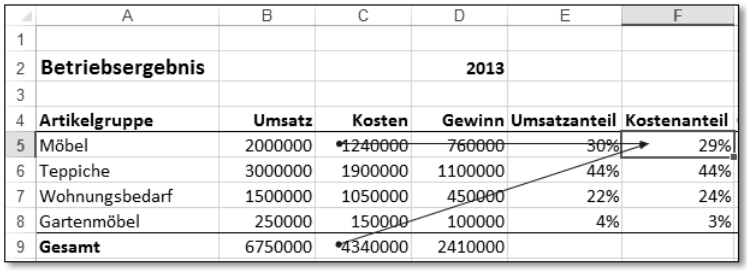


Abbildung 1.79 Schrittweise Fehlerüberprüfung im Blatt

1.19.7 Spuren verfolgen

Wollen Sie prüfen, von welchen Zellen eine gerade ausgewählte Formel abhängig ist, benutzen Sie in der Gruppe **Formelüberwachung** das Symbol **Spur zum Vorgänger**. Excel macht die Verbindungen zwischen Zellen mit Pfeilen sichtbar. Das Pfeilende wird jeweils durch einen Punkt gekennzeichnet. Hat der Vorgänger selbst noch einen Vorgänger, wiederholen Sie den Befehl. Sie können dieses Verfahren für mehrere Zellen ausführen, die Pfeile bleiben stehen, bis sie gelöscht werden. Wenn Sie die Zelle am Pfeilende markieren wollen, klicken oder tippen Sie doppelt auf den Spurpfeil. Ein erneuter Doppelklick oder -tipp markiert wieder die Zelle an der Pfeilspitze. Das ist praktisch, wenn Pfeilspitze und Pfeilende weit auseinanderliegen.



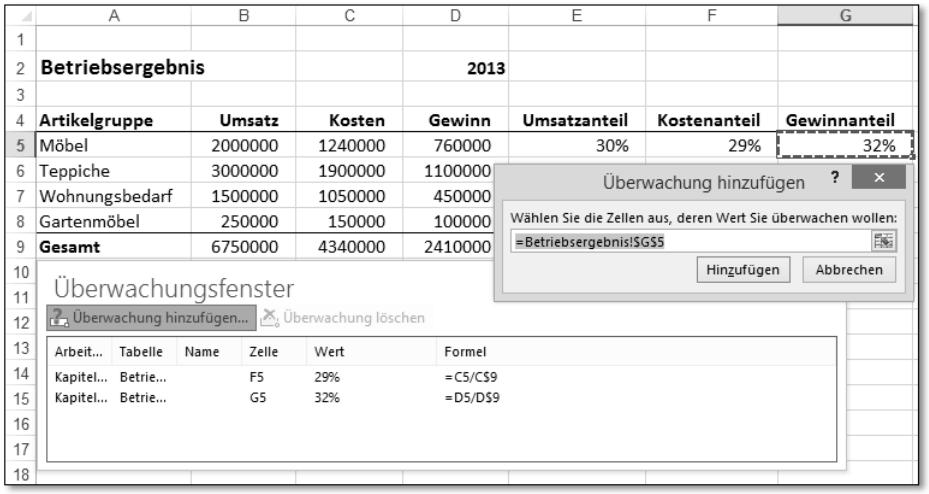
	A	B	C	D	E	F
1						
2	**Betriebsergebnis**			**2013**		
3						
4	**Artikelgruppe**	**Umsatz**	**Kosten**	**Gewinn**	**Umsatzanteil**	**Kostenanteil**
5	Möbel	2000000	1240000	760000	30%	29%
6	Teppiche	3000000	1900000	1100000	44%	44%
7	Wohnungsbedarf	1500000	1050000	450000	22%	24%
8	Gartenmöbel	250000	150000	100000	4%	3%
9	**Gesamt**	6750000	4340000	2410000		

Abbildung 1.80 Anzeige der Zellen, von denen die Formel in F5 abhängig ist

Wollen Sie umgekehrt wissen, welche Zellen die markierte Zelle beeinflussen, benutzen Sie das Symbol **Spur zum Nachfolger**. Gelöscht werden die Pfeile mit dem Befehl **Pfeile entfernen**, wobei Sie über das Menü der Schaltfläche entscheiden können, ob Sie gleich alle Pfeile entfernen wollen oder nur die Vorgänger- oder Nachfolgerpfeile.

1.19.8 Werteprüfung im Überwachungsfenster

Eine andere Methode der Fehlerentdeckung ist die Überwachung von Formeln. In Excel 2013 wird mit **Formeln • Formelüberwachung • Überwachungsfenster** eben dieses eingeblendet. Zunächst lassen sich dann über die dortige Schaltfläche **Überwachung hinzufügen** beliebige Zellen oder Zellbereiche auswählen, deren jeweils aktuelle Werte im Überwachungsfenster kontrolliert werden (siehe Abbildung 1.81). Jede Änderung von Werten wird anschließend sofort in diesem Fenster angezeigt.



	A	B	C	D	E	F	G
1							
2	**Betriebsergebnis**			**2013**			
3							
4	**Artikelgruppe**	**Umsatz**	**Kosten**	**Gewinn**	**Umsatzanteil**	**Kostenanteil**	**Gewinnanteil**
5	Möbel	2000000	1240000	760000	30%	29%	32%
6	Teppiche	3000000	1900000	1100000			
7	Wohnungsbedarf	1500000	1050000	450000			
8	Gartenmöbel	250000	150000	100000			
9	**Gesamt**	6750000	4340000	2410000			

Abbildung 1.81 Werteprüfung im Überwachungsfenster

1.19.9 Zirkuläre Formeln

Gewarnt werden Sie von Excel auch, wenn eine Formel zirkulär ist. Hier ein einfaches Beispiel. In der Zelle B10 ist die Funktion `=SUMME(B6:B10)` eingetragen. Die Zelle B10, die die Summe anzeigen soll, gehört selbst zu der Kolonne, die summiert wird. Wenn Sie versuchen, die Summenformel zu bestätigen, erhalten Sie folgende Fehlermeldung:

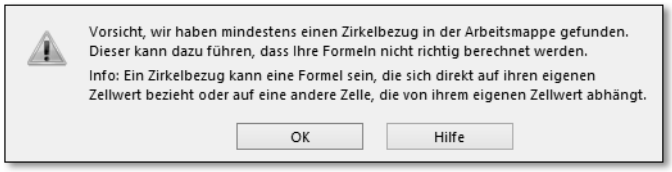


Abbildung 1.82 Fehlermeldung bei Zirkelbezug

In der Statusleiste wird die Zelle angezeigt, die den Zirkelbezug verursacht. Die Zelle, die die zirkuläre Formel enthält, wird auf den Wert Null gesetzt.

1.19.10 Formeln schrittweise prüfen

Eine weitere Möglichkeit, Fehler aufzuspüren, besteht darin, sich für einzelne Teile der Formel jeweils das Ergebnis anzeigen zu lassen. Markieren Sie den betreffenden Teil der Formel und benutzen Sie dann F9. Excel zeigt das Ergebnis für diesen Teil der Formel an. Benutzen Sie Esc, um die Formel wieder in den alten Zustand zu versetzen.

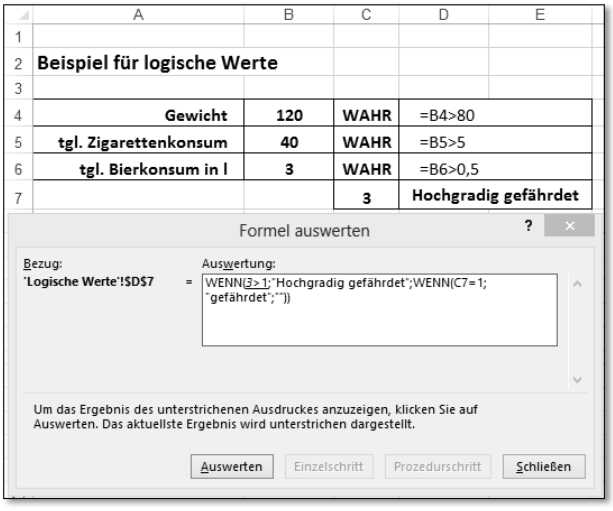


Abbildung 1.83 Schrittweise Auswertung einer verschachtelten Funktion

Excel 2013 bietet alternativ dazu den Befehl **Formelauswertung**, den Sie über **Formeln • Formelüberwachung** erreichen (früher: **Extras • Formelüberwachung • Formelauswertung**). Im Dialog wird die vorher markierte Formel angezeigt. Mit der Schaltfläche **Auswerten** lässt sich der jeweils unterstrichene Teil einer Formel berechnen. **Einzelschritt** zeigt den Wert der markierten Stelle, **Prozedurschritt** setzt diesen Wert in die Formel ein. Die Formel in der Zelle selbst bleibt bei dieser Prüfung aber im Unterschied zu dem Verfahren mit F9 unverändert.

1.20 Berechnungseinstellung und -optimierung

Normalerweise berechnet Excel unaufgefordert alle die Formeln neu durch, die von der Neueingabe oder Änderung von Werten oder Formeln oder von neuen Namensfestlegungen betroffen sind. Um die Rechenvorgänge zu beschleunigen, werden Formeln, die nicht von einer Änderung berührt sind, also nicht jedes Mal neu berechnet.

1.20.1 Berechnungsoptionen

Alle Optionen, die die Berechnungsmethode von Excel 2013 beeinflussen, sind auf der Seite **Formeln** zusammengestellt, das Sie über **Datei • Optionen** erreichen (in früheren Versionen: **Extras • Optionen • Berechnung**).

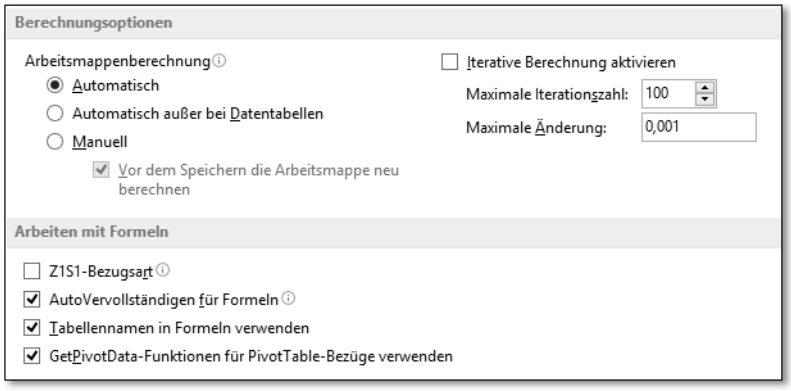


Abbildung 1.84 Wahl der Berechnungsoptionen

Hauptsächlich besteht unter **Berechnungsoptionen** die Wahl zwischen **Automatisch** und **Manuell**. Diese Einstellung gilt immer so lange, bis sie geändert wird. Die automatische Neuberechnung ist in Excel 2013 zwar auf die Zellen beschränkt, die von einer Daten-

änderung berührt werden, dennoch kann es beispielsweise bei Tabellen mit Zehntausenden von Zeilen sinnvoll sein, die automatische Neuberechnung auszuschalten, damit nicht bei jeder neu eingegebenen Zahl etwa die Spaltensummen nachgerechnet werden. Der Kompromiss für diesen Fall ist die Einstellung **Automatisch außer bei Datentabellen**.

Bei der Option **Manuell** sollte allerdings **Vor dem Speichern die Arbeitsmappe neu berechnen** in der Regel aktiviert bleiben; dies garantiert, dass die Datei immer vollständig durchgerechnet ist, wenn sie abgespeichert wird. Wenn Sie allerdings an einer großen Arbeitsmappe arbeiten und häufiger zwischendurch abspeichern, kann es doch angenehmer sein, das Kontrollkästchen leer zu lassen.

Für den schnellen Wechsel zwischen den Berechnungseinstellungen können Sie auf dem Register **Formeln** auch das Menü der Schaltfläche **Berechnungsoptionen** in der Gruppe **Berechnung** nutzen.

Ist die manuelle Neuberechnung eingeschaltet, wird die Neuberechnung der gerade offenen Arbeitsmappen mit F9 oder über das Symbol **Neu berechnen** in der Gruppe **Formeln • Berechnung** gestartet. Soll nur das aktive Arbeitsblatt neu berechnet werden, kann auch die Schaltfläche **Blatt berechnen** verwendet oder die Tastenkombination ⇧ + F9 gedrückt werden. Sobald in der Arbeitsmappe ein Wert eingegeben wird, der eine Formel betrifft, erscheint in der Statusleiste die Aufforderung **Berechnen**. Klick oder Tipp auf diese Schaltfläche startet ebenfalls die Neuberechnung.

Wenn Sie eine Neuberechnung aller Formeln einer Arbeitsmappe erzwingen wollen, benutzen Sie Strg + Alt + ⇧ + F9.

1.20.2 Kontrolle iterativer Berechnungen

Iterative Berechnungen sind wiederholte Berechnungen, die mit Zirkelbezügen arbeiten. Von Zirkelbezügen wird bei Excel gesprochen, wenn z.B. in einer Formel in der Zelle C7 ein Operand oder ein Argument `C7` auftaucht. Die Formel bezieht sich also in diesem Fall auf sich selbst, die Berechnung läuft im Kreis. Solche Zirkelbezüge können irrtümlich – durch die Markierung der falschen Zelle – entstehen. Sie können aber auch bewusst eingesetzt werden, um z.B. Näherungswerte zu errechnen. Dabei wird angenommen, dass durch Wiederholung der Berechnung allmählich eine Annäherung an einen Wert erfolgt, der sich auch bei erneuter Neuberechnung nicht mehr oder nicht mehr wesentlich verändert.

Wird unter **Berechnungsoptionen** das Kontrollkästchen **Iterative Berechnung aktivieren** abgehakt, ist eine solche wiederholte Berechnung zugelassen. Dabei kann die maximale

Anzahl der Wiederholungen bestimmt werden. Excel gibt unter **Maximale Iterationszahl** den Wert 100 vor. Die Berechnung wird also spätestens nach der 100. Wiederholung gestoppt und das erreichte Ergebnis angezeigt.

Der zweite Wert, mit dem die iterative Berechnung gesteuert werden kann, ist **Maximale Änderung**. Vorgabe ist 0,001. Damit ist gemeint, dass die Berechnung dann nicht mehr wiederholt wird, wenn die verschiedenen Berechnungen zu Ergebnissen führen, deren Differenz kleiner ist als der Änderungshöchstwert. Welcher Wert hier der richtige ist, hängt natürlich von der Dimension ab, die das erwartete Ergebnis hat. Erwarten Sie z. B. Werte mit sechs Nachkommastellen, ist 0,001 ein zu grobes Maß.

1.20.3 Berechnungsoptionen für die Arbeitsmappe

Während sich die bisher behandelten Optionen auf das Verhalten in der gesamten Arbeitssitzung beziehen, können Sie im Dialog **Excel-Optionen** unter **Erweitert** auch Berechnungsoptionen für jede einzelne Arbeitsmappe festlegen. In den älteren Versionen finden Sie diese Einstellungen auf dem Register **Extras • Optionen • Berechnung** unter **Arbeitsmappenoptionen**.

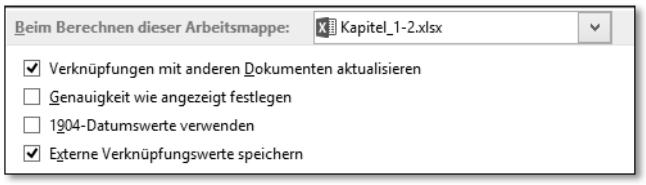


Abbildung 1.85 Berechnungsoptionen für eine Arbeitsmappe

Interessant für die praktische Arbeit ist insbesondere die Option **Genauigkeit wie angezeigt festlegen**, die normalerweise nicht ausgewählt ist. Wird die Option gewählt, ersetzt Excel in der betreffenden Arbeitsmappe die intern gespeicherten Werte durch die aufgrund der gewählten Formatierung angezeigten Werte.

Die Wahl der Option **1904-Datumswerte verwenden** bedeutet, dass die Datumsseriennummer nicht ab dem 1.1.1900 (Tag 1), sondern ab dem 1.1.1904 (Tag 0) berechnet wird. Dieses Datumssystem wird auf dem *Apple Macintosh* verwendet. Apple vermied damit das Problem, dass das Jahr 1900 kein Schaltjahr war. Bei der Berechnung ab 1900 wurde dies ignoriert, weshalb das Datum 29.2.1900 fälschlicherweise akzeptiert wird. (In dem neuen Datenformat *Strict Open XML* ist das – wie bereits erwähnt – nicht mehr der Fall.)

Wenn Sie häufiger Dateien mit Anwendern austauschen, die Office auf einem System von Apple verwenden, kann es praktisch sein, diese Einstellung zu verwenden. Beachten Sie aber, dass ein nachträglicher Wechsel für bereits eingetragene Datumswerte in einer Arbeitsmappe zu einer entsprechenden Verschiebung um 1462 Tage führt, so dass Korrekturen durch Addieren oder Subtrahieren mit diesem Wert notwendig werden. Am besten verwenden Sie dazu die Funktion **Einfügen • Inhalte einfügen • Addieren** oder **Subtrahieren**.

Ein weiterer Grund, das alternative Datumssystem zu verwenden, hängt damit zusammen, dass das von Microsoft bevorzugte Datumssystem ab dem 1.1.1900 auch in der Version Excel 2013 – es ist schwer zu begreifen – immer noch nicht in der Lage ist, negative Zeitwerte darzustellen. Wenn Sie beispielsweise in Zelle B4 die Zeit `20:30` ablegen und in B5 `23:30`, liefert die Formel `=B4-B3` einen Wert, den das Programm nicht anzeigen kann, die Zelle wird mit #-Zeichen gefüllt. Verwenden Sie dagegen das alternative Datumssystem, liefert die Formel als Ergebnis einen negativen Zeitwert: `-03:00`. Bei Arbeitszeitberechnungen, wo häufig mit negativen Zeitwerten zu rechnen ist, sollten Sie also die 1904-Datumswerte verwenden.

In der Gruppe **Formeln** finden Sie auf dem Register **Erweitert** außerdem eine Option, die die Beschleunigung der Berechnungen in großen Arbeitsmappen betrifft. Seit Excel 2007 besteht die Möglichkeit, die Berechnungsarbeit auf parallele Threads – Ausführungsstränge – aufzuteilen. Wenn Sie mehrkernige Prozessoren verwenden, sollte diese Option deshalb aktiviert sein. Insbesondere bei Arbeitsmappen mit zahlreichen voneinander unabhängigen Formelketten kann die Verteilung der Lösung auf mehrere Kerne für deutlich schnellere Berechnungen sorgen.

1.21 Arbeit mit bedingten Formaten

Einer der Bereiche, in denen Excel seit der Version 2007 gegenüber den vorhergehenden Versionen wesentlich zugelegt hat, ist der Einsatz bedingter Formate. In Excel 2013 wurden diese Optionen noch einmal stark erweitert. Die Schaltfläche **Bedingte Formatierung** in der Gruppe **Start • Formatvorlagen** bietet gleich mehrere Paletten an, um bestimmte Zellen aufgrund ihrer Werte in besonderer Weise hervorzuheben. Über die Möglichkeiten hinaus, die bis Excel 2003 über **Format • Bedingte Formatierung** zur Verfügung standen, wird der Einsatz von Datenbalken, Farbskalen und Symbolsätzen angeboten. Abbildung 1.86 zeigt ein einfaches Beispiel für diese drei Varianten:

	A	B	C	D
1		Spalte mit Datenbalken	Spalte mit Farbskala	Spalte mit Symbolen
2		⇩	⇩	⇩
3	**Warengruppe**	**2010**	**2011**	**2012**
4	CD-ROM-Laufwerke	120000	132000	108000
5	Soundkarten	90000	88000	144000
6	Scanner	145000	159500	160800
7	Videorecorder	156000	13200	174000
8	Camcorder	230000	253000	187200
9	Videokarten	134000	147400	276000

Abbildung 1.86 Varianten für bedingte Formatierung

Die erste Variante, die als **Datenbalken** bezeichnet wird, verwendet eine Visualisierung der Wertgrößen, die dem Einsatz von Balkendiagrammen ähnlich ist. Jede Änderung von Werten in einer Spalte führt sofort zu einer entsprechenden Anpassung der Farbbalken. Änderungen der Spaltenbreite werden dabei ebenfalls automatisch berücksichtigt.

Wenn es um die Kennzeichnung bestimmter Werte geht, können Farbskalen genutzt werden, etwa um allen Werten, die 10% über dem Mittelwert liegen, eine bestimmte Farbe zuzuordnen. Dazu kann eine entsprechende Formel benutzt werden, wie Abbildung 1.87 zeigt:

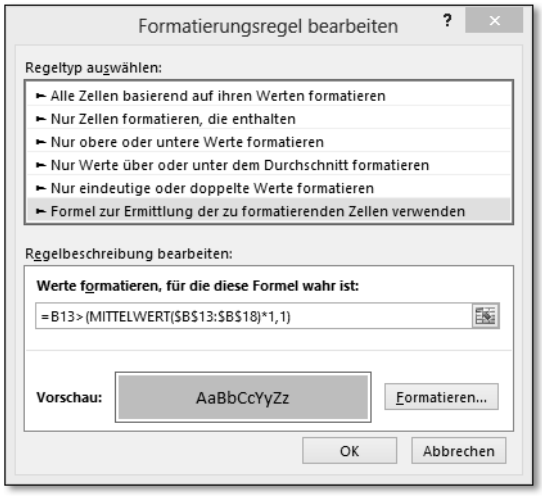


Abbildung 1.87 Bedingtes Format auf der Basis einer Formel

Die Formel vergleicht die erste Zelle in der Spalte mit dem Mittelwert der Spalte. Die erste Adresse muss relativ, die zweite absolut eingegeben werden, damit Excel in allen Zellen der Spalte die Formel korrekt auswerten kann.

Eine weitere Möglichkeit für bedingte Formate ist der Einsatz von Symbolen, etwa Pfeile, die anzeigen, welche Werte über oder unter bestimmten Schwellenwerten liegen. Diese Schwellenwerte können Sie in dem Dialog für die Formatierungsregeln in mehreren Stufen definieren. Abbildung 1.88 zeigt ein Beispiel.

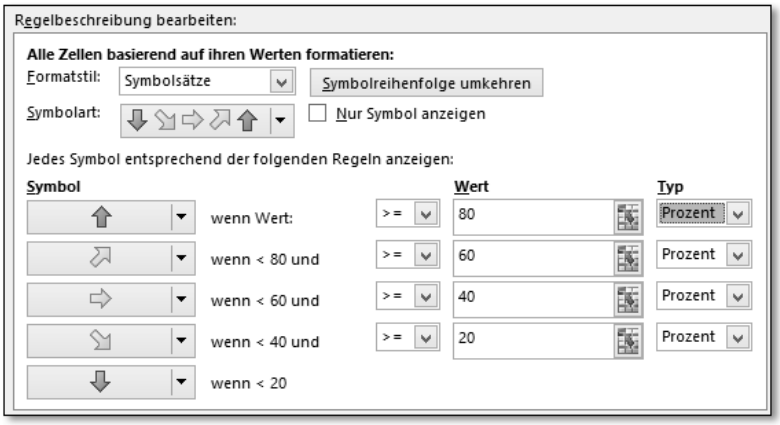


Abbildung 1.88 Zuordnen von Wertbereichen zu den Symbolen

Der so formatierte Bereich zeigt die Symbole vor den Werten an.

12	**Warengruppe**	**2012**
13	CD-ROM-Laufwerke	108000
14	Soundkarten	144000
15	Scanner	160800
16	Videorecorder	174000
17	Camcorder	187200
18	Videokarten	276000

Abbildung 1.89 Bedingtes Format mit Pfeilsymbolen

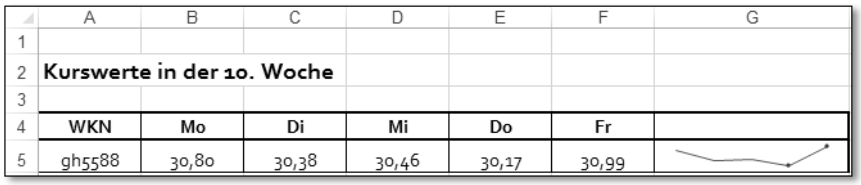
1.22 Exkurs über die Visualisierung von Daten: Sparklines und Diagramme

Die im letzten Abschnitt beschriebene Möglichkeit, Tabellen mit bedingten Formaten zu belegen, ist bereits eine der Formen, Informationen mit Hilfe von visuellen Elementen aufzubereiten. Dieses Thema leitet über zu den beiden Hauptmethoden, die Excel für die Visualisierung von Daten bereitstellt. Die eine sind Sparklines, eine erst seit Excel 2010 angebotene Möglichkeit, die andere sind Diagramme.

Im Rahmen dieses Buches, das sich ja bewusst auf die Berechnungsmethoden konzentriert, soll wenigstens kurz auf die grundlegenden Verfahren eingegangen werden.

1.22.1 Sparklines

Sparklines sind eine Alternative zu ausgewachsenen Diagrammen, wenn es um eine kompakte Darstellung geht. Sie sind an der Vorstellung orientiert, dass der Text und die Daten, die den Bedeutungszusammenhang liefern, und die Visualisierung der Daten möglichst eng verknüpft bleiben sollen. Hilfreich sind Sparklines hauptsächlich da, wo Zahlen vorhanden sind, die Entwicklungsreihen oder Trends darstellen. Ein praktisches Beispiel ist eine Zeile mit Kursdaten für eine Woche:



	A	B	C	D	E	F	G
1							
2	**Kurswerte in der 10. Woche**						
3							
4	**WKN**	**Mo**	**Di**	**Mi**	**Do**	**Fr**	
5	gh5588	30,80	30,38	30,46	30,17	30,99	

Abbildung 1.90 Datenzeile mit einer Sparkline

Die erste Zelle in Zeile 5 gibt an, um welches Wertpapier es geht, die fünf folgenden Zellen liefern die Tageswerte, und die Sparkline in der Nachbarzelle visualisiert diese Daten. Die Reihenfolge und die Anordnung der drei Elemente lassen sich variieren, normalerweise sollten Beschriftung, Zahlen und Grafik aber nicht zu weit auseinandergerissen werden. Ändern sich die Zahlen, wird die Grafik wie üblich sofort angepasst. Beim Drucken werden die Sparklines wie Zellinhalte behandelt, also auf jeden Fall mitgedruckt.

Anders als Diagramme sind Sparklines aber keine grafischen Objekte, sie sind Inhalt einer Zelle, auch wenn dieser Inhalt in der Bearbeitungszeile nicht erscheint. Die Sparkline bildet einen zusätzlichen Hintergrund einer Zelle. Es ist also möglich, in diese Zelle auch noch einen Text einzugeben, etwa um die Grafik zu beschriften. Als unterste Ebene darf aber auch noch ein anderer Hintergrund unter die Sparkline gelegt werden!

Die Bindung an die Zelle hat auch zur Folge, dass jede Änderung der Höhe oder Breite der Zelle sofort von der Sparkline mitgemacht wird. Außerdem lässt sich auch ein Zellverbund als Raum für eine Sparkline verwenden. Wenn Sie also mehrere Zellen mit dem Symbol **Verbinden und Zentrieren** zusammenfügen, wird eine Sparkline, die in der ersten Zelle angesiedelt ist, auf den ganzen Bereich ausgedehnt. Wird eine Zelle mit einer Sparkline verschoben, ändern sich bei relativen Zellbezügen eben auch diese.

Wozu lassen sich Sparklines verwenden? Typische Anwendungsbereiche sind Daten, die Entwicklungen anzeigen, saisonale Schwankungen oder Verteilungen im Zeitverlauf, beispielsweise:

- Kursentwicklungen
- Preisentwicklungen
- Umsatz-, Absatz- oder Gewinnentwicklungen
- Temperaturdaten im Zeitverlauf
- politische Verteilungen

Im Folgenden wird gezeigt, wie Sparklines einer Tabelle mit mehreren Kurswerten zugeordnet werden. Die erste Spalte enthält jeweils die WKN-Nummer der Papiere, die folgenden Spalten die Tageskurse.

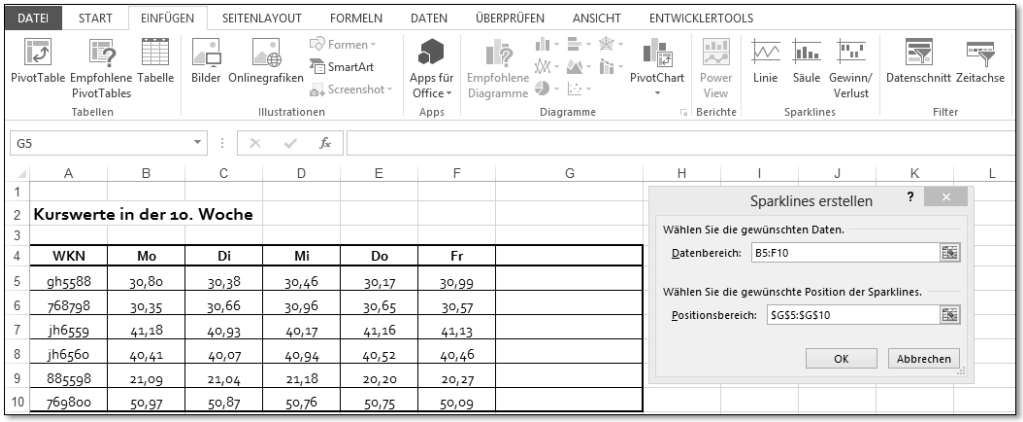


Abbildung 1.91 Erstellen einer Gruppe von Sparklines

1 Zunächst markieren Sie den Zellbereich, der die Sparklines aufnehmen soll, im Beispiel ist es G5:G10.

2 Wählen Sie über **Einfügen • Sparklines** das Symbol für **Linie**.

3 Im Dialog **Sparklines erstellen** geben Sie unter **Datenbereich** den Bereich der Kursdaten an, die als Sparklines visualisiert werden sollen.

4 Da der Positionsbereich für die Sparklines durch die Auswahl in Schritt 1 bereits bestimmt ist, kann der Dialog quittiert werden.

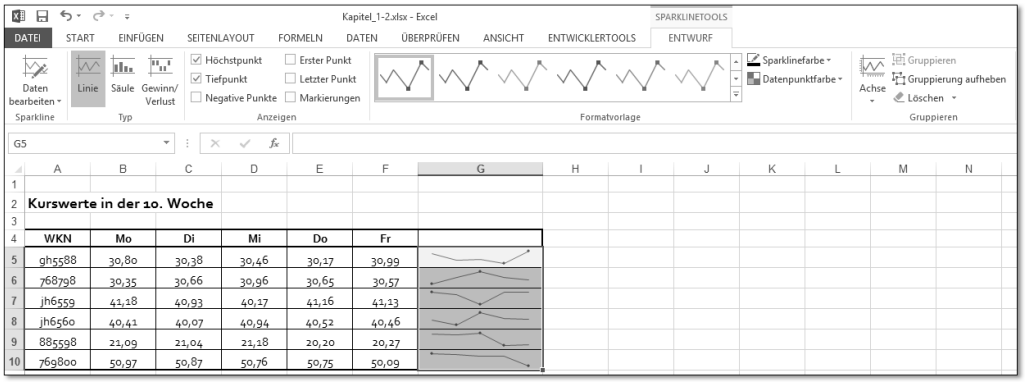


Abbildung 1.92 Die Gruppe der Sparklines und die Werkzeuge zu ihrer Bearbeitung

Excel 2013 fügt die Sparklines in den ausgewählten Zellbereich als Gruppe ein und bietet anschließend im Menüband das Register **Sparklinetools Entwurf** an. Hier finden Sie unter **Anzeigen** verschiedene Optionen, um bestimmte Punkte der Sparklines hervorzuheben. In diesem Fall ist es sinnvoll, zumindest den Höchstpunkt und den Tiefpunkt hervorzuheben.

Je nach der Wahl unter **Anzeigen** werden in der Gruppe **Formatvorlage** unterschiedliche Gestaltungsmuster angeboten. Ein Klick oder Tipp ordnet das Format zu. Alternativ oder ergänzend zu den Formaten einer gewählten Vorlage lassen sich über die Schaltflächen **Sparklinefarbe** und **Datenpunktfarbe** eigene Farben aus der Farbpalette auswählen. Bei den Datenpunkten geschieht dies separat für jeden Punkttyp, wobei die aktuelle Einstellung in dem Menü dieser Typen angezeigt wird. Auf dem Touchscreen sind die entsprechenden Paletten vergrößert, damit Sie die Muster leichter treffen können.

In der Gruppe **Gruppieren** wird noch ein Menü zu der Schaltfläche **Achse** angeboten, das Einstellungen für die Gestaltung der horizontalen und der vertikalen Achse anbietet. Hier lassen sich insbesondere die Minimal- und Höchstwerte der vertikalen Achse ändern, um eine andere Skalierung zu erreichen, etwa um die Unterschiede deutlicher zu machen. Wenn negative Werte vorkommen, lässt sich auch die horizontale Achse einblenden, um den Übergang zu den Werten unter null deutlich zu machen. Durch Veränderung der Spaltenbreite und der Zeilenhöhe kann die Zelle bei Bedarf in eine Größe gebracht werden, die die Diagrammlinien möglichst gut lesbar macht.

1.22.2 Darstellungsvarianten

Excel 2013 stellt drei Typen von Sparklines zur Verfügung, wobei es für jeden Typ wieder zahlreiche Formatvorlagen und Möglichkeiten gibt, wie er angepasst werden kann.

1. Der Typ **Linie** entspricht einem normalen Liniendiagramm aus den diskreten Werten des als Datenbasis angegebenen Zellbereichs, eingebettet in eine Zelle. Allerdings fehlen alle Details, die Achsen sind in der Regel nicht zu sehen, Legenden oder Betitelungen fehlen ebenfalls.
2. Beim Typ **Säule** wird ein einfaches Balkendiagramm ohne Beschriftungen erzeugt. Die negativen Werte werden von der gedachten Achse nach unten angezeigt. Bei Bedarf kann diese Achse in Form einer einfachen Linie auch eingeblendet werden. Dies geschieht über das Menü **Achse** in der Gruppe **Entwurf • Gruppieren**.
3. Beim Typ **Gewinn/Verlust** wird für jede Zelle im Datenbereich mit gleichbleibend großen Balken nur angezeigt, ob es sich um einen Gewinn oder um einen Verlust handelt. Auf diese etwas grobe Weise können beispielsweise verschiedene Jahre im Vergleich gekennzeichnet werden.

Abbildung 1.93 zeigt die beiden letzten Varianten an einem Beispiel.

	A	B	C	D	E	F
3	Artikelgruppe	Umsatz	Kosten	Gewinn	**Säulen**	**Gewinn/Verlust**
4	Möbel	2000000	1240000	760000		
5	Teppiche	3000000	1900000	1100000		
6	Wohnungsbedarf	1500000	1800000	-300000		
7	Gartenmöbel	250000	150000	100000		

Abbildung 1.93 Sparklines vom Typ **Säule** mit eingeblendeter Achse bei negativen Werten und vom Typ **Gewinn/Verlust**

1.22.3 Neuerungen für die Diagrammgestaltung

Das bevorzugte Instrument, um Zahlenmaterial auf seinen Kern zu konzentrieren, ist die Übersetzung in eine grafische Präsentation. Was die Zahlen zu sagen haben, kann im Diagramm auf einen Blick erfasst werden. Für Excel 2013 hat Microsoft die Werkzeuge dafür noch einmal rundum überarbeitet. Neu ist ein Ratgeber, den Sie über die Schaltfläche **Empfohlene Diagramme** auf dem Register **Einfügen** starten. Er unterstützt Sie bei der Wahl des für die ausgewählten Daten am besten geeigneten Diagrammtyps.

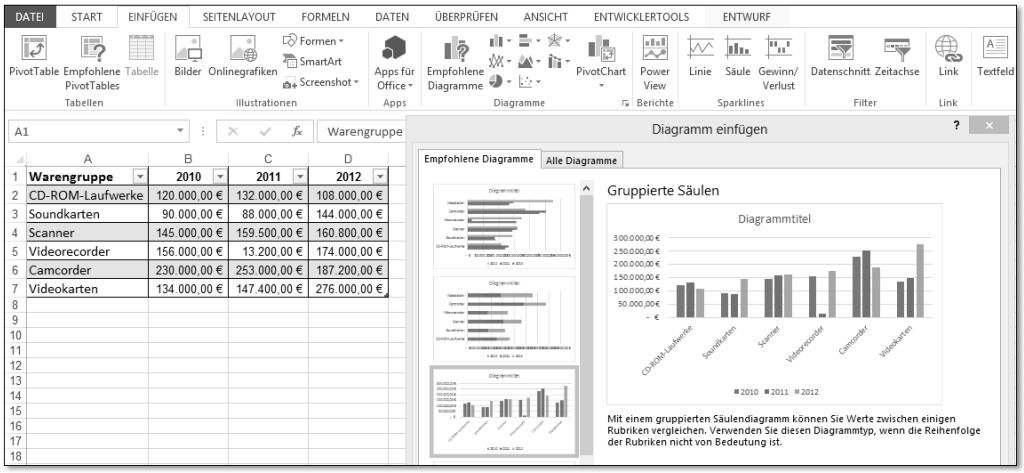


Abbildung 1.94 Der Befehl **Empfohlene Diagramme** hilft bei der Wahl des passenden Diagrammtyps.

Die Werkzeuge im Menüband, die der Gestaltung von Diagrammen dienen, sind jetzt auf zwei Register konzentriert worden, was die Bedienung übersichtlicher macht. Die Formatierung des Diagramms wird nicht mehr über Dialogfelder, sondern über entsprechende Aufgabenbereiche vorgenommen, was insbesondere auch die Bedienung auf einem Touchscreen sicher erleichtert. Neu sind auch erweiterte Möglichkeiten, Informationen zu Datenpunkten übersichtlich einzufügen. Eine praktische Hilfe sind auch die drei neuen Bearbeitungsschaltflächen, die jetzt an der rechten Seite eines ausgewählten Diagramms erscheinen. Sie beschleunigen das Einfügen fehlender Elemente, die Auswahl von Formatvorlagen und Farbzusammenstellungen und das Filtern von Diagramminhalten.

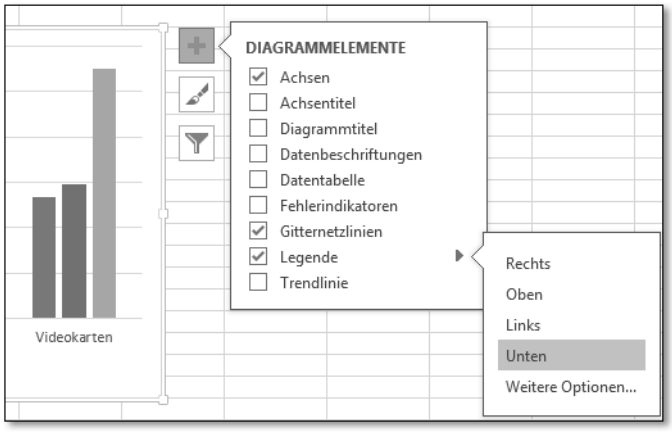


Abbildung 1.95 Die neuen Bearbeitungssymbole am rechten Rand eines markierten Diagramms

1.22.4 Von der Tabelle zum Diagramm

Als äußeren Bezugsrahmen verwenden die meisten Diagramme das kartesische Koordinatensystem. Die Geraden werden als Achsen bezeichnet, die horizontale als x-Achse und die vertikale als y-Achse. Diese Achsen werden als Skalen für die Darstellung von Werten benutzt, wobei üblicherweise die x-Achse für die unabhängige Variable und die y-Achse für die abhängige Variable dient. In dreidimensionalen Diagrammen kommt noch eine dritte hinzu, die als z-Achse oder Tiefenachse bezeichnet wird. Die Skalierung der Achsen wird meist durch eine Unterteilung der Achsen mittels Strichen und Teilstrichen und durch eine Beschriftung deutlich gemacht.

Neben den rechtwinkligen Koordinatensystemen finden auch noch zwei andere Koordinatensysteme Verwendung. Das eine wird bei Kreisdiagrammen (oft auch Tortendiagramme genannt) und Ringdiagrammen (gestaffelte Kreisdiagramme) verwendet. Hier wird statt der Rubrikenachse ein Kreis eingesetzt, der im Gegensatz zu einer Achse keinen Anfangspunkt hat. An die Stelle der Größenachse tritt dann der Winkel im Kreis, so dass den verschiedenen Datengrößen unterschiedliche Winkelgrößen entsprechen.

Ein anderes Koordinatensystem verwenden Netzdiagramme. Hier werden die Rubriken wieder kreisförmig angeordnet. Für jede Rubrik wird eine eigene Größenachse erstellt, auf der der jeweilige Datenpunkt oder die Datenpunkte markiert werden.

Das Koordinatensystem liefert den Bezugsrahmen, in dem die Daten einer Tabelle grafisch repräsentiert werden. Die Daten selbst werden im einfachsten Fall durch Datenpunkte dargestellt, an deren Stelle je nach Diagrammtyp Säulen, Balken, Flächen, Linien etc. treten. In Excel wird deshalb eine Säule oder ein Balken ebenfalls – es sollte Sie nicht verwirren – als Datenpunkt bezeichnet.

Die Daten, die zusammengehören, bilden jeweils eine Datenreihe. Eine Datenreihe besteht entweder aus Werten, die in einer Spalte, oder aus Werten, die in einer Zeile der Tabelle zusammengestellt sind. Excel erlaubt es, maximal 255 Datenreihen in einem Diagramm auszugeben. Die Begrenzung auf 32.000 Datenpunkte pro Datenreihe, die noch für Excel 2007 galt, ist aber inzwischen aufgehoben worden.

Ein Koordinatensystem mit Datenreihen und Datenpunkten ist jedoch nur die Mindestausstattung für ein Diagramm. Um ein Diagramm wirklich in eine aussagekräftige bildliche Darstellung zu verwandeln, sind insbesondere Beschriftungen notwendig, damit klar wird, was beispielsweise eine Säule oder ein Balken überhaupt darstellen soll. Folgende Elemente stehen zur Verfügung:

- Achsenbeschriftungen
- Achsenunterteilungen
- Gitternetzlinien
- Legenden
- Diagrammtitel
- Datenbeschriftungen

Wenn Sie eine Tabelle in ein Diagramm umsetzen wollen, stellt sich immer die Frage, welcher der geeignete Typ dafür ist.

1.22.5 Diagrammtypen

Für die meisten Diagramme sind die Typen brauchbar, die ein rechtwinkliges Koordinatensystem verwenden: Säulen- oder Balkendiagramme, Linien- oder Flächendiagramme. Sie lassen sich danach unterteilen, ob die x-Achse numerisch unterteilt ist oder nicht. Eine numerische Unterteilung kann zeitlich sein (Jahreszahlen, Monatszahlen etc.), es kommen aber – besonders im naturwissenschaftlichen Bereich – auch viele andere Teilungen vor. Des Weiteren kann die numerische Unterteilung diskret oder kontinuierlich sein. Wenn Sie etwa Jahresumsätze grafisch darstellen, dann verwenden Sie diskrete Werte, die Jahreszahlen. Die Darstellung der Durchbiegung eines Trägers in Abhängigkeit von seiner Länge dagegen bietet stetige Werte auf der x-Achse: die Länge. Bei solchen Werten lassen sich Zwischenwerte anhand der Grafik interpolieren, bei diskreten Werten macht die Interpolation dagegen keinen Sinn.

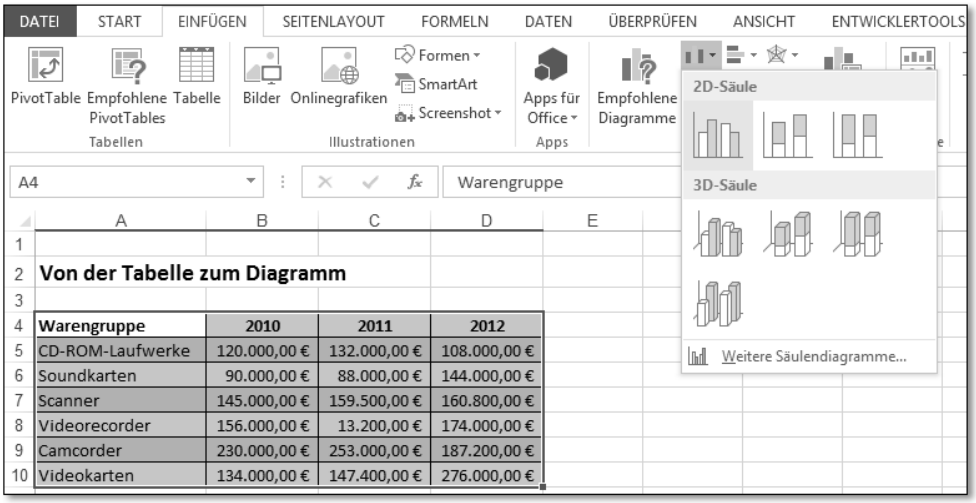
Nicht numerische Einteilungen der x-Achse kommen ebenfalls häufig vor. Beispiele sind etwa der Umsatz eines Unternehmens in verschiedenen Regionen; die Produktivität in unterschiedlichen Branchen; die Lebenserwartung nach Regionen.

Die gebräuchlichsten Diagramme ohne rechtwinkliges Koordinatensystem sind die Kreisdiagramme, die häufig auch als Anteilsdiagramme bezeichnet werden. Diesen Namen verdanken sie dem Umstand, dass immer ein vollständiger Kreis zur Verfügung steht, der anteilig nach den vorliegenden Werten aufgeteilt wird. Hierdurch sind Kreisdiagramme besonders geeignet für alle Daten, bei denen sich die einzelnen Daten zu einer Gesamtheit summieren. Das bekannteste Beispiel hierfür ist die grafische Darstellung der Prozentanteile von Parteien bei Wahlen.

1.22.6 Ein Diagramm erstellen

Als Beispiel für die Entwicklung eines Diagramms soll hier eine einfache Umsatzauswertung für mehrere Jahre dienen. Die Schrittfolge sieht so aus:

1 Um die Umsatztabelle grafisch umzusetzen, markieren Sie zunächst den Datenbereich, der in der Grafik ausgewertet werden soll. In dem abgebildeten Beispiel ist es der Bereich A4 bis D10. Dieser Bereich umfasst nicht nur die Umsatzwerte für die einzelnen Jahre, sondern auch die Spalte mit den Warengruppenbezeichnungen, die für die Beschriftung der x-Achse benötigt werden, und die Zeile mit den Jahreszahlen, die als Legende dienen sollen.



Warengruppe	2010	2011	2012
CD-ROM-Laufwerke	120.000,00 €	132.000,00 €	108.000,00 €
Soundkarten	90.000,00 €	88.000,00 €	144.000,00 €
Scanner	145.000,00 €	159.500,00 €	160.800,00 €
Videorecorder	156.000,00 €	13.200,00 €	174.000,00 €
Camcorder	230.000,00 €	253.000,00 €	187.200,00 €
Videokarten	134.000,00 €	147.400,00 €	276.000,00 €

2 Die Basiswerkzeuge, um aus einer Tabelle ein Diagramm zu erzeugen, finden Sie auf der Registerkarte **Einfügen** in der Gruppe **Diagramme**. Hier werden Symbole für die gängigsten Diagrammtypen angeboten. Um ein Säulendiagramm zu zeichnen, klicken oder tippen Sie auf die Schaltfläche **Säulendiagramm einfügen**, die eine Palette von entsprechenden Diagrammtypen öffnet.

3 In diesem Fall soll zunächst ein einfaches 2D-Säulendiagramm gewählt werden, und zwar gleich das erste in der Palette. Ist das gewünschte Diagrammmuster per Klick oder Tipp ausgewählt, wird mitten im Tabellenblatt sofort ein der Tabelle entsprechendes Diagramm gezeichnet. Wenn die Größe nicht gefällt, kann das Diagramm an den Rahmenanfassern angepasst werden. Abbildung 1.96 zeigt das von Excel 2013 erzeugte Standarddiagramm. Die Breite wurde etwas verändert, damit die Beschriftungen der x-Achse nicht schräg ausgegeben werden müssen.

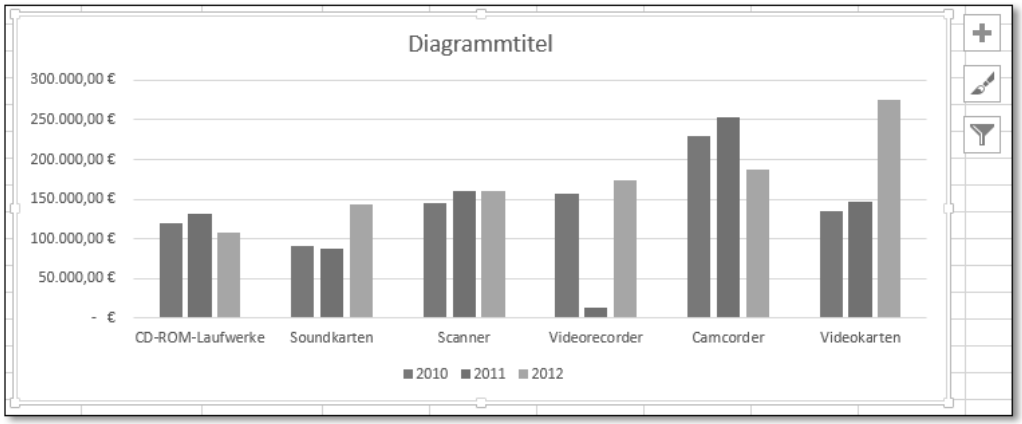


Abbildung 1.96 Das Standarddiagramm zu den markierten Daten

Excel geht bei diesem Verfahren von folgender Annahme aus: Wenn in der markierten Tabelle mehr Zeilen als Spalten vorhanden sind, nimmt Excel an, dass die erste Spalte die Rubriken für die x-Achse enthält und die Datenreihen dazu in den Spalten daneben angeordnet sind. Im anderen Fall schlägt Excel die Verwendung der ersten Zeile für die Rubriken vor.

Im Fall der Umsatzauswertung kann diese Vorgabe übernommen werden. Im Beispiel sind die Daten ja so angeordnet, dass die Umsätze für jedes Jahr untereinanderstehen, also in einer Spalte, die Umsätze der einzelnen Warengruppen dagegen nebeneinander in einer Zeile. Das automatisch erzeugte Diagramm ist schon ziemlich brauchbar, es fehlt nur noch ein Diagrammtitel. Excel gibt einen Dummy-Titel vor, den Sie einfach überschreiben können.

1.22.7 Achsenskalierung

Verbesserungswürdig ist allerdings in diesem Fall das Aussehen der y- oder Werteachsen. Excel übernimmt bei der automatischen Skalierung den Wertebereich mit dem aktuellen Höchstwert und dem Zahlenformat, das in der Tabelle gegeben ist. Wenn Sie nachträglich die Nachkommastellen aus der Tabelle entfernen, werden diese auch im Diagramm weggenommen.

Weitere Möglichkeiten haben Sie, wenn Sie die Achse auswählen und den entsprechenden Aufgabenbereich dazu öffnen. Dazu reicht ein Doppelklick oder -tipp auf die Werteachse. Die andere Methode ist die Auswahl über das Menüband. Wenn das Diagramm per Klick oder Tipp ausgewählt ist, finden Sie auf dem dann angebotenen Register **For-**

mat in der ersten Gruppe ein Listenfeld, in dem alle Elemente, aus denen das Diagramm besteht, ausgewählt werden können.

Wenn Sie anschließend in derselben Gruppe den Befehl **Auswahl formatieren** verwenden, öffnet Excel 2013 den entsprechenden Aufgabenbereich. Diese Aufgabenbereiche ersetzen die in früheren Versionen für Formatangelegenheiten verwendeten Dialogfelder.

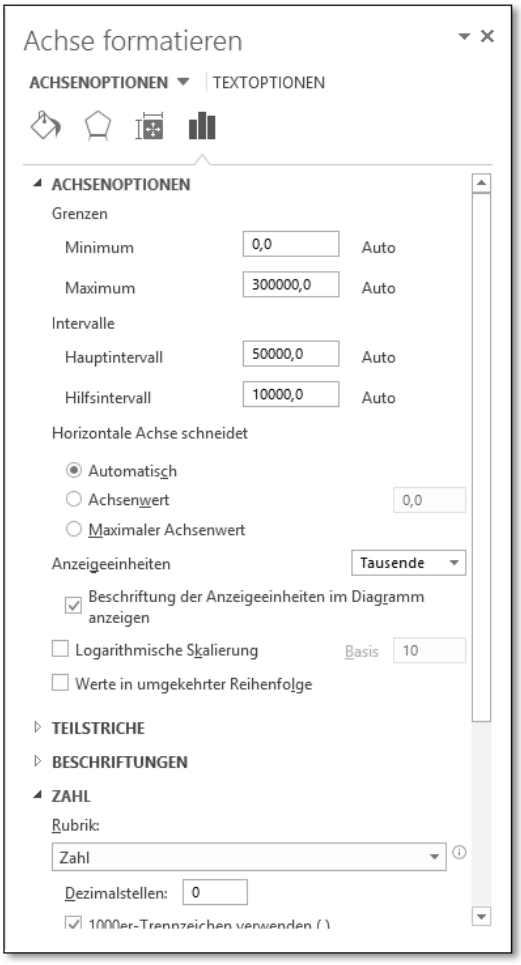


Abbildung 1.97 Der Aufgabenbereich für die Formatierung der Werteachse

Bei numerischen Achsen enthält die hier angebotene Seite **Achsenoptionen** vor allem die Einstellungen zur Skalierung der Achse. Wenn Sie andere Werte für **Minimum** oder **Maximum** eingeben, wird die Achse entsprechend verkürzt oder verlängert. Bei hohen

Werten, wie in diesem Beispiel, bietet es sich an, unter **Anzeigeeinheiten** beispielsweise **Tausende** auszuwählen. Allerdings sollten Sie dann unter **Zahl** ein Format einstellen, das nicht mit dem Währungszeichen arbeitet.

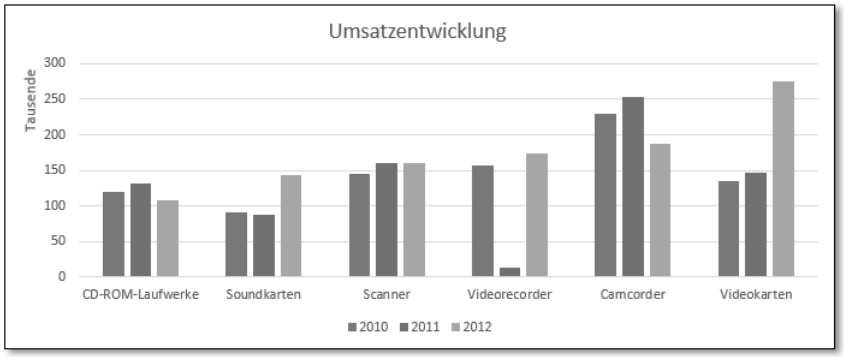


Abbildung 1.98 Das Diagramm mit einer überarbeiteten Werteachse

1.22.8 Ändern der Diagrammdaten und des Diagrammtyps

Der zuerst gewählte Diagrammtyp muss nicht in jedem Fall der beste für die Darstellung der aktuellen Daten sein. Ist das Diagramm ausgewählt, kann aber jederzeit auch nachträglich ein anderer Typ zugewiesen werden. Dazu verwenden Sie auf dem Register **Entwurf** die Schaltfläche **Diagrammtyp ändern**.

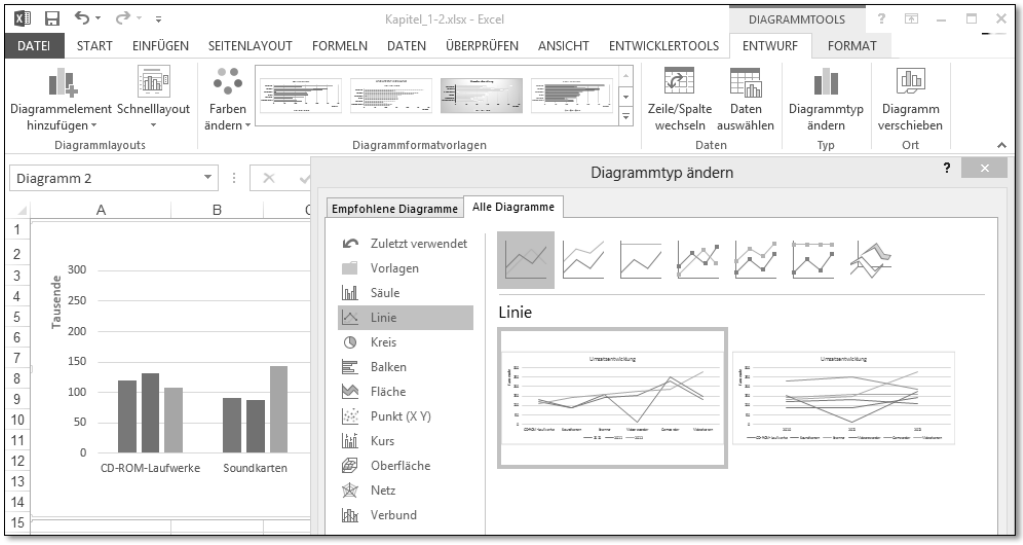


Abbildung 1.99 Ändern des Diagrammtyps

Wenn Sie dagegen die Datenbasis des Diagramms ändern wollen, können Sie dies entweder dauerhaft oder vorübergehend mit Hilfe eines Filters tun. Dazu verwenden Sie auf dem Register **Entwurf** die Schaltfläche **Daten auswählen**. Im Dialog können Sie in diesem Fall beispielsweise die Daten für die weiter zurückliegenden Jahre entfernen. Soll dies nur vorübergehend geschehen, löschen Sie nur die Häkchen vor den Legendeneinträgen.

Abbildung 1.100 Ändern der Datenbasis des Diagramms

Wenn Sie nur die Daten für ein Jahr anzeigen wollen, ist es naheliegend, als Diagrammtyp ein Kreisdiagramm zu wählen, weil so die Verteilung des Umsatzkuchens auf die verschiedenen Warengruppen besonders schön zu sehen ist.

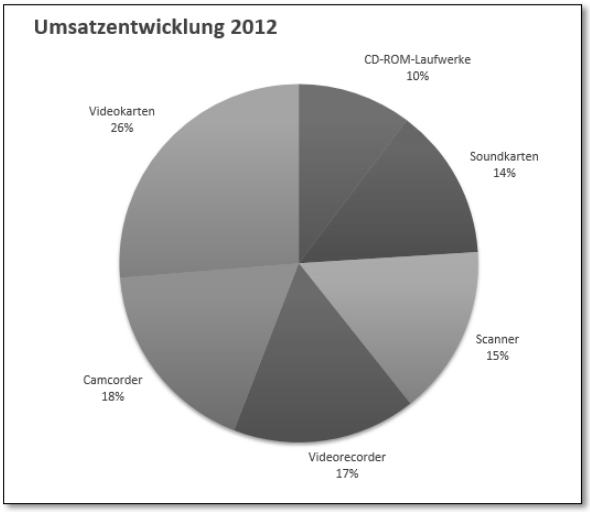


Abbildung 1.101 Die Umsatzwerte eines Jahres als Kreisdiagramm

In diesem Fall kann die Legende durch entsprechende Datenbeschriftungen ersetzt werden. Die Einstellungen dafür nehmen Sie im Aufgabenbereich **Datenbeschriftung formatieren** vor. Die Möglichkeiten zur Gestaltung dieser Beschriftungen sind in Excel 2013 wesentlich erweitert worden.

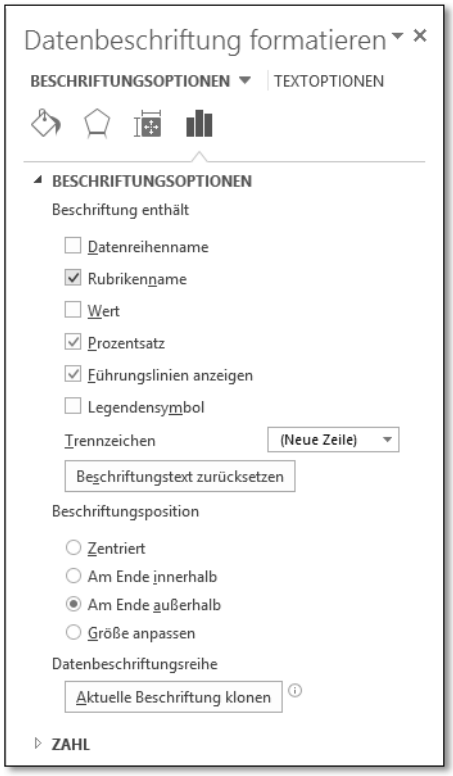
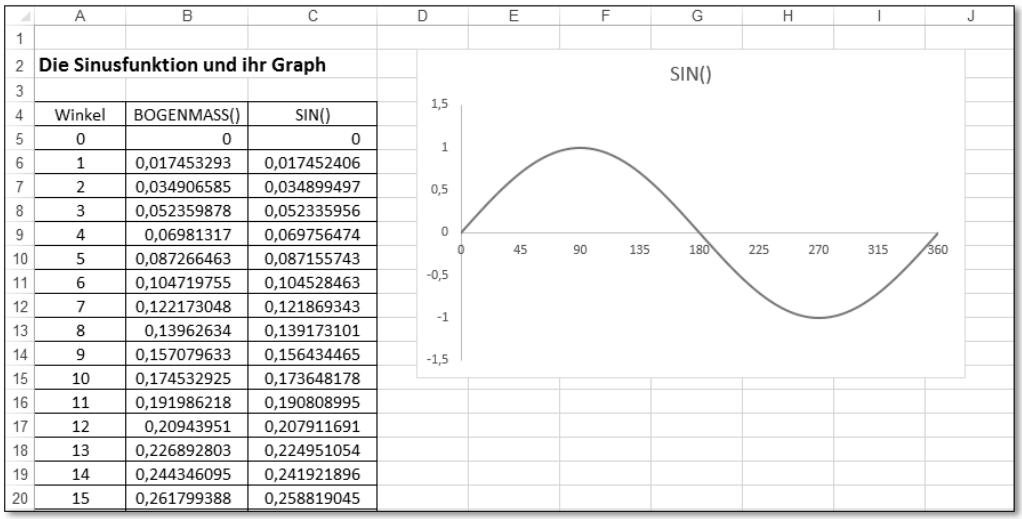


Abbildung 1.102 Der Aufgabenbereich für die Formatierung der Datenbeschriftungen

1.22.9 Bessere Lesbarkeit mit Gitternetzlinien

Bei der Beschreibung der mathematischen oder statistischen Funktionen in diesem Buch werden Sie zahlreiche Diagramme finden, die mit dem Typ **Punkt (XY)** arbeiten, etwa um den Graphen für eine Funktion zu zeigen. Dank dieses Diagrammtyps arbeiten Sie immer mit mindestens zwei numerischen Achsen. Damit Sie die Daten in einem solchen Diagramm besser lesen können, ist es in der Regel sinnvoll, in das Diagramm Gitternetzlinien einzufügen, bei denen die Teilstriche einer oder beider Achsen auf dem Hintergrund des Diagramms durchgezogen sind.

Als Beispiel soll die Darstellung der Sinusfunktion dienen. Um den Graphen zu erstellen, markieren Sie in diesem Fall zunächst die beiden Zellbereiche A4:A365 und C4:C365. Benutzen Sie über **Einfügen** in der Gruppe **Diagramme** die Option **Punkte mit interpolierten Linien** im Menü der Schaltfläche **Punkt (XY)- oder Blasendiagramm einfügen**. Abbildung 1.103 zeigt den Graphen zunächst ohne Gitternetzlinien.



Winkel	BOGENMASS()	SIN()
0	0	0
1	0,017453293	0,017452406
2	0,034906585	0,034899497
3	0,052359878	0,052335956
4	0,06981317	0,069756474
5	0,087266463	0,087155743
6	0,104719755	0,104528463
7	0,122173048	0,121869343
8	0,13962634	0,139173101
9	0,157079633	0,156434465
10	0,174532925	0,173648178
11	0,191986218	0,190808995
12	0,20943951	0,207911691
13	0,226892803	0,224951054
14	0,244346095	0,241921896
15	0,261799388	0,258819045

Abbildung 1.103 XY-Diagramm der Sinusfunktion ohne Gitternetzlinien

Wenn das Diagramm ohne Gitternetzlinien ausgegeben wird, sind die Werte der einzelnen Punkte auf der Kurve nur schwer zu schätzen. Abhilfe können Sie sehr schnell über das Plussymbol neben dem ausgewählten Diagramm finden. Unter **Gitternetzlinien** finden Sie eine Palette, in der für die horizontale und für die vertikale Achse Haupt- und Hilfsgitternetze ausgewählt werden können.

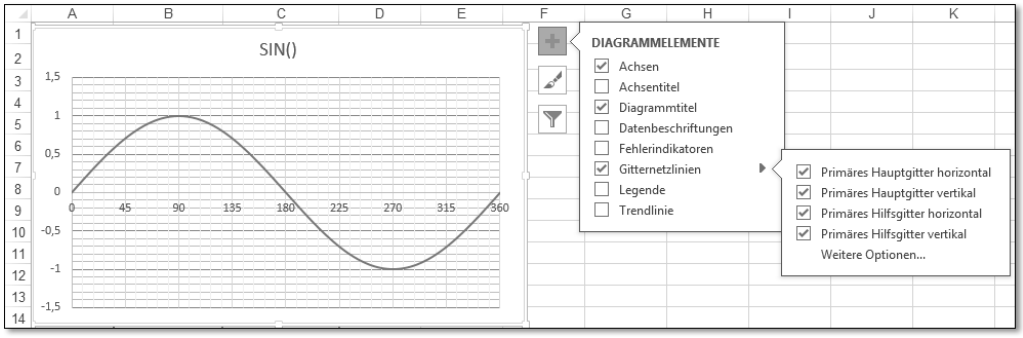


Abbildung 1.104 Einfügen der Gitternetzlinien über das Plussymbol

Wenn Ihnen die Farben der einzelnen Gitternetzlinien zu blass oder zu stark erscheinen, benutzen Sie die Optionen im Aufgabenbereich **Hauptgitternetz formatieren** oder **Hilfsgitternetz formatieren**, die Sie ebenfalls mit einem Doppelklick oder -tipp auf die Linien erreichen. Unter **Linie** finden Sie dort die gewünschten Einstellungen zur Farbe und Stärke der Linien.

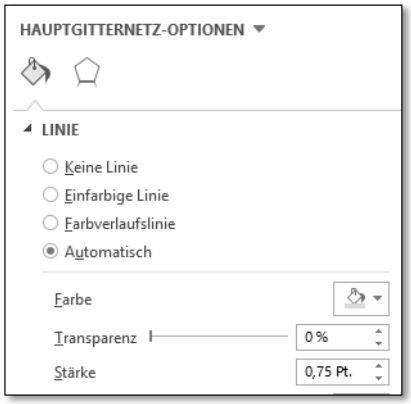


Abbildung 1.105 Hier wählen Sie die Farbe und die Stärke der Gitternetzlinien aus.

2 Berechnungstools

Formeln sind nicht die einzige Form, um Excel zu Berechnungen zu veranlassen. Im Folgenden soll eine kurze Übersicht über die wichtigsten Berechnungstools gegeben werden.

2.1 Zielwertsuche

Bei dem, was Excel als Zielwertsuche anbietet, handelt es sich mathematisch gesehen um die Lösung einer Gleichung mit einer Unbekannten. Wenn es beispielsweise darum geht, den Zins, den ein bestimmter Betrag nach einer bestimmten Zeit ergibt, auszurechnen, wobei der Betrag, der Zinssatz und die Laufzeit gegeben sind, kann diese Aufgabe mit der Formel `Zins = Betrag * Zinssatz * Jahre` gelöst werden. Zins ist hier die Unbekannte.

Mit der gleichen Formel kann gearbeitet werden, wenn sich die Fragestellung verändert. Ist z. B. bekannt, wie hoch der Zins sein soll und welcher Betrag für wie viele Jahre zur Verfügung steht, und ist gefragt, wie hoch dann der Zinssatz sein soll, dann muss die Gleichung nur nach dem Zinssatz umgestellt werden: `Zinssatz = Zins / (Betrag * Jahre)`.

Würden Sie in Excel mit diesen Formeln arbeiten, müssten Sie allerdings auf die eingebauten Tabellenfunktionen verzichten, da diese ja nicht einfach umgestellt werden können. Zum anderen müssten Sie die Gleichung von Hand umstellen (dazu muss aber das mathematische Verfahren bekannt sein), da Excel keine Methode bietet, um Gleichungen umzustellen. Diese ganze Mühe lässt sich mit der Zielwertsuche vermeiden. Deren Einsatz lohnt sich insbesondere, wenn es nicht bloß um derart einfache Gleichungen geht.

Bei der Zielwertsuche rechnet Excel mit den eingebauten Funktionen, stellt aber nicht mehr die Aufgabe, aus den vorgegebenen Werten das Ergebnis zu berechnen, sondern gibt das Ergebnis vor und probiert aus, wie groß einer der anderen Werte für das gewünschte Ergebnis werden muss.

Dazu ein typisches Beispiel aus der Finanzmathematik: Angenommen, Sie wollen einen größeren Kredit aufnehmen und ihn im Lauf von 20 Jahren abbezahlen. Für die monat-

liche Abzahlung stehen 1.000 € zur Verfügung, die Bankzinsen für einen Kredit liegen bei 6,7%.

Die Aufgabe kann mit der Tabellenfunktion für die Barwertberechnung gelöst werden:

```
=BW(6,7%/12; 20*12; -1000)
```

Die Abzahlung wird in diesem Fall als negativer Betrag angegeben – dieses Geld wird ja ausgegeben und nicht eingenommen; da die monatliche Abzahlung angegeben wird, muss der Zinssatz durch 12 geteilt und die Jahresperioden mit 12 multipliziert werden.

Die Funktion liefert den Wert 132.031,63 €, Sie könnten sich also einen Kredit in dieser Höhe leisten. Nun stellen Sie aber fest, dass Sie mit diesem Kredit nicht auskommen. Sie benötigen beispielsweise 150.000 €. Da die Bank über den Zinssatz nicht mit sich verhandeln lässt, gibt es nur zwei Möglichkeiten: Entweder muss die Laufzeit vergrößert werden oder der monatlich bezahlte Betrag.

Um Ihnen die Möglichkeiten der Zielwertsuche zu zeigen, finden Sie in Abbildung 2.1 eine Tabelle, in der die Berechnung zunächst einfach zweimal kopiert worden ist, um die beiden anderen Fragen zu beantworten.

	A	B	C	D
1				
2	**Zielwertsuche**			
3				
4	**Berechnung des Barwerts**			
5	Zinssatz	Jahre	mon. Abzahlung	Barwert
6	6,70%	20	- 1.000,00 €	132.031,63 €
7				
8	**Berechnung der monatlichen Zahlung**			
9	Zinssatz	Jahre	mon. Abzahlung	Barwert
10	6,70%	20	- 1.136,09 €	150.000,00 €
11				
12	**Berechnung der Laufzeit**			
13	Zinssatz	Jahre	mon. Abzahlung	Barwert
14	6,70%	27	- 1.000,00 €	150.000,00 €

Abbildung 2.1 Zielwertsuche für ein Kreditproblem

Die Überschriften für die Berechnung sind entsprechend angepasst. Um die monatlich erforderliche Zahlung für den 150.000-€-Kredit zu ermitteln, wird der Zellzeiger auf die Ergebniszelle (im Beispiel D10) gesetzt. Dann kann in Excel 2013 mit **Daten • Datentools • Was-wäre-wenn-Analyse • Zielwertsuche** der Rechenprozess gestartet werden. Die älteren Versionen bieten hierfür den Befehl **Extras • Zielwertsuche** an.

Die Zielzelle ist schon richtig angegeben, als Zielwert wird 150000 eingetragen. Die veränderbare Zelle ist die mit der monatlichen Zahlung (im Beispiel C10). **OK** führt in ein Dialogfeld, in dem Excel meldet, dass es das Problem gelöst hat. Ein weiteres **OK** trägt den Zielwert und den gesuchten Monatsbetrag in die Tabelle ein.

Auf die gleiche Weise lässt sich ermitteln, wie der Kredit durch eine Verlängerung der Laufzeit zu finanzieren wäre. Als Zielwert wird wieder 150000 eingetragen, die veränderbare Zelle ist diesmal die Zelle mit der Zahl der Jahre (im Beispiel B14).

Der Versuch, auf die gleiche Weise zu ermitteln, ob durch längere Abzahlung auch ein Kredit von 250.000 € zu finanzieren wäre, führt allerdings zu dem Hinweis, dass eine nicht zulässige Lösung gefunden wurde. Das liegt einfach daran, dass der Kredit so nicht finanzierbar wäre, da ja bei 250.000 € die monatlichen Zinsen schon über 1.000 € liegen, also für eine Tilgung gar nichts übrig bleiben würde.

2.2 Lösungen mit dem Solver suchen

Ganz ähnlich wie die Zielwertsuche funktioniert der Solver, nur mit wesentlich umfangreicheren Optionen. Wieder lässt sich ein Zielwert festlegen, nur dass diesmal der Zielwert nicht nur ein bestimmter fester Wert sein kann, sondern auch ein Maximal- oder Minimalwert. Der Solver kann also insbesondere da eingesetzt werden, wo es um Optimierungen geht. Statt einer einzelnen veränderbaren Zelle können jetzt mehrere Zellen angegeben werden. Der Solver ist ein Add-In zu Excel, steht also nur dann zur Verfügung, wenn er auch in die Liste der Add-Ins aufgenommen wurde. Dafür gibt es zwei Wege.

Abbildung 2.2 Der Dialog **Add-Ins**

Sie gehen über **Datei • Optionen** auf die Seite **Add-Ins**. In dem Listenfeld zu **Verwalten** wählen Sie **Excel-Add-Ins** aus und benutzen die Schaltfläche **Gehe zu**. Im Dialog **Add-Ins**

aktivieren Sie unter **Verfügbare Add-Ins** den Eintrag **Solver**. Anschließend erscheint das Symbol für den Start des Solvers in der Gruppe **Analyse** auf dem Register **Daten**.

Für die Version Excel 2010 ist der Solver überarbeitet worden, um bessere und genauere Ergebnisse zu ermöglichen. Die Dialoge wurden etwas verändert. Außerdem wurde ein neues *Evolutionary-Modul* hinzugefügt, das genetische Algorithmen verwendet. Das neue Add-In steht auch für die 64-Bit-Version von Excel 2013 zur Verfügung, was bei sehr komplexen Lösungsversuchen in puncto Rechengeschwindigkeit natürlich Vorteile verspricht.

2.2.1 Zur Arbeitsweise des Solvers

Mathematisch gesehen werden mit dem Solver Probleme bearbeitet, die sich als Gleichungen bzw. als Gleichungssysteme mit mehreren Unbekannten formulieren lassen, wobei auch sogenannte Ungleichungen erlaubt sind. Um derartige Aufgaben von Hand zu lösen, würde zunächst eine Gleichung nach einer Unbekannten aufgelöst, das Ergebnis in die nächste Gleichung eingesetzt werden usw., bis am Ende eine Gleichung mit nur noch einer Unbekannten erreicht ist, die dann nach dieser aufgelöst werden kann.

Das eigentliche Problem bei der Arbeit mit dem Solver besteht darin, einen Zusammenhang so zu formulieren und aufzubereiten, dass er dem Solver zur Lösung angeboten werden kann. Der einfachste Punkt ist die Bestimmung des Zielwerts. Hier handelt es sich darum, festzulegen, welches Ergebnis eine festgelegte Berechnung haben soll. Da der Zielwert in der Tabelle ohnehin immer das Ergebnis einer Berechnung ist, definiert durch die Formel, die in der Zelle mit dem Zielwert abgelegt ist, kann deshalb allenfalls die Formulierung der Berechnung Schwierigkeiten bereiten.

Etwas komplizierter verhält es sich mit den Nebenbedingungen. Um zu vermeiden, dass der Solver triviale Ergebnisse ausgibt, müssen die Nebenbedingungen möglichst vollständig definiert werden. Sonst kann es leicht passieren, dass der Solver nach längeren Berechnungen ein Ergebnis liefert, das entweder von vornherein klar war oder aber völlig unrealistisch ist oder auch zu einer Fehlermeldung führt.

2.2.2 Beispiel Materialkostenoptimierung

Als praktisches Beispiel für die Arbeit mit dem Solver soll hier eine Aufgabe verwendet werden, die Sie von Hand mit den Mitteln der Differenzialrechnung bearbeiten müssten. Es handelt sich um ein Verpackungsproblem: Es sollen Konservendosen hergestellt werden, die ein vorgegebenes Volumen (z.B. 1.000 ml) haben, wobei der Materialauf-

wand (und damit auch die Materialkosten) so gering wie möglich sein sollen. Das Volumen einer Konservendose wird berechnet, indem die Grundfläche (das ist ein Kreis) mit der Höhe der Dose multipliziert wird. Die Grundfläche selbst wird als Kreisfläche berechnet. Also gilt:

```
Vol = r^2 * PI * h
```

Der Materialverbrauch für eine derartige Dose setzt sich aus der Bodenfläche, dem Mantel und dem Deckel zusammen. Der Boden und der Deckel werden jeweils als Kreis (`r^2 * PI`) gerechnet, der Mantel hat dann als Fläche den Umfang der Dose (`2 * r * PI`) multipliziert mit der Höhe. Also gilt insgesamt:

```
Fläche = 2 * r^2 * PI + 2 * r * PI * h
```

1 Um das Problem zu bearbeiten, wird eine Tabelle aufgebaut, in der die Berechnung des Materials und des Dosenvolumens zunächst mit ganz willkürlichen Zahlen vorgenommen wird.

	A	B	C	D	E
1					
2	**Materialverbrauch pro Dose**				
3					
4	**Radius**	**Höhe**	**Material**		
5	4,00	6,00	251,327		=2*A5^2*PI()+2*A5*PI()*B5
6	**Dosenvolumen:**		301,593		=A5^2*PI()*B5
7					

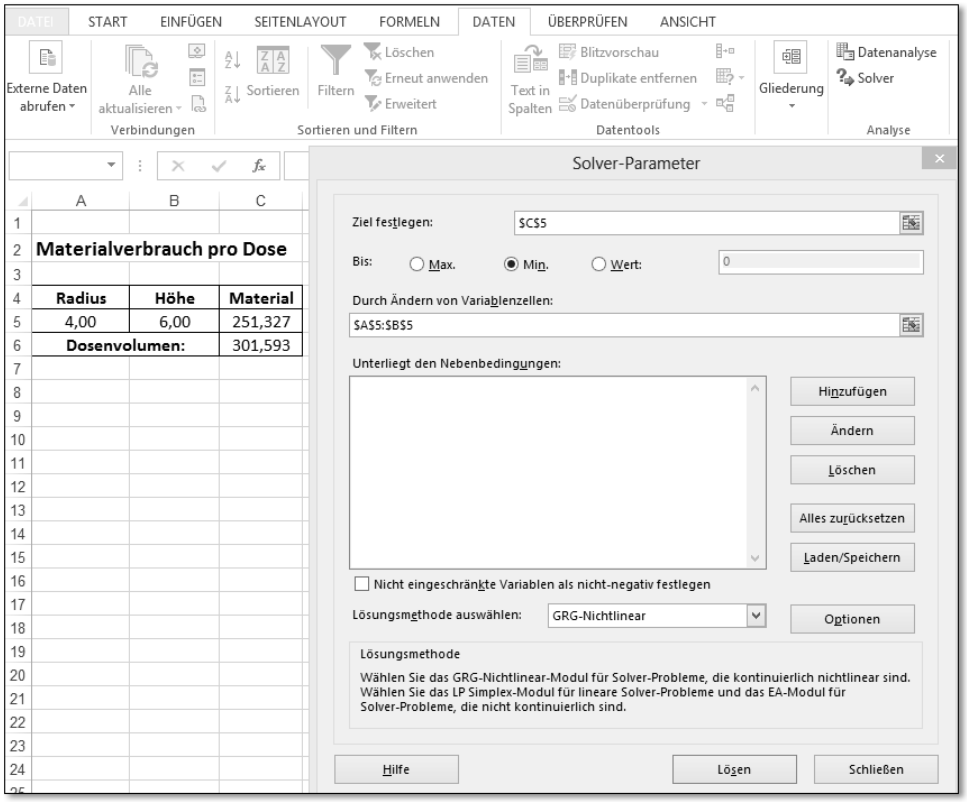
2 In die Zellen für den Materialverbrauch und für das Volumen werden die entsprechenden Formeln eingetragen, die in der Abbildung noch einmal als Text angezeigt werden. Mit den willkürlich eingetragenen Werten für den Radius und die Höhe der Dose liefern die Formeln zunächst ebenso willkürliche Werte für den Materialverbrauch und das Volumen. Werden die Dosenmaße in Zentimeter angegeben, ergibt sich der Materialverbrauch in Quadratzentimeter und das Volumen in Milliliter.

3 Nach diesen Vorbereitungen können Sie den Solver mit **Daten • Analyse • Solver** aufrufen. Wurde der Solver schon einmal in dem aktuellen Tabellenblatt benutzt, hat er noch die alten Einträge gespeichert. Dies ist sehr nützlich, wenn Sie an einem angefangenen Problem weiterarbeiten wollen, etwa um den Zielwert zu ändern oder weitere Nebenbedingungen zu formulieren. Soll dagegen ein neues Problem behandelt werden, empfiehlt es sich, nach Aufruf des Solvers zunächst die Schaltfläche **Alles zurücksetzen** zu benutzen, um die alten Einträge insgesamt zu entfernen.

4 Wenn beim Aufruf des Solvers die Zielzelle schon markiert ist, kann sie unter **Ziel festlegen:** einfach übernommen werden.

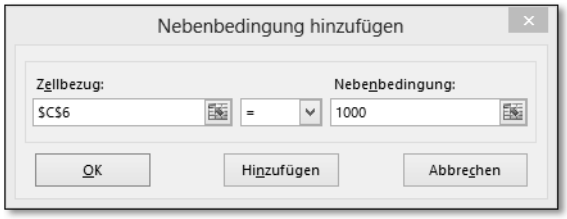
5 Unter **Bis:** muss in diesem Beispiel die Option **Min.** (für Minimum) gewählt werden, da es ja darum geht, die Lösung mit dem geringsten Materialverbrauch zu finden.

6 Die veränderbaren Zellen sind in diesem Fall die beiden Zellen, in denen der Radius und die Höhe stehen. Diese Zellen werden in dem Feld zu **Durch Ändern von Variablenzellen:** angegeben oder ausgewählt. In diesem Fall sind die veränderbaren Zellen in einem zusammenhängenden Bereich angeordnet. Liegen die veränderbaren Zellen nicht in einem zusammenhängenden Bereich, dann müssen die einzelnen Zellen oder Bereiche durch Semikola getrennt werden. Am einfachsten lassen sie sich durch die Auswahl in der Tabelle übernehmen, wobei zusätzliche Zellen oder Bereiche mit gedrückter Strg-Taste markiert werden müssen.



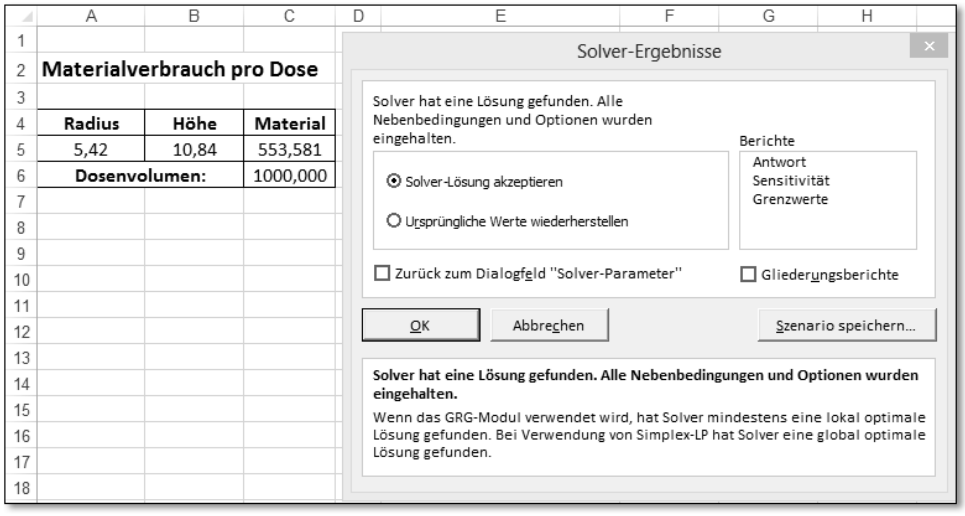
7 In diesem Beispiel gibt es nur eine einzige, aber entscheidende Nebenbedingung: Das Volumen der Dose soll 1.000 ml betragen. Ohne diese Bedingung würde der Solver sich in negative Werte für den Materialbedarf verirren. Mit **Hinzufügen** wird

das Dialogfeld für die Nebenbedingungen geöffnet. Als **Zellbezug** wird die Zelle mit dem Dosenvolumen eingetragen (am besten wieder durch Markieren in der Tabelle). Als Wert muss dann unter **Nebenbedingung** 1000 angegeben werden. In dem kleinen Listenfeld für den Operator wird = ausgewählt, da das Volumen ja genau diesen Wert erreichen soll.



8 Sobald der Eintrag der Nebenbedingungen abgeschlossen ist, wählen Sie im Dialogfeld des Solvers zunächst noch die Lösungsmethode **GRG-Nichtlinear** aus und benutzen dann die Schaltfläche **Lösen**.

9 Nach einiger Rechenarbeit liefert der Solver das gewünschte Ergebnis, wobei unter Umständen in der Statuszeile eine Anzahl von Zwischenergebnissen angezeigt wird.



10 Im Dialog **Solver-Ergebnisse** entscheiden Sie, ob Sie das Ergebnis in das Arbeitsblatt übernehmen wollen oder nicht.

Als Ergebnis werden ein Radius von ca. 5,42 und eine Höhe von ca. 10,84 ausgegeben, was der Berechnung mit den Mitteln der Differenzialrechnung sehr genau entspricht (danach muss die Höhe doppelt so groß sein wie der Radius).

2.2.3 Lösungsmethoden und Optionen

Die zuletzt beschriebene Vorgehensweise ist bei allen Problemen sinnvoll, die keine besonders komplexen Berechnungen erfordern. Es kann aber durchaus sein, dass so noch nicht das gewünschte Ergebnis erreicht wird, sei es, dass die Rechenzeit nicht ausreicht, sei es, dass das Ergebnis nicht hinreichend genau ist. In solchen Fällen müssen unter Umständen die Vorgaben für den Solver geändert werden. Hierzu wird im Dialogfeld des Solvers die Schaltfläche **Optionen** verwendet.

Damit wird ein weiteres Dialogfeld eingeblendet, in dem sich die Arbeit des Solvers in weitem Umfang beeinflussen lässt (siehe Abbildung 2.3). Der Dialog **Optionen** bietet ein Register für alle Lösungsmethoden und zusätzlich Register für die Methoden **GRG-Nichtlinear** und **EA (Evolutionärer Algorithmus)**.

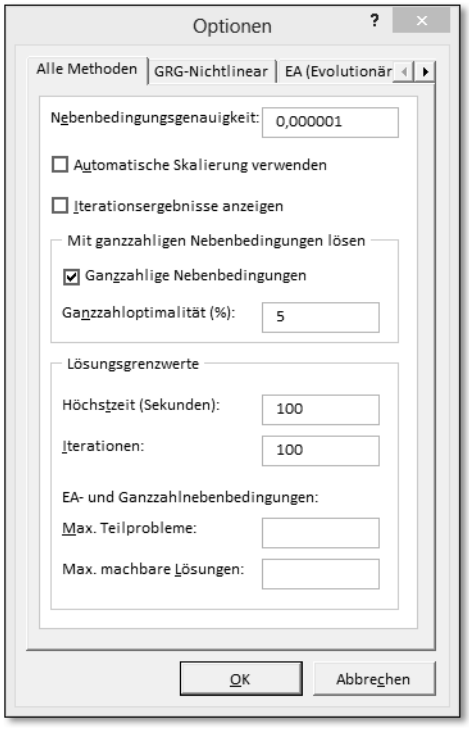


Abbildung 2.3 Solver-Optionen

2.2.4 Hinweise zu den Lösungsmethoden

Die Wahl zwischen verschiedenen Lösungsmethoden wird im Dialog **Solver-Parameter** angeboten. Die Unterschiede dieser Methoden können hier nur angedeutet werden:

- **Simplex-LP** ist die Lösungsmethode für lineare Optimierungen. Wenn die zu berechnenden Zusammenhänge alle linear sind, dann ist dies die richtige Methode. Das bedeutet, dass veränderbare Zellen nur mit Konstanten multipliziert oder durch Konstanten dividiert, aber nicht miteinander multipliziert oder dividiert werden. Es werden auch keine Potenzen, Wurzeln oder Funktionen wie Sinus oder Logarithmus verwendet. Nehmen Sie diese Methode also nur, wenn Sie sich wirklich sicher sind, dass ausschließlich lineare Zusammenhänge vorliegen.
- **GRG-Nichtlinear** ist die vorgegebene Lösungsmethode. Sie erlaubt kontinuierliche nicht lineare Optimierungen. Sie lässt für den Zielwert Formeln zu, die Potenzen, Wurzeln oder Funktionen wie Sinus, Logarithmus etc. verwenden. Die veränderbaren Zellen dürfen miteinander multipliziert, dividiert oder potenziert werden.
- **EA (Evolutionärer Algorithmus)** ist eine Lösungsmethode für nichtkontinuierliche Optimierungen, die sich an der biologischen Evolution und an Modellen für die Beschreibung der Veränderung des Erbguts orientiert. Dabei werden wiederholt Lösungsvorschläge generiert, die sich schrittweise immer besser an die gewünschten Zielwerte anpassen.

2.2.5 Allgemeine Optionen

Die erste der allgemeinen Einstellungen im Dialog **Optionen** ist die **Nebenbedingungsgenauigkeit**. Sie legt fest, bei welcher Genauigkeit der Solver eine Bedingung als erfüllt betrachtet. Je größer die vorgegebene Genauigkeit ist (je kleiner also der angegebene Wert), umso länger wird der Solver für seine Berechnungen benötigen.

Die Option **Automatische Skalierung verwenden** ist zu empfehlen, wenn in einer Berechnung gleichzeitig sehr große und sehr kleine Zahlen vorkommen. Der Solver bringt dann die Zahlen intern in eine vergleichbare Größenordnung und rechnet am Ende die Ergebnisse zurück. So lässt sich unter Umständen die Genauigkeit erhöhen.

Mit der Option **Iterationsergebnisse anzeigen** erreichen Sie, dass der Solver nach jedem »Ausprobieren« das Ergebnis anzeigt und per Dialog abfragt, ob die Suche nach einer Lösung fortgesetzt werden soll. Das kann bei komplexen und langwierigen Berechnungen sinnvoll sein, da sich dann verfolgen lässt, wie der Solver arbeitet.

Wird die Option **Ganzzahlige Nebenbedingungen** aktiviert, kann noch ein Prozentsatz für die **Ganzzahloptimalität** angegeben werden. Werden Nebenbedingungen formuliert, die ganzzahlige Werte bei den veränderbaren Zellen fordern, kann die Rechenzeit erheblich zunehmen. Durch Erhöhung der Toleranz – Vorgabe ist 1 % – lässt sich die benötigte Zeit wieder vermindern, wobei allerdings auch höhere Abweichungen in Kauf zu nehmen sind.

Die untere Optionengruppe auf dem Register betrifft die Lösungsgrenzwerte, die die Arbeit des Solvers einfassen. Unter **Höchstzeit** lässt sich die Zeit limitieren, die dem Solver insgesamt für die Lösung einer Aufgabe zur Verfügung stehen soll. Sie wird in Sekunden angegeben. Die Zahl für die erlaubte Wiederholung von Lösungsversuchen kann unter **Iterationen** limitiert werden. Schließlich kann noch die maximale Zahl der Teilprobleme und der machbaren Lösungen begrenzt werden.

2.2.6 Spezielle Optionen

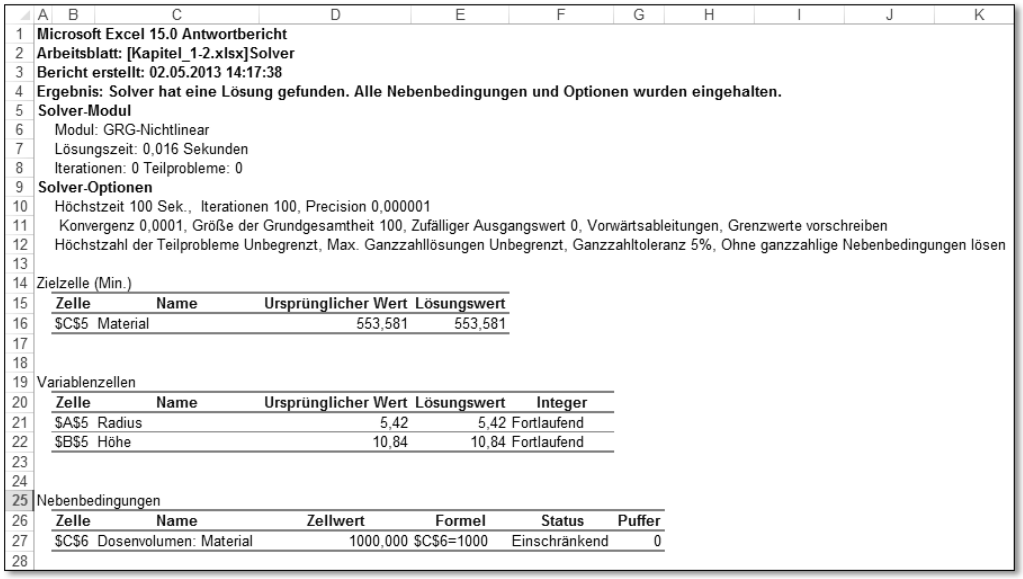
Auf dem Register **GRG-Nichtlinear** kann für nicht lineare Aufgaben unter **Konvergenz** eine Bruchzahl zwischen 0 und 1 angegeben werden. Unterschreitet die relative Änderung in der Zielzelle bei den letzten fünf Iterationen diesen Wert, stoppt der Solver.

Unter **Ableitungen** wird bestimmt, ob bei der Einschätzung von partiellen Ableitungen mit Vorwärtsdifferenzen oder mit zentralen Differenzen gearbeitet wird. Zentrale Differenzen können ausprobiert werden, wenn Werte in der Nähe eines Grenzwerts stark schwanken und der Solver die Meldung ausgibt, er könne das Ergebnis nicht weiter verbessern.

Das Register **EA (Evolutionärer Algorithmus)** bietet neben einem Eingabefeld für **Konvergenz** noch Felder für **Mutationsrate**, **Größe der Grundgesamtheit**, **Zufälliger Ausgangswert** und **Höchstzeit ohne Verbesserung**.

2.2.7 Auswertung der Ergebnisse und Berichte

Nach Abschluss der Arbeit mit dem Solver erfolgt – wie schon erwähnt – eine Meldung, dass der Solver eine Lösung gefunden hat, oder eine Meldung, dass dies nicht gelungen ist. Zusätzlich können Sie die Arbeit des Solvers noch in verschiedenen Berichten dokumentieren. Der Antwortbericht fasst die Ergebnisse in einer übersichtlichen Tabelle zusammen. Hierbei werden sowohl die Ausgangswerte als auch die vom Solver gefundenen Werte angegeben.



Microsoft Excel 15.0 Antwortbericht
Arbeitsblatt: [Kapitel_1-2.xlsx]Solver
Bericht erstellt: 02.05.2013 14:17:38
Ergebnis: Solver hat eine Lösung gefunden. Alle Nebenbedingungen und Optionen wurden eingehalten.
Solver-Modul
Modul: GRG-Nichtlinear
Lösungszeit: 0,016 Sekunden
Iterationen: 0 Teilprobleme: 0
Solver-Optionen
Höchstzeit 100 Sek., Iterationen 100, Precision 0,000001
Konvergenz 0,0001, Größe der Grundgesamtheit 100, Zufälliger Ausgangswert 0, Vorwärtsableitungen, Grenzwerte vorschreiben
Höchstzahl der Teilprobleme Unbegrenzt, Max. Ganzzahllösungen Unbegrenzt, Ganzzahltoleranz 5%, Ohne ganzzahlige Nebenbedingungen lösen

Zielzelle (Min.)

Zelle	Name	Ursprünglicher Wert	Lösungswert
$C$5	Material	553,581	553,581

Variablenzellen

Zelle	Name	Ursprünglicher Wert	Lösungswert	Integer
$A$5	Radius	5,42	5,42	Fortlaufend
$B$5	Höhe	10,84	10,84	Fortlaufend

Nebenbedingungen

Zelle	Name	Zellwert	Formel	Status	Puffer
$C$6	Dosenvolumen: Material	1000,000	$C$6=1000	Einschränkend	0

Abbildung 2.4 Der Antwortbericht zu dem Lösungsbeispiel

Die beiden anderen Berichte, der Sensitivitäts- und der Grenzwertbericht, geben Auskunft über die Arbeitsweise, die vom Solver angewandt wurde. Wird die lineare Simplex-LP-Methode in einem Fall angewendet, der sich nicht linear lösen lässt, hilft ein dann angebotener **Linearitätsbericht**, die Formeln zu finden, die nicht linear sind. Bei der EA-Methode steht noch ein **Grundgesamtheits-Bericht** zur Verfügung, der statistische Werte zu den Variablen liefert.

Interessant sind diese Berichte in erster Linie, wenn die Ergebnisse nicht überzeugend sind. Sie können dann Ausgangspunkt für eine andere Formulierung des Problems sein.

2.3 Was wäre, wenn ... mit Datentabellen

Excel stellt Ihnen unter dem Namen **Datentabelle** noch eine spezielle Variante von Was-wäre-wenn-Analysen zur Verfügung, die mit der Auswertung von Formeln zu tun hat. Zwei Versionen davon unterscheiden sich im Kern durch die Anzahl der Variablen, die zugelassen sind. In der ersten Version kann nur ein Wert variiert werden, in der zweiten zwei.

Die unterschiedliche Anzahl der Variablen hat Folgen für die Anordnung der Daten, die in die Berechnung eingehen. Zudem ist die Anzahl der Formeln, die ausgewertet werden

können, von der Zahl der Variablen abhängig. Nur die Version mit einer Variablen kann gleichzeitig mehrere Formeln auswerten.

2.3.1 Datentabelle mit einer Variablen

Zunächst ein einfaches Beispiel. In Abbildung 2.5 finden Sie eine kleine Anwendung mit einer Barwertfunktion. Die Fragestellung könnte lauten: Wie viel muss am 1.1.2014 auf der Bank eingezahlt werden, um nach Ablauf von drei Jahren den Betrag zu erhalten, der drei jährlichen Einzahlungen von 40.000 € entspricht? Der Zinssatz beträgt 3%.

Die Formel mit der Funktion `BW()` in Zelle C7 enthält keinen konstanten Zinssatz, sondern die Adresse der Zelle B6. Dort sind zunächst `3,00%` eingetragen. Um nun die Wirkung anderer Zinssätze beobachten zu können, wird in eine freie Spalte eine Reihe von Zinsvarianten eingetragen. Dies ist in der Tabelle in Abbildung 2.5 in der Spalte B geschehen. Die Beschriftung **andere Zinssätze** weist darauf hin, wozu die Werte gedacht sind. Die alternativen Werte müssen untereinander und in eine Spalte eingetragen werden. Die Reihenfolge ist nicht vorgeschrieben, aber eine auf- oder absteigende Reihenfolge macht das Ergebnis übersichtlicher.

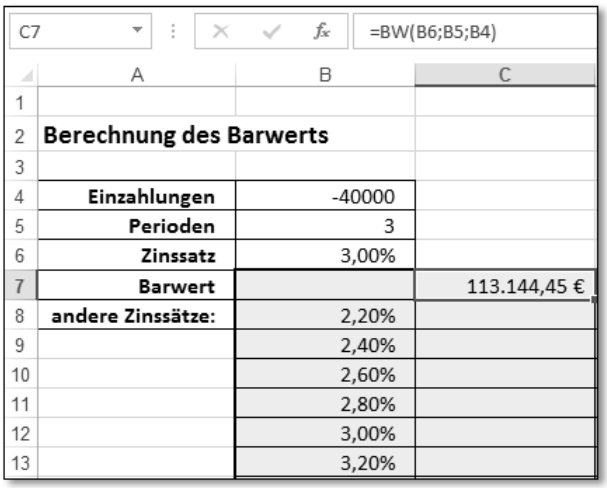


Abbildung 2.5 Vorbereitung der Barwertberechnung als Mehrfachoperation

Der nächste Schritt der Vorbereitung betrifft die Formel, die mit den alternativen Werten gefüttert werden soll. In diesem Beispiel ist es die Barwertfunktion. Diese Formel muss in die Zeile eingetragen werden, die direkt über der ersten Zeile liegt, die alternative Werte enthält. Sie muss aber um mindestens eine Spalte nach rechts versetzt wer-

den, weil das Programm die neu errechneten Werte direkt unter dieser Formel ablegen wird. In Abbildung 2.5 ist deshalb die Zelle C7 der richtige Ort für die Formel.

Im nächsten Schritt muss der Bereich markiert werden, der für die Datentabelle benutzt werden soll (in früheren Versionen wurde hier der Begriff »Mehrfachoperation« verwendet). In der Abbildung ist dieser Bereich zur besseren Verdeutlichung mit einem Hintergrundmuster gekennzeichnet. Der Bereich wird bestimmt durch die Spalte mit den alternativen Werten und durch die Zeile, die die Formel oder die Formeln enthält, die der Befehl mehrfach durchrechnen soll. Im Unterschied zu einer normalen Formel wird das Ergebnis bei dieser Operation eben nicht in der Zelle angezeigt, die die Formeln enthält, sondern in einer fortlaufenden Reihe von Zellen.

Nun kann in Excel 2013 aus der Gruppe **Datentools** auf dem Register **Daten** über das Menü der Schaltfläche **Was-wäre-wenn-Analyse** der Befehl **Datentabelle** aufgerufen werden. Das entspricht dem früheren Befehl **Daten • Tabelle**. In dem kleinen Dialogfeld wird in diesem Beispiel die Eingabe der Zelladresse für den Zinssatz, also B6, erwartet. Das ist die Eingabezelle, in die die verschiedenen Zinssätze nacheinander »eingeschoben« werden, um die Barwertformel jedes Mal neu zu berechnen. Da die alternativen Werte für den Zinssatz in einer Spalte angeordnet sind, muss die Adresse `$B$6` in das Feld **Werte aus Spalte** eingetragen werden. Wären die alternativen Zinssätze nur in einer Zeile angeordnet, etwa D6 bis L6, müssten Sie die Zelladresse in das Feld **Werte aus Zeile** eintragen.

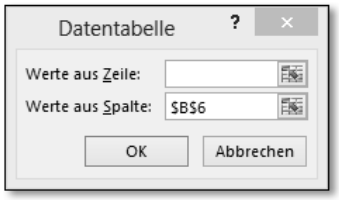
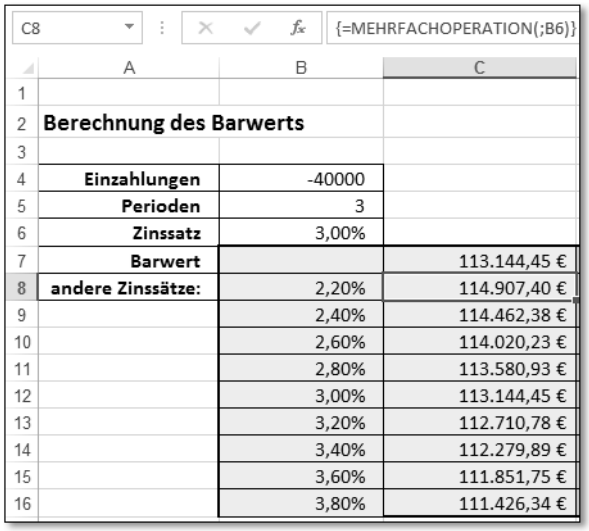


Abbildung 2.6 Auswahl der Eingabezelle

Wenn Sie die Eingaben im Dialogfeld bestätigen, rechnet Excel 2013 für alle alternativen Zinssätze die Barwertformel durch und stellt die Ergebnisse in der Spalte unter der Formel zur Verfügung.

Die Tabelle kann folgendermaßen gelesen werden: Der alternative Zinssatz von 2,4% würde einen Barwert von 114.462 € ergeben, der alternative Zinssatz von 3,6% einen Barwert von 111.851 € usw. Die Originalformel in Zelle C7 und der ursprünglich angenommene Zinssatz von 3% in Zelle B6 bleiben von dieser Operation unberührt.



C8 {=MEHRFACHOPERATION(;B6)}

	A	B	C
1			
2	**Berechnung des Barwerts**		
3			
4	**Einzahlungen**	-40000	
5	**Perioden**	3	
6	**Zinssatz**	3,00%	
7	**Barwert**		113.144,45 €
8	**andere Zinssätze:**	2,20%	114.907,40 €
9		2,40%	114.462,38 €
10		2,60%	114.020,23 €
11		2,80%	113.580,93 €
12		3,00%	113.144,45 €
13		3,20%	112.710,78 €
14		3,40%	112.279,89 €
15		3,60%	111.851,75 €
16		3,80%	111.426,34 €

Abbildung 2.7 Die berechneten Alternativen

Wenn Sie sich die Zelle C8 ansehen, werden Sie in der Bearbeitungsleiste eine Matrixformel mit der Funktion `MEHRFACHOPERATION()` finden. Diese Funktion steht übrigens im Dialog **Funktion einfügen** nicht zur Verfügung. Die Zellen, die die Ergebnisse der Mehrfachoperation anzeigen, bilden eine Ergebnismatrix. Einzelne Zellen dieses Bereichs können also nicht geändert oder gelöscht werden.

Dagegen können die eingegebenen alternativen Zinssätze beliebig verändert werden. Wenn Sie auch die anderen Argumente der Barwertfunktion durch Zellbezüge festgelegt haben, können Sie nun auch andere Werte für die Anzahl der Perioden oder die regelmäßigen Zahlungen ausprobieren. Die gesamte Tabelle der Mehrfachoperation wird jedes Mal neu durchgerechnet.

2.3.2 Auswertung mehrerer Formeln

Im letzten Beispiel ist nur eine einzige Formel mit alternativen Werten durchgespielt worden. Solange Sie mit einer einzigen Variablen arbeiten, können Sie aber, wie schon erwähnt, auch das Verhalten mehrerer Formeln gleichzeitig beobachten. Zum Beispiel könnten Sie zusätzlich den Barwert berechnen lassen für den Fall, dass die Einzahlungen schon am Anfang der Periode fällig sind. Die Formel muss dazu nur um das Argument für die Fälligkeit erweitert werden. In diesem Fall muss als Bereich für die Mehrfachoperation B7 bis D16 markiert werden.

	A	B	C	D
1				
2	**Berechnung des Barwerts**			
3				
4	**Einzahlungen**	-40000		
5	**Perioden**	3		
6	**Zinssatz**	3,00%		
7	**Barwert**		113.144,45 €	116.538,79 €
8	**andere Zinssätze:**	2,20%	114.907,40 €	117.435,37 €
9		2,40%	114.462,38 €	117.209,47 €
10		2,60%	114.020,23 €	116.984,75 €
11		2,80%	113.580,93 €	116.761,19 €
12		3,00%	113.144,45 €	116.538,79 €
13		3,20%	112.710,78 €	116.317,53 €
14		3,40%	112.279,89 €	116.097,41 €
15		3,60%	111.851,75 €	115.878,42 €
16		3,80%	111.426,34 €	115.660,54 €

Abbildung 2.8 Vorschüssige und nachschüssige Barwertberechnung

2.3.3 Mehrfachoperation mit zwei Variablen

Die zweite Version der Mehrfachoperation lässt zwei Variablen zu, also zwei »Eingabezellen«, kann aber nur jeweils eine Formel auswerten. Der Aufbau des Tabellenbereichs ist ein wenig anders als bei der ersten Version. Die Formel wird in die bisher leere linke obere Eckzelle des für die Mehrfachoperation markierten Tabellenbereichs eingetragen.

Die alternativen Werte für die erste Variable werden wie gehabt in die linke Spalte des Tabellenbereichs eingepflegt. Die Werte für die zweite Variable dagegen werden in die erste Zeile eingegeben, beginnend mit der Zelle rechts neben der Formelzelle. Dort, wo in der ersten Version die Formeln standen, müssen bei dieser Version also die Werte für die zweite Variable stehen.

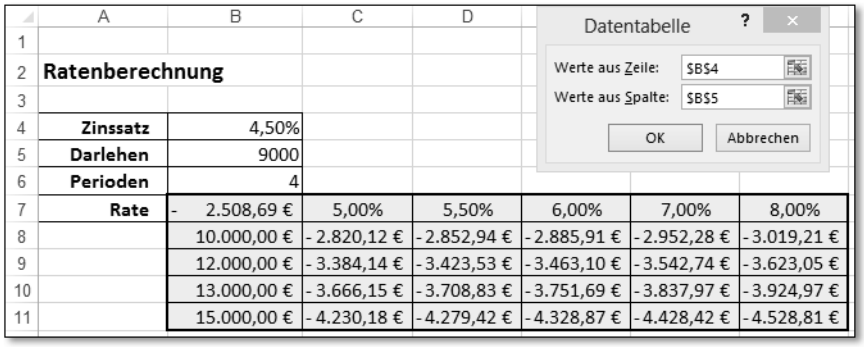
2.3.4 Beispiel Ratenberechnung

Ein Blick auf Abbildung 2.9 zeigt den Aufbau dieses speziellen Tabellenbereichs. Als Beispiel soll wieder eine Finanzfunktion dienen. Eine Ratenberechnung arbeitet mit drei Argumenten, einem Kapitalbetrag, z.B. einem Darlehen, einem Zinssatz und der Anzahl der Perioden. Variiert werden sollen das Darlehen und der Zinssatz.

Die Werte für die erste Variable, das Darlehen, sind in der Spalte B eingetragen, die Werte für die zweite Variable, also Zinssatz, in der Zeile 7. Die Ratenformel steht in der linken oberen Eckzelle des Tabellenbereichs, also in B7.

Wieder können Sie zunächst den Tabellenbereich markieren, von B7 bis G11. Rufen Sie den Befehl **Datentabelle** auf. Diesmal müssen beide Eingabefelder ausgefüllt werden. Bei **Werte aus Zeile** geben Sie die Adresse der Zelle an, die den Zinssatz enthält, also B4, bei **Werte aus Spalte** die Adresse B5, die den Wert für das Darlehen enthält.

Der Befehl liefert eine Tabelle von Werten, die Sie lesen können wie ein XY-Diagramm. Um die Rate für das Darlehen von 13.000 € zu finden, die bei einem Zinssatz von 7% zu zahlen ist, können Sie zunächst die Zeile aufsuchen, die mit 13000 beginnt, und dann so viele Zellen nach rechts gehen, bis Sie in der Spalte angekommen sind, über der 7% steht. Diese Zelle enthält die entsprechende Rate.



	A	B	C	D			
1							
2	**Ratenberechnung**						
3							
4	**Zinssatz**	4,50%					
5	**Darlehen**	9000					
6	**Perioden**	4					
7	**Rate**	- 2.508,69 €	5,00%	5,50%	6,00%	7,00%	8,00%
8		10.000,00 €	- 2.820,12 €	- 2.852,94 €	- 2.885,91 €	- 2.952,28 €	- 3.019,21 €
9		12.000,00 €	- 3.384,14 €	- 3.423,53 €	- 3.463,10 €	- 3.542,74 €	- 3.623,05 €
10		13.000,00 €	- 3.666,15 €	- 3.708,83 €	- 3.751,69 €	- 3.837,97 €	- 3.924,97 €
11		15.000,00 €	- 4.230,18 €	- 4.279,42 €	- 4.328,87 €	- 4.428,42 €	- 4.528,81 €

Abbildung 2.9 Ratenberechnung mit zwei Variablen

2.4 Arbeit mit Szenarios

Wenn es um den Vergleich mehrerer Lösungen für eine bestimmte Aufgabe geht, gibt es verschiedene Möglichkeiten, damit umzugehen. Vielleicht kommen Sie auf die Idee, in einer Arbeitsmappe für jede Lösung ein Blatt anzulegen. Sie würden das erste Lösungsmodell auf einem Blatt zusammenstellen, dann das Blatt einige Male kopieren und die abweichenden Werte entsprechend ändern. Dieses Verfahren ist durchaus in Ordnung, hat aber einen Nachteil: Um die Lösungen zu vergleichen, müssen Sie in der Mappe immer hin und her blättern.

Eine andere Möglichkeit wäre, die Daten für jede Lösung jeweils in einer Spalte zusammenzustellen. Vorteil: Die Daten lassen sich gut vergleichen, weil sie ja nebeneinanderstehen. Aber dieses Verfahren lässt sich bei komplizierteren Kalkulationsmodellen nicht so leicht anwenden, weil immer alle Daten in einer Spalte untergebracht werden müssten. Das ist aber nicht immer möglich.

Glücklicherweise hat Excel hier noch eine andere Lösung anzubieten: die Arbeit mit dem **Szenario-Manager**. Das Kalkulationsmodell kann dabei völlig frei aufgebaut werden. Es genügt ein einziges Modell, das dann jedes Mal mit anderen Zahlen gefüttert wird. Für den übersichtlichen Vergleich sorgen Berichte, die der Szenario-Manager automatisch erstellt.

2.4.1 Erst vergleichen, dann entscheiden

Mit dem Begriff *Szenario* ist ein Gedankenexperiment oder Planspiel gemeint. Als Beispiel wird im Folgenden eine mehrstufige Deckungsbeitragsrechnung vorgestellt. Es soll die Frage geprüft werden, bei welchen variablen Kosten für die Herstellung verschiedener Artikel ein ausreichender Gewinn erwirtschaftet werden kann.

2.4.2 Aufbau des Modells

Die Vorgehensweise, um die Beantwortung einer solchen Frage mit Hilfe des Szenario-Managers anzugehen, beginnt mit dem Aufbau eines entsprechenden Kalkulationsmodells, in dem bereits alle Dateien und Berechnungen für eine der möglichen Lösungen enthalten sind.

	A	B	C	D	E	F
1						
2	**Mehrstufige Deckungsbeitragsrechnung**					
3						
4		**Artikel A-1**	**Artikel A-2**	**Artikel B-1**	**Artikel B-2**	**Gesamt**
5	Umsatzerlöse	400.000	600.000	200.000	300.000	1.500.000
6	Variable Kosten	190.000	290.000	90.000	170.000	740.000
7	Deckungsbeitrag I	210.000	310.000	110.000	130.000	760.000
8	Produktfixe Kosten	70.000	120.000	100.000	40.000	330.000
9	Deckungsbeitrag II	140.000	190.000	10.000	90.000	430.000
10	Bereichsfixe Kosten		125.000		30.000	155.000
11	Deckungsbeitrag III		205.000		70.000	275.000
12	Unternehmensfixe Kosten				20.000	20.000
13	Betriebsergebnis				255.000	255.000

Abbildung 2.10 Modell für eine mehrstufige Deckungsbeitragsrechnung

1. Beginnen Sie mit der Berechnung der ersten Stufe des Deckungsbeitrags für je zwei Artikel aus zwei unterschiedlichen Artikelgruppen. Tragen Sie in Zeile 5 die geschätzten Umsatzerlöse ein und in Zeile 6 die geschätzten beschäftigungsabhängigen variablen Kosten.
2. In Zeile 7 kann dann durch eine einfache Subtraktion der Deckungsbeitrag der ersten Stufe errechnet werden. In der Spalte F bilden Sie davon noch die Quersumme, um jeweils die Gesamtbeträge zu erhalten.
3. In Zeile 8 lassen sich die auf den einzelnen Artikel bezogenen fixen Kosten ablegen.
4. Zeile 9 ermittelt die zweite Stufe des Deckungsbetrags, indem diese artikelbezogenen Kosten von Deckungsbeitrag I abgezogen werden.
5. Zeile 10 nimmt die fixen Kosten auf, die für jede der beiden Artikelgruppen entstehen, und legt sie in den Zellen B10 und D10 ab.
6. Die dritte Stufe des Deckungsbeitrags wird dann berechnet, indem jeweils von der Summe der beiden Deckungsbeiträge II der einzelnen Artikel die bereichsfixen Kosten abgezogen werden.
7. In Zeile 12 werden noch die fixen Kosten auf der Ebene des Unternehmens eingetragen.
8. Das Betriebsergebnis ergibt sich, wenn von der Summe der Deckungsbeiträge III die unternehmensfixen Kosten abgezogen werden.

Damit ist das Kalkulationsmodell abgeschlossen. Es liegt an Ihnen, ob Sie bei der Erstellung erst mal nur mit ungefähren Testdaten arbeiten oder gleich die Daten für das erste Szenario angeben.

2.4.3 Welche Werte sind veränderbar?

Bei der Arbeit mit Szenarios unterscheidet Excel zwischen veränderbaren Zellen und Ergebniszellen. Veränderbar sind die Zellen, für die jeweils unterschiedliche Werte eingegeben werden sollen, um zu sehen, welche Auswirkungen diese auf bestimmte berechnete Werte und auf das Gesamtergebnis des Kalkulationsmodells haben. In diesem Fall sollen nur die Zellen für variable Herstellungskosten der einzelnen Artikel als veränderbare Zellen angegeben werden.

Die Ergebniszellen sind die Zellen, die über Formeln von den veränderbaren Zellen abhängen. Als Ergebniszellen sind hier insbesondere die Zellen in der Spalte F mit den Gesamtbeträgen der Deckungsbeitragsstufen und des Betriebsergebnisses interessant.

2.4.4 Besser mit Namen

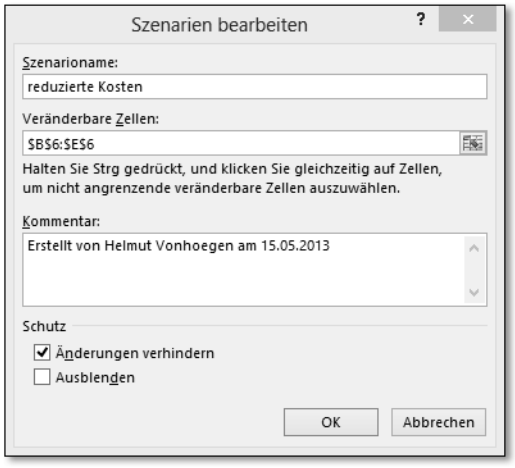
Es ist zwar nicht zwingend erforderlich, aber doch vorteilhaft, beide Zellgruppen vor dem Aufruf des Szenario-Managers mit Namen zu belegen. Sie werden das sofort verstehen, wenn Sie die Werte der alternativen Berechnungsmodelle eingeben wollen. Sind die Zellen benannt, fügt Excel die Namen automatisch in das Dialogfeld für die Dateneingabe ein. Außerdem sind dann auch die Berichte, die der Szenario-Manager zusammenstellt, besser lesbar.

Da die Zellen in diesem Fall nicht direkt unter oder neben ihren entsprechenden Beschriftungen liegen, ist es notwendig, die Zellen einzeln zu benennen. Markieren Sie nacheinander die Zellen B6 bis E6 und die Zellen F7, F9, F11 und F13. Benutzen Sie jedes Mal auf dem Register **Formeln** in der Gruppe **Definierte Namen** die Option **Namen definieren**.

2.4.5 Einrichten verschiedener Szenarios

Damit sind alle Vorbereitungen getroffen, um verschiedene Szenarios ins Spiel zu bringen.

1. Benutzen Sie auf dem Register **Daten** in der Gruppe **Datentools** aus dem Menü der Schaltfläche **Was-wäre-wenn-Analyse** den Befehl **Szenario-Manager**.
2. Wählen Sie im Dialog die Schaltfläche **Hinzufügen**.



3 Zunächst vergeben Sie einen passenden Namen für das Szenario. In diesem Fall ist es sinnvoll, dafür Namen zu verwenden, die sich auf den Status der variablen Kosten beziehen, etwa »reduzierte Kosten«, »mittleres Kostenniveau« und »erhöhte Kosten«.

4 Geben Sie nun unter **Veränderbare Zellen** die Adressen der Zellen mit den variablen Kosten pro Artikel an. Da sie alle nebeneinanderliegen, können Sie hier einfach den Bereich B6:E6 markieren.

5 Unter **Kommentar** erscheinen automatisch Ihr Name und das Datum der Erstellung. Ergänzen Sie den Kommentar, wenn nötig.

6 Wird unter **Schutz** die Option **Änderungen verhindern** abgehakt, verhindern Sie, dass das Szenario geändert werden kann, vorausgesetzt, das Blatt wird anschließend insgesamt gegen Änderungen gesperrt. Dies ist möglich über das Register **Überprüfen** mit der Schaltfläche **Blatt schützen** in der Gruppe **Änderungen**. Wenn Sie **Ausblenden** abhaken, wird das aktuelle Szenario in der Liste im ersten Dialogfeld erst gar nicht mehr aufgeführt, wenn die Datei insgesamt mit **Überprüfen • Arbeitsmappe schützen** geschützt worden ist. Diese Variante bleibt also für Unbefugte unsichtbar.

7 Wenn Sie das Dialogfeld mit **OK** bestätigen, erscheint die Eingabemaske mit den veränderbaren Zellen, die Sie angegeben haben. Wenn im Tabellenblatt die Werte dieses ersten Szenarios schon eingetragen sind, werden diese Werte hier angezeigt. Sie können einfach mit **OK** übernommen werden.

Abbildung 2.11 Eingabe der Werte für die veränderbaren Zellen

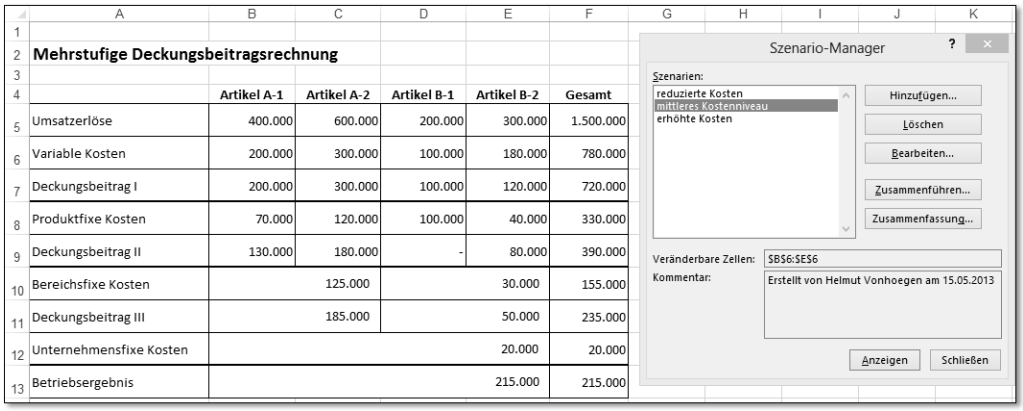
2.4.6 Werte für die anderen Angebote eingeben

Im Dialogfeld des Szenario-Managers wird nun der Name des ersten Angebots angezeigt.

1. Um weitere Angebote einzugeben, verwenden Sie wieder **Hinzufügen** und wiederholen die letzten Schritte. Mit **Schließen** beenden Sie die Eingabe der Szenarios.

2. Wenn Sie ein bestimmtes Szenario in der Tabelle sehen wollen, wählen Sie den Namen des Szenarios aus der Liste und benutzen dann **Anzeigen**.

Um ein Szenario zu ändern, nehmen Sie im Dialog die Schaltfläche **Bearbeiten**, um es zu entfernen, die Schaltfläche **Löschen**. Mit **Zusammenführen** lassen sich auch Daten aus anderen Arbeitsmappen einlesen, vorausgesetzt, dort sind gleich aufgebaute Tabellen angelegt.



Mehrstufige Deckungsbeitragsrechnung

	Artikel A-1	Artikel A-2	Artikel B-1	Artikel B-2	Gesamt
Umsatzerlöse	400.000	600.000	200.000	300.000	1.500.000
Variable Kosten	200.000	300.000	100.000	180.000	780.000
Deckungsbeitrag I	200.000	300.000	100.000	120.000	720.000
Produktfixe Kosten	70.000	120.000	100.000	40.000	330.000
Deckungsbeitrag II	130.000	180.000	-	80.000	390.000
Bereichsfixe Kosten		125.000		30.000	155.000
Deckungsbeitrag III		185.000		50.000	235.000
Unternehmensfixe Kosten				20.000	20.000
Betriebsergebnis				215.000	215.000

Abbildung 2.12 Anzeige eines anderen Szenarios

Wenn das Dialogfeld beim Ansehen der Ergebnisse im Weg ist, ziehen Sie es einfach an der Titelleiste ganz nach unten. Das Szenario, das Sie zuletzt angeschaut haben, bleibt jeweils in der Tabelle stehen, wenn Sie den Dialog schließen.

2.4.7 Zusammenfassende Berichte

Um zu einer Entscheidung zwischen den verschiedenen Szenarios zu kommen, ist es nützlich, die Ergebnisse auf einen Blick nebeneinander zu sehen. Dazu gibt es zwei Möglichkeiten: Die erste ist ein Bericht, der die verschiedenen Werte der veränderbaren Zellen und der Ergebniszellen zusammenstellt. Die andere Möglichkeit ist, dieselben Daten in einer Pivot-Tabelle auszugeben.

1. Benutzen Sie im Dialogfeld des Szenario-Managers dazu die Schaltfläche **Zusammenfassung**.

2 Wählen Sie unter **Berichtstyp** die Option **Szenariobericht**, und markieren Sie unter **Ergebniszellen** die Zellen F7, F9, F11 und F13.

3 Wenn Sie die Eingaben bestätigen, werden die Daten auf einem neuen Blatt in derselben Arbeitsmappe ausgegeben. Für jedes Szenario wird eine Spalte verwendet, so dass Sie die Werte leicht vergleichen können.

4 Der Bericht wird automatisch gegliedert. Mit den kleinen Minus-Schaltflächen können Sie Details ausblenden.

Szenariobericht	Aktuelle Werte:	reduzierte Kosten	mittleres Kostenniveau	erhöhte Kosten
Veränderbare Zellen:				
Variable_Kosten_A1	190.000	190.000	200.000	210.000
Variable_Kosten_A2	290.000	290.000	300.000	310.000
Variable_Kosten_B1	90.000	90.000	100.000	110.000
Variable_Kosten_B2	170.000	170.000	180.000	190.000
Ergebniszellen:				
DB_I	760.000	760.000	720.000	680.000
DB_II	430.000	430.000	390.000	350.000
DB_III	275.000	275.000	235.000	195.000
Betriebsergebnis	255.000	255.000	215.000	175.000

Abbildung 2.13 Szenariobericht

Wenn Sie im Dialog **Szenariobericht** stattdessen einen **PivotTable-Szenariobericht** anfordern, werden als Felder die Autoren, die Szenarionamen und die Ergebniszellen angeboten, so dass Sie in diesem Fall eine sehr kompakte Zusammenfassung erhalten. Allerdings sollte die Beschriftung der Felder der Autoren und der Szenarionamen geändert werden, weil Excel als Vorgabe nur Zellbezüge anzeigt.

von	(Alle)			
Szenario	DB_I	DB_II	DB_III	Betriebsergebnis
erhöhte Kosten	680000	350000	195000	175000
mittleres Kostenniveau	720000	390000	235000	215000
reduzierte Kosten	760000	430000	275000	255000

Abbildung 2.14 Leicht überarbeiteter Pivot-Bericht

3 Finanzmathematische Funktionen

Funktion	Seite	Funktion	Seite
AMORDEGRK()	184	RENDITE()	220
AMORLINEARK()	186	RENDITEDIS()	221
AUFGELZINS()	187	RENDITEFÄLL()	222
AUFGELZINSF()	189	RMZ()	223
AUSZAHLUNG()	190	TBILLÄQUIV()	225
BW()	191	TBILLKURS()	226
DIA()	194	TBILLRENDITE()	227
DISAGIO()	195	UNREGER.KURS()	228
DURATION()	196	UNREGER.REND()	229
EFFEKTIV()	198	UNREGLE.KURS()	230
GDA()	199	UNREGLE.REND()	232
GDA2()	200	VDB()	233
IKV()	202	XINTZINSFUSS()	234
ISPMT()	203	XKAPITALWERT()	235
KAPZ()	204	ZINS()	236
KUMKAPITAL()	205	ZINSSATZ()	239
KUMZINSZ()	206	ZINSTERMNZ()	240
KURS()	207	ZINSTERMTAGE()	241
KURSDISAGIO()	208	ZINSTERMTAGNZ()	242
KURSFÄLLIG()	210	ZINSTERMTAGVA()	243
LIA()	211	ZINSTERMVZ()	244
MDURATION()	212	ZINSTERMZAHL()	245
NBW()	213	ZINSZ()	246
NOMINAL()	215	ZSATZINVEST()	247
NOTIERUNGBRU()	215	ZW()	248
NOTIERUNGDEZ()	216	ZW2()	250
PDURATION()	217	ZZR()	251
QIKV()	218		

3.1 Einsatzbereiche der finanzmathematischen Funktionen

Wie die Finanzmathematik insgesamt decken auch die von Excel angebotenen finanzmathematischen Funktionen eine Reihe von Themen ab, die hier zunächst kurz im Überblick vorgestellt wird.

3.1.1 Einfache Zinsrechnung

Der erste Themenbereich betrifft die einfache Zinsrechnung. Das Adjektiv »einfach« spielt darauf an, dass die Zinsen nach erfolgter Zahlung nicht zum verzinsten Kapital geschlagen werden. Der Zins wird als Gebühr für geliehenes Kapital verstanden.

Der Zins bezieht sich dabei immer auf eine bestimmte Laufzeit, die in Form von Jahren – p. a. – oder kleineren Perioden – Quartale, Monate – angegeben wird. Dabei wird noch unterschieden, ob die Zinsen jeweils am Ende einer Zinsperiode oder am Anfang gezahlt werden. Das erste wird als nachschüssige, das zweite als vorschüssige Verzinsung bezeichnet. Dabei geht es häufig um Zinsberechnungen, die sich auf kurze, also unterjährige Zeiträume beziehen.

3.1.2 Zinseszinsrechnung

Ein weiterer Bereich betrifft den Fall, dass die Zinsen nach Ablauf der Zinsperioden jeweils zum Ausgangskapital geschlagen werden, also die Zinseszinsrechnung. Die hier vorkommenden Berechnungen sind wesentlich komplexer, zumal es viele Variationsmöglichkeiten in Bezug auf die Zinszahlungstermine gibt.

3.1.3 Rentenrechnung

Ein umfangreicher Bereich der Finanzmathematik ist die Rentenrechnung. Hier geht es um regelmäßig wiederkehrende Auszahlungen in gleicher Höhe. Bei diesen Renten geht es also nicht um sozialrechtliche Versorgungsrenten, sondern um privatwirtschaftlich vereinbarte regelmäßige Zahlungen. Die einfachste und für das Verständnis dieser Funktionen durchschaubarste Form hierfür ist folgendes Modell:

Sie zahlen auf der Bank eine bestimmte Geldsumme ein, die dort nach einem vereinbarten Zinssatz verzinst wird. Die Bank zahlt Ihnen aus diesem Guthaben so lange regelmä-

ßig einen bestimmten Betrag (»Rente«), bis das Guthaben verbraucht ist. Auch hier wird unterschieden zwischen vorschüssigen und nachschüssigen Rentenzahlungen, je nachdem, ob die Rente am Anfang oder am Ende der jeweiligen Periode gezahlt wird. Die von Excel hier zur Verfügung gestellten Funktionen gehen dabei immer davon aus, dass die Rentenzahlungstermine mit den Zinsterminen zusammenfallen.

3.1.4 Tilgungsrechnung

Über die Zinsberechnung hinaus geht die Tilgungsrechnung. Dabei geht es um die Rückzahlung von Schulden, sei dies ein Kredit, ein Darlehen, eine Hypothek oder eine Anleihe. Bei diesen Berechnungen wird die Ermittlung der anfallenden Zinsen kombiniert mit der Feststellung des Tilgungsbetrags, wobei es verschiedene Verfahren gibt, diesen Ablauf zu gestalten. Während bei der Annuitätentilgung die Summe aus Zinszahlung und Tilgung immer gleich bleibt, wird bei einer Tilgung mit gleichen Tilgungsraten die Annuität mit jeder Periode geringer.

3.1.5 Investitionsrechnung

Das Thema Investitionsrechnung wird meist untergliedert in statische und dynamische Investitionsanalysen. Dabei geht es in der Regel um die Frage, ob es sich lohnt, für eine Investition das dafür notwendige Kapital über einen entsprechenden Zeitraum für ein bestimmtes Vorhaben zu binden. Dabei werden in der dynamischen Analyse auch die Zinseszinseffekte berücksichtigt.

3.1.6 Abschreibungsrechnung

Bei der Abschreibungsrechnung geht es darum, Wertminderungen von Anlagegütern buchmäßig zu erfassen. Dafür stehen verschiedene Verfahren zur Verfügung, von der einfachen linearen Abschreibung bis zu verschiedenen Formen der degressiven Abschreibung. Dabei muss beachtet werden, dass nicht alle möglichen Berechnungsverfahren auch steuerrechtlich in dem jeweiligen Land zugelassen sind. Das Steuerrecht in Deutschland erlaubt nur die lineare und mit bestimmten Einschränkungen die geometrisch-degressive Methode. Es ist außerdem ein Wechsel von der zweiten zur ersten Methode erlaubt, aber nicht ein Wechsel von linear zu degressiv.

3.1.7 Kursrechnung

Die Kursrechnung beschäftigt sich mit Berechnungen zu Wertpapieren. Meist geht es um festverzinsliche Papiere, in einigen Fällen aber auch um unverzinsliche. Excel stellt hier zahlreiche Funktionen zur Verfügung, wobei einige speziell für amerikanische Schatzanweisungen ausgelegt sind.

3.2 Zur Berechnung von Zins und Zinseszins

Die Höhe des Zinses wird in Form des Zinsfußes oder des Zinssatzes angegeben. Der Zinsfuß ist der Bruchteil des Anfangskapitals, der pro 100 Einheiten zu zahlen ist; der Zinssatz ist der Prozentsatz, den dieser Bruchteil ausmacht.

3.2.1 Die Formeln für die einfache Verzinsung

Die Berechnung einfacher Zinsen bei einer jährlichen Zinsperiode erfolgt üblicherweise nach der Formel:

```
Zins = Anfangskapital * Zinssatz * Laufzeit in Jahren
```

oder in der Schreibweise mit Buchstaben:

$z = K_0 * i * n$

Die Zinsen sind dabei in jeder Periode gleich. Die Gleichung lässt sich leicht umstellen, wenn die Zinsen bekannt sind und ein anderer Wert ermittelt werden soll.

$K_0 = z / (i * n)$
$i = z / (K_0 * n)$
$n = z / (K_0 * i)$

Das Kapital nach Ablauf der Laufzeit berechnen Sie nach der Formel:

$K_n = K_0 * (1 + n * 1)$

Sind Endkapital, Zins und Laufzeit bekannt, ergibt sich das Anfangskapital nach der Formel:

$K_0 = K_n / (1 + n * 1)$

Die anderen Umstellungen der Formel lauten entsprechend:

$n = 1/i * (K_n / K_0 - 1)$
$i = 1/n * (K_n / K_0 - 1)$

Abbildung 3.1 bestätigt die angegebenen Formeln. Zur Verdeutlichung sind die Zellen – soweit es geht – mit den entsprechenden Namen benannt.

	A	B	C	D	E
1					
2	**Einfache Verzinsung**				
3					
4	**Zins**	**Anfangskapital**	**Zinssatz**	**Laufzeit**	**Endkapital**
5	z	K0	i	n	Kn
6	150	1000	3%	5	1150
7	**=K0*i*n**				**=K0*(1+n*i)**
8		1000			1000
9		**=z_/(i*n)**			**=Kn/(1+n*i)**
10			3%		3%
11			**=z_/(K0*n)**		**=1/n*(Kn/K0-1)**
12				5	5
13				**=z_/(K0*i)**	**=1/i*(Kn/K0-1)**

Abbildung 3.1 Belege für die Formeln zur einfachen Verzinsung

Soll die Verzinsung für kürzere Zeiträume berechnet werden, gilt für die Monatsberechnung die Formel:

$z = (K_0 * i * m) / 12$

wobei `m` die Anzahl der Monate angibt.

Für Berechnungen mit Tagesperioden wird folgende Formel verwendet:

$z = (K_0 * i * t) / 360$

wobei `t` die Anzahl der Tage angibt. Statt mit `360` kann auch mit `365` gerechnet werden, je nachdem, welche Methode für die Tageszählung verwendet wird.

Diese Formeln können in Excel leicht nachgebildet werden. Einige der finanzmathematischen Funktionen, die im Folgenden behandelt werden, verwenden die entsprechenden Formeln intern, wenn es um Vorgänge geht, bei denen einfache Zinsen ohne Zinseszinseffekt berechnet werden. Dazu gehören Funktionen wie `AUFGELZINS()`, `AUFGELZINSF()`, `AUSZAHLUNG()`, `ISPMT()`, `ZINSSATZ()`.

3.2.2 Die Formel für den Zinseszinseffekt

Werden die Zinsen nach Ablauf der Zinsperiode dem Kapital zugeschlagen, kommt es zu dem schon genannten Zinseszinseffekt. Auch Renten und Tilgungsraten werden mit Zinseszinsen berechnet. Die allgemeine Formel für die Berechnung von Zinseszinsen ist:

$K_n = K_0 * q^n$

Das Kapital nach n Jahren ist gleich dem Startkapital multipliziert mit der n-ten Potenz von q, wobei q der Aufzinsungsfaktor ist. Dieser wird mit

$q = (1 + i)$

aus dem Zinssatz i berechnet.

Das Anfangskapital ist dann:

$K_0 = K_n / q^n$

Der Zinssatz ist:

$i = ((K_n / K_0)^{1/n}) - 1$

Und für die Laufzeit gilt:

$n = (\ln(K_n) - \ln(K_0))/\ln(q)$

Abbildung 3.2 bestätigt wieder die angegebenen Formeln.

	A	B	C	D	E	F
1						
2	**Formel für die Berechnung von Zinseszinen**					
3						
4	**Zins**	**Anfangskapital**	**Zinssatz**	**Zinsfuss**	**Laufzeit**	**Endkapital**
5	z	K0	i	q	n	Kn
6	2472,25	10000	3%	103%	8	12472,25
7					8	**=K0*q^n**
8			3%		**=(LN(Kn)-LN(K0))/LN(q)**	
9		10000	**=((Kn/K0)^(1/n)) -1**			
10	2472,25	**=(Kn/q^n)**				
11	**=(K0*q^n)-K0**					

Abbildung 3.2 Formeln für die Berechnung von Zinseszinsen

Wie unterschiedlich sich ein Kapital bei Zinseszinsen im Vergleich zur einfachen Verzinsung entwickelt, deutet Abbildung 3.3 für einen Zeitraum von 20 Jahren an:

Vergleich: einfache Verzinsung und Zinseszinsen

Zinssatz	5%	
Jahre	Kapital bei einfacher Verzinsung	Kapital bei Zinseszinsen
0	10000	10000
1	10500	10500
2	11000	11025
3	11500	11576
4	12000	12155
5	12500	12763
6	13000	13401
7	13500	14071
8	14000	14775
9	14500	15513
10	15000	16289
11	15500	17103
12	16000	17959
13	16500	18856
14	17000	19799
15	17500	20789
16	18000	21829
17	18500	22920
18	19000	24066
19	19500	25270
20	20000	26533
	ZW()	26533

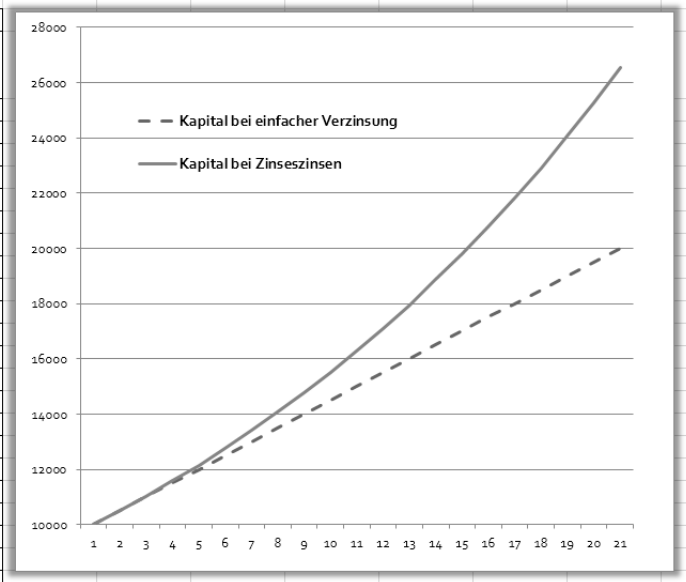


Abbildung 3.3 Einfache Verzinsung und Zinseszins im Vergleich

3.3 Häufig benötigte Argumente und ihr Zusammenhang

Bei zahlreichen finanzmathematischen Funktionen, insbesondere bei der Renten- und Tilgungsrechnung, werden bestimmte Argumente benötigt, so dass es vernünftig scheint, diese vorweg vorzustellen.

Funktionsargument	Erklärung
`Bw [Barwert]`	der gegenwärtige rechnerische Wert einer Reihe regelmäßiger, gleichbleibender Zahlungen, z. B. einer Rente
`Rmz [Ratenzahlung]`	regelmäßig zu leistende oder zu erhaltende Zahlungen
`Zins [Zinssatz]`	Zinssatz für einen Zahlungszeitraum. Er wird dezimal (z. B. `0,08`) oder in Prozent (z. B. `8%`) angegeben.
`Zr [Zeitraum]`	die einzelne Zahlungsperiode im Gesamtzeitraum

Tabelle 3.1 Argumente in der Rentenberechnung

Funktionsargument	Erklärung
`Zw [Zukunftswert]`	Der zukünftige Wert einer Investition. Wird `Zw` als optionales Argument verwandt, setzt Excel den Wert `0` als Vorgabe.
`Zzr [Zahlungszeiträume]`	Anzahl der Zahlungsperioden. Es ist darauf zu achten, dass die Zahlungszeiträume mit den anderen Argumenten in den Einheiten übereinstimmen, d.h., dass einheitlich Jahre, Monate oder Tage in einer Funktion benutzt werden. Werden z.B. Monate verwendet, muss auch der Monatszins (Jahreszins / 12) eingesetzt werden. Beachten Sie hierbei bitte, dass in Deutschland das Zinsjahr in 12 Zinsmonate mit jeweils 30 Tagen eingeteilt wird. In anderen Ländern gelten teilweise andere Regelungen.
`F [Fälligkeit]`	Das Argument `F` für Fälligkeit legt fest, ob die Berechnung für vorschüssige oder nachschüssige Rentenzahlungen etc. erfolgen soll. Wird `F` nicht oder mit `0` angegeben, wird die Funktion als nachschüssig berechnet. Die Zahlungen erfolgen also am Ende der Periode. Wird `F` mit `1` angegeben, so erfolgt die Berechnung für eine vorschüssige Rente.
`q`	Der Zinsfaktor ist gleich `1 + Zins`.

Tabelle 3.1 Argumente in der Rentenberechnung (Forts.)

Der Zusammenhang der in Tabelle 3.1 aufgeführten Werte ist in einer Grundformel definiert, von der die Berechnungsmethoden für die einzelnen Werte abgeleitet werden. Diese Formel lautet:

```
Bw * (1+Zins)^Zzr + Rmz * (1+Zins * F)*(((1+Zins)^Zzr)-1 / Zins) + Zw = 0
```

Abbildung 3.4 zeigt einen Beleg für diese Formel in Zelle A7. Um die Formel nachzubilden, sind die Zellen in der Zeile 5 mit den in Zeile 4 angegebenen Namen benannt. In Zelle A9 ist die Formel als Beispiel auf die Berechnung des Zukunftswerts umgestellt.

Bei allen Funktionen, die mit den in Tabelle 3.1 aufgeführten Argumenten arbeiten, ist immer zu beachten, wer das Subjekt des zugrunde liegenden Vorgangs ist und in welche Richtung das Geld fließt. Wenn Sie Zinsen oder Tilgungen für einen Kredit zu zahlen haben, sind Sie das Subjekt des Vorgangs, und die Zahlungen werden als negative Werte

angegeben, weil das Geld von Ihnen wegfließt. Bei einer Rente ist es umgekehrt, die regelmäßigen Zahlungen werden positiv angegeben, weil Ihnen das Geld ja zufließt.

	A	B	C	D	E	F	G
1							
2	**Barwert und Zukunftswert**						
3							
4	**Bw**	**Zins**	**Rmz**	**Zw**	**F**	**Zzr**	**q**
5	-83,87 €	7%	10,00	10,00	0	12	1,07
6							
7	0,00	**=Bw*(1+Zins)^Zzr+Rmz*(1+Zins*F)*(((1+Zins)^Zzr-1)/Zins)+Zw**					
8							
9	10	**=(Bw*(1+Zins)^Zzr+Rmz*(1+Zins*F)*(((1+Zins)^Zzr)-1)/Zins)*-1**					
10							

Abbildung 3.4 Die Grundformel für die Berechnungen rund um Barwert und Zukunftswert

3.4 Berechnungen zu Darlehen

Ein Gegenstück zu Rentenberechnungen ist die Tilgungsberechnung. In diesem Fall geht es darum, der Bank ein Darlehen in periodischen, gleichbleibenden Raten zurückzuzahlen. Für die klassische Annuitätentilgung wird die Höhe dieser Zahlung mit der Funktion `RMZ()` berechnet. Die Raten bestehen jeweils aus einem Zinsanteil (berechnet mit `ZINSZ()`) und einem Tilgungsanteil, um den sich die Gesamtschuld jeweils verringert. Dieser Anteil wird mit der Funktion `KAPZ()` berechnet. Abbildung 3.5 zeigt ein einfaches Beispiel mit einem Tilgungsplan und einem Diagramm, das noch einmal zeigt, dass der Anteil der Zinsen an der Rate mit jedem Jahr geringer wird.

Tilgung von Darlehen

Jährliche Zahlungen (RMZ) = Tilgung (KAPZ) + Zinsen (ZINSZ)

Zins	Jahr	Zzr	Bw	KAPZ()	ZINSZ()	RMZ()
6,00%	1	10	20.000 €	-1.517,36 €	-1.200,00 €	-2.717,36 €
6,00%	2	10	20.000 €	-1.608,40 €	-1.108,96 €	-2.717,36 €
6,00%	3	10	20.000 €	-1.704,90 €	-1.012,45 €	-2.717,36 €
6,00%	4	10	20.000 €	-1.807,20 €	-910,16 €	-2.717,36 €
6,00%	5	10	20.000 €	-1.915,63 €	-801,73 €	-2.717,36 €
6,00%	6	10	20.000 €	-2.030,57 €	-686,79 €	-2.717,36 €
6,00%	7	10	20.000 €	-2.152,40 €	-564,96 €	-2.717,36 €
6,00%	8	10	20.000 €	-2.281,55 €	-435,81 €	-2.717,36 €
6,00%	9	10	20.000 €	-2.418,44 €	-298,92 €	-2.717,36 €
6,00%	10	10	20.000 €	-2.563,55 €	-153,81 €	-2.717,36 €

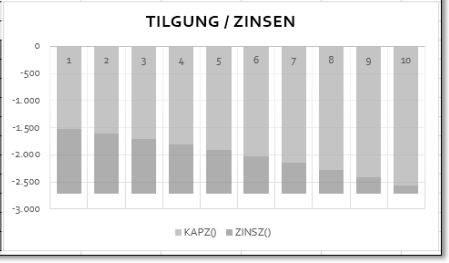


Abbildung 3.5 Tilgungsplan für ein Darlehen

3.5 Die Berechnung von Abschreibungen

Abschreibungen, also Investitionen, z. B. in Maschinen, Fahrzeuge, Gebäude etc., sind betriebswirtschaftlich in zweierlei Hinsicht zu berücksichtigen. Zum einen ist es betriebsintern so, dass mit einer Anschaffung das Anlagevermögen zunächst um den Wert dieser Anschaffung vermehrt wird, dass aber der Wert dieser Anschaffung durch Abnutzung kontinuierlich sinkt, bis am Ende nur noch ein Restwert oder Schrottwert übrig bleibt.

Zum anderen kann eine derartige Investition steuerlich geltend gemacht werden, sie kann »von der Steuer abgesetzt« werden. Die einfachste Form ist hierbei, dass die Anschaffung in dem Jahr, in dem sie getätigt wird, in vollem Umfang steuerlich geltend gemacht wird – hierzu bedarf es natürlich keiner weiteren Berechnungen. Bei langlebigen Investitionen ist es dagegen üblich, die Abschreibung auf mehrere Jahre zu verteilen.

Hierfür gibt es verschiedene Verfahren. Das einfachste ist die lineare Abschreibung: Der abzuschreibende Betrag wird einfach gleichmäßig auf den gesamten Abschreibungszeitraum verteilt. Die anderen Verfahren, für die Excel Funktionen zur Verfügung stellt, sind sogenannte degressive Abschreibungen: Der Abschreibungsbetrag sinkt von Jahr zu Jahr, so dass am Anfang ein großer Betrag abgeschrieben wird, in den folgenden Jahren wird der Betrag kontinuierlich kleiner. Auf die Frage, welche Methode am sinnvollsten und welche zu welchem Zeitraum in einem Land steuerrechtlich zulässig ist, kann Excel natürlich keine Antwort geben.

In diesem Abschnitt wird ein Tabellenblatt aufgebaut, mit dessen Hilfe Sie die Abschreibungsrate für jedes Jahr der Nutzungsdauer für ein beliebiges Wirtschaftsgut mit den verschiedenen Abschreibungsmethoden berechnen können. Welche Steuermethode die aktuelle Steuergesetzgebung zulässt, muss allerdings immer erst geprüft werden, weil es hier in der letzten Zeit einige Veränderungen gegeben hat, insbesondere was die degressive Abschreibung betrifft. In den Zellen C4 bis C7 werden zunächst die Daten abgelegt, die für die verschiedenen Funktionen als Argumente benötigt werden.

In der Spalte B wird die lineare Abschreibung für jedes Jahr der Nutzungsdauer berechnet. Die Formel ist für alle Jahre gleich und arbeitet mit den absoluten Adressen der Zellen C4 bis C6:

```
=LIA($C$4;$C$5;$C$6)
```

Bei der geometrisch-degressiven Abschreibung ist die Formel etwas komplexer, damit sie nach unten kopiert werden kann:

```
=GDA($C$4;$C$5;$C$6;$C$6-($A$19-A10);$C$7)
```

Beachten Sie, dass der Bezug auf die jeweilige Jahreszahl relativ ist, während die anderen Bezüge alle absolut sind. Ähnlich wird auch bei der Formel für die digitale Abschreibung verfahren:

```
=DIA($C$4;$C$5;$C$6;$C$6-($A$19-A10))
```

Die Formel für die vierte Methode ist etwas komplizierter:

```
=VDB($C$4;$C$5;$C$6;$A10-$A$10;$A11-$A$10;$C$7;FALSCH)
```

Wenn Sie die Spalten C und E vergleichen, sehen Sie, dass in den beiden letzten Jahren die geometrisch-degressive Methode geringere Beträge ergibt als die lineare. In diesen Jahren ist also ein Wechsel zur linearen Methode bei den vorliegenden Zahlen sinnvoll.

	A	B	C	D	E
1					
2	**Vergleich der Abschreibungsmethoden**				
3					
4		**Anschaffungswert**	22000		
5		**Restwert**	2000		
6		**Nutzungsdauer Jahre**	10		
7		**Faktor**	2		
8					
9	**Jahr**	**Linear**	**Geometrisch degressiv**	**Arithmetisch degressiv/digital**	**Degressiv mit Wechsel zu linear**
10	1	2000,00	4400,00	3636,36	4400,00
11	2	2000,00	3520,00	3272,73	3520,00
12	3	2000,00	2816,00	2909,09	2816,00
13	4	2000,00	2252,80	2545,45	2252,80
14	5	2000,00	1802,24	2181,82	1802,24
15	6	2000,00	1441,79	1818,18	1441,79
16	7	2000,00	1153,43	1454,55	1153,43
17	8	2000,00	922,75	1090,91	922,75
18	9	2000,00	738,20	727,27	845,49
19	10	2000,00	590,56	363,64	845,49
20	11	20000,00	19637,77	20000,00	20000,00

Abbildung 3.6 Mit diesem Modell können Sie die Wirkung der verschiedenen Abschreibungsmethoden beobachten.

Das Diagramm in Abbildung 3.7 zeigt die unterschiedlichen Verläufe.

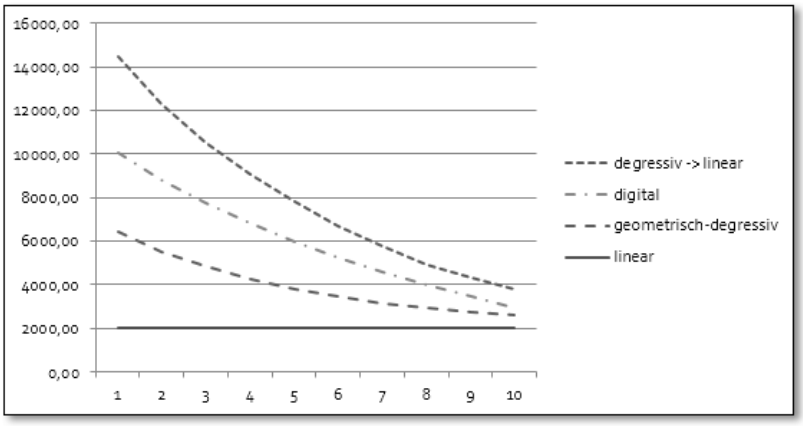


Abbildung 3.7 Grafische Darstellung des Abschreibungsverlaufs

Um das Modell an kürzere oder längere Nutzungsdauern anzupassen, brauchen Sie vor der Zeile mit den Summen nur entsprechend viele Zeilen einzufügen oder zu löschen und die Formeln, wenn nötig, einige Zeilen weiter nach unten zu kopieren. Statt Jahr 1, 2, 3 ... können Sie auch die Jahreszahlen verwenden. Die Formeln in Spalte E brauchen den Wert, der in Zelle A20 steht, also ein Jahr, das über das letzte Jahr des Abschreibungszeitraums hinausreicht.

3.6 Funktionen für Wertpapierberechnungen

Eine große Gruppe von Funktionen kreist um festverzinsliche und unverzinsliche Wertpapiere. Auch hierzu einige Bemerkungen.

Festverzinsliche Wertpapiere sind Wertpapiere, die für einen bestimmten Zeitraum Gültigkeit haben und in dieser Zeit regelmäßig verzinst werden. Das können Sparbriefe, Schuldverschreibungen, private oder öffentliche Anleihen und Obligationen sein. Festverzinsliche Anleihen werden auch Kuponanleihen genannt, wegen der Zinsscheine (Kupons), die früher das Recht auf die Zinszahlung verbrieften.

Unverzinsliche Wertpapiere sind solche, die zu einem bestimmten Kurs ausgegeben und später zu einem höheren Kurs wieder zurückgenommen werden. De facto könnte natürlich auch hier von einer Verzinsung geredet werden, da der Wert des Papiers ja um einen bestimmten Betrag steigt, der sich durchaus auf einen jährlichen prozentualen Steigerungsbetrag umrechnen ließe. Solche Papiere werden auch als Nullkuponanleihen oder Zerobonds bezeichnet.

3.6.1 Termine

Ausgehend von dem Modell, dass ein derartiges Wertpapier (festverzinslich oder unverzinslich) bei einem Geldinstitut erworben und später wieder veräußert wird, sind folgende Termine von zentraler Bedeutung:

- **Emission**: der Ausgabetermin des Wertpapiers. Das ist der Termin, zu dem das Papier auf den Markt gebracht wird und ab dem sein Wertzuwachs bzw. seine Verzinsung läuft.
- **Abrechnung**: der Termin, zu dem ein Käufer ein Wertpapier erwirbt. Er kann mit dem Emissionstermin zusammenfallen, er kann aber auch – und das ist für die Berechnungen natürlich der interessantere Fall – ein späterer Termin sein.
- **Fälligkeit**: der Termin, zu dem die Bank das Wertpapier zurücknimmt und dem Eigentümer einen vorher vereinbarten Betrag dafür bezahlt.

Alle Werte für solche Termine verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Für Ausgabe und Rücknahme derartiger Wertpapiere gibt es drei Möglichkeiten:

1. Die Papiere werden zum Nennwert verkauft. Anschließend werden für die Laufzeit regelmäßig Zinsen ausgeschüttet (üblich ist ein- oder zweimal jährlich). Die Zinszahlungstermine werden in der Regel vom Emissionstermin her berechnet. Am Ende der Laufzeit wird das Papier zum Nennwert zurückgenommen. Bisweilen wird bei der Ausgabe noch ein Disagio vorgenommen, d. h., vom Nennwert wird ein bestimmter Prozentsatz (eventuell noch orientiert an der Laufzeit) abgezogen.
2. Das Papier wird abgezinst verkauft, d. h. zum Nennwert vermindert um den Betrag, der durch Zinsen während der Laufzeit hinzukommt, und ohne zwischenzeitliche Zinsausschüttungen am Ende der Laufzeit zum Nennwert zurückgenommen.
3. Das Papier wird zum Nennwert verkauft und ohne zwischenzeitliche Zinsausschüttungen am Ende der Laufzeit aufgezinst, d. h., zum Nennwert vermehrt um die aufgelaufenen Zinsen zurückgenommen.

Die zweite und die dritte Methode gelten sinngemäß so auch für unverzinsliche Papiere.

Da derartige Wertpapiere zum Teil auch während der Laufzeit gehandelt werden, stellt sich die Frage nach ihrem Augenblickswert, das ist der Kurs, zu dem sie gehandelt wer-

den können. Dieser Kurs sollte natürlich marktgerecht sein, d. h., ein potenzieller Käufer muss die Möglichkeit haben, zu überprüfen, ob sich die Investition in das betreffende Papier im Vergleich zu anderen möglichen Investitionen lohnt. Hierzu wird als Vergleichsgröße die Rendite eingeführt, der effektive jährliche Gewinn in Prozent vom Kapitaleinsatz. Die Wahl dieser Größe ist auf eine Art willkürlich: Der Kurs wird auf der Grundlage der Rendite berechnet, die mit anderen Anlageformen erzielt werden könnte.

Findet ein Besitzwechsel zwischen zwei Zinsperioden statt, fallen bei festverzinslichen Anleihen **Stückzinsen** an. Der Verkäufer erhält vom Käufer die Zinsen erstattet, die in dem Zeitraum zwischen dem letzten Zinstermin und dem Kauftermin anfallen, der Käufer erhält dafür beim nächsten Zinstermin die Zinsen für die komplette Zinsperiode.

3.6.2 Zeitbasis

In einer Anzahl von Funktionen taucht das Argument `Basis` auf. Gemeint ist damit bei Zinsberechnungen die Zeitbasis, auf der gerechnet wird. Die angebotenen Optionen sind:

Option	Bedeutung
`0`	US-amerikanisches System: Die Monate werden mit 30 Tagen, das Jahr wird mit 360 Tagen gerechnet. Ein Enddatum zum 31. eines Monats wird, falls das Startdatum vor dem 30. eines Monats liegt, als 1. Tag des Folgemonats behandelt, sonst als 30. des Monats.
`1`	Die tatsächlichen Tage der Monate und die tatsächlichen Tage der Jahre werden gerechnet.
`2`	Es wird mit den tatsächlichen Tagen des Monats gerechnet, für das Jahr werden aber 360 Tage angesetzt.
`3`	Es wird mit den tatsächlichen Tagen des Monats gerechnet, für das Jahr werden aber 365 Tage angesetzt.
`4`	Europäisches System: Die Monate werden mit 30 Tagen, das Jahr wird mit 360 Tagen gerechnet. Fällt ein Start- oder Enddatum auf den 31. eines Monats, wird es als 30. Tag behandelt.

Tabelle 3.2 Optionen für das Argument `Basis`

Wird `Basis` nicht angegeben, rechnet Excel mit `Basis = 0`.

3.6.3 Verkauf eines festverzinslichen Wertpapiers

Wenn Sie festverzinsliche Wertpapiere besitzen und einen Teil davon zu einem bestimmten Termin verkaufen wollen, können Sie sich mit Hilfe der finanzmathematischen Funktionen selbst ausrechnen, was Sie ausgezahlt bekommen. Sie brauchen dazu natürlich die Angabe über den Kurs für den Abrechnungstermin, also den Tag, an dem verkauft werden soll. Der Abrechnung beim Kauf der Papiere können Sie die übrigen Daten entnehmen, die nötig sind: das Datum der Emission des Papiers, das Datum der Fälligkeit, den Nominalzinssatz, den Nennwert und die Anzahl der Zinstermine pro Jahr. Aus der Anzahl der Zinstermine ergibt sich der erste Zinstermin.

Excel braucht zusätzlich noch die Angabe über die `Basis`, auf der die Zinstage berechnet werden. Das ist hier immer der Wert `4`, also 360 Tage pro Jahr, 30 Tage pro Monat. In der Zelle B13 wird der aktuelle Kurswert einfach mit

```
=Nennwert * Kurs/100
```

errechnet. Zusätzlich wird noch die Anzahl der Zinstage berechnet. Dafür kann die Formel

```
=ZINSTERMTAGVA(Abrechnungstermin;Faelligkeit; 1; 4)
```

verwendet werden. Das Ergebnis dient hier aber nur zur Information, für die Berechnung der inzwischen aufgelaufenen Zinsen wird dieser Wert nicht benötigt.

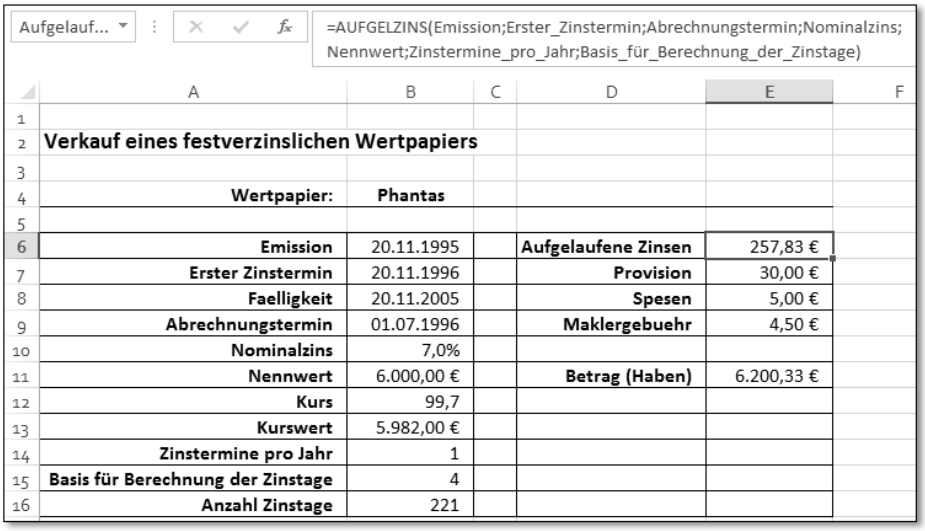
Neben dem Kurs spielt beim Verkauf der Betrag der aufgelaufenen Zinsen eine Rolle. Der wird in E6 mit der Funktion `AUFGELZINS()` berechnet. Die Formel lautet:

```
=AUFGELZINS(Emission; Erster_Zinstermin;
Abrechnungstermin;
Nominalzins; Nennwert;
Zinstermine_pro_Jahr;
Basis_für_Berechnung_der_Zinstage)
```

Nun kommen in der Regel aber noch ein paar Kosten hinzu:

Als Provision berechnet die Bank beispielsweise 0,5 % des Nennwerts. Als Maklergebühr sind 0,075 % des Nennwerts zu zahlen. Dann muss noch eine fixe Spesengebühr entrichtet werden. Der tatsächlich ausgezahlte Betrag ergibt sich somit aus dem Kurswert plus aufgelaufene Zinsen abzüglich der drei genannten Posten.

Beachten Sie, dass bei ausländischen Papieren meist leicht abweichende Maklergebühren berechnet werden.



	A	B	C	D	E	F
1						
2	**Verkauf eines festverzinslichen Wertpapiers**					
3						
4	**Wertpapier:**	**Phantas**				
5						
6	**Emission**	20.11.1995		**Aufgelaufene Zinsen**	257,83 €	
7	**Erster Zinstermin**	20.11.1996		**Provision**	30,00 €	
8	**Faelligkeit**	20.11.2005		**Spesen**	5,00 €	
9	**Abrechnungstermin**	01.07.1996		**Maklergebuehr**	4,50 €	
10	**Nominalzins**	7,0%				
11	**Nennwert**	6.000,00 €		**Betrag (Haben)**	6.200,33 €	
12	**Kurs**	99,7				
13	**Kurswert**	5.982,00 €				
14	**Zinstermine pro Jahr**	1				
15	**Basis für Berechnung der Zinstage**	4				
16	**Anzahl Zinstage**	221				

Abbildung 3.8 Verkauf eines festverzinslichen Wertpapiers

3.7 Referenz der finanzmathematischen Funktionen

AMORDEGRK()

AMORDEGRC()

Syntax:	`AMORDEGRK(`**`Ansch_Wert`**`; `**`Kaufdatum`**`; `**`Erster_Zinstermin`**`;`**`Restwert`**`; `**`Termin`**`; `**`Satz`**`; Basis)`
Beispiel:	`=AMORDEGRK(10000;DATUM(2012;6;6);DATUM(2012;12;31);0;0;0,25;4)`
Ergebnis:	`2125`

Die Funktion `AMORDEGRK()` liefert den Abschreibungsbetrag nach dem französischen Buchhaltungssystem für eine bestimmte Periode.

Das Argument `Ansch_Wert` nennt den Anschaffungswert, `Kaufdatum` ist das Datum der Anschaffung, `Erster_Zinstermin` meint das Datum zum Ende der ersten Periode. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Restwert` ist der Wert des Wirtschaftsguts nach Ablauf aller Abschreibungsperioden. `Termin` gibt die Periode an, für die der Abschreibungsbetrag errechnet werden soll.

Die erste Periode wird als `0` angegeben. `Satz` ist die Abschreibungsrate für die zunächst linear angegebene Abschreibung als Prozentsatz. Sie ist abhängig von der geplanten Nutzungsdauer. 20% entsprechen 5 Jahren, 10% entsprechen 10 Jahren.

`Basis` gibt die Zeitbasis an, die bei der Berechnung verwendet wird. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben. Der Wert `2` ist aber bei dieser Funktion nicht erlaubt.

Ungültige Datumswerte oder nicht numerische Werte für `Ansch_Wert`, `Restwert` oder `Satz` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Basis` zu dem Fehler `#ZAHL!`. Die Nutzungsdauer wird bei dieser Funktion nicht direkt angegeben, sondern indirekt über den Wert für `Satz`. Ist dieser Wert zu groß, ist die Nutzungsdauer zu kurz angesetzt. Bei Werten bis zu zwei Jahren liefert die Funktion den Fehler `#ZAHL!`.

	A	B	C	D	E
1					
2	**Abschreibungsbetrag** (französisches Buchführungssystem)				
3					
4	Anschaffungswert	10000	10000	10000	10000
5	Kaufdatum	06.06.2012	06.06.2012	06.06.2012	06.06.2012
6	Erste Periode	31.12.2012	31.12.2012	31.12.2012	31.12.2012
7	Restwert	0	0	0	0
8	Periode	0	1	2	3
9	Abschreibungssatz	25%	25%	25%	25%
10	Basis	4	4	4	4
11	AMORDEGRK()	2125	2953	2461	2461

Abbildung 3.9 Degressive Abschreibung nach dem französischen System

Die Funktion arbeitet anders als die folgende Funktion `AMORLINEARK()` zusätzlich noch mit einem Abschreibungskoeffizienten, der für den degressiven Verlauf der Abschreibung sorgt. Dieser Koeffizient ist von der Nutzungsdauer abhängig.

Nutzungsdauer	Abschreibungskoeffizient
Zwischen 3 und 4 Jahre	1,5
Zwischen 5 und 6 Jahre	2
Mehr als 6 Jahre	2,5

Tabelle 3.3 Abschreibungskoeffizienten in Abhängigkeit von der Nutzungsdauer

Ist die erste Periode nicht vollständig, wird eine zusätzliche Periode eingeführt und ein anteiliger Abschreibungsbetrag errechnet. Für die erste Periode wird der Abschreibungssatz verwendet. Für die weiteren Perioden wird der Abschreibungsbetrag nach der Formel

```
Buchwert * (Abschreibungssatz * Koeffizient)
```

ermittelt. Ist der Restwert auf 0 angesetzt, kann dieser nach der vorgesehenen Nutzungsdauer nicht erreicht werden. Deshalb wird der Abschreibungsbetrag auf die beiden letzten Perioden so verteilt, dass die Gesamtsumme der Abschreibungsbeträge die Anschaffungskosten nicht überschreitet.

AMORLINEARK()

AMORLINC()

Syntax: `AMORLINEARK(`**`Ansch_Wert`**`; `**`Kaufdatum`**`; `**`Erster_Zinstermin`**`; `**`Restwert`**`; `**`Termin`**`; `**`Satz`**`; Basis)`

Beispiel:
```
=AMORLINEARK(10000;DATUM(2012;7;1);
DATUM(2012;12;31);0;0;0,33;4)
```

Ergebnis: `1657`

Die Funktion `AMORLINEARK()` liefert den Abschreibungsbetrag nach dem französischen Buchhaltungssystem für eine bestimmte Periode.

Das Argument `Ansch_Wert` nennt den Anschaffungswert, `Kaufdatum` ist das Datum der Anschaffung, `Erster_Zinstermin` meint das Datum zum Ende der ersten Periode. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Restwert` ist der Wert des Wirtschaftsguts nach Ablauf aller Abschreibungsperioden. `Termin` gibt den Zeitraum an, für den der Abschreibungsbetrag errechnet werden soll.

Die erste Periode wird als `0` angegeben. `Satz` ist die Abschreibungsrate für die zunächst linear angegebene Abschreibung als Prozentsatz. Sie ist abhängig von der geplanten Nutzungsdauer. 20% entsprechen 5 Jahren, 10% entsprechen 10 Jahren.

`Basis` gibt die Zeitbasis an, die bei der Berechnung verwendet wird. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben. Der Wert `2` ist aber bei dieser Funktion nicht erlaubt.

Ungültige Datumswerte oder nicht numerische Werte für `Ansch_Wert`, `Restwert` oder `Satz` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Basis` zu dem Fehler `#ZAHL!`. Die Nutzungsdauer wird bei dieser Funktion nicht direkt angegeben, sondern indirekt über den Wert für `Satz`. Ist dieser Wert zu groß, ist die Nutzungsdauer zu kurz angesetzt. Bei Werten bis zu zwei Jahren liefert die Funktion den Fehler `#ZAHL!`.

	A	B	C	D	E
1					
2	**Abschreibungsbetrag** (französisches Buchführungssystem)				
3					
4	Anschaffungswert	10000	10000	10000	10000
5	Kaufdatum	01.07.2012	01.07.2012	01.07.2012	01.07.2012
6	Erste Periode	31.12.2012	31.12.2012	31.12.2012	31.12.2012
7	Restwert	0	0	0	0
8	Periode	0	1	2	3
9	Abschreibungssatz	33%	33%	33%	33%
10	Basis	4	4	4	4
11	AMORLINEARK()	1657	3333	3333	1677

Abbildung 3.10 Lineare Abschreibung nach dem französischen System

Ist die erste Periode nicht vollständig, wird eine zusätzliche Periode eingeführt und ein anteiliger Abschreibungsbetrag errechnet. In der letzten Periode wird dann der aus der ersten Periode übrig gebliebene Rest abgeschrieben.

AUFGELZINS()

ACCRINT()

Syntax: `AUFGELZINS(`**`Emission`**`; `**`Erster_Zinstermin`**`; `**`Abrechnung`**`; `**`Satz`**`; `**`Nennwert`**`; `**`Häufigkeit`**`; `**`Basis`**`; Berechnungsmethode)`

Beispiel: `=AUFGELZINS(DATUM(2013;1;1); DATUM(2014;1;1);DATUM(2013;8;3); 3%;10000;1;4;WAHR)`

Ergebnis: `176,67`

Die Funktion `AUFGELZINS()` liefert für festverzinsliche Wertpapiere die Summe der aufgelaufenen Zinsen (Stückzinsen) für einen bestimmten Zeitraum bei periodischen Zinszahlungen. Bei solchen Papieren bezahlt der Käufer des Papiers Stückzinsen, wenn der Kauf vor dem Termin einer Zinszahlung stattfindet. Die Funktion dient dazu, den Kaufpreis von festverzinslichen Papieren zu kalkulieren, der sich aus dem aktuellen Kurswert und den Stückzinsen zusammensetzt.

Emission ist das Datum der Ausgabe des Wertpapiers (ab diesem Termin wird das Papier verzinst). Erster_Zinstermin ist der Termin, an dem zum ersten Mal Zinsen fällig sind. Mit Abrechnung wird der Tag angegeben, an dem das Wertpapier den Besitzer wechselt. Das ist zugleich der Stichtag, für den die aufgelaufenen Zinsen berechnet werden sollen. Diese drei Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Es folgen mit Satz der Zinssatz des Wertpapiers und der Nennwert. Wird Nennwert nicht angegeben, wird mit 1000 gerechnet.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Berechnungsmethode ist ein Wahrheitswert, über den bestimmt wird, wie gerechnet werden soll, wenn das Abrechnungsdatum nach dem ersten Zinstermin liegt. WAHR oder 1 oder nicht angegeben bedeutet, dass die insgesamt zwischen Emission und Abrechnung aufgelaufenen Zinsen ausgegeben werden. FALSCH oder 0 bewirkt, dass nur die Zinsen von Erster_Zinstermin bis Abrechnung ausgegeben werden.

Ungültige Datumswerte oder nicht numerische Werte für Satz, Nennwert oder Berechnungsmethode führen zu dem Fehler #WERT!, nicht erlaubte Werte für Häufigkeit oder Basis zu dem Fehler #ZAHL!. Das gilt auch, wenn Satz oder Nennwert ≤ 0 oder Emission ≥ Abrechnung.

	A	B
1		
2	**Aufgelaufene Zinsen bei Wertpapieren**	
3		
4	Emission	01.01.2013
5	Erster_Zinstermin	01.01.2014
6	Abrechnung	03.08.2013
7	Zinssatz	3%
8	Nennwert	10.000,00 €
9	Häufigkeit	1
10	Basis	4
11	Berechnungsmethode	1
12	AUFGELZINS()	176,67 €

Abbildung 3.11 Berechnung der aufgelaufenen Zinsen bei einem festverzinslichen Wertpapier

Die Funktion berechnet die Stückzinsen im unterjährigen Bereich. Liegt der Abrechnungstermin nach dem Termin der ersten Zinszahlung, muss der Zins für frühere Perioden abgezogen werden, um die beim Kauf fälligen Stückzinsen zu errechnen. Im Beispiel in Abbildung 3.11 liefert die Funktion beispielsweise Stückzinsen von 476,67, wenn als Abrechnungstermin der 03.08.2014 genommen wird. Der Jahreszins von 300 muss also abgezogen werden, um die beim Kauf fälligen Stückzinsen von 176,67 zu erhalten. Am einfachsten erreichen Sie das dadurch, dass Sie anstelle des Termins für die Emission den Termin der letzten Zinszahlung angeben.

AUFGELZINSF()

ACCRINTM()

Syntax:	AUFGELZINSF(**Emission**; **Abrechnung**; **Nominalzins**; Nennwert; Basis)
Beispiel:	=AUFGELZINSF(DATUM(2013;1;4); DATUM(2013;07;22);0,03;10000;4)
Ergebnis:	165,00

Die Funktion AUFGELZINSF() ist eine vereinfachte Variante zu AUFGELZINS(). Sie liefert für Wertpapiere die Summe der aufgelaufenen Zinsen (Stückzinsen) für einen bestimmten Zeitraum bei einmaliger Zahlung im Jahr. Sie dient dazu, den Kaufpreis von festverzinslichen Papieren zu kalkulieren.

Emission ist das Datum der Ausgabe des Wertpapiers (ab diesem Termin wird das Papier verzinst). Mit Abrechnung wird der Stichtag angegeben, für den die aufgelaufenen Zinsen berechnet werden. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Nominalzins ist der Zinssatz des Wertpapiers und Nennwert der Nominalwert des Wertpapiers. Wird der Nennwert nicht angegeben, wird mit 1000 gerechnet. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für Nominalzins und Nennwert führen zu dem Fehler #WERT!, nicht erlaubte Werte für Basis zu dem Fehler #ZAHL!. Das gilt auch, wenn Nominalzins oder Nennwert ≤ 0 oder Emission ≥ Abrechnung.

	A	B	C	D	E	F
1						
2	**Aufgelaufene Zinsen bei jährlicher Zinszahlung**					
3						
4	Emission	04.01.2013	04.01.2013	04.01.2013	04.01.2013	04.01.2013
5	Abrechnung	22.07.2013	22.07.2013	22.07.2013	22.07.2013	22.07.2013
6	Nominalzins	3,0%	3,0%	3,0%	3,0%	3,0%
7	Nennwert	10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €
8	Basis	0	1	2	3	4
9	AUFGELZINSF()	165,00 €	163,56 €	165,83 €	163,56 €	165,00 €

Abbildung 3.12 Vereinfachte Berechnung der aufgelaufenen Zinsen

Die Funktion berechnet die Stückzinsen im unterjährigen Bereich. Liegt der Abrechnungstermin mehr als ein Jahr nach dem Termin der Emission, muss der Zins für frühere Zinsperioden abgezogen werden, um die beim Kauf fälligen Stückzinsen zu errechnen. Am einfachsten erreichen Sie das dadurch, dass Sie anstelle des Termins für die Emission den Termin der letzten Zinszahlung angeben.

AUSZAHLUNG()

INTRATE()

Syntax: `AUSZAHLUNG(`**`Abrechnung; Fälligkeit; Anlage; Disagio`**`; Basis)`

Beispiel: `=AUSZAHLUNG(DATUM(2013;1;1); DATUM(2014;1;1);10000;5,25%;4)`

Ergebnis: `10554,09`

Die Funktion `AUSZAHLUNG()` liefert den Rückzahlungsbetrag, den Sie für eine Einlage mit einfacher vorschüssiger Abzinsung zu einem bestimmten Zeitpunkt erhalten.

Mit `Abrechnung` wird der Zeitpunkt der Einzahlung angegeben, mit `Fälligkeit` der Zeitpunkt, zu dem die Rückzahlung erfolgt und zu dem der Auszahlungsbetrag berechnet werden soll. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Anlage` ist die Höhe der Einlage. (Die Funktion setzt voraus, dass der Gesamtbetrag in die Einlage investiert wird.) `Disagio` ist der prozentuale Abschlag, der der vorschüssigen Verzinsung entspricht (bei 5% Disagio z.B. wird für einen Auszahlungsbetrag von

100 € nur 95 € eingezahlt). `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D	E	F
1						
2	**Auszahlungsbetrag einer Einlage**					
3						
4	Abrechnung	01.01.2013	01.01.2013	01.01.2013	01.01.2013	
5	Fälligkeit	01.01.2014	01.01.2014	01.01.2014	01.01.2014	
6	Anlage	10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €	
7	Disagio	5,25%	5,25%	5,25%	5,25%	
8	Basis	1	2	3	4	
9	AUSZAHLUNG()	10.554,09 €	10.562,22 €	10.554,09 €	10.554,09 €	
10						

Abbildung 3.13 Berechnung des Auszahlungsbetrags

Ungültige Datumswerte oder nicht numerische Werte für `Anlage` oder `Disagio` führen zu dem Fehler `#WERT!`. Nicht erlaubte Werte für `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn `Anlage` oder `Disagio` ≤ 0 oder `Abrechnung` ≥ `Fälligkeit`.

BW()

PV()

Syntax: `BW(`**`Zins; Zzr; Rmz`**`; Zw; F)`
Beispiel: `=BW(0,04;20;10000;;0)`
Ergebnis: `-135.903,26`

Die Funktion `BW()` berechnet den Barwert einer Investition. Gemeint ist damit der Gesamtbetrag, mit dem der Wert einer Reihe von regelmäßigen zukünftigen Zahlungen angegeben werden kann. Ein einfaches Beispiel: Wer 20 Jahre lang jedes Jahr 10.000 € von der Bank erhalten will, müsste heute 135.903,26 € zu 4% Zinsen dort einzahlen. Das Ergebnis im oben angegebenen Beispiel ist negativ, weil ja dieser Betrag zu zahlen wäre.

Mit `Zins` wird der Zinssatz und mit `Zzr` werden die Zahlungszeiträume – Perioden – und damit die Anzahl der Zahlungen und Zinszahlungen angegeben. Dabei ist darauf zu achten, dass jeweils die passenden Zeiteinheiten verwendet werden. Als Vorgabe wird `Zins` als Jahreszinssatz interpretiert. Soll mit einer halbjährlichen, vierteljährlichen oder mo-

natlichen Verzinsung gearbeitet werden, muss die Anzahl der Jahreszeiträume mit `2`, `4` oder `12` multipliziert, der Jahreszinssatz entsprechend durch `2`, `4` oder `12` dividiert werden.

`Rmz` ist der Betrag, der periodisch gezahlt wird. Dieser Betrag ist, wenn er angegeben wird, während der ganzen Laufzeit konstant. Zusätzlich kann mit `Zw` noch ein Zukunftswert angegeben werden, der nach der letzten Zahlung erreicht werden soll (im obigen Beispiel wäre das der Betrag, der nach 20 Jahren auf der Bank übrig sein soll). Wird `Zw` nicht angegeben, so wird er als `0` angenommen.

Die Funktion benötigt, wenn `Zw` nicht angegeben wird, einen Wert für `Rmz`. Umgekehrt kann aber auch ohne einen Wert für `Rmz` gerechnet werden, wenn `Zw` angegeben ist. Mit `F` wird noch angegeben, ob die Zahlungen jeweils – nachschüssig – am Ende einer Periode erfolgen (`F = 0` oder keine Angabe) oder – vorschüssig – also am Anfang (`F = 1`).

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`.

Die Berechnung des Barwerts in Excel basiert auf der in Abschnitt 3.3, »Häufig benötigte Argumente und ihr Zusammenhang«, bereits vorgestellten Grundformel:

```
Bw * (1+Zins)^Zzr + Rmz * (1+Zins * F) * (((1+Zins)^Zzr)-1 / Zins) + Zw = 0
```

3.7.1 Barwert regelmäßiger Zahlungen

In Abbildung 3.14 wird verglichen, wie sich der Barwert ändert, wenn regelmäßige Auszahlungen vorschüssig oder nachschüssig verzinst werden, und welchen Einfluss kürzere Zahlungs- und Zinsperioden haben. Bei der quartalsweisen Zahlung und Verzinsung wird der Jahreszins durch 4 dividiert und die Zahl der Perioden mit 4 multipliziert. Bei der monatlichen Zahlung und Verzinsung wird durch 12 dividiert bzw. multipliziert.

	A	B	C	D	E	F	G
1							
2	**Den Barwert regelmäßiger Zahlungen berechnen**						
3							
4	Zins	4%	4%	1%	1%	0,33%	0,33%
5	Zzr	20	20	80	80	240	240
6	Rmz	10.000,00 €	10.000,00 €	2.500,00 €	2.500,00 €	833,33 €	833,33 €
7	Zw						
8	F	0	1	0	1	0	1
9	BW()	-135.903,26 €	-141.339,39 €	-137.220,52 €	-138.592,72 €	-137.518,22 €	-137.976,61 €
10		Jährliche Zahlung		Quartalsweise Zahlung		Monatliche Zahlung	

Abbildung 3.14 Barwertberechnung mit regelmäßigen Zahlungen

Die Ergebnisse der BW()-Funktion sind in diesem Fall negativ, da ja ein Betrag errechnet wird, der eingezahlt werden müsste, um am Ende einen Wert zu erreichen, der allen Auszahlungen plus Zins und Zinseszinsen entspricht. Die Unterschiede in den errechneten Barwerten für die unterschiedlichen Perioden sind deutlich.

3.7.2 Tilgungsrechnung

Während in dem letzten Beispiel die Bank Ihnen regelmäßig etwas zahlen soll, kann die Funktion BW() auch den umgekehrten Fall abbilden. Sie überlegen, einen Kredit aufzunehmen, und gehen davon aus, dass Sie sich dafür im Monat mit 500 € belasten können. Wie hoch kann der Kredit sein, wenn Sie eine Laufzeit von 5, 6 oder 8 Jahren anvisieren?

In diesem Fall muss der Wert für Rmz als negativer Wert angegeben werden, da sich das Geld ja, anders als vorhin, von Ihnen weg in Richtung Bank bewegt. Das Ergebnis der Funktion dagegen ist diesmal positiv, denn es entspricht ja der Höhe der Kredits, also des Betrags, der sich von der Bank zu Ihnen bewegt. Der Wert für Zins wird durch die Division des Jahreszinssatzes durch 12 ermittelt. Die Werte für Zzr werden dagegen mit 12 multipliziert.

	A	B	C	D
1				
2	**Barwert für einen Zukunftswert berechnen**			
3				
4	Zins	0,50%	0,50%	0,50%
5	Zzr	60	72	96
6	Rmz	500,00 €	500,00 €	500,00 €
7	Zw			
8	F	0	0	0
9	BW()	-25.862,78 €	-30.169,76 €	-38.047,61 €

Abbildung 3.15 Barwertberechnung zur Abschätzung einer möglichen Kreditsumme

3.7.3 Barwert für einen zukünftigen Wert berechnen

Wie schon angesprochen, kann die Funktion auch eingesetzt werden, um zu berechnen, welcher einmalige Betrag eingesetzt werden müsste, um nach einer bestimmten Anzahl von Perioden einen angestrebten Endwert zu erreichen. In dem Beispiel in Abbildung 3.16 findet eine solche Rechnung für verschiedene Zeiträume statt. Zeile 9 zeigt, dass in diesem Fall die Frage der Fälligkeit keine Rolle spielt.

	A	B	C	D	E	F
1						
2	**Barwert für einen Zukunftswert berechnen**					
3						
4	Zins	4,00%	4,00%	4,00%	4,00%	
5	Zzr	10	10	12	12	
6	Rmz					
7	Zw	150.000,00 €	150.000,00 €	150.000,00 €	150.000,00 €	
8	F	0	1	0	1	
9	BW()	-101.334,63 €	-101.334,63 €	-93.689,56 €	-93.689,56 €	
10						

Abbildung 3.16 Barwertberechnung für einen anvisierten Zukunftswert

DIA()

SYD()

Syntax: `DIA(Ansch_Wert; Restwert; Nutzungsdauer; Zr)`
Beispiel: `=DIA(20000;2000;3;2)`
Ergebnis: `6666,67`

Die Funktion `DIA()` berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der arithmetisch-degressiven Methode, die auch als digitale Methode bezeichnet wird. Für diese Methode gilt: Wenn `n` die Anzahl der Jahre ist, dann ist der Abschreibungsbetrag im ersten Jahr `n` mal so hoch wie der Betrag im letzten Jahr, im zweiten Jahr n-1 mal so hoch etc., mit anderen Worten, die Abschreibung wird jedes Jahr genau um den Betrag vermindert, der im letzten Jahr abgeschrieben wird.

Diese Methode verbindet den Vorteil der degressiven Methode, dass nämlich in den ersten Jahren höhere Beträge abgeschrieben werden können, mit dem Merkmal der linearen Methode, dass auch in den letzten Jahren noch nennenswerte Abschreibungsbeträge anfallen, da die Degression linear vorgenommen wird. Diese Methode ist allerdings nach deutschem Steuerrecht nicht zulässig.

Das Argument `Ansch_Wert` nennt die Anschaffungskosten, die für die Investition aufgewandt wurden. `Restwert` ist der Wert, den die Investition nach der Abschreibung noch hat. `Nutzungsdauer` entspricht der Anzahl der Perioden bis zum Ende der Abschreibung. `Zr` gibt das Jahr an, also den Zeitraum, für den die Abschreibung errechnet werden soll.

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`. Ist einer der Werte negativ oder `Zr > Nutzungsdauer`, erscheint der Fehler `#ZAHL!`.

	A	B	C	D
1				
2	**Digitale Abschreibung**			
3				
4	Anschaffungskosten	22.000,00 €	22.000,00 €	22.000,00 €
5	Restwert	2.000,00 €	2.000,00 €	2.000,00 €
6	Nutzungsdauer in Jahren	3	3	3
7	Jahr	1	2	3
8	DIA()	10.000,00 €	6.666,67 €	3.333,33 €
9			Gesamt	20.000,00 €

Abbildung 3.17 Berechnung der digitalen Abschreibung bei einer Nutzungsdauer von drei Jahren

DISAGIO()

DISC()

Syntax: `DISAGIO(`**`Abrechnung; Fälligkeit; Kurs; Rückzahlung`**`; Basis)`

Beispiel: `=DISAGIO(DATUM(2013;10;12); DATUM(2014;9;1);95;100;4)`

Ergebnis: `5,64%`

Die Funktion `DISAGIO()` berechnet das Disagio, also den Abzinsungssatz beim Handel mit einem Wertpapier. Dabei handelt es sich um einen Preisabschlag auf den Nennwert eines Papiers, der auch als Abgeld bezeichnet wird, im Gegensatz zum Agio, dem Aufgeld auf den Nennwert.

Dieses Verfahren wird beispielsweise bei den von der Bundesrepublik Deutschland angebotenen Finanzierungsschätzen verwendet. Der Erwerber zahlt beim Kauf einen geringeren Betrag, als er später bei der Einlösung zum festgelegten Fälligkeitstag zurückerhält. Die Zinsen für die Zeit zwischen Kauf und Rückzahlung werden also – tagesgenau berechnet – vorschüssig vom Nennwert abgezogen.

Die Funktion benötigt die folgenden Argumente: `Abrechnung` ist der Termin des Kaufs, `Fälligkeit` ist der Termin, zu dem das Wertpapier abläuft. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Kurs` ist der Wert zum Kauftermin pro 100 € Nennwert, und `Rückzahlung` ist der Wert am Fälligkeitstag pro 100 € Nennwert. `Basis` gibt an, nach welchem Zeitsystem die Zins-

tage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D	E
1					
2	**Abschlag eines Wertpapiers**				
3					
4	Abrechnung	12.10.2013	12.10.2013	12.10.2013	12.10.2013
5	Fälligkeit	01.09.2014	01.09.2014	01.09.2014	01.09.2014
6	Kurs	95,00 €	95,00 €	95,00 €	95,00 €
7	Rückzahlung	100,00 €	100,00 €	100,00 €	100,00 €
8	Basis	1	2	3	4
9	DISAGIO()	5,63%	5,56%	5,63%	5,64%

Abbildung 3.18 Berechnungsvarianten für den Abschlag bei einem Wertpapier

Ungültige Datumswerte führen zu dem Fehler `#WERT!`. Ist `Kurs` oder `Rückzahlung` ≤ `0` oder `Abrechnung` ≥ `Fälligkeit` oder der Wert für `Basis` ungültig, liefert die Funktion den Fehler `#ZAHL!`.

DURATION()

DURATION()

Syntax:	`DURATION(`**`Abrechnung`**`; `**`Fälligkeit`**`; `**`Nominalzins`**`; `**`Rendite`**`; `**`Häufigkeit`**`; Basis)`
Beispiel:	`=DURATION(DATUM(2013;6;19); DATUM(2018;8;27);0,04;0,0349;1;4)`
Ergebnis:	`4,644`

Die Funktion `DURATION()` liefert für einen Nennwert von 100 Einheiten die hypothetische durchschnittliche Kapitalbindungsdauer eines festverzinslichen Wertpapiers in Jahren. Diese Kennzahl wird nach ihrem Erfinder auch Macaulay-Dauer oder Macaulay-Duration genannt. Sie wird verwendet, um das Kursänderungsrisiko abzuschätzen, das als Folge von Schwankungen des Marktzinses zu erwarten ist.

Dabei wird ein gewichteter Mittelwert der einfachen Barwerte aller in dem betreffenden Zeitraum anfallenden Zahlungsvorgänge ermittelt. Die zweite Tabelle in Abbildung 3.19 zeigt das Verfahren. In Spalte C werden zunächst die Barwerte der bis zum Fälligkeitstermin anfallenden Zahlungen berechnet, in diesem Fall also die der Zinszahlun-

gen und der Rückzahlung bei Fälligkeit. Diese werden dann in Spalte D gewichtet, indem sie einfach mit der Periodenzahl multipliziert werden.

Mit Hilfe dieser Kennzahl wird es möglich, verschiedene Anlagemöglichkeiten miteinander zu vergleichen. Je niedriger die durchschnittliche Bindungsdauer ist, umso risikoärmer ist ein Wertpapier einzuschätzen.

`Abrechnung` ist das Kaufdatum, `Fälligkeit` ist das Datum, zu dem das Wertpapier fällig ist. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit `Nominalzins` wird der Zinssatz angegeben, mit `Rendite` die am Abrechnungstag geltende jährliche Rendite.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Nominalzins` oder `Rendite` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn für `Nominalzins` oder `Rendite` negative Werte angegeben werden oder wenn die Reihenfolge der Termine nicht eingehalten wird: `Abrechnung` < `Fälligkeit`.

In dem Beispiel in Abbildung 3.19 wird die Duration einer festverzinslichen Anleihe mit einer Laufzeit von 10 Jahren errechnet, wobei davon ausgegangen wird, dass der Emissions- und der Abrechnungstermin zusammenfallen. Das Ergebnis von 7,5 Jahren kann dann so interpretiert werden: Verkauft der Anleger die Anleihe nach 7,5 Jahren, kann er die angegebene Rendite von 7% realisieren, unabhängig von späteren Schwankungen des Marktzinses. Dabei wird allerdings vorausgesetzt, dass der Anleger die erhaltenen periodischen Zinszahlungen immer sofort reinvestiert.

Die Börsen geben in den Daten zu einem festverzinslichen Papier, das zum Handel angeboten wird, die Duration üblicherweise als einen der Kennwerte an.

Eine Variante von `DURATION()` ist die Funktion `MDURATION()`, die die modifizierte Macaulay-Dauer liefert.

	A	B	C	D	E	F
1						
2	**Hypothetische Kapitalbindungsdauer**					
3						
4	Abrechnung	01.01.2003				
5	Fälligkeit	01.01.2013				
6	Nominalzins	7%				
7	Rendite	7%				
8	Häufigkeit	1				
9	Basis	0				
10	DURATION()	7,52				
11	MDURATION	7,02				
12						
13	Periode	Zahlungen	Barwerte	gewichtete Zahlungszeitpunkte	7,52	**=D24*1/100**
14	1	7	6,54	6,54	7,02	**=B10/(1+B6)**
15	2	7	6,11	12,23		
16	3	7	5,71	17,14		
17	4	7	5,34	21,36		
18	5	7	4,99	24,95		
19	6	7	4,66	27,99		
20	7	7	4,36	30,51		
21	8	7	4,07	32,59		
22	9	7	3,81	34,27		
23	10	107	54,39	543,93		
24			100,00	751,52		

Abbildung 3.19 Berechnung von `DURATION()` und `MDURATION()`

EFFEKTIV()

EFFECT()

Syntax: `EFFEKTIV(`**`Nominalzins; Perioden`**`)`

Beispiel: `=EFFEKTIV(6%;4)`

Ergebnis: `6,14%`

Die Funktion `EFFEKTIV()` berechnet aus einem nachschüssigen Nominalzinssatz den effektiven Jahreszins für eine Anlage oder ein Darlehen. `Nominalzins` ist der jährliche Nominalzins, `Perioden` ist die Zahl der jährlichen Zinszahlungen. Da bei mehreren jährlichen anteiligen Zinszahlungen ein Teil der Jahreszinsen praktisch vorweg gezahlt wird, erhöht sich der Effektivzins mit der Zahl der jährlichen Zinszahlungen.

	A	B	C
1			
2	**Berechnung der Effektivverzinsung**		
3			
4	Nominalzins	Perioden pro Jahr	EFFEKTIV()
5	4,00%	2	4,04%
6	4,50%	2	4,55%
7	5,00%	2	5,06%
8	5,50%	4	5,61%
9	6,00%	4	6,14%
10	6,50%	4	6,66%
11	7,00%	12	7,23%
12	7,50%	12	7,76%
13	8,00%	12	8,30%

Abbildung 3.20 Effektivzins im Vergleich zum Nominalzins

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Nominalzins` ≤ `0` oder `Perioden` < `1`, ist der Fehler `#ZAHL!`.

Diese Funktion ist die Umkehrfunktion von `NOMINAL()`. Es gilt:

```
Effektivzins = (1 + (Nominalzins / Perioden) * Perioden) -1
```

GDA()

DDB()

Syntax: `GDA(`**`Ansch_Wert; Restwert; Nutzungsdauer; Periode`**`; Faktor)`

Beispiel: `=GDA(20000;2000;8;1;2)`

Ergebnis: `5000`

Die Funktion `GDA()` berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der geometrisch-degressiven Methode, die auch als Mehrfachraten-Abschreibung bezeichnet wird. Bei diesem Verfahren wird der Abschreibungsbetrag durch Multiplikation des nach jeder Abschreibung übrig bleibenden Buchwerts mit einem gleich bleibenden Prozentsatz ermittelt.

Das Argument `Ansch_Wert` ist der Betrag, der für die Investition aufgewandt wurde. `Restwert` ist der Wert, den die Investition am Ende der Abschreibungsperioden noch hat. `Nutzungsdauer` entspricht der Anzahl von Zeiträumen bis zum Ende der Abschreibung. `Periode` ist der Zeitraum, für den die Abschreibung ermittelt werden soll.

Mit `Faktor` wird die Stärke der Degressivität angegeben. Fehlt dieser Wert, wird `2` angesetzt (Doppelratenabschreibung). Mit diesem Wert wird der hypothetische lineare Abschreibungssatz multipliziert. Beträgt die Laufzeit beispielsweise 8 Jahre, wäre der lineare Satz 12,50%, multipliziert mit einem Faktor `2` ergeben sich 25%. (Der angegebene Restwert wird dabei nicht berücksichtigt.)

Alle Argumente benötigen positive numerische Werte. Nicht numerische Werte führen zu dem Fehler `#WERT!`, negative Werte zu dem Fehler `#ZAHL!`.

	A	B	C	D	E	F	G	H	I
1									
2	**Geometrisch-degressive Abschreibung**								
3									
4	Anschaffungskosten	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €
5	Restwert	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €
6	Nutzungsdauer in Jahren	8	8	8	8	8	8	8	8
7	Jahr	1	2	3	4	5	6	7	8
8	Faktor	2	2	2	2	2	2	2	2
9	GDA()	5.000 €	3.750 €	2.813 €	2.109 €	1.582 €	1.187 €	890 €	667 €

Abbildung 3.21 Verlauf der geometrisch-degressiven Abschreibung

Da diese Form der Abschreibung unter Umständen dazu führt, dass am Ende der Perioden nicht der komplette Wert abgeschrieben ist, wird für die letzte(n) Periode(n) häufig ein Wechsel zur linearen Abschreibungsmethode vorgenommen.

Nach dem augenblicklichen deutschen Steuerrecht ist die geometrisch-degressive Abschreibung nur noch für Güter, die zwischen dem 1.1.2008 und dem 31.12.2009 erworben wurden, erlaubt, allerdings darf der Abschreibungssatz höchstens das 2½-fache des linearen Abschreibungssatzes betragen und 25% nicht übersteigen. Der Übergang zur linearen Abschreibung ist zulässig. Diese Bedingung ist in dem Beispiel in Abbildung 3.21 erfüllt.

GDA2()

DB()

Syntax: `GDA2(`**`Ansch_Wert; Restwert; Nutzungsdauer; Periode;`**`Monate)`

Beispiel: `=GDA2(20000;2000;8;1;6)`

Ergebnis: `2500`

Die Funktion GDA2() berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der geometrisch-degressiven Methode. Sie gleicht der Funktion GDA(), erlaubt es aber zusätzlich, für die erste Periode gebrochene Jahresanteile zu berücksichtigen.

Das Argument Ansch_Wert ist der Betrag, der für die Investition aufgewandt wurde. Restwert ist der Wert, den die Investition nach der Abschreibung noch hat. Nutzungsdauer entspricht der Anzahl von Zeiträumen bis zum Ende der Abschreibung. Periode ist der Zeitraum, für den der Abschreibungsbetrag ermittelt werden soll. Mit dem optionalen Argument Monate kann für das erste Jahr die Anzahl der Monate angegeben werden. Wird es weggelassen, wird der Wert 12 angenommen.

Alle Argumente benötigen positive numerische Werte. Nicht numerische Werte führen zu dem Fehler #WERT!, negative Werte zu dem Fehler #ZAHL!.

Diese Funktion arbeitet anders als GDA() mit einem nicht veränderbaren Abschreibungssatz, der aus der Anzahl der Perioden errechnet wird:

```
Abschreibungsrate = 1 - ((Restwert/Anschaffungswert) ^ (1/Nutzungsdauer))
```

Aus der Formel lässt sich ableiten, warum der Restwert nicht mit 0 angegeben werden darf. Sie müssen also wenigstens einen Minimalwert von 1 € angeben.

Da diese Form der Abschreibung unter Umständen dazu führt, dass am Ende der Perioden nicht der komplette Wert abgeschrieben ist, wird für die letzte(n) Periode(n) häufig ein Wechsel zur linearen Abschreibungsmethode vorgenommen.

	A	B	C	D	E	F	G	H	I
1									
2	**Geometrisch-Degressive Abschreibung II**								
3									
4	Anschaffungskosten	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €
5	Restwert	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €
6	Nutzungsdauer in Jahren	8	8	8	8	8	8	8	8
7	Jahr	1	2	3	4	5	6	7	8
8	Monate	6	6	6	6	6	6	6	6
9	GDA2()	2.500 €	4.375 €	3.281 €	2.461 €	1.846 €	1.384 €	1.038 €	779 €

Abbildung 3.22 Berechnung der geometrisch-degressiven Abschreibung für einen Zeitraum

Zur Zulässigkeit dieser Methode siehe den Hinweis bei GDA().

IKV()

IRR()

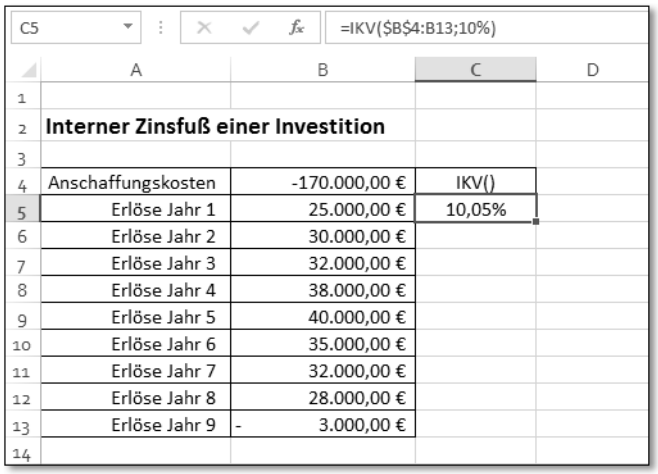
Syntax: `IKV(`**`Werte`**`; Schätzwert)`

Beispiel: siehe Abbildung 3.23

Die Funktion `IKV()` berechnet den internen Ertragszins einer Investition. Das ist der Zinssatz, bei dem die Summe der Barwerte aller Aus- und Einzahlungen den Wert 0 ergibt. Mit dieser Kennzahl kann die Rentabilität einer Investition geprüft werden. Dabei wird der interne Zinssatz mit einem kalkulatorischen Zinssatz verglichen, also mit Zinsen, die bei einer Anlage auf dem Kapitalmarkt erzielt werden können. Eine Investition kann als vorteilhaft eingeschätzt werden, wenn der errechnete interne Ertragszins den Kalkulationszinssatz übersteigt.

Das Argument `Werte` ist eine Matrix oder ein Bereich, der mindestens einen positiven und einen negativen Wert enthält, andernfalls liefert die Funktion den Fehler `#ZAHL!`.

Der erste Wert stellt die getätigte Investition in der Regel als Ausgabe dar, erscheint also negativ, die weiteren Zellen stellen die periodischen Rückflüsse dar, also die Überschüsse aus Aus- und Einzahlungen. Wenn innerhalb des angegebenen Zellbereichs oder der Matrix nicht numerische Werte, Wahrheitswerte oder leere Zellen vorkommen, werden sie bei der Berechnung ignoriert.



	A	B	C	D
1				
2	**Interner Zinsfuß einer Investition**			
3				
4	Anschaffungskosten	-170.000,00 €	IKV()	
5	Erlöse Jahr 1	25.000,00 €	10,05%	
6	Erlöse Jahr 2	30.000,00 €		
7	Erlöse Jahr 3	32.000,00 €		
8	Erlöse Jahr 4	38.000,00 €		
9	Erlöse Jahr 5	40.000,00 €		
10	Erlöse Jahr 6	35.000,00 €		
11	Erlöse Jahr 7	32.000,00 €		
12	Erlöse Jahr 8	28.000,00 €		
13	Erlöse Jahr 9	- 3.000,00 €		
14				

Abbildung 3.23 Berechnen des internen Zinssatzes

Die angegebenen Werte müssen fortlaufend in einer Spalte oder Zeile angeordnet sein, wobei jede Zelle den Wert für eine Periode angibt. Es wird also unterstellt, dass die Zah-

lungen in der entsprechenden Reihenfolge erfolgen. Die Beträge können unterschiedlich groß sein, es wird aber davon ausgegangen, dass die Verzinsung in regelmäßigen Abständen nachschüssig erfolgt. Dies können Jahre oder auch Monate sein.

Das optionale Argument Schätzwert kann mit einem Wert belegt werden, der in einer Schätzung dem erwarteten Ergebnis schon möglichst nahekommt. Wird er nicht eingetragen, so wird sein Wert von Excel mit 10% angenommen.

Bei der Berechnung setzt die Funktion – ähnlich wie die Funktion ZINS() – ein Iterationsverfahren ein, das ausgehend von dem angegebenen oder vorgegebenen Schätzwert so oft durchlaufen wird, bis das Ergebnis auf 0,00001% genau ist.

Falls allerdings nach 20 Berechnungsrunden immer noch kein brauchbares Ergebnis gefunden ist, liefert die Funktion den Fehlerwert #ZAHL!.

Da es sich um eine Näherungsrechnung handelt, kann es auch vorkommen, dass die Funktion unbrauchbare Ergebnisse wie etwa einen negativen Zinssatz liefert.

Beachten Sie, dass die Funktion IKV() nur für periodische Zahlungsvorgänge brauchbar ist. Wollen Sie auch unregelmäßige Zahlungsvorgänge berücksichtigen, können Sie stattdessen mit der Funktion XINTZINSFUSS() arbeiten.

ISPMT()

ISPMT()

Syntax: ISPMT(**Rate; Per; Nper; Pv**)
Beispiel: =ISPMT(0,07;4;12;-10000)
Ergebnis: 466,67

Die Funktion ISPMT() berechnet die Zinsen, die auf einen bestimmten Teil eines Jahres entfallen, wobei eine einfache nachschüssige Verzinsung vorausgesetzt wird.

Mit Rate wird der nachschüssige Jahreszinssatz angegeben. Per gibt die Periode an, ab deren Ende die Zinsen bis zum Jahresende berechnet werden sollen; Nper ist die Anzahl der Zahlungsperioden im Jahr. Per und Nper müssen die gleichen Perioden verwenden. Pv ist der Gegenwartswert, also der eingezahlte oder geliehene Betrag. Dieser Wert muss mit einem Minuszeichen angegeben werden.

Ist einer der Werte nicht numerisch, liefert die Funktion den Fehler #WERT!.

Das Beispiel in Abbildung 3.24 entspricht in der Spalte B dem Fall, dass eine Summe von 10.000 € am Ende des 4. Monats auf ein Sparbuch eingezahlt wird, das 5% Zinsen ab-

wirft. Die Funktion errechnet die Zinsen für die restlichen 8 Monate des Jahres. In der Spalte C werden dieselben Zeiträume in Tagen angegeben. Die Funktion wird aus Gründen der Kompatibilität mit *Lotus 1-2-3* bereitgestellt, daher die anderen Namen für die Argumente.

	A	B	C
1			
2	**Berechnung der unterjährigen Zinsen**		
3			
4	Zins	5%	4%
5	Zr	4	120
6	Zzr	12	360
7	BW	- 10.000,00 €	- 10.000,00 €
8	ISPMT()	333,33 €	266,67 €

Abbildung 3.24 Summe der Zinsen für einen bestimmten Zeitraum innerhalb eines Jahres

KAPZ()

PPMT()

Syntax: `KAPZ(`**`Zins; Zr; Zzr; Bw`**`; Zw; F)`

Beispiel: `=KAPZ(0,1;1;10;20000)`

Ergebnis: `-1254,91`

Die Funktion `KAPZ()` berechnet den Tilgungsanteil für die Abzahlung eines Annuitätendarlehens. Ein solches Darlehen wird in periodischen, gleich bleibenden Raten abbezahlt, deren Höhe mit `RMZ()` berechnet wird. Die Raten bestehen jeweils aus einem Zinsanteil (berechnet mit `ZINSZ()`) und einem Tilgungsanteil, um den sich die Gesamtschuld jeweils verringert (siehe Abbildung 3.25).

	A	B	C	D	E	F	G	H	I	J	K
1											
2	**Tilgung von Darlehen**										
3											
4	Zins	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%
5	Zr	1	2	3	4	5	6	7	8	9	10
6	Zzr	10	10	10	10	10	10	10	10	10	10
7	Bw	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €
8	KAPZ()	-1.447,55 €	-1.548,88 €	-1.657,30 €	-1.773,31 €	-1.897,44 €	-2.030,26 €	-2.172,38 €	-2.324,45 €	-2.487,16 €	-2.661,26 €
9	ZINSZ()	-1.400,00 €	-1.298,67 €	-1.190,25 €	-1.074,24 €	-950,11 €	-817,29 €	-675,17 €	-523,10 €	-360,39 €	-186,29 €
10	RMZ()	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €	-2.847,55 €
11											
12	Jährliche Zahlungen (RMZ) = Tilgung (KAPZ) + Zinsen (ZINSZ)										

Abbildung 3.25 Verlauf der Tilgung eines Darlehens

Argumente sind Zins für den konstanten Nominalzinssatz, Zr für den Zeitraum, für den die Zahlung berechnet werden soll, Zzr für die Anzahl der Abzahlungszeiträume und Bw, also der Barwert, der bei einem Darlehen der Darlehenssumme entspricht. Dazu können Sie noch Zw angeben für einen Zukunftswert, also einen Betrag, der am Ende der letzten Zahlung noch an der vollständigen Rückzahlung fehlt. Ist dieser Wert nicht angegeben, wird er als 0 angenommen.

Der Wert F ist optional und gibt die Fälligkeit der Zahlungen an: 0 oder nicht angegeben = nachschüssig, 1 = vorschüssig).

Alle Werte müssen numerisch sein, sonst liefert die Funktion den Fehler #WERT!. Negative Werte für Zr und Zzr führen zu dem Fehler #ZAHL!.

KUMKAPITAL()

CUMPRINC()

Syntax: KUMKAPITAL(**Zins; Zzr; Bw; Zeitraum_Anfang; Zeitraum_Ende; F**)
Beispiel: =KUMKAPITAL(6%;10;30000;1;5;0)
Ergebnis: -12830,24

Die Funktion KUMKAPITAL() berechnet den Betrag, der für die Tilgung eines Annuitätendarlehens in einem bestimmten Zeitraum aufgebracht wird (nicht zu verwechseln mit den Zahlungsbeträgen). Die Funktion kann beispielsweise verwendet werden, um die verbleibende Restschuld zu berechnen.

Mit Zins wird der Zinssatz pro Zinszeitraum angegeben; mit Zzr die Anzahl der Zahlungsperioden, mit Bw die Darlehenssumme. Zeitraum_Anfang und Zeitraum_Ende sind die Nummern der beiden Zahlungsperioden, zwischen denen die Gesamttilgung berechnet werden soll. Mit F muss angegeben werden, ob die Zahlungen vorschüssig (1) oder nachschüssig (0) erfolgen sollen. Das Argument ist in diesem Fall also ausnahmsweise nicht optional.

Für Zins, Zzr und Bw werden positive Werte > 0 erwartet, andernfalls liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn Zeitraum_Anfang und Zeitraum_Ende < 1 oder Zeitraum_Anfang > Zeitraum_Ende sind. Nicht numerische Werte liefern den Fehler #WERT!.

	A	B	C	D	E	F	G
1							
2	**Berechnung der aufgelaufenen Tilgung eines Darlehens**						
3							
4	Zins	6%	6%	6%	6%	6%	6%
5	Zzr	10	10	10	10	10	10
6	Bw	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €
7	Zeitraum_Anfang	1	1	1	1	1	1
8	Zeitraum_Ende	5	6	7	8	9	10
9	F	0	0	0	0	0	0
10	KUMKAPITAL()	-12.830,24 €	-15.876,10 €	-19.104,70 €	-22.527,02 €	-26.154,68 €	-30.000,00 €

Abbildung 3.26 Berechnung des Tilgungsanteils zwischen zwei Zeitpunkten

Beachten Sie, dass für `Zins` und `Zzr` zueinander passende Zeiteinheiten verwendet werden müssen. Werden die Zahlungsperioden nicht in Jahren, sondern in Monaten angegeben, muss also der Jahreszinssatz mit 12 dividiert und `Zzr` mit 12 multipliziert werden.

Während mit `KUMKAPITAL()` der Tilgungsanteil berechnet wird, liefert die folgende Funktion `KUMZINSZ()` den Zinsanteil für den betreffenden Zeitraum. Beides zusammen ist der Betrag, der für den betreffenden Zeitraum insgesamt aufgebracht werden muss. Analog verhalten sich `KAPZ()` und `ZINSZ()`, zusammen ergeben sie `RMZ()`.

KUMZINSZ()

CUMIPMT()

Syntax: `KUMZINSZ(`**`Zins; Zzr; Bw; Zeitraum_Anfang; Zeitraum_Ende; F`**`)`

Beispiel: `=KUMZINSZ(6%;10;30000;1;5;0)`

Ergebnis: `-7549,95`

Die Funktion `KUMZINSZ()` berechnet den Betrag, der für die Zinsen eines Annuitätendarlehens in einem bestimmten Zeitraum aufgebracht wird (nicht zu verwechseln mit den Zahlungsbeträgen).

Mit `Zins` wird der Zinssatz pro Zinszeitraum angegeben, mit `Zzr` die Anzahl der Zahlungsperioden, mit `Bw` die Darlehenssumme. `Zeitraum_Anfang` und `Zeitraum_Ende` sind die Nummern der beiden Zahlungsperioden, zwischen denen die aufgelaufenen Zinsen berechnet werden sollen. Mit `F` muss angegeben werden, ob die Zahlungen vorschüssig (`1`) oder nachschüssig (`0`) erfolgen sollen. Das Argument ist in diesem Fall also ausnahmsweise nicht optional.

Für Zins, Zzr und Bw werden positive Werte > 0 erwartet, andernfalls liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn Zeitraum_Anfang und Zeitraum_Ende < 1 oder Zeitraum_Anfang > Zeitraum_Ende sind. Nicht numerische Werte liefern den Fehler #WERT!.

	A	B	C	D	E	F	G
1							
2	**Berechnung der aufgelaufenen Zinsen eines Darlehens**						
3							
4	Zins	6%	6%	6%	6%	6%	6%
5	Zzr	10	10	10	10	10	10
6	Bw	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €	30.000,00 €
7	Zeitraum_Anfang	1	1	1	1	1	1
8	Zeitraum_Ende	5	6	7	8	9	10
9	F	0	0	0	0	0	0
10	KUMZINSZ()	- 7.549,95 €	- 8.580,14 €	- 9.427,57 €	-10.081,29 €	-10.529,67 €	-10.760,39 €

Abbildung 3.27 Aufgelaufene Zinsen zwischen zwei Zeitpunkten

Während KUMZINSZ() den Zinsanteil für den betreffenden Zeitraum berechnet, liefert KUMKAPITAL() den Tilgungsanteil. Beides zusammen ist der Betrag, der für den betreffenden Zeitraum insgesamt aufgebracht werden muss. Analog verhalten sich KAPZ() und ZINSZ(), zusammen ergeben sie RMZ().

KURS()

PRICE()

Syntax:	KURS(**Abrechnung; Fälligkeit; Zins; Rendite; Rückzahlung; Häufigkeit**; Basis)
Beispiel:	=KURS(DATUM(2010;1;3); DATUM(2012;2;10);0,06;0,05;100;1;4)
Ergebnis:	101,94

Die Funktion KURS() berechnet den Kurswert eines festverzinslichen Wertpapiers mit dem Nennwert 100 € abhängig von einer anvisierten Rendite. Sie gibt also Antwort auf die Frage: Welchen Kurs müsste ein Papier haben, um eine bestimmte Rendite zu erreichen?

Abrechnung ist das Datum des Besitzwechsels; Fälligkeit ist das Fälligkeitsdatum des Wertpapiers. Diese beiden Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein

solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit `Zins` wird der Nominalzins angegeben, mit `Rendite` die jährliche Rendite, die das Papier bringen soll. `Rückzahlung` ist der Betrag, zu dem das Wertpapier zum Fälligkeitstermin ausbezahlt wird (bei Anleihen im Allgemeinen `100`). Alle drei Argumente müssen positive Werte angeben, sonst liefert die Funktion den Fehler `#ZAHL!`.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Zins`, `Rendite` oder `Rückzahlung` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` oder `Abrechnung` > `Fälligkeit` ergeben den Fehler `#ZAHL!`.

Mit fast denselben Argumenten lässt sich, wenn der Kurs bekannt ist, mit Hilfe der Funktion `RENDITE()` die hier vorgegebene Rendite errechnen. Siehe `RENDITE()`.

	A	B
1		
2	**Kurswert bei festverzinslichen Papieren**	
3		
4	Abrechnung	03.01.2010
5	Fälligkeit	10.02.2012
6	Zins	6%
7	Rendite	5%
8	Rückzahlung	100
9	Häufigkeit	1
10	Basis	4
11	KURS()	101,94 €

Abbildung 3.28 Kurswert von festverzinslichen Papieren

KURSDISAGIO()

PRICEDISC()

Syntax: `KURSDISAGIO(`**`Abrechnung`**`; `**`Fälligkeit`**`; `**`Disagio`**`; `**`Rückzahlung`**`; Basis)`

Beispiel: `=KURSDISAGIO(DATUM(2012;1;3); DATUM(2014;2;10);3%;100;4)`

Ergebnis: `93,69`

Die Funktion KURSDISAGIO() berechnet den Ausgabekurs eines unverzinslichen Wertpapiers. Mit dem etwas irreführenden Adjektiv »unverzinslich« wird gesagt, dass es bei diesem Papier keine periodischen Zinszahlungen gibt. Stattdessen wird beim Kauf ein Abschlag – Disagio – gewährt, d.h., der Zinsbetrag wird vorweg abgezogen.

Abrechnung ist das Datum des Besitzwechsels, Fälligkeit das Fälligkeitsdatum des Wertpapiers. Diese beiden Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Disagio legt als Prozentsatz fest, mit welchem Abschlag (aufs Jahr gerechnet) das Papier verkauft werden soll. Rückzahlung ist der Betrag, zu dem das Wertpapier zum Fälligkeitstermin ausbezahlt wird (bei Anleihen im allgemeinen 100). Beide Argumente müssen positive Werte angeben, sonst liefert die Funktion den Fehler #ZAHL!. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für Disagio oder Rückzahlung führen zu dem Fehler #WERT!, nicht erlaubte Werte für Basis oder Abrechnung > Fälligkeit ergeben den Fehler #ZAHL!.

Die Funktion lässt sich beispielsweise anwenden, um den Kaufpreis von sogenannten Zerobonds zu berechnen. Abbildung 3.29 zeigt ein Beispiel.

	A	B
1		
2	**Kurswert eines unverzinslichen Wertpapiers**	
3		
4	Abrechnung	03.01.2012
5	Fälligkeit	10.02.2014
6	Disagio	3%
7	Rückzahlung	100,00
8	Basis	4
9	KURSDISAGIO()	93,69

Abbildung 3.29 Berechnen des Kurses eines Zerobonds

KURSFÄLLIG()

PRICEMAT()

Syntax: KURSFÄLLIG(**Abrechnung**; **Fälligkeit**; **Emission**; **Zins**; **Rendite**; Basis)

Beispiel: =KURSFÄLLIG(DATUM(2013;3;1); DATUM(2014;1;1); DATUM(2013;1;1);4%;3%;4)

Ergebnis: 100,80

Die Funktion KURSFÄLLIG() liefert den Kurs eines Wertpapiers bezogen auf einen Nennwert von 100 Einheiten, bei dem die Zinsen erst zum Fälligkeitsdatum, also mit der Rückzahlung, ausgezahlt werden. Dabei handelt es sich um eine einfache nachschüssige Verzinsung ohne Zinseszinseffekt.

Abrechnung ist das Kaufdatum, für den der Kurswert errechnet werden soll, Fälligkeit das Datum, zu dem das Wertpapier zurückgenommen wird. Emission ist das Ausgabedatum des Papiers, ab dem die Zinsen berechnet werden. Alle drei Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Zins ist der jährliche Nominalzinssatz. Mit Rendite wird angegeben, wie viel das Wertpapier jährlich einbringen soll (in Prozent vom Kaufpreis). Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für Zins oder Rendite führen zu dem Fehler #WERT!, nicht erlaubte Werte für Basis ergeben den Fehler #ZAHL!. Das gilt auch, wenn die Werte Emission, Abrechnung und Fälligkeit nicht aufeinanderfolgen.

	A	B
1		
2	**Kurs bei nachschüssiger Verzinsung**	
3		
4	Abrechnung	01.03.2013
5	Fälligkeit	01.01.2014
6	Emission	01.01.2013
7	Zins	4%
8	Rendite	3%
9	Basis	4
10	KURSFÄLLIG()	100,80

Abbildung 3.30 Berechnung des Kurses bei nachschüssiger Verzinsung

LIA()

SLN()

Syntax: `LIA(Ansch_Wert; Restwert; Nutzungsdauer)`
Beispiel: `=LIA(22000;2000;5)`
Ergebnis: `4000`

Die Funktion `LIA()` berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der linearen Methode. Dies ist eine der beiden steuerrechtlich möglichen Methoden der »Absetzung für Abnutzung« – kurz *AFA*. Bei dieser Methode wird über den gesamten Abschreibungszeitraum mit periodisch gleich bleibenden Beträgen abgeschrieben.

Das Argument `Ansch_Wert` nennt die Anschaffungskosten, die für die Investition aufgewandt wurden. `Restwert` ist der Wert, den die Investition nach der Abschreibung noch hat. `Nutzungsdauer` entspricht der Anzahl von Zeiträumen bis zum Ende der Abschreibung. Der jährliche Abschreibungsbetrag wird ermittelt, indem die Differenz zwischen Anschaffungskosten und Restwert durch den Wert für Nutzungsdauer geteilt wird. Dieser Betrag wird jeweils vom Buchwert der Vorperiode abgezogen.

Abbildung 3.31 zeigt den gleichmäßigen Verlauf der Abschreibung über die gesamte Nutzungsdauer.

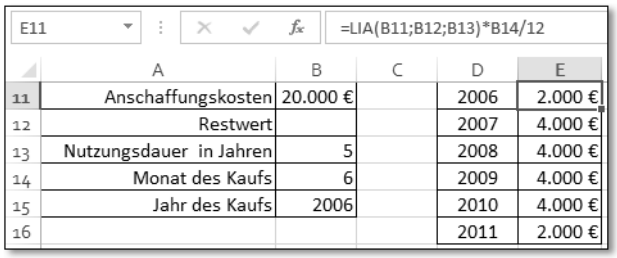
	A	B	C	D	E	F
1						
2	**Lineare Abschreibung**					
3						
4	Anschaffungskosten	22.000 €	22.000 €	22.000 €	22.000 €	22.000 €
5	Restwert	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €
6	Nutzungsdauer in Jahren	5	5	5	5	5
7	Jahr	1	2	3	4	5
8	LIA()	4.000 €	4.000 €	4.000 €	4.000 €	4.000 €
9					Gesamt:	20.000 €

Abbildung 3.31 Verlauf der linearen Abschreibung

Das Beispiel in Abbildung 3.31 geht allerdings davon aus, dass die Nutzungsdauer nur ganze Jahre umfasst. In der Praxis muss aber berücksichtigt werden, dass Anschaffungen ab dem Monat abgeschrieben werden müssen, in dem sie getätigt worden sind, wobei es keine Rolle spielt, um welchen Tag des Monats es sich handelt. Seit 2004 gilt die früher übliche Halbjahresregel nicht mehr.

Um einen entsprechenden Abschreibungsplan zu erstellen, kann für das erste Jahr mit einer Formel gearbeitet werden, die den jährlichen Abschreibungsbetrag mit dem Faktor Anzahl Monate / 12 multipliziert. Die folgenden Jahre werden wie oben mit Hilfe von `LIA()` berechnet. Für den am Ende übrig bleibenden Rest wird dann eine zusätzliche Periode angehängt.

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`.



	A	B	C	D	E
11	Anschaffungskosten	20.000 €		2006	2.000 €
12	Restwert			2007	4.000 €
13	Nutzungsdauer in Jahren	5		2008	4.000 €
14	Monat des Kaufs	6		2009	4.000 €
15	Jahr des Kaufs	2006		2010	4.000 €
16				2011	2.000 €

Abbildung 3.32 Abschreibungsplan bei Kauf innerhalb des Jahres

MDURATION()

MDURATION()

Syntax: `MDURATION(`**`Abrechnung; Fälligkeit; Coupon; Rendite; Häufigkeit;`** `Basis)`

Beispiel: `=MDURATION(DATUM(2013;6;19); DATUM(2018;8;27);0,04;0,0349;1;4)`

Ergebnis: `4,49`

Die Funktion `MDURATION()` liefert die als modifizierte Duration bezeichnete Kennzahl für festverzinsliche Wertpapiere. Sie gibt an, wie sehr sich der Anleihekurs prozentual ändert, wenn das Marktzinsniveau um 1% angehoben wird. Dadurch kann geprüft werden, wie stark sich der Gesamtertrag einer Anleihe unter Umständen ändern kann, wenn sich der Zinssatz am Markt ändert.

Wird das Ergebnis der Funktion beispielsweise mit einer erwarteten Marktzinsänderung von 0,5% multipliziert, ergibt sich ein Prozentsatz für eine relative Kursänderung gegenüber dem bisherigen Kurs. Anders als bei Aktien sind bei Anleihen, je näher der Abrechnungstermin dem Fälligkeitstermin kommt, umso geringere Kursschwankungen zu erwarten. Ein geringerer Wert für `MDURATION()` bedeutet also, dass auch geringere Kursgewinne bzw. -verluste bei fallenden bzw. steigenden Marktzinsen zu erwarten sind.

`Abrechnung` ist das Kaufdatum, `Fälligkeit` ist das Datum, zu dem das Wertpapier fällig ist. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist. Mit `Coupon` wird der Zinssatz angegeben, mit `Rendite` die am Abrechnungstag geltende jährliche Rendite.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Coupon` oder `Rendite` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn für `Coupon` oder `Rendite` negative Werte angegeben werden oder wenn die Reihenfolge der Termine nicht eingehalten wird: `Abrechnung` < `Fälligkeit`.

Die Funktion steht zu `DURATION()` in folgender Beziehung:

```
MDURATION = DURATION / 1 + (Rendite / Jährliche Kuponzahlungen)
```

	A	B
1		
2	**Modifizierte Duration**	
3		
4	Abrechnung	19.06.2013
5	Fälligkeit	27.08.2018
6	Nominalzins	4%
7	Rendite	3,49428%
8	Häufigkeit	1
9	Basis	4
10	MDURATION	4,49
11		

Abbildung 3.33 Berechnung der modifizierten Duration für ein festverzinsliches Papier

NBW()

NPV()

Syntax: `NBW(`**`Zins; Wert1`**`; Wert2; ...)`

Beispiel: `=NBW(0,1;-20000;6000;8000;12000)`

Ergebnis: `984`

Die Funktion NBW() berechnet den Nettobarwert von zukünftigen Periodenüberschüssen (*Cashflows*), die aufgrund einer Investition erwartet werden. Diese Berechnung gehört zu den Methoden der dynamischen Investitionsrechnung, mit deren Hilfe geprüft werden kann, ob sich eine Investition lohnt oder nicht. Nur Investitionen mit einem positiven NBW() können als lohnend eingestuft werden, bei einem negativen Ergebnis wäre es besser, das Geld zum angegebenen Zinssatz auf die Bank zu tragen.

Mit dem Argument Zins muss zunächst ein konstanter Kalkulationszinssatz für den gesamten betrachteten Zeitraum vorgegeben werden. Der hier sinnvolle Wert ist eine brauchbare Vergleichsrendite, die sich aus der Überlegung ergibt, wie viel bei der gegebenen Marktsituation durch eine andere Anlage des investierten Kapitals erreicht werden könnte.

Wert1, Wert2, ... sind dann die erwarteten Zahlungsvorgänge (Einzahlung positiv, Auszahlung negativ), die jeweils am Ende einer Periode erfolgen. Die Werte werden in der Reihenfolge verarbeitet, in der sie in einer Spalte oder Zeile angeordnet sind. Es muss also darauf geachtet werden, dass die zeitliche Abfolge dabei eingehalten wird.

Nicht numerische Argumente oder Bezüge auf leere Zellen werden so behandelt, als ob sie nicht existieren. Der Wert #NV führt allerdings auch zu dem Funktionsergebnis #NV. Gleiches gilt für andere Fehlerwerte.

Anders als bei der BW()-Funktion erlaubt NBW() unterschiedliche Zahlungsbeträge, die aber immer am Ende der Periode anfallen.

Abbildung 3.34 zeigt ein Beispiel mit einer Anschaffung in Höhe von 20.000 € am Anfang und unterschiedlichen Überschüssen in den folgenden drei Jahren. NBW() ermittelt einen positiven Wert, d.h., dass die Investition auf jeden Fall die anvisierten 10% übertreffen wird. Das Ergebnis in C9 dagegen ist negativ. Eine interne Zinsrate von 15% ist also so nicht zu erreichen.

	A	B	C	D
1				
2	**Berechnen des Kapitalwerts einer Investition**			
3				
4	Jährlicher Abzinsungsfaktor	10%	15%	
5	Anfangskosten	-20000	-20000	
6	Überschuss Jahr 1	6000	6000	
7	Überschuss Jahr 2	8000	8000	
8	Überschuss Jahr 3	12000	12000	
9	NBW()	984	- 733	
10				

Abbildung 3.34 Nettobarwert zu einem Cashflow

NOMINAL()

NOMINAL()

Syntax: `NOMINAL(Effektiver_Zins; Perioden)`
Beispiel: `=NOMINAL(6,3%;4)`
Ergebnis: `6,16%`

Die Funktion `NOMINAL()` berechnet den Nominalzins (Jahreszins auf den Nennwert). Diese Funktion ist die Umkehrfunktion von `EFFEKTIV()`. Es gilt:

```
Effektivzins = (1 + (Nominalzins / Perioden)) ^ Perioden) -1
```

Mit `Effektiver_Zins` wird der tatsächlich erzielte Jahreszins angegeben, der sich aus der Berücksichtigung von unterjährigen Zinseszinseffekten ergibt; `Perioden` gibt die Zahl der im Jahr vorkommenden Zinszahlungen an. Bei einer jährlichen Zahlung sind Nominalzins und Effektivzins identisch.

	A	B	C
1			
2	**Nominalzins aus Effektivzins berechnen**		
3			
4	Effektiver_Zins	Perioden im Jahr	NOMINAL()
5	8%	1	8,00%
6	8%	2	7,85%
7	8%	4	7,77%
8	8%	12	7,72%

Abbildung 3.35 Rückrechnen auf den Nominalzins

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Effektiver_Zins` ≤ 0 oder `Perioden` < 1, ist der Fehler `#ZAHL!`.

NOTIERUNGBRU()

DOLLARFR()

Syntax: `NOTIERUNGBRU(Zahl; Teiler)`
Beispiel: `=NOTIERUNGBRU(4,125;8)`
Ergebnis: `4,1`

Die Funktion `NOTIERUNGBRU()` liefert eine als Bruch interpretierbare Darstellung einer Dezimalzahl. `Zahl` ist die dezimale Zahl, die als Bruchzahl dargestellt werden soll.

`Teiler` gibt den Nenner an, der für die Bruchzahl verwendet werden soll. Er muss eine ganze Zahl > `0` sein, Dezimalstellen werden abgeschnitten.

Beide Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`. Ist `Teiler` negativ, erscheint der Fehler `#ZAHL!`.

Abbildung 3.36 zeigt einige Beispiele. Das Ergebnis der Funktion in B6 ist so zu lesen: Der Dezimalzahl 4,125 entspricht die Bruchzahl 4 und 1/8.

Beachten Sie, dass das Ergebnis nur eine Lesehilfe ist und kein Wert, mit dem Sie weiterrechnen können. Da in den USA bis 2001 Aktiennotierungen mit Hilfe natürlicher Brüche vorgenommen wurden, können mit der Funktion erstellte Tabellen beim Lesen älterer Notierungen helfen.

	A	B	C	D
1				
2	**Darstellung von Dezimalzahlen als Bruchzahlen**			
3				
4		Teiler		
5	Zahl	8	4	2
6	4,125	4,1	4,05	4,025
7	4,250	4,2	4,1	4,05
8	4,375	4,3	4,15	4,075
9	4,500	4,4	4,2	4,1
10	4,625	4,5	4,25	4,125
11	4,750	4,6	4,3	4,15
12	4,875	4,7	4,35	4,175

Abbildung 3.36 Umwandlung von Dezimalzahlen in Bruchzahlen

NOTIERUNGDEZ()

DOLLARDE()

Syntax: `NOTIERUNGDEZ(`**`Zahl; Teiler`**`)`

Beispiel: `=NOTIERUNGDEZ(2,1;8)`

Ergebnis: `2,125`

Die Funktion `NOTIERUNGDEZ()` konvertiert einen als Bruch interpretierten Ausdruck `Zahl` in eine Dezimalzahl. `Zahl` ist in diesem Fall ein Wert, dessen Dezimalstellen als Zähler eines Bruchs zu interpretieren sind, also ein Wert, wie ihn die oben beschriebene Funktion `NOTIERUNGBRU()` als Ergebnis liefert. `NOTIERUNGDEZ()` ist die Umkehrung von `NOTIERUNGBRU()`.

`Teiler` gibt den Nenner an, für den der Dezimalteil von `Zahl` als Zähler zu verstehen ist. Er muss eine ganze Zahl > `0` sein, Dezimalstellen werden abgeschnitten.

Beide Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`. Ist `Teiler` negativ, erscheint der Fehler `#ZAHL!`.

Abbildung 3.37 zeigt einige Beispiele. Das Ergebnis der Funktion in B8 ist so zu lesen: Der Bruchzahl 2,1, die für 2 und 1/8 steht, entspricht die Dezimalzahl 2,125. Anders als bei der Funktion `NOTIERUNGBRU()` kann mit dem Ergebnis der Funktion `NOTIERUNGDEZ()` weitergerechnet werden, weil sie ja wieder eine korrekte Dezimalzahl liefert.

Siehe auch `NOTIERUNGSBRU()`.

	A	B	C	D	E
1					
2	**Darstellung von Bruchzahlnotierungen als Dezimalzahl**				
3					
4	Teiler	2,1	2,2	2,01	2,02
5	2	2,500	3,000	2,0500	2,1000
6	4	2,250	2,500	2,0250	2,0500
7	8	2,125	2,250	2,0125	2,0250

Abbildung 3.37 Umwandlung von Bruchzahlnotierungen in Dezimalzahlen

PDURATION()

PDURATION()

Syntax: `PDURATION(`**`Zins; Bw; Zw`**`)`

Beispiel: `=PDURATION(5%;10000;13000)`

Ergebnis: `5,3774`

Mit der in Excel 2013 neuen Funktion `PDURATION()` wird die Anzahl der Perioden berechnet, bis eine Investition den angegebenen Zielwert erreicht hat. Dabei wird angenommen, dass nach der ersten Zahlung keine Zuzahlungen oder Entnahmen erfolgen.

Die Funktion verlangt drei Argumente. Das Argument `Zins` gibt den Zinssatz pro Zahlungsperiode an. Mit `Bw` wird der aktuelle Wert der Investition angegeben. `Zw` gibt den gewünschten zukünftigen Wert der Investition an. Dabei wird folgende Gleichung verwendet:

```
PDURATION = (LOG(Zw)-LOG(Bw))/LOG(1+Zins)
```

Alle Argumente erfordern, anders als bei verwandten Funktionen wie BW() oder ZW(), positive numerische Werte. Ist das nicht der Fall, liefert die Funktion den Fehlerwert #ZAHL!. Ist ein Argument nicht numerisch, ist der Fehlerwert #WERT!.

Soll das Ergebnis in Monatsperioden geliefert werden, wird der Wert für Zins durch 12 geteilt.

	A	B	C
1			
2	**PDURATION**		
3			
4	Zins	4%	4,50%
5	Barwert	100000	100000
6	Zukunftswert	120000	120000
7	PDURATION()	4,65	4,14

Abbildung 3.38 Beispiele für die Funktion PDURATION()

Die Funktion liefert das gleiche Ergebnis wie die Funktion ZZR(), wenn dort das Argument Rmz auf 0 gesetzt und entweder der Barwert oder der Zukunftswert mit negativem Vorzeichen angegeben ist.

QIKV()

MIRR()

Syntax: QIKV(**Werte**; **Investition**; **Reinvestition**)
Beispiel: =QIKV({-80000;-2000;12000;25000;26000;30000};8%;10%)
Ergebnis: 5%

Mit der Funktion QIKV() kann der qualifizierte interne Zinssatz einer Reihe von Periodenüberschüssen ermittelt werden, bei dem die negativen Cashflows mit einem anderen Zinssatz bewertet werden können als die positiven. Der Grund für unterschiedliche Zinssätze kann beispielsweise darin liegen, dass negative Cashflows mit hohen Überziehungszinsen finanziert werden müssen, während die Anlage von Überschüssen wesentlich geringere Zinsen abwirft.

Drei Argumente müssen für die Funktion eingetragen werden: Das Argument Werte ist eine Matrix oder ein Bereich, der mindestens einen positiven und einen negativen Wert enthält. Der erste Wert stellt beispielsweise die getätigte Investition als Ausgabe dar, erscheint also negativ, die weiteren Werte stellen die periodischen Rückflüsse dar, also die

Überschüsse aus Aus- und Einzahlungen. Nicht numerische Werte werden ignoriert. Fehlerwerte dagegen führen dazu, dass die Funktion diesen Wert auch als Ergebnis ausgibt.

Die Werte müssen, falls sie nicht als Matrix direkt eingegeben werden, fortlaufend in einer Spalte oder auch Zeile angeordnet sein, wobei jede Zelle den Wert für eine Periode angibt. Es wird also unterstellt, dass die Zahlungsvorgänge in der entsprechenden Reihenfolge jeweils am Ende der Periode erfolgen. Die Beträge können unterschiedlich groß sein, es wird aber davon ausgegangen, dass die Verzinsung in regelmäßigen Abständen erfolgt. Dies können Jahre oder auch Monate sein.

Mit `Investition` wird der Zinssatz angegeben, mit dem negative Beträge, also Ausgaben, abgezinst werden; `Reinvestition` ist der Zinssatz, mit dem positive Beträge, also Einnahmen, aufgezinst werden, also der Zinssatz, der realisiert werden kann, wenn etwa erzielte Gewinne aus der Investition wieder angelegt werden. Die Funktion ermittelt jeweils den Barwert der Ausgaben und den Zukunftswert der Einnahmen. Das Ergebnis der Funktion gibt also an, zu welchem Zinssatz das ausgegebene Geld am Ende wieder hereingekommen ist.

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`. Enthält `Werte` nicht mindestens einen negativen Wert, liefert die Funktion den Fehler `#DIV/0!`.

Abbildung 3.39 zeigt ein Beispiel, in dem die Anschaffungskosten und die Verluste im ersten Jahr mit einem Kredit zu 6% finanziert wurden, während die Überschüsse in den folgenden Jahren nur mit einer Verzinsung von 3% angelegt werden konnten.

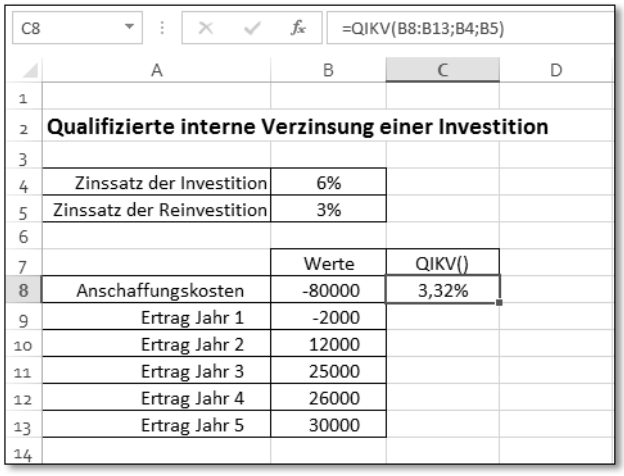


Abbildung 3.39 Berechnung der qualifizierten internen Kapitalverzinsung

RENDITE()

YIELD()

Syntax:	RENDITE(**Abrechnung**; **Fälligkeit**; **Zins**; **Kurs**; **Rückzahlung**; **Häufigkeit**; Basis)
Beispiel:	=RENDITE(DATUM(2010;1;3); DATUM(2014;2;10);3%;96;100;2;4)
Ergebnis:	4,07%

Die Funktion RENDITE() berechnet die jährliche Rendite eines festverzinslichen Wertpapiers, also einer Anleihe oder Obligation.

Abrechnung ist das Datum des Besitzwechsels, Fälligkeit das Fälligkeitsdatum des Wertpapiers. Diese beiden Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit Zins wird der Nominalzins angegeben, mit Kurs der aktuelle Kurs des Papiers bezogen auf einen Nennwert von 100 Einheiten. Rückzahlung ist der Betrag, zu dem das Wertpapier zum Fälligkeitstermin ausbezahlt wird (bei Anleihen im Allgemeinen 100). Alle drei Argumente müssen positive Werte angeben, sonst liefert die Funktion den Fehler #ZAHL!.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C
1			
2	**Rendite bei festverzinslichen Papieren**		
3			
4	Abrechnung	03.01.2010	03.01.2010
5	Fälligkeit	10.02.2014	10.02.2014
6	Zins	3%	3%
7	Kurs	96,00	96,00
8	Rückzahlung	100,00	100,00
9	Häufigkeit	1	2
10	Basis	4	4
11	RENDITE()	4,08%	4,07%

Abbildung 3.40 Berechnen der Rendite einer Anleihe

Ungültige Datumswerte oder nicht numerische Werte für Zins, Kurs oder Rückzahlung führen zu dem Fehler #WERT!, nicht erlaubte Werte für Häufigkeit oder Basis oder Abrechnung > Fälligkeit ergeben den Fehler #ZAHL!.

Mit fast denselben Argumenten lässt sich, wenn die Rendite bekannt ist, mit Hilfe der Funktion KURS() der entsprechende Kurs errechnen. Siehe KURS().

RENDITEDIS()

YIELDDISC()

Syntax:	RENDITEDIS(**Abrechnung**; **Fälligkeit**; **Kurs**; **Rückzahlung**; Basis)
Beispiel:	=RENDITEDIS(DATUM(2012;1;3); DATUM(2013;1;3);96,5;100;4)
Ergebnis:	3,63%

Die Funktion RENDITEDIS() berechnet die jährliche Rendite eines unverzinslichen Wertpapiers. Mit dem etwas irreführenden Adjektiv »unverzinslich« wird gesagt, dass es bei diesem Papier keine periodischen Zinszahlungen gibt. Stattdessen wird beim Kauf ein Abschlag – Disagio – gewährt, d.h., der Zinsbetrag wird vorweg abgezogen. Ein Beispiel sind Zerobonds, die seit 1985 auch in Deutschland gehandelt werden dürfen. Bei diesen Papieren ist die Bonität des Schuldners besonders zu beachten, da ja bei Fälligkeit auf einen Schlag der Anleihebetrag plus Zinsen gezahlt werden müssen.

Abrechnung ist der Kauftermin, Fälligkeit der Rückzahlungstermin. Diese beiden Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Das Papier wird zum Preis von Kurs erworben und zum Fälligkeitstermin zum Preis von Rückzahlung eingelöst. Beide Werte werden bezogen auf 100 Einheiten angegeben. Beide müssen positive Werte sein, sonst liefert die Funktion den Fehler #ZAHL!.

Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für Kurs oder Rückzahlung führen zu dem Fehler #WERT!, nicht erlaubte Werte für Basis oder Abrechnung > Fälligkeit ergeben den Fehler #ZAHL!.

	A	B	C	D	E	F
1						
2	**Jährliche Rendite eines unverzinslichen Wertpapiers**					
3						
4	Abrechnung	03.01.2012	03.01.2012	03.01.2012	03.01.2012	03.01.2012
5	Fälligkeit	03.01.2013	03.01.2013	03.01.2013	03.01.2013	03.01.2013
6	Kurs	96,50	96,50	96,50	96,50	96,50
7	Rückzahlung	100,00	100,00	100,00	100,00	100,00
8	Basis	0	1	2	3	4
9	RENDITEDIS()	3,63%	3,63%	3,57%	3,62%	3,63%

Abbildung 3.41 Berechnung der Rendite eines Wertpapiers

RENDITEFÄLL()

YIELDMAT()

Syntax: `RENDITEFÄLL(`**`Abrechnung`**`; `**`Fälligkeit`**`; `**`Emission`**`; `**`Zins`**`; `**`Kurs`**`; Basis)`

Beispiel: `=RENDITEFÄLL(DATUM(2012;4;1); DATUM(2012;10;1); DATUM(2011;10;1);0,045;99;4)`

Ergebnis: `6,42%`

Die Funktion `RENDITEFÄLL()` berechnet im unterjährigen Bereich, also ohne Zinseszinseffekte, die jährliche Rendite eines Wertpapiers, dessen Zinsen nachschüssig zum Fälligkeitstermin ausgezahlt werden.

	A	B	C	D
1				
2	**Rendite eines Wertpapiers, dessen Zinsen bei Fälligkeit gezahlt werden**			
3				
4	Abrechnung	01.04.2012	01.04.2012	01.04.2012
5	Fälligkeit	01.10.2012	01.10.2012	01.10.2012
6	Emission	01.10.2011	01.10.2011	01.10.2011
7	Zins	5,00%	5,00%	4,50%
8	Kurs	99,00	98,00	99,00
9	Basis	4	4	4
10	RENDITEFÄLL()	6,90%	8,96%	6,42%

Abbildung 3.42 Berechnung der Rendite bei Fälligkeit

`Abrechnung` ist der Kauftermin; das ist gleichzeitig der Zeitpunkt, für den die Rendite berechnet wird. `Fälligkeit` ist der Rücknahme- und damit Auszahlungstermin, `Emission` ist der Ausgabetermin des Papiers, ab dem die Verzinsung berechnet wird. Alle drei Ar-

gumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Zins` ist der vereinbarte jährliche Nominalzinssatz. Das Papier wird zum Preis von `Kurs` am Abrechnungstermin erworben und wird zum Fälligkeitstermin zum Nennwert zusammen mit den aufgelaufenen Zinsen eingelöst. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Zins` oder `Rendite` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn die Datumswerte `Emission`, `Abrechnung` und `Fälligkeit` nicht aufeinanderfolgen oder wenn `Zins` < `0` oder `Kurs` ≤ `0` ist.

RMZ()

PMT()

Syntax: `RMZ(`**`Zins`**`; `**`Zzr`**`; `**`Bw`**`; Zw; F)`

Beispiel: `=RMZ(0,09/12;8;8000)`

Ergebnis: `-1034,04`

Die Funktion `RMZ()` liefert die kontinuierliche Rente, wenn sie innerhalb einer Rentenrechnung verwendet wird. Bei einem Annuitätendarlehen berechnet die Funktion die Annuität, die sich jeweils aus einem Zinsanteil (berechnet mit `ZINSZ()`) und einem Tilgungsanteil (berechnet mit `KAPZ()`), um den sich die Gesamtschuld jeweils verringert, zusammensetzt.

Argumente sind `Zins` für den konstanten Nominalzinssatz, `Zzr` für die Anzahl der Zahlungszeiträume und `Bw`, also der Barwert, der bei einem Darlehen der Darlehenssumme entspricht. Bei der Rentenrechnung gibt dieser Wert den Kontostand am Beginn der Rentenzahlungen an.

Dazu können Sie noch `Zw` angeben für einen Zukunftswert, also einen Betrag, der am Ende der letzten Zahlung noch an der vollständigen Rückzahlung eines Darlehens fehlt. Bei der Rentenrechnung ist es der Kontostand, der am Ende aller Rentenzahlungen erreicht werden soll.

Ist dieser Wert nicht angegeben, wird er als `0` angenommen. Ist `Zw` angegeben, kann `Bw` auch fehlen. Wird dagegen `Bw` nicht angegeben, muss `Zw` angegeben werden.

Der Wert `F` ist optional und gibt die Fälligkeit der Zahlungen an: `0` oder nicht angegeben = nachschüssig, `1` = vorschüssig).

Alle Werte müssen numerisch sein, sonst liefert die Funktion den Fehler `#WERT!`.

Abbildung 3.43 zeigt eine Tabelle, in der der monatliche Betrag für die Annuitätentilgung für verschiedene Jahreszinssätze berechnet wird. Die Formel für B7 zeigt, dass dabei der Jahreszinssatz durch 12 dividiert und die Anzahl der Perioden mit 12 multipliziert werden muss. Das Ergebnis ist negativ, weil diese Beträge ja an den Kreditgeber abfließen. Die Funktion ist also aus der Sicht des Schuldners zu verstehen.

	A	B	C	D	E	F
1						
2	**Berechnen des monatlichen Betrags bei Annuitätentilgung**					
3						
4	Zins	7,00%	6,50%	6,00%		
5	Zzr	10	10	10		
6	Bw	20.000 €	20.000 €	20.000 €		
7	RMZ()	-232,22 €	-227,10 €	-222,04 €		
8	Gesamt	-27.866,04 €	-27.251,51 €	-26.644,92 €		
9						

Abbildung 3.43 Berechnen der Annuität bei einem Darlehen

Wenn Sie wissen wollen, wie viel Sie insgesamt für den Kredit zu zahlen haben, multiplizieren Sie den Wert für `RMZ()` mit der Anzahl der Perioden, in diesem Fall rechnen Sie also: `10 * 12`.

Das Argument `Zw` kann auch verwendet werden, wenn Sie eine Tilgung vereinbaren wollen, wo am Ende ein größerer Restbetrag gezahlt werden soll. Abbildung 3.44 zeigt ein Beispiel, in dem am Ende ein solcher Betrag fällig ist. Er muss als negativer Betrag berücksichtigt werden, da er ja zu zahlen ist.

	A	B	C	D	E
1					
2	**Berechnen des monatlichen Betrags bei einem Restbetrag**				
3					
4	Zins	7,00%	6,50%	6,00%	
5	Zzr	10	10	10	
6	Bw	20.000 €	20.000 €	20.000 €	
7	Zw	- 2.000 €	- 2.000 €	- 2.000 €	
8	RMZ()	-220,66 €	-215,22 €	-209,84 €	
9					

Abbildung 3.44 Berechnen der Annuität bei einem Darlehen mit einem Restbetrag

Die Berechnung von RMZ() basiert auf der in Abschnitt 3.3, »Häufig benötigte Argumente und ihr Zusammenhang«, bereits vorgestellten Grundformel:

```
Bw * (1+Zins)^Zzr + Rmz * (1+Zins * F) * (((1+Zins)^Zzr)-1 / Zins) + Zw = 0
```


TBILLÄQUIV()

TBILLEQ()

Syntax: TBILLÄQUIV(**Abrechnung**; **Fälligkeit**; **Abzinsungssatz**)
Beispiel: =TBILLÄQUIV(DATUM(2012;3;31); DATUM(2012;6;1);0,1)
Ergebnis: 10,32%

Die Funktion TBILLÄQUIV() berechnet vergleichsweise die jährliche nachschüssige Verzinsung eines Schatzwechsels (*Treasury Bill*) auf der Basis von 365 Tagen für einen gegebenen vorschüssigen Jahreszinssatz auf der Basis von 360 Tagen um.

Ein Schatzwechsel ist eine kurzfristige Schuldverschreibung, die zum Termin Abrechnung mit einem Abschlag (Disagio) erworben wird. Abzinsungssatz gibt die Höhe dieses Abschlags in Prozent auf ein Jahr umgerechnet an. Zum Termin Fälligkeit wird das Papier zum vollen Wert eingelöst.

Da Abzinsungssatz praktisch ein vorschüssiger Zins ist, rechnet die Funktion also um, wie der Abzinsungssatz sich in einem nachschüssigen Zins widerspiegeln würde.

Sind Abrechnung und Fälligkeit keine gültigen Datumswerte, liefert die Funktion den Fehler #WERT!. Ist Abzinsungssatz < 0 oder Abrechnung später als Fälligkeit oder Fälligkeit mehr als ein Jahr später als Abrechnung, führt das zu dem Fehler #ZAHL!.

	A	B	C	D
1				
2	**Umrechnung eines vorschüssigen in den äquivalenten nachschüssigen Zinssatz**			
3				
4	Abrechnung	31.03.2012	31.03.2012	31.03.2012
5	Fälligkeit	01.06.2012	01.06.2012	01.06.2012
6	Diskont	8,00%	9,00%	10,00%
7	TBILLÄQUIV()	8,22%	9,27%	10,32%

Abbildung 3.45 Berechnen der vergleichsweisen Verzinsung eines Schatzwechsels

TBILLKURS()

TBILLPRICE()

Syntax: TBILLKURS(**Abrechnung; Fälligkeit; Abzinsungssatz**)
Beispiel: =TBILLKURS(DATUM(2012;3;31); DATUM(2012;6;1);0,10)
Ergebnis: 98,28

Die Funktion TBILLKURS() liefert den Ausgabekurs für einen Schatzwechsel als Prozentsatz bezogen auf einen Nennwert von 100 Einheiten.

Ein Schatzwechsel ist eine kurzfristige Schuldverschreibung, die zum Termin Abrechnung mit einem Abschlag (Disagio) erworben wird. Abzinsungssatz gibt die Höhe dieses Abschlags in Prozent auf ein Jahr umgerechnet an. Zum Termin Fälligkeit wird das Papier zum vollen Wert eingelöst.

Sind Abrechnung und Fälligkeit keine gültigen Datumswerte, liefert die Funktion den Fehler #WERT!. Ist Abzinsungssatz < 0 oder Abrechnung später als Fälligkeit oder Fälligkeit mehr als ein Jahr später als Abrechnung, führt das zu dem Fehler #ZAHL!.

Die Kursberechnung erfolgt durch eine einfache Prozentrechnung, bei der der per Abzinsungssatz angegebene vorschüssige Jahreszinssatz auf die tatsächliche Laufzeit heruntergebrochen wird. Dabei wird das Jahr mit 360 Tagen und der Monat mit den tatsächlichen Tagen berechnet:

```
TBILLKURS=(1-(Fälligkeit-Abrechnung)/360)*Abzinsungssatz)
```

	A	B	C	D
1				
2	**Kurs eines Schatzwechsels pro 100 € Nennwert**			
3				
4	Abrechnung	31.03.2012	31.03.2012	31.03.2012
5	Fälligkeit	01.06.2012	01.06.2012	01.06.2012
6	Disagio	8,00%	9,00%	10,00%
7	TBILLKURS()	98,62	98,45	98,28

Abbildung 3.46 Berechnen des Kurses eines abgezinsten Wertpapiers

TBILLRENDITE()

TBILLYIELD()

Syntax: `TBILLRENDITE(Abrechnung; Fälligkeit; Pr)`
Beispiel: `=TBILLRENDITE(DATUM(2012;3;31); DATUM(2012;6;1);99,5)`
Ergebnis: `2,92%`

Die Funktion `TBILLRENDITE()` liefert die Rendite eines Schatzwechsels, der zu dem mit dem Argument `Pr` angegebenen Ausgabekurs erworben wurde. `Pr` wird als Prozentsatz angegeben, bezogen auf einen Nennwert von 100 Einheiten.

Ein Schatzwechsel ist eine kurzfristige Schuldverschreibung, die zum Termin `Abrechnung` mit einem Abschlag (Disagio) erworben wird. Zum Termin `Fälligkeit` wird das Papier zum vollen Wert eingelöst.

Sind `Abrechnung` und `Fälligkeit` keine gültigen Datumswerte, liefert die Funktion den Fehler `#WERT!`. Ist `Pr` ≤ `0` oder `Abrechnung` später als `Fälligkeit` oder `Fälligkeit` mehr als ein Jahr später als `Abrechnung`, führt das zu dem Fehler `#ZAHL!`.

Die Funktion arbeitet mit der Formel:

```
TBILLRENDITE = ((100 - Pr) / Pr) * (360 / (Fälligkeit - Abrechnung))
```

	A	B	C	D	E
1					
2	**Rendite eines Schatzwechsels**				
3					
4	Abrechnung	31.03.2012	31.03.2012	31.03.2012	
5	Fälligkeit	01.06.2012	01.06.2012	01.06.2012	
6	Kurs	98,50	99,00	99,50	
7	TBILLRENDITE()	8,84%	5,87%	2,92%	
8					

Abbildung 3.47 Berechnen der Rendite eines Schatzwechsels

UNREGER.KURS()

ODDFPRICE()

Syntax: `UNREGER.KURS(`**`Abrechnung; Fälligkeit; Emission; Erster_Zinstermin; Zins; Rendite; Rückzahlung; Häufigkeit;`** `Basis)`

Beispiel: `=UNREGER.KURS(DATUM(2008;11;11); DATUM(2013;3;1); DATUM(2008;10;15); DATUM(2009;3;1); 0,0785;0,0625;100;2;1)`

Ergebnis: `105,96`

Die Funktion `UNREGER.KURS()` kalkuliert den Kurswert eines festverzinslichen Wertpapiers bezogen auf den Nennwert von 100 Einheiten, bei dem die erste Zinsperiode kürzer oder auch länger als die folgenden regelmäßigen Perioden ausfällt. Die Funktion liefert eine alternative Berechnungsmethode für die Berücksichtigung der ersten (gebrochenen) Zinsperiode.

`Abrechnung` ist der Kauftermin, für den der Kurs errechnet werden soll, `Fälligkeit` der Rückzahlungstermin. `Emission` ist der Ausgabetermin, ab dem die Zinsberechnung läuft. Der Termin, ab dem die regelmäßigen Zinszahlungen einsetzen, wird mit `Erster_Zinstermin` angegeben. All diese Termine verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

	A	B	C
1			
2	**Kurs/Rendite bei unregelmäßiger 1. Periode**		
3			
4	Abrechnung	11.11.2008	
5	Fälligkeit	01.03.2013	
6	Emission	15.10.2008	
7	Erster Zinstermin	01.03.2009	
8	Zins	7,85%	
9	*Rendite	6,25%	
10	*Kurs	101	
11	Rückzahlung	100	
12	Häufigkeit	2	
13	Basis	1	
14	UNREGER.KURS()	105,96	
15	KURS()	105,94	
16	UNREGER.REND()	7,57%	
17	RENDITE()	7,57%	
18			

Abbildung 3.48 Kurs und Rendite bei Papieren mit kürzerer erster Zinsperiode

Der Nominalzins wird mit Zins angegeben. Rendite ist die jährliche Rendite des Papiers; Rückzahlung ist der auf 100 Einheiten bezogene Betrag, der zum Fälligkeitstermin ausbezahlt wird.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für Zins, Rendite oder Rückzahlung führen zu dem Fehler #WERT!, nicht erlaubte Werte für Häufigkeit oder Basis ergeben den Fehler #ZAHL!. Das gilt auch, wenn die Reihenfolge der Termine nicht eingehalten wird: Emission < Abrechnung < Erster_Zinstermin < Fälligkeit.

Das in Abbildung 3.48 mit * bezeichnete Argument Kurs wird bei der folgenden Funktion UNREGER.REND() benötigt, das Argument Rendite bei der vorliegenden Funktion UNREGER.KURS().

UNREGER.REND()

ODDFYIELD()

Syntax:	UNREGER.REND(**Abrechnung**; **Fälligkeit**; **Emission**; **Erster_Zinstermin**; **Zins**; **Kurs**; **Rückzahlung**; **Häufigkeit**; Basis)
Beispiel:	=UNREGER.REND(DATUM(2008;11;11); DATUM(2013;3;1); DATUM(2008;10;15); DATUM(2009;3;1); 0,0785;101;100;2;1)
Ergebnis:	7,57%

Die Funktion UNREGER.REND() liefert die Rendite eines festverzinslichen Wertpapiers bezogen auf den Nennwert von 100 Einheiten, bei dem die erste Zinsperiode kürzer oder auch länger als die folgenden regelmäßigen Perioden ausfällt. Die Funktion liefert eine alternative Berechnungsmethode für die Berücksichtigung der ersten (gebrochenen) Zinsperiode.

Abrechnung ist der Kauftermin, Fälligkeit der Rückzahlungstermin. Emission ist der Ausgabetermin, ab dem die Zinsberechnung läuft. Der Termin, ab dem die regelmäßigen Zinszahlungen einsetzen, wird mit Erster_Zinstermin angegeben. Alle diese Termine verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text

übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Der Nominalzins wird mit `Zins` angegeben. `Kurs` ist der Kurs zum Termin `Abrechnung`. `Rückzahlung` ist der auf 100 Einheiten bezogene Betrag, der zum Fälligkeitstermin ausbezahlt wird.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Zins`, `Kurs` oder `Rückzahlung` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn die Reihenfolge der Termine nicht eingehalten wird: `Emission` < `Abrechnung` < `Erster_Zinstermin` < `Fälligkeit`.

Siehe auch Abbildung 3.48 und die Funktion `UNREGER.KURS()`.

UNREGLE.KURS()

ODDLPRICE()

Syntax:	`UNREGLE.KURS(`**`Abrechnung; Fälligkeit; Letzter_Zinstermin; Zins; Rendite; Rückzahlung; Häufigkeit`**`; Basis)`
Beispiel:	`=UNREGLE.KURS(DATUM(2013;3;12); DATUM(2013;6;1); DATUM(2013;1;1); 0,038;0,05;100;2;0)`
Ergebnis:	`99,72`

Die Funktion `UNREGLE.KURS()` kalkuliert den Kurswert eines festverzinslichen Wertpapiers bezogen auf den Nennwert von 100 Einheiten in einer letzten Zinsperiode, deren Länge sich von den vorhergehenden gleichmäßigen Perioden unterscheidet. Dabei wird mit einfacher Verzinsung gerechnet. Die Funktion kann nur für diese Periode verwendet werden.

`Abrechnung` ist der Kauftermin, für den der Kurs errechnet werden soll, `Fälligkeit` der Rückzahlungstermin. Der Termin, an dem die letzte regelmäßige Zinszahlung vor dem Kauftermin stattfindet, wird mit `Letzter_Zinstermin` angegeben. Alle diese Termine verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Da-

tumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Der Nominalzins wird mit `Zins` angegeben. `Rendite` ist die jährliche Rendite des Papiers; `Rückzahlung` ist der auf 100 Einheiten bezogene Betrag, der zum Fälligkeitstermin ausbezahlt wird.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Zins`, `Rendite` oder `Rückzahlung` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn die Reihenfolge der Termine nicht eingehalten wird: `Erster_Zinstermin` < `Abrechnung` < `Fälligkeit`.

Von den im Beispiel in Abbildung 3.49 gezeigten Argumenten wird `*Kurs` nur für die folgende Funktion `UNREGLE.REND()` benötigt, bei der dann naturgemäß `*Rendite` wegfällt.

	A	B
1		
2	**Kurs/Rendite bei unregelmäßiger letzter Zinsperiode**	
3		
4	Abrechnung	12.03.2013
5	Fälligkeit	01.06.2013
6	Letzter Zinstermin	01.01.2013
7	Zins	3,8%
8	*Rendite	5%
9	*Kurs	99,87
10	Rückzahlung	100
11	Häufigkeit	2
12	Basis	0
13	UNREGLE.KURS()	99,72
14	UNREGLE.REND()	4,32%

Abbildung 3.49 Kurs und Rendite für Wertpapiere mit unregelmäßiger letzter Zinsperiode

UNREGLE.REND()

ODDLYIELD()

Syntax: `UNREGLE.REND(`**`Abrechnung`**`; `**`Fälligkeit`**`; `**`Letzter_Zinstermin`**`; `**`Zins`**`; `**`Kurs`**`; `**`Rückzahlung`**`; `**`Häufigkeit`**`; Basis)`

Beispiel: `=UNREGLE.REND(DATUM(2013;3;12); DATUM(2013;6;1); DATUM(2013;1;1);0,038;99,87;100;2;0)`

Ergebnis: `4,32%`

Die Funktion `UNREGLE.REND()` berechnet die Rendite eines festverzinslichen Wertpapiers bezogen auf den Nennwert von 100 Einheiten in einer letzten Zinsperiode, deren Länge sich von den vorhergehenden gleichmäßigen Perioden unterscheidet. Dabei wird mit einfacher Verzinsung gerechnet. Die Funktion kann nur für diese Periode verwendet werden.

`Abrechnung` ist der Kauftermin, `Fälligkeit` der Rückzahlungstermin. Der Termin, an dem die letzte regelmäßige Zinszahlung vor dem Kauftermin stattfindet, wird mit `Letzter_Zinstermin` angegeben. Alle diese Termine verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Der Nominalzins wird mit `Zins` angegeben. `Kurs` ist der Kurs zum Termin `Abrechnung`. `Rückzahlung` ist der auf 100 Einheiten bezogene Betrag, der zum Fälligkeitstermin ausbezahlt wird.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte oder nicht numerische Werte für `Zins`, `Rendite` oder `Rückzahlung` führen zu dem Fehler `#WERT!`, nicht erlaubte Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`. Das gilt auch, wenn die Reihenfolge der Termine nicht eingehalten wird: `Abrechnung` < `Erster_Zinstermin` < `Fälligkeit`.

Siehe auch Abbildung 3.49 und die Funktion `UNREGLE.KURS()`.

VDB()

VDB()

Syntax:	VDB(**Ansch_Wert; Restwert; Nutzungsdauer; Anfang; Ende;** Faktor; Nicht_wechseln)
Beispiel:	=VDB(22000;2000;5;0;1)
Ergebnis:	4000

Die Funktion VDB() berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der variabel-degressiven Methode.

Diese Funktion ist eine Variation der geometrisch-degressiven Abschreibung. Sie ermöglicht es, sobald die lineare Abschreibung höhere Abschreibungsbeträge liefert, in diese überzuwechseln.

Das Argument Ansch_Wert ist der Betrag, der für die Investition aufgewandt wurde. Restwert ist der Wert, den die Investition nach der Abschreibung noch hat. Nutzungsdauer entspricht der Anzahl von Zeiträumen bis zum Ende der Abschreibung. Mit Faktor wird die Stärke der Degressivität angegeben (ohne Angabe wird 2 angesetzt, was der Doppelratenabschreibung entspricht). Dabei handelt es sich um die Rate, um die jeweils der Restbuchwert abnimmt.

Anfang ist der Anfang der Periode, für die die Abschreibung berechnet werden soll, Ende der Endzeitpunkt. Soll z. B. die Abschreibung für das erste Jahr berechnet werden, wären 0 und 1 einzusetzen. Die Abschreibung kann so auch über mehrere Perioden berechnet werden.

Nicht_wechseln wird durch einen Wahrheitswert geschaltet. Wenn die nach der linearen Methode erzielten Abschreibungen höher liegen als bei der degressiven, wechselt die Funktion auf die lineare Abschreibung, falls Nicht_wechseln mit FALSCH belegt wird oder nicht angegeben ist. Ist das Argument mit WAHR belegt, so wird das Abschreibungsverfahren nicht geändert.

Alle Argumente außer Nicht_wechseln benötigen positive numerische Werte. Nicht numerische Werte führen zu dem Fehler #WERT!, negative Werte zu dem Fehler #ZAHL!.

Steuerrechtlich gelten für die variabel-degressive Abschreibung die Einschränkungen, die schon für die Funktion GDA() beschrieben wurden.

	A	B	C	D	E	F
1						
2	**Variabel-degressive Abschreibung**					
3						
4	Anschaffungskosten	22.000 €	22.000 €	22.000 €	22.000 €	22.000 €
5	Restwert	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €
6	Nutzungsdauer in Jahren	5	5	5	5	5
7	Anfang	0	1	2	3	4
8	Ende	1	2	3	4	5
9	Faktor	2	2	2	2	2
10	Nicht_wechseln					
11	VDB()	8.800 €	5.280 €	3.168 €	1.901 €	851 €
12					Gesamt:	20.000,00 €

Abbildung 3.50 Verlauf der variabel-degressiven Abschreibung

XINTZINSFUSS()

XIRR()

Syntax: `XINTZINSFUSS(`**`Werte; Zeitpkte`**`; Schätzwert)`

Beispiel: `=XINTZINSFUSS(B4:B8;A4:A8;B10)`
siehe Abbildung 3.51

Die Funktion `XINTZINSFUSS()` berechnet – etwas im Widerspruch zum Funktionsnamen – den internen Zinssatz für eine Reihe von Zahlungsvorgängen im unterjährigen Bereich, die in unterschiedlicher Höhe zu nicht regelmäßigen Zeitpunkten erfolgen kann. Mit dieser Funktion lassen sich auch Investitionen und Zahlungsströme überprüfen, die sonst sehr undurchschaubar bleiben.

`Werte` sind die Zahlungsvorgänge, die in unregelmäßigen Abständen erfolgen können, wobei mindestens ein positiver und ein negativer Wert enthalten sein müssen. Die Daten werden in einer Spalte oder Zeile lückenlos abgelegt oder als Matrix übergeben. Die erste Zahlung ist optional und entspricht einer Auszahlung zu Beginn einer Investition. Alle folgenden Zahlungen werden ausgehend von einem Jahr mit 365 Tagen abgezinst.

`Zeitpkte` sind die zu den Zahlungen gehörenden Termine. Der erste Termin ist der Beginn des Zahlungsplans, alle anderen müssen später liegen, brauchen aber nicht unbedingt in zeitlicher Reihenfolge angeordnet zu sein.

Mit `Schätzwert` kann eine Vorgabe für den erwarteten internen Zinsfuß gemacht werden, ohne Angabe geht Excel von 10% aus.

Ungültige Datumswerte für `Zeitpkte` oder nicht numerische Werte für `Werte` oder `Schätzwert` führen zu dem Fehler `#WERT!`, Liegt einer der Datumswerte vor dem Anfangstermin, liefert die Funktion den Fehler `#ZAHL!`. Das gilt auch, wenn `Werte` und `Zeitpkte` unterschiedlich viele Elemente enthalten. Bei dieser Funktion wird ein Näherungsverfahren verwendet, das spätestens nach 100 Iterationsschritten mit der Fehlermeldung `#ZAHL!` abgebrochen wird, wenn die Unterschiede nachfolgender Ergebnisse nicht gegen 0,0000001 konvergieren.

	A	B
1		
2	**Interner Zinsfuß und Kapitalwert**	
3		
4	01.01.2012	-100000
5	01.02.2012	30000
6	01.06.2012	20000
7	15.09.2012	30000
8	18.12.2012	25000
9	Summe der Zahlungen	5000
10	Schätzwert	10,00%
11	XINTZINSFUSS()	9,64%
12	XKAPITALWERT()	0,00

Abbildung 3.51 Interner Zinsfuß und Kapitalwert bei unregelmäßigen Zahlungen

Der von `XINTZINSFUSS()` errechnete Zinssatz ist genau der Zinssatz, für den die verwandte Funktion `XKAPITALWERT()` den Wert `0` liefert, wie Abbildung 3.51 belegt.

XKAPITALWERT()

XNPV()

Syntax: `XKAPITALWERT(`**`Zins; Werte; Zeitpkte`**`)`

Beispiel: `=XKAPITALWERT(B11;B4:B8;A4:A8)`
siehe Abbildung 3.51

Die Funktion `XKAPITALWERT()` berechnet den Nettokapitalwert für eine Reihe von Zahlungsvorgängen im unterjährigen Bereich, die in unterschiedlicher Höhe zu nicht regelmäßigen Zeitpunkten erfolgen kann. Mit dieser Funktion lassen sich auch Investitionen und Zahlungsströme überprüfen, die sonst sehr undurchschaubar bleiben.

`Zins` gibt den Jahreszinssatz an, der in die Berechnung eingehen soll. `Werte` sind die Zahlungsvorgänge, die in unregelmäßigen Abständen erfolgen können, wobei mindestens ein positiver und ein negativer Wert enthalten sein muss. Die Daten werden in einer

Spalte oder Zeile lückenlos abgelegt oder als Matrix übergeben. Die erste Zahlung ist optional und entspricht einer Auszahlung für eine Investition. Alle folgenden Zahlungen werden ausgehend von einem Jahr mit 365 Tagen abgezinst.

`Zeitpkte` sind die zu den Zahlungen gehörenden Termine. Der erste Termin ist der Beginn des Zahlungsplans, alle anderen müssen später liegen, brauchen aber nicht unbedingt in zeitlicher Reihenfolge angeordnet zu sein.

Ungültige Datumswerte für `Zeitpkte` oder nicht numerische Werte für `Zins` oder `Werte` führen zu dem Fehler `#WERT!`, Liegt einer der Datumswerte vor dem Anfangstermin, liefert die Funktion den Fehler `#ZAHL!`. Das gilt auch, wenn `Werte` und `Zeitpkte` unterschiedlich viele Elemente enthalten.

Siehe auch `XINTZINSFUSS()`.

ZINS()

RATE()

Syntax:	`ZINS(`**`Zzr`**`; `**`Rmz`**`; `**`Bw`**`; Zw; F; Schätzwert)`
Beispiel:	`=ZINS(15;;-10000;22000)`
Ergebnis:	`5,4%`

Die Funktion `ZINS()` berechnet den vorliegenden Zinssatz bei regelmäßigen Zahlungen. Sie kann sowohl im Bereich der Zinseszinsrechnung als auch bei der Renten- und Tilgungsrechnung verwendet werden.

Die Funktion ist hilfreich, wenn die anderen Daten eines Finanzierungsvorhabens bekannt sind, etwa die regelmäßigen Zahlungen, die Anzahl der Perioden, der Barwert oder der Zukunftswert, und es darum geht, herauszufinden, bei welchem Zinssatz die angestrebten Zielsetzungen realisiert werden können. Oder umgekehrt, um zu prüfen, ob das Vorhaben mit den von der Bank angebotenen Zinssätzen überhaupt realistisch ist.

Mit `Zzr` wird die Zahl der Zahlungsperioden angegeben, an deren Anfang oder Ende jeweils Zahlungen erfolgen. Soll die Funktion den Zinssatz für eine monatliche Verzinsung berechnen, muss der Jahreswert von `Zzr` mit 12 multipliziert angegeben werden, also etwa `3 * 12` für 36 Monate.

`Rmz` ist der über die Zahlungszeiträume konstante regelmäßige Zahlungs- oder Rentenbetrag. Das Vorzeichen hängt ab von der Richtung, in die das Geld fließt. Für Einzahlun-

gen benutzen Sie ein Minuszeichen vor dem Betrag, Auszahlungen werden als positive Werte angegeben.

Bw ist der Barwert, also der aktuelle Wert, z.B. die anfängliche Einzahlung, der Kontostand oder die Kreditsumme, die getilgt werden soll.

Zw ist der Betrag, der in Zukunft, genauer am Ende der Zahlungsperioden, erreicht werden soll. Wird der Zukunftswert nicht angegeben, wird er von der Funktion als 0 angenommen. Bei der Abzahlung eines Kredits entspricht dies der vollständigen Tilgung.

Wird F (Fälligkeit der Zahlungen: 1 für am Anfang, 0 für am Ende der Zahlungszeiträume) nicht gesetzt, so wird sie ebenfalls als 0 angenommen.

Es müssen mindestens zwei der drei Argumente Rmz, Bw, und Zw angegeben werden. Wenn also für Rmz kein Wert angegeben wird, muss für Zw ein Wert angegeben werden.

Mit Schätzwert können Sie angeben, wie hoch Sie den Zins einschätzen (das kann das Rechenverfahren verkürzen). Geben Sie keinen Wert ein, so wird er von der Funktion mit 10% veranschlagt.

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler #WERT!. Rmz, Bw und Zw dürfen nicht alle das gleiche Vorzeichen haben, sonst erscheint der Fehler #ZAHL!. Bei der Berechnung wird ein Iterationsverfahren angewendet, das nach maximal 20 Schritten mit der Fehlermeldung #ZAHL! abgebrochen wird, wenn die Unterschiede nachfolgender Ergebnisse nicht gegen 0,0000001 konvergieren.

In den Spalten B und C der Abbildung 3.52 wird der Zins für den Fall berechnet, dass eine Summe von 10.000 € für 15 Jahre angelegt wird. Berechnet wird der Zinssatz, der benötigt wird, damit das Geld auf 22.000 € anwächst.

Der Wert für den Barwert wird mit einem Minuszeichen angegeben, da es sich ja um eine Einzahlung (weg von mir, hin zur Bank) handelt. Der Zukunftswert wird als positiver Wert angegeben, da es sich um eine erwartete Auszahlung handelt (weg von der Bank, hin zu mir).

Die beiden Spalten unterscheiden sich nur durch den Wert von F, der aber in diesem Fall keine Rolle spielt, das Ergebnis ist in beiden Fällen gleich. Die angezeigten Ergebnisse sind nur durch die Formatierung gerundet, Excel rechnet aber genau, was sofort sichtbar wird, wenn Sie weitere Dezimalstellen hinzufügen.

In den Spalten D und E wird dagegen am Anfang ein Betrag von 55.000 € einbezahlt, in der Absicht, davon 15 Jahre lang jährlich 5.000 € abzuheben, bis alles verbraucht ist. Die Funktion rechnet aus, welche Verzinsung notwendig ist, damit dies gelingt. In die-

sem Fall macht es einen deutlichen Unterschied, ob für `F` mit dem Wert `0` – Zahlung am Ende der Periode – oder mit `1` – Zahlung am Anfang der Periode – gearbeitet wird.

In den Spalten F und G werden jährlich 5.000 € eingezahlt. Nach 15 Jahren soll ein Gesamtbetrag von 110.000 € erreicht werden. Die Funktion berechnet, zu welchem Zinssatz dies möglich ist. Auch hier werden die Ergebnisse durch den Wert für `F` beeinflusst.

	A	B	C	D	E	F	G
1							
2	**Zinssatz pro Periode**						
3							
4	Zzr	15	15	15	15	15	15
5	Rmz			5.000,00 €	5.000,00 €	-5.000,00 €	-5.000,00 €
6	Bw	-10.000,00 €	-10.000,00 €	-55.000,00 €	-55.000,00 €		
7	Zw	22.000,00 €	22.000,00 €			110.000,00 €	110.000,00 €
8	F	0	1	0	1	0	1
9	Schätzwert						
10	ZINS()	5,40%	5,40%	4,15%	4,84%	5,26%	4,65%

Abbildung 3.52 Berechnung des Zinssatzes bei regelmäßigen Zahlungen

3.7.4 Anpassung an monatliche Zahlungen

Wenn es um monatliche Auszahlungen geht, muss darauf geachtet werden, dass die Daten an diese Periode angepasst werden. Abbildung 3.53 zeigt die Formel, die dazu in C10 verwendet wird. Als Argument für die Zahlungszeiträume wird die Zahl der Jahresperioden aus C4 mit 12 multipliziert. Da der Wert für das Argument `Rmz` schon als Monatswert angegeben ist, muss hier nichts geändert werden. Die Funktion berechnet dann den Monatszinssatz. In C11 wird dieser Wert mit 12 multipliziert, um den Jahreszinssatz zu erhalten.

	A	B	C	D
1				
2	**Zinssatz bei monatlicher Zahlung**			
3				
4	Zzr	15	15	
5	monatliche Rmz	416,67 €	416,67 €	
6	Bw	-55.000,00 €	-55.000,00 €	
7	Zw			
8	F	0	1	
9	Schätzwert			
10	ZINS()	0,36%	0,37%	**=ZINS(C4*12;C5;C6;C7;C8;C9)**
11	Jahreszins	4,35%	4,41%	**=C10*12**

Abbildung 3.53 Berechnung des Zinssatzes bei monatlichen Auszahlungen

ZINSSATZ()

RECEIVED()

Syntax: `ZINSSATZ(`**`Abrechnung; Fälligkeit; Anlage; Rückzahlung`**`; Basis)`
Beispiel: `=ZINSSATZ(DATUM(2011;10;12); DATUM(2012;9;1);9600;10000;4)`
Ergebnis: `4,70%`

Die Funktion `ZINSSATZ()` berechnet den entsprechenden nachschüssigen (jährlichen) Zinssatz für eine Investition, bei der zwischen Abrechnung und Rückzahlung keine Zinsen ausgeschüttet werden, sondern vorschüssig ein Abschlag (Disagio) stattfindet. `ZINSSATZ()` geht von einer Laufzeit unter einem Jahr aus, Zinseszinseffekte werden nicht berücksichtigt. Die Funktion ermittelt also, welchem Zinssatz der Abschlag entspricht, wenn die Zinsen am Ende der Laufzeit gezahlt würden.

Bei der Investition handelt es sich nach der Terminologie der Argumente um den Kauf von Wertpapieren, die Funktion kann aber auf jede Situation angewendet werden, in der ein Betrag eingezahlt und ein anderer Betrag nach einer Frist zurückgezahlt wird.

`Abrechnung` ist der Kauftermin, `Fälligkeit` der Rückzahlungstermin. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Anlage` ist der Betrag, der angelegt wurde, `Rückzahlung` der Betrag, der zum Fälligkeitstermin zurückgezahlt wird. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D	E
1					
2	**Zinssatz eines voll investierten Wertpapiers**				
3					
4	Abrechnung	12.10.2011	12.10.2011	12.10.2011	12.10.2011
5	Fälligkeit	01.09.2012	01.09.2012	01.09.2012	01.09.2012
6	Anlage	9.600,00 €	9.600,00 €	9.600,00 €	9.600,00 €
7	Rückzahlung	10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €
8	Basis	1	2	3	4
9	ZINSSATZ()	4,69%	4,62%	4,68%	4,70%

Abbildung 3.54 Zinssatz eines Wertpapiers

Ungültige Datumswerte führen zu dem Fehler #WERT!. Werte für Anlage oder Rückzahlung, die ≤ 0 sind, ergeben den Fehler #ZAHL!. Denselben Fehler liefert die Funktion, wenn Abrechnung ≥ Fälligkeit oder für Basis ein ungültiger Wert angegeben ist.

ZINSTERMNZ()

COUPNCD()

Syntax: ZINSTERMNZ(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)
Beispiel: =ZINSTERMNZ(DATUM(2013;10;12); DATUM(2014;9;1);4;4)
Ergebnis: 01.12.2013

Die Funktion ZINSTERMNZ() berechnet bei einem festverzinslichen Wertpapier das Datum der ersten Zinsausschüttung nach dem Kauftermin.

Dabei ist Abrechnung der Kauftermin, Fälligkeit der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D
1				
2	**Berechnen des nächsten Zinstermins**			
3				
4	Abrechnung	12.10.2013	12.10.2013	12.10.2013
5	Fälligkeit	01.09.2014	01.09.2014	01.09.2014
6	Häufigkeit	1	2	4
7	Basis	4	4	4
8	ZINSTERMNZ()	01.09.2014	01.03.2014	01.12.2013

Abbildung 3.55 Zinstermine bei unterschiedlichen Häufigkeiten

Ungültige Datumswerte führen zu dem Fehler #WERT!. Ungültige Werte für Häufigkeit oder Basis ergeben den Fehler #ZAHL!.

Die Funktion ZINSTERMNZ() ist eine von sechs Funktionen mit dem Präfix »ZINSTERM«, die alle mit Berechnungen zu festverzinslichen Wertpapieren zu tun haben. Abbildung 3.56 zeigt ein Tabellenblatt, in dem die Daten eines solchen Wertpapiers zusammengestellt sind:

	A	B	C
1			
2	**Daten eines festverzinslichen Wertpapiers**		
3			
4	Abrechnung	15.05.2007	
5	Kaufkurs	98,48	
6	Rückzahlung	100,00	
7	Fälligkeit	06.05.2013	
8	Nominalzins	4,63%	
9	Zeitbasis	4	
10	Häufigkeit der Zinszahlungen	1	
11	Letzter Zinszahlungstermin	06.05.2007	=ZINSTERMVZ(B4;B7;B10;B9)
12	Nächster Zinszahlungstermin	06.05.2008	=ZINSTERMNZ(B4;B7;B10;B9)
13	Jahresrest in Tagen	351	=ZINSTERMTAGNZ(B4;B7;B10;B9)
14	Tage für Stückzinsen	9	=ZINSTERMTAGVA(B4;B7;B10;B9)
15	Tage im Jahr	360	=ZINSTERMTAGE(B4;B7;B10;B9)
16	Stückzinsen	0,12	=B8*B14/B15*B6
17	Anzahl der Zinstermine	6,00	=ZINSTERMZAHL(B4;B7;B10;B9)
18	Kaufpreis	98,60	=B5+B16
19	**Rendite p.a.**	**4,92%**	
20	**Ergebnis der Funktion RENDITE():**	**4,92%**	=RENDITE(B4;B7;B8;B5;B6;1;B9)

Abbildung 3.56 Daten zu einem festverzinslichen Wertpapier, die mit Hilfe der ZINSTERM*-Funktionen berechnet werden können

ZINSTERMTAGE()

COUPDAYS()

Syntax: ZINSTERMTAGE(**Abrechnung; Fälligkeit; Häufigkeit**; Basis)

Beispiel: =ZINSTERMTAGE(DATUM(2013;1;27); DATUM(2014;11;17);4;4)

Ergebnis: 90

Die Funktion ZINSTERMTAGE() berechnet bei einem festverzinslichen Wertpapier die Anzahl der Tage in derjenigen Zinsperiode, in die der Abrechnungszeitpunkt fällt.

Dabei ist Abrechnung der Kauftermin, Fälligkeit der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argu-

ment auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Häufigkeit` ist die Zahl der jährlichen Zinsausschüttungen. Zulässig sind die Werte: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D	E
1					
2	**Dauer der Zinsperiode, in der der Abrechnungstermin liegt**				
3					
4	Abrechnung	25.01.2013	26.01.2013	27.01.2013	
5	Fälligkeit	15.11.2014	16.11.2014	17.11.2014	
6	Häufigkeit	1	2	4	
7	Basis	1	3	4	
8	ZINSTERMTAGE()	365	183	90	
9					

Abbildung 3.57 Tage der Zinsperiode mit dem Abrechnungstermin

Ungültige Datumswerte führen zu dem Fehler `#WERT!`. Ungültige Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`.

ZINSTERMTAGNZ()

COUPDAYSNC()

Syntax: `ZINSTERMTAGNZ(`**`Abrechnung; Fälligkeit; Häufigkeit`**`; Basis)`

Beispiel: `=ZINSTERMTAGNZ(DATUM(2013;1;25); DATUM(2014;11;15);4;4)`

Ergebnis: `20`

Die Funktion `ZINSTERMTAGNZ()` berechnet bei einem festverzinslichen Wertpapier die Tage vom Kauftermin bis zum ersten Zinstermin. Das Ergebnis der Funktion kann dann verwendet werden, um die Zinsen für diesen Zeitraum zu berechnen, die dem neuen Besitzer zustehen.

Dabei ist `Abrechnung` der Kauftermin, `Fälligkeit` der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D
1				
2	**Tage vom Abrechnungstermin bis zum nächsten Zinstermin**			
3				
4	Abrechnung	25.01.2013	25.01.2013	25.01.2013
5	Fälligkeit	15.11.2014	15.11.2014	15.11.2014
6	Häufigkeit	1	2	4
7	Basis	4	4	4
8	ZINSTERMTAGNZ()	290	110	20

Abbildung 3.58 Anzahl Tage bis zum 1. Zinstermin nach dem Kauf des Wertpapiers

Ungültige Datumswerte führen zu dem Fehler #WERT!. Ungültige Werte für Häufigkeit oder Basis ergeben den Fehler #ZAHL!.

ZINSTERMTAGVA()

COUPDAYBS()

Syntax: ZINSTERMTAGVA(**Abrechnung; Fälligkeit; Häufigkeit**; Basis)

Beispiel: =ZINSTERMTAGVA(DATUM(2013;3;25); DATUM(2014;6;15);4;4)

Ergebnis: 10

Die Funktion ZINSTERMTAGVA() berechnet bei einem festverzinslichen Wertpapier die Anzahl der Tage vom letzten Zinstermin vor der Abrechnung bis zur Abrechnung. Das Ergebnis der Funktion kann verwendet werden, um die Stückzinsen auszurechnen, die beim Kauf zusätzlich zum Kurs zu zahlen sind.

Dabei ist Abrechnung der Kauftermin, Fälligkeit der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit Häufigkeit wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: 1 – jährlich, 2 – halbjährlich, 4 – vierteljährlich. Basis gibt an, nach welchem

Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D
1				
2	**Tage vom letzten Zinstermin bis zum Abrechnungstermin**			
3				
4	Abrechnung	25.03.2013	25.03.2013	25.03.2013
5	Fälligkeit	15.06.2014	15.06.2014	15.06.2014
6	Häufigkeit	1	2	4
7	Basis	4	4	4
8	ZINSTERMTAGVA()	280	100	10

Abbildung 3.59 Dauer der letzten Zinsperiode vor dem Kauf

Ungültige Datumswerte führen zu dem Fehler `#WERT!`. Ungültige Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`.

ZINSTERMVZ()

COUPPCD()

Syntax: `ZINSTERMVZ(`**`Abrechnung`**`; `**`Fälligkeit`**`; `**`Häufigkeit`**`; Basis)`

Beispiel: `=ZINSTERMVZ(DATUM(2013;7;25); DATUM(2014;11;15);1;4)`

Ergebnis: `15.05.2013`

Die Funktion `ZINSTERMVZ()` berechnet bei einem festverzinslichen Wertpapier das Datum des letzten Zinstermins vor der Abrechnung.

Dabei ist `Abrechnung` der Kauftermin, `Fälligkeit` der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

Ungültige Datumswerte führen zu dem Fehler `#WERT!`. Ungültige Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`.

	A	B	C	D
1				
2	**Datum der letzten Zinszahlung vor dem Kauf**			
3				
4	Abrechnung	25.07.2013	25.07.2013	25.07.2013
5	Fälligkeit	15.11.2014	15.11.2014	15.11.2014
6	Häufigkeit	1	2	4
7	Basis	4	4	4
8	ZINSTERMVZ()	15.11.2012	15.05.2013	15.05.2013

Abbildung 3.60 Rückrechnen auf den letzten Zinstermin vor dem Kauf

ZINSTERMZAHL()

COUPNUM()

Syntax: `ZINSTERMZAHL(`**`Abrechnung; Fälligkeit; Häufigkeit`**`; Basis)`

Beispiel: `=ZINSTERMZAHL(DATUM(2013;02;25); DATUM(2016;11;15);4;4)`

Ergebnis: `15`

Die Funktion `ZINSTERMZAHL()` berechnet bei einem festverzinslichen Wertpapier die Zahl der Zinstermine zwischen dem Kaufdatum und dem Fälligkeitsdatum.

Dabei ist `Abrechnung` der Kauftermin, `Fälligkeit` der Termin, an dem die Rückzahlung erfolgt. Beide Argumente verlangen gültige Datumswerte. Sie können als serielle Werte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann ein solches Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Mit `Häufigkeit` wird angegeben, wie oft die Zinsen im Jahr fällig werden. Zulässige Werte sind: `1` – jährlich, `2` – halbjährlich, `4` – vierteljährlich. `Basis` gibt an, nach welchem Zeitsystem die Zinstage berechnet werden. Die möglichen Werte sind in Abschnitt 3.6, »Funktionen für Wertpapierberechnungen«, beschrieben.

	A	B	C	D
1				
2	**Zinszahlungstermine zwischen Kauf und Fälligkeit**			
3				
4	Abrechnung	25.02.2013	25.02.2013	25.02.2013
5	Fälligkeit	15.11.2016	15.11.2016	15.11.2016
6	Häufigkeit	1	2	4
7	Basis	4	4	4
8	ZINSTERMZAHL()	4	8	15

Abbildung 3.61 Berechnen der Anzahl der Zinszahlungstermine

Ungültige Datumswerte führen zu dem Fehler `#WERT!`. Ungültige Werte für `Häufigkeit` oder `Basis` ergeben den Fehler `#ZAHL!`.

ZINSZ()

IPMT()

Syntax: `ZINSZ(Zins; Zr; Zzr; Bw; Zw; F)`
Beispiel: `=ZINSZ(0,08;5;6;20000)`
Ergebnis: `-617,20`

Die Funktion `ZINSZ()` ermittelt den mit jeder Periode abnehmenden Zinsanteil für die Abzahlung eines Annuitätendarlehens bei gleich bleibenden Zahlungen und bei gleich bleibendem Zinssatz (vergleiche `KAPZ()` und `RMZ()`). Für die erste Periode wird der Zinsanteil einfach durch Multiplikation der Darlehenssumme mit dem Zinssatz errechnet, in den folgenden Perioden bezieht sich die Zinsberechnung dann jeweils auf die Restschuld, die nach Abzug der Tilgung übrig bleibt.

Argumente sind `Zins` für den konstanten Nominalzinssatz, `Zr` für die Angabe der Periode, für die der Zinsanteil berechnet werden soll, `Zzr` für die Anzahl der Abzahlungszeiträume und `Bw`, also der Barwert, der bei einem Darlehen der Darlehenssumme entspricht.

Dazu können Sie noch `Zw` angeben für einen Zukunftswert, also einen Betrag, der am Ende der letzten Zahlung noch an der vollständigen Rückzahlung eines Darlehens fehlt. Ist dieser Wert nicht angegeben, wird er als `0` angenommen. Das Argument `F` für Fälligkeit legt fest, ob die Berechnung für vorschüssige – `1` – oder nachschüssige Zinszahlungen – `0` – erfolgen soll.

Alle Werte müssen numerisch sein, sonst liefert die Funktion den Fehler `#WERT!`. `Zr` und `Zzr` müssen > `0` und `Zr` darf nicht > `Zzr` sein, sonst erscheint der Fehler `#ZAHL!`.

	A	B	C	D	E	F	G
1							
2	**Zinsanteil bei der Abzahlung von Darlehen**						
3							
4	Zins	7,00%	7,00%	7,00%	7,00%	7,00%	7,00%
5	Zr	1	2	3	4	5	6
6	Zzr	6	6	6	6	6	6
7	Bw	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €	20.000 €
8	ZINSZ()	-1.400,00 €	-1.204,29 €	-994,87 €	-770,80 €	-531,04 €	-274,50 €

Abbildung 3.62 Berechnen der Zinsen bei einem Darlehen

ZSATZINVEST()

RRI()

Syntax: `ZSATZINVEST(Zzr; Bw; Zw)`

Beispiel: `=ZSATZINVEST(10;10000;15000)`

Ergebnis: `0,04138`

Die in Excel 2013 neue Funktion `ZSATZINVEST()` berechnet den effektiven Jahreszins für den Wertzuwachs einer Investition. Dabei wird angenommen, dass nach der ersten Zahlung keine Zuzahlungen oder Entnahmen erfolgen.

Alle drei Argumente sind erforderlich. Mit `Zzr` wird die Anzahl der Perioden angegeben, `Bw` gibt den aktuellen Wert der Investition, `Zw` den angestrebten Zukunftswert an.

Die Funktion arbeitet mit folgender Gleichung:

```
ZSATZINVEST = (Zw / Bw)^(1 / Zrr) - 1
```


Alle Argumente erfordern, anders als bei verwandten Funktionen wie `BW()` oder `ZW()`, positive numerische Werte. Ist das nicht der Fall, liefert die Funktion den Fehlerwert `#ZAHL!`. Ist ein Argument nicht numerisch, ist der Fehlerwert `#WERT!`.

	A	B	C	D	E
1					
2	**ZSATZINVEST()**				
3					
4	Zzr	10	10	10	10
5	Bw	100000	100000	100000	100000
6	Zw	120000	130000	140000	150000
7	ZSATZINVEST()	1,84%	2,66%	3,42%	4,14%

Abbildung 3.63 Berechnen des effektiven Jahreszinses für den Wertzuwachs einer Investition

Die Funktion liefert das gleiche Ergebnis wie die Funktion `ZINS()`, wenn dort das Argument `Rmz` auf `0` gesetzt und entweder der Barwert oder der Zukunftswert mit negativem Vorzeichen angegeben ist.

ZW()

FV()

Syntax:	ZW(**Zins**; **Zzr**; **Rmz**; Bw; F)
Beispiel:	=ZW(0,06;10;-1000;;1)
Ergebnis:	13.971,64

Die Funktion ZW() liefert als Ergebnis den zukünftigen Wert auf der Grundlage gleicher Zahlungen zu gleichem Zinssatz über die angegebene Laufzeit.

Mit Zins wird der Zinssatz und mit Zzr werden die Zahlungszeiträume – Perioden – und damit die Anzahl der Zahlungen angegeben. Dabei ist darauf zu achten, dass jeweils die passenden Zeiteinheiten verwendet werden. Als Vorgabe wird Zins als Jahreszins interpretiert. Soll mit einer halbjährlichen, vierteljährlichen oder monatlichen Verzinsung gearbeitet werden, muss die Anzahl der Jahreszeiträume mit 2, 4 oder 12 multipliziert, der Jahreszinssatz entsprechend mit 2, 4 oder 12 dividiert werden.

Rmz ist der Betrag, der periodisch gezahlt wird. Dieser Betrag ist, wenn er angegeben wird, während der ganzen Laufzeit konstant. Handelt es sich um eine Rente, ist dies der ausgezahlte Betrag, handelt es sich um ein Darlehen, ist dies der Betrag der Annuität, die jährlich gezahlt wird. Zusätzlich kann mit Bw noch ein Barwert angegeben werden, der zu Beginn in Form eines Guthabens oder als Darlehenssumme vorhanden ist. Wird Bw nicht angegeben, so wird er als 0 angenommen.

Die Funktion benötigt, wenn Bw nicht angegeben wird, unbedingt einen Wert für Rmz. Umgekehrt kann aber auch ohne einen Wert für Rmz gerechnet werden, wenn Bw angegeben ist. Mit F wird noch angegeben, ob die Zahlungen jeweils am Ende einer Periode erfolgen (F = 0 oder weggelassen) oder am Anfang (F = 1).

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler #WERT!.

Die Berechnung des Zukunftswerts basiert auf der in Abschnitt 3.3, »Häufig benötigte Argumente und ihr Zusammenhang«, bereits vorgestellten Grundformel:

```
Bw * (1+Zins)^Zzr + Rmz * (1+Zins * F) * (((1+Zins)^Zzr)-1 / Zins) + Zw = 0
```

3.7.5 Zukünftiger Wert regelmäßiger Zahlungen

Abbildung 3.64 zeigt die Berechnung des Zukunftswerts jährlicher Zahlungen bei unterschiedlichen Zinssätzen. Der Betrag für Rmz wird negativ eingefügt, da Sie ja diesen Be-

trag pro Periode zahlen. Das Ergebnis dagegen ist positiv, denn das ist der Betrag, der Ihnen am Ende ausgezahlt wird.

	A	B	C	D
1				
2	**Zukunftswert von regelmäßigen Zahlungen**			
3				
4	Zins	3%	2,5%	
5	Zzr	10	10	
6	Rmz	- 1.000,00 €	- 1.000,00 €	
7	Bw			
8	F	1	1	
9	ZW()	11.807,80 €	11.483,47 €	
10				

Abbildung 3.64 Berechnen des zukünftigen Werts auf der Basis regelmäßiger Zahlungen

3.7.6 Zukünftiger Wert einer einmaligen Einzahlung

Anstelle regelmäßiger Zahlungen kann auch ein einmaliger Betrag für eine Anzahl von Perioden angelegt werden. Die Frage stellt sich dann, wie schnell das Kapital wächst (oder wie langsam, wenn wir die heute leider üblichen Sparzinsen ansehen). Das Beispiel in Abbildung 3.65 zeigt eine entsprechende Berechnung für zwei verschiedene Zinssätze.

	A	B	C	D
1				
2	**Zukunftswert einer Einmalzahlung**			
3				
4	Zins	3%	2,5%	
5	Zzr	10	10	
6	Rmz			
7	Bw	- 20.000,00 €	- 20.000,00 €	
8	F	1	1	
9	ZW()	26.878,33 €	25.601,69 €	
10				

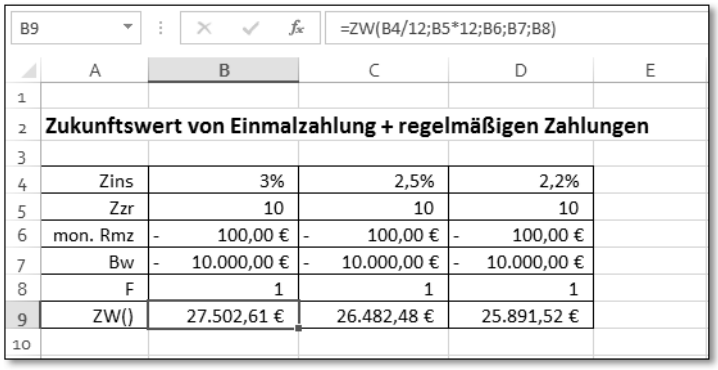
Abbildung 3.65 Berechnen des zukünftigen Werts bei einer Einmalzahlung

3.7.7 Zukunftswert bei regelmäßigen Zahlungen und Einmalzahlung

Die Funktion kann ohne Weiteres auch den Fall abdecken, dass Sie zunächst eine Einmalzahlung tätigen und anschließend regelmäßige Zahlungen folgen lassen. Dafür ge-

ben Sie sowohl einen Betrag für das Argument `Rmz` also auch für `Bw` an. Beide Werte werden als negative Werte angegeben.

Abbildung 3.66 zeigt das Ergebnis wieder für verschiedene Zinssätze an. Diesmal sind die Zahlungen als Monatszahlungen eingetragen, weshalb der Zinssatz durch 12 dividiert und die Periodenzahl mit 12 multipliziert werden muss.



	A	B	C	D	E
1					
2	**Zukunftswert von Einmalzahlung + regelmäßigen Zahlungen**				
3					
4	Zins	3%	2,5%	2,2%	
5	Zzr	10	10	10	
6	mon. Rmz	- 100,00 €	- 100,00 €	- 100,00 €	
7	Bw	- 10.000,00 €	- 10.000,00 €	- 10.000,00 €	
8	F	1	1	1	
9	ZW()	27.502,61 €	26.482,48 €	25.891,52 €	
10					

Abbildung 3.66 Berechnen des zukünftigen Werts bei Einmalzahlung und regelmäßigen Zuzahlungen

ZW2()

FVSCHEDULE()

Syntax: `ZW2(`**`Kapital; Zinsen`**`)`
Beispiel: `=ZW2(10000;{0,03;0,05;0,03;0,08})`
Ergebnis: `12031`

Die Funktion `ZW2()` berechnet den Endwert eines Kapitals, das über mehrere Jahre mit jährlich wechselnden Zinsen nachschüssig verzinst wird. Die Zinsen werden jedes Mal zum Kapital geschlagen, so dass Zinseszinsen anfallen.

`Kapital` ist das zum Anfang eingesetzte Kapital, `Zinsen` sind die jeweiligen Zinssätze, die in der Reihenfolge, in der sie tatsächlich anfallen, in eine Spalte oder Zeile eingetragen werden müssen. Dabei werden leere Zellen als `0` gewertet. Stattdessen kann auch eine entsprechende Matrixkonstante angegeben werden. Implizit werden die Jahre aus der Zahl der eingegebenen Zinssätze errechnet.

Nicht numerische Argumente führen zu dem Fehler `#WERT!`.

In Abbildung 3.67 sind in der Spalte A vier verschiedene Jahreszinssätze angegeben, das Anfangskapitel wird also über vier Jahre nachschüssig verzinst. In Spalte B wird zur Kontrolle das jährliche Wachstum des Kapitals mit Hilfe der Funktion

```
=ZW(Jahreszins;1;;-Kapital am Anfang der Periode)
```

berechnet.

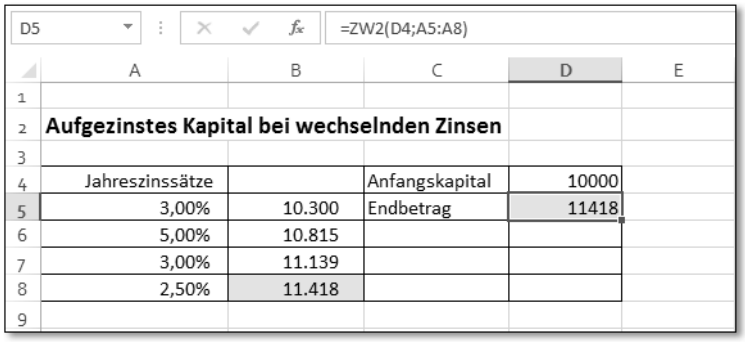


Abbildung 3.67 Verzinsung mit wechselnden Zinssätzen

ZZR()

NPER()

Syntax: `ZZR(`**`Zins; Rmz; Bw`**`; Zw; F)`

Beispiel: `=ZZR(5%;-1000;10000)`

Ergebnis: `14,2`

Die Funktion `ZZR()` berechnet die Zahl der Zahlungsperioden, also die Laufzeit. Sie kann bei der Zinseszinsrechnung, bei der Rentenrechnung und bei der Tilgungsrechnung bei einem Darlehen eingesetzt werden, wenn die anderen Argumente bekannt sind.

Mit `Zins` wird der Zinssatz der Periode angegeben, wobei von einer nachschüssigen Verzinsung ausgegangen wird. Handelt es sich um Monatsperioden, muss der Wert für den Jahreszins durch 12 dividiert werden.

`Rmz` ist der Betrag, der periodisch gezahlt wird. Dieser Betrag ist, wenn er angegeben wird, während der ganzen Laufzeit konstant. Handelt es sich um eine Rente, ist dies der ausgezahlte Betrag, handelt es sich um ein Darlehen, ist dies der Betrag der Annuität, die jährlich gezahlt wird, also die Summe aus dem Tilgungs- und dem Zinsanteil.

Zusätzlich kann mit `Bw` noch ein Barwert angegeben werden, der zu Beginn in Form eines Guthabens oder als Darlehenssumme vorhanden ist. Wird `Bw` nicht angegeben, so wird er als `0` angenommen. Außerdem kann mit `Zw` ein Zukunftswert angegeben werden, der einen bestimmten Kontostand am Ende von Rentenzahlungen oder eine Restschuld bei einem Darlehen angibt.

Die Funktion benötigt immer wenigstens zwei der drei Argumente `Rmz`, `Bw` und `Zw`. Mit `F` wird noch angegeben, ob die Zahlungen jeweils am Ende einer Periode erfolgen (`F = 0` oder weggelassen) oder am Anfang (`F = 1`).

Alle Werte müssen numerische Werte sein, sonst liefert die Funktion den Fehler `#WERT!`. `Rmz`, `Bw` und `Zw` dürfen nicht alle das gleiche Vorzeichen haben, sonst erscheint der Fehler `#ZAHL!`.

Die Berechnung von `ZZR()` basiert auf der in Abschnitt 3.3, »Häufig benötigte Argumente und ihr Zusammenhang«, bereits vorgestellten Grundformel:

```
Bw * (1+Zins)^Zzr + Rmz * (1+Zins * F) * (((1+Zins)^Zzr)-1 / Zins) + Zw = 0
```

3.7.8 Zinsperioden bei Einmalzahlung

Abbildung 3.68 zeigt in Spalte B, dass eine einmalige Zahlung von 10.000 € nach 23 Jahren bei einer Verzinsung von 3% einen Endwert von 20.000 € erreicht. In Spalte C wird eine entsprechende Berechnung für einen Jahreszinssatz von 2,5% ausgeführt.

	A	B	C	D
1				
2	**Berechnen der Zahlungsperioden**			
3				
4	Zins	3%	2,5%	
5	Rmz			
6	Bw	- 10.000,00 €	- 10.000,00 €	
7	Zw	20.000,00 €	20.000,00 €	
8	F	0	0	
9	ZZR()	23	28	
10				

Abbildung 3.68 Berechnen der jährlichen Zahlungsperioden bei unterschiedlichen Zinssätzen

3.7.9 Zahlungsperioden bei regelmäßigen Zahlungen

Ein anderer Fall, der sich mit `ZZR()` berechnen lässt: Sie haben sich eine Lebensversicherung über 100.000 € auszahlen lassen und bei der Bank angelegt. Sie wollen wissen, wie viele Jahre daraus eine monatliche Zusatzrente von 1.000 € abgezweigt werden kann.

Der Wert für RMZ() wird als negativer Wert angegeben, da dieses Geld ja an Sie ausgezahlt wird. Die 100.000 € werden als Barwert zu Beginn der Zahlungsperioden angegeben. Da Sie die Anzahl der Monate interessiert, lautet die Formel in B9:

```
= ZZR(B4/12;B5;B6;;B8)
```


	A	B	C	D
1				
2	**Berechnen möglicher Rentenzahlungen**			
3				
4	Zins	3,0%	2,5%	
5	Rmz	- 1.000,00 €	- 1.000,00 €	
6	Bw	100.000,00 €	100.000,00 €	
7	Zw			
8	F	0	0	
9	ZZR()	115	112	
10				

Abbildung 3.69 Berechnen möglicher Rentenzahlungen

3.7.10 Geld liegen lassen

Ein drittes Beispiel soll den Fall behandeln, dass Sie eine größere Summe bei der Bank deponieren wollen, um nach einigen Jahren davon Ihre Traumreise zu finanzieren. In diesem Fall geben Sie die einmalige Zahlung als Barwert an und den vermuteten Preis der Reise als Zukunftswert. Wie lange müssen Sie nun warten, bis das Geld genügend gewachsen ist?

In diesem Fall hängt alles vom Zinssatz ab. In Abbildung 3.70 sind die Ergebnisse der Funktion auf volle Jahre gerundet. Wenn Sie zur Kontrolle noch einmal mit der ZW()-Funktion nachrechnen, erhalten Sie für

```
=ZW(B4;B9;;-10000)
```

den Wert 15000.

	A	B	C	D
1				
2	**Berechnen der Wartezeit**			
3				
4	Zins	3%	2,5%	
5	Rmz			
6	Bw	- 10.000,00 €	- 10.000,00 €	
7	Zw	15.000,00 €	15.000,00 €	
8	F	0	0	
9	ZZR()	14	16	
10				

Abbildung 3.70 Berechnen der Wartezeit bis zum Erreichen eines Zielwerts

4 Datums- und Zeitfunktionen

Funktion	Seite	Funktion	Seite
ARBEITSTAG()	269	MONAT()	286
ARBEITSTAG.INTL()	271	MONATSENDE()	288
BRTEILJAHRE()	273	NETTOARBEITSTAGE()	289
DATEDIF()	274	NETTOARBEITSTAGE.INTL()	290
DATUM()	275	SEKUNDE()	292
DATWERT()	277	STUNDE()	293
EDATUM()	278	TAG()	294
HEUTE()	279	TAGE()	294
ISOKALENDERWOCHE()	280	TAGE360()	295
JAHR()	281	WOCHENTAG()	297
JETZT()	282	ZEIT()	299
KALENDERWOCHE()	283	ZEITWERT()	300
MINUTE()	286		

4.1 Einsatzbereich der Datums- und Zeitfunktionen

Berechnungen, in denen Datums- und Zeitwerte verwendet werden, spielen in vielen Anwendungsgebieten von Excel eine wichtige Rolle. Bei kommerziellen Lösungen geht es häufig darum, Fälligkeiten von Zahlungen festzustellen oder die Zeiträume zu berechnen, für die Zinsen anfallen. Bei der Analyse der gewerblichen Abläufe werden Umsätze, Kosten oder Gewinne bestimmten Zeitintervallen zugeordnet. Im Personalwesen werden Leistungen nach der Zeit abgerechnet, Urlaubsberechtigungen kontrolliert etc. In der Produktionsplanung kommt es auf die korrekte Verteilung der anfallenden Produktionsabläufe auf entsprechende Zeiteinheiten an.

4.1.1 Serielle Datums- und Zeitwerte

Excel rechnet in den Datums- und Zeitfunktionen mit »seriellen Zahlen«, die auch als *fortlaufende Zahl* bezeichnet werden. Alle Datumsangaben beziehen sich normalerweise auf den 1.1.1900. Dieser Tag entspricht der seriellen Zahl 1, der darauf folgende

Tag wird durch die serielle Zahl 2 repräsentiert usw., bis zur Zahl 2958465, die für den 31.12.9999 steht.

Alternativ kann noch eine andere Zählung der Datumswerte verwendet werden, die der auf dem Mac üblichen Zählung entspricht und den Austausch von Arbeitsmappen mit dem Mac vereinfacht. Sie geht von dem Zeitpunkt 1.1.1904 als dem Tag 0 aus. Mit diesem späteren Einstieg in die Zählung hat Apple das Problem vermieden, dass das Jahr 1900 kein Schaltjahr war, was bei der Zählung ab 1.1.1900 unterstellt wird. Das falsche Datum 29.2.1900 kann bei dieser Zählung eingegeben werden, obwohl es diesen Tag nicht gegeben hat. (Erst mit dem neuen Dateiformat *Strict Open XML* wird dieser Fehler vermieden.)

Seit Excel 2007 können Sie eine Einstellung auf das alternative Datumsformat über **Excel-Optionen • Erweitert • Beim Berechnen dieser Arbeitsmappe** vornehmen, indem Sie das Kontrollkästchen **1904-Datumswerte verwenden** ankreuzen. In den älteren Versionen finden Sie eine entsprechende Option über **Extras • Optionen • Berechnung**. Die Verwendung der Zählung ab 1904 ist auch unter Windows nützlich, wenn mit negativen Werten für Zeiten gerechnet werden soll, wie weiter unten im Zusammenhang mit der Arbeitszeitberechnung noch gezeigt wird.

Uhrzeiten werden in Excel auch als serielle Zahlen dargestellt, alle Zeitangaben werden durch Werte zwischen 0 und 1 repräsentiert, und zwar ist 0,00001 (gerundet) die erste Sekunde des Tages, und 0,5 entspricht 12 Uhr mittags. Werden Zeitwerte mit einem Datumswert kombiniert, bilden sie den Dezimalteil der seriellen Zahl: 41422,5 entspricht also 12 Uhr mittags am 28. Mai 2013.

Durch die Umwandlung in serielle Zahlen kann mit Datums- und Zeitangaben sehr einfach gerechnet werden. Da diese sich nun intern als normale numerische Werte darstellen, können mit ihnen die üblichen Rechenoperationen vorgenommen werden, vornehmlich Subtraktionen und Additionen können so ausgeführt werden.

Allerdings ist für den Benutzer nicht unbedingt sichtbar, dass die Datums- und Zeitfunktionen mit seriellen Zahlen rechnen, denn für die Ausgabe erscheint das Ergebnis zumeist schon in einem entsprechenden Datums- oder Zeitformat. Um die Ausgabe als serielle Zahl zu sehen, braucht aber nur das Standardformat aktiviert zu werden, beispielsweise durch Löschen des zugewiesenen Datums- und Zeitformats.

4.1.2 Die Rolle der Datums- und Zeitformate

Bei der Eingabe von Datums- und Zeitwerten in eine Zelle werden alle Zeichenfolgen, die ein gültiges Datums- oder Zeitformat darstellen, korrekt interpretiert. Galt für die

Zelle vorher das Standardformat, gilt nach der Eingabe das entsprechende Datums- oder Zeitformat. Mehr zu diesen Formaten finden Sie im Abschnitt »Datums- und Zeitformate« in Kapitel 1, »Einstieg in Berechnungen mit Excel«.

Die Möglichkeit, Datums- und Zeitwerte als Zeichenfolgen einzugeben, besteht auch, wenn sie als Argumente für Funktionen eingegeben werden. Nötig ist dann nur, dass eine solche Zeichenfolge einem gültigen Datums- bzw. Zeitformat entspricht. Excel erkennt unterschiedliche Schreibweisen bei Datumswerten:


```
11.12.2013
11-12-2013
11/12/2013
```

werden alle gleichermaßen akzeptiert.

Dennoch empfiehlt die Hilfe zu Excel, solche Argumente möglichst in Form von Funktionen anzugeben, die korrekte Datums- und Zeitwerte liefern. Dieser Empfehlung wird in den folgenden Beispielen gefolgt.

Zwei Funktionen in dieser Kategorie verwenden die Systemzeit Ihres Geräts: `HEUTE()` und `JETZT()`. Solange die automatische Neuberechnung eingestellt ist, werden die Ergebnisse dieser Funktionen ständig aktualisiert. Ob sie korrekte Werte liefern, hängt allerdings davon ab, dass Sie die Systemzeit korrekt eingestellt haben.

Beim Formatieren der Ergebnisse von Datums- und Zeitfunktionen sollte darauf geachtet werden, dass nicht aus Versehen ein falsches Format zugewiesen wird und damit fehlerhafte Ergebnisse erzeugt werden.

Bei der Eingabe der Funktion `=JAHR(HEUTE())` im Jahr 2013 liefert Excel das korrekte Ergebnis 2013. Wird die betreffende Zelle jedoch irrtümlich mit einem Datumsformat belegt, ergibt sich der 05.07.1905, der zweitausenddreizehnte Tag seit dem 1.1.1900. Das liegt daran, dass die Funktion `JAHR()` eine Zahl für das Jahr liefert und keinen Datumswert.

4.2 Periodische Datumsreihen berechnen

Im folgenden Beispiel wird gezeigt, wie mit der Funktion `DATUM()` Datumsberechnungen angestellt werden können. Es sollen periodische Datumsreihen beliebiger Art berechnet werden. Sie brauchen dazu nur ein Ausgangsdatum einzugeben und das gewünschte Intervall in Tagen, Wochen oder Monaten. In der ersten Zeile der Tabelle mit den Datumsreihen wird einfach nur das Ausgangsdatum durch einen absoluten Bezug übernommen.

Die einzelnen Termine der Datumsreihe sind in Spalte A fortlaufend nummeriert. Auf diese Nummer nehmen die Formeln der Reihe Bezug. Die Nummerierung kann durch Ziehen des Ausfüllkästchens bei gedrückter Strg-Taste oder mit dem Befehl **Start • Bearbeiten • Füllbereich • Reihe** erzeugt werden.

Ab der zweiten Zeile werden Formeln verwendet, um den Zeitsprung jeweils zu berechnen. Diese Formeln müssen nur einmal eingegeben und können dann nach unten kopiert werden.

Die Formel für ein Tagesintervall lautet:

```
=DATUM(JAHR($A$6);MONAT($A$6);TAG($A$6)+(A10-1)*$B$6)
```

Vergessen Sie nicht, die absoluten Bezüge zu verwenden, wenn Sie das Beispiel nachvollziehen wollen.

Die Formeln für das Monatsintervall sind entsprechend. Es wird immer ein bestimmtes Vielfaches des angegebenen Intervalls zu dem entsprechenden Bestandteil des Datums addiert:

```
=DATUM(JAHR($A$6);MONAT($A$6)+(A10-1)*$D$6;TAG($A$6))
```

	A	B	C	D
1				
2	**Periodische Datumsreihen**			
3				
4		**Intervall**		
5	**Ausgangsdatum**	**in Tagen**	**in Wochen**	**in Monaten**
6	05.05.2013	14	2	3
7				
8	**Termin**			
9	1	05.05.2013	05.05.2013	05.05.2013
10	2	19.05.2013	19.05.2013	05.08.2013
11	3	02.06.2013	02.06.2013	05.11.2013
12	4	16.06.2013	16.06.2013	05.02.2014
13	5	30.06.2013	30.06.2013	05.05.2014
14	6	14.07.2013	14.07.2013	05.08.2014
15	7	28.07.2013	28.07.2013	05.11.2014
16	8	11.08.2013	11.08.2013	05.02.2015
17	9	25.08.2013	25.08.2013	05.05.2015
18	10	08.09.2013	08.09.2013	05.08.2015
19	11	22.09.2013	22.09.2013	05.11.2015
20	12	06.10.2013	06.10.2013	05.02.2016
21	13	20.10.2013	20.10.2013	05.05.2016
22	14	03.11.2013	03.11.2013	05.08.2016
23	15	17.11.2013	17.11.2013	05.11.2016
24	16	01.12.2013	01.12.2013	05.02.2017

Abbildung 4.1 Datumsreihen berechnen

Etwas mehr Aufwand macht das Wochenintervall:

```
=DATUM(JAHR($A$6);MONAT($A$6);TAG($A$6)+(A10-1)*$C$6*7)
```

Ist die Tabelle einmal aufgebaut, können Sie in den Zellen B6 bis D6 beliebige Werte für das Intervall eingeben und erhalten sofort die gewünschte Reihe.

Es gibt bei dieser Tabelle allerdings ein Problem mit der Monatsspalte, wenn das Ausgangsdatum auf den 31. eines Monats fällt. Abbildung 4.2 zeigt, dass Excel in diesem Fall einen Termin, der drei Monate später liegt, immer dann auf den 1. des Folgemonats setzt, wenn der letzte Monat weniger als 31 Tage hat.

5	Ausgangsdatum	in Tagen	in Wochen	in Monaten
6	31.12.2013	14	2	3
7				
8	Termin			
9	1	31.12.2013	31.12.2013	31.12.2013
10	2	14.01.2014	14.01.2014	31.03.2014
11	3	28.01.2014	28.01.2014	01.07.2014
12	4	11.02.2014	11.02.2014	01.10.2014
13	5	25.02.2014	25.02.2014	31.12.2014
14	6	11.03.2014	11.03.2014	31.03.2015
15	7	25.03.2014	25.03.2014	01.07.2015

Abbildung 4.2 Addition der Monate beim Ausgang von einem Monat mit 31 Tagen

Intern errechnet Excel bei der Addition von 3 Monaten auf den 31. Januar das Datum 31. April; da es den nicht gibt, wird der 1. Mai ausgegeben.

Wenn Sie dieses Ergebnis vermeiden wollen und bei der Addition auf den letzten Tag des Monats immer auch den letzten Tag des neuen Monats sehen wollen, können Sie die oben verwendete Formel durch eine Formel ersetzen, die mit der Funktion `EDATUM()` arbeitet.

```
=EDATUM($A$6;(A10-1)*$D$6)
```

	A	B	C	D
5	Ausgangsdatum	in Tagen	in Wochen	in Monaten
6	31.01.2014	14	2	3
7				
8	Termin			
9	1	31.01.2014	31.01.2014	31.01.2014
10	2	14.02.2014	14.02.2014	30.04.2014
11	3	28.02.2014	28.02.2014	31.07.2014
12	4	14.03.2014	14.03.2014	31.10.2014
13	5	28.03.2014	28.03.2014	31.01.2015
14	6	11.04.2014	11.04.2014	30.04.2015

Abbildung 4.3 Die Addition der Monate mit der Funktion `EDATUM()`

4.3 Periodische Zeitreihen berechnen

In dem nächsten Beispiel werden, ähnlich wie oben in dem Beispiel zu Datumsreihen, periodische Zeitreihen beliebiger Art berechnet. Sie geben eine Ausgangszeit ein und dann das gewünschte Intervall in Stunden, Minuten oder Sekunden.

In der ersten Zeile der Tabelle mit den Zeitreihen wird einfach nur die Ausgangszeit durch einen absoluten Bezug übernommen. Die einzelnen Termine der Zeitreihe sind in Spalte A fortlaufend nummeriert. Auf diese Nummern nehmen die Formeln der Reihe Bezug. Die Nummerierung kann durch Ziehen des Ausfüllkästchens bei gedrückter Strg-Taste oder mit dem Befehl **Start • Bearbeiten • Füllbereich • Reihe** erzeugt werden.

	A	B	C	D
1				
2	**Periodische Zeitreihen**			
3				
4		**Intervall**		
5	**Ausgangszeit**	**in Stunden**	**in Minuten**	**in Sekunden**
6	9:43:43	6	30	30
7				
8	**Termin**			
9	1	9:43:43	9:43:43	9:43:43
10	2	15:43:43	10:13:43	9:44:13
11	3	21:43:43	10:43:43	9:44:43
12	4	3:43:43	11:13:43	9:45:13
13	5	9:43:43	11:43:43	9:45:43
14	6	15:43:43	12:13:43	9:46:13
15	7	21:43:43	12:43:43	9:46:43
16	8	3:43:43	13:13:43	9:47:13
17	9	9:43:43	13:43:43	9:47:43
18	10	15:43:43	14:13:43	9:48:13
19	11	21:43:43	14:43:43	9:48:43
20	12	3:43:43	15:13:43	9:49:13
21	13	9:43:43	15:43:43	9:49:43
22	14	15:43:43	16:13:43	9:50:13
23	15	21:43:43	16:43:43	9:50:43
24	16	3:43:43	17:13:43	9:51:13

Abbildung 4.4 Zeitreihen berechnen

In der zweiten Zeile werden Formeln verwendet, um jeweils den Zeitsprung zu berechnen. Diese Formeln müssen nur einmal eingegeben und können dann nach unten kopiert werden.

Die Formel für ein Stundenintervall lautet:

```
=ZEIT(STUNDE($A$6)+(A10-1)*$B$6;MINUTE($A$6);SEKUNDE($A$6))
```

Vergessen Sie nicht, die absoluten Bezüge zu verwenden, wenn Sie das Beispiel nachvollziehen wollen. Die Formeln für das Minuten- bzw. Sekundenintervall sind entsprechend. Es wird immer ein bestimmtes Vielfaches des angegebenen Intervalls auf den entsprechenden Bestandteil der Zeit – hier die Minuten oder die Sekunden – addiert.

Ist die Tabelle einmal aufgebaut, können Sie in den Zellen B6 bis D6 beliebige Werte für das Intervall eingeben und erhalten sofort die gewünschte Reihe.

4.4 Uhrzeit und Dauer

Das folgende Beispiel behandelt die Frage, wie Zeiten addiert und subtrahiert werden. Die Zeitfunktionen von Excel liefern ja normalerweise die Uhrzeit für einen bestimmten Zeitpunkt und nicht die Angabe über eine Dauer. Es ist aber kein Problem, Zeitdifferenzen zu ermitteln. Etwas trickreich ist dagegen die Addition von Zeiten.

Eine kleine Tabelle wurde aufgebaut, um Arbeitszeiten für eine Woche einzutragen. Es werden immer der Beginn und das Ende der Arbeitszeit in den Spalten B und C eingegeben. Dabei werden drei verschiedene Arbeitszeiten unterschieden: Normalarbeitszeit, Samstagsarbeitszeit und Sonntagsarbeitszeit.

Um die Differenz zu ermitteln, kann wieder mit der Zeitfunktion gearbeitet werden. Die Formel für den ersten Tag lautet:

```
=ZEIT(STUNDE(C5-B5)-1;MINUTE(C5-B5);)
```

D12 =SUMME(D5:D11)

	A	B	C	D	E	F
1						
2	**Arbeitszeitberechnung**				**Woche:**	21
3						
4		**Beginn**	**Ende**	**Normale Arbeitszeit**	**Samstags-arbeit**	**Sonntags-arbeit**
5	Montag, 21. Mai 2007	8:40	16:30	6:50		
6	Dienstag, 22. Mai 2007	8:30	16:30	7:00		
7	Mittwoch, 23. Mai 2007	12:30	21:30	8:00		
8	Donnerstag, 24. Mai 2007	14:30	21:30	6:00		
9	Freitag, 25. Mai 2007	8:30	16:30	7:00		
10	Samstag, 26. Mai 2007	8:30	12:30		4:00	
11	Sonntag, 27. Mai 2007	8:30	10:30			2:00
12				10:50	4:00	2:00
13	* eine Stunde Pause					

Abbildung 4.5 Die Zeitrechnung liefert eine »falsche« Summe.

Mit `-1` bei Stunden wird eine einstündige Pause berücksichtigt.

Wenn nun in der Tabelle die Summen für die drei Zeitarten gebildet werden, zeigt Excel bei normalem Datumsformat nicht die erwartete Zeitsumme von 34 Stunden und 50 Minuten an, sondern eine Uhrzeit: `10:50`. Da die Differenz zwischen dem angezeigten und dem erwarteten Wert 24 ist, wird erkennbar, dass Excel durchaus nicht falsch gerechnet hat, sondern lediglich das Ergebnis als die Uhrzeit interpretiert, die nach 34 Stunden und 50 Minuten erreicht wird.

Um in der Zellanzeige zu erreichen, dass auch eine Stundenzahl über 24 angezeigt wird, kann ein spezielles Format genutzt werden. Markieren Sie den Bereich D5 bis F12, benutzen Sie im Dialog **Zellen formatieren • Zahlen** die Kategorie **Benutzerdefiniert**. Geben Sie folgendes Format ein:

```
[h]:mm
```

Nun werden die Zeitsummen korrekt angezeigt.

	A	B	C	D	E	F
1						
2	**Arbeitszeitberechnung**				**Woche:**	21
3						
4		**Beginn**	**Ende**	**Normale Arbeitszeit**	**Samstags-arbeit**	**Sonntags-arbeit**
5	Montag, 21. Mai 2007	8:40	16:30	6:50		
6	Dienstag, 22. Mai 2007	8:30	16:30	7:00		
7	Mittwoch, 23. Mai 2007	12:30	21:30	8:00		
8	Donnerstag, 24. Mai 2007	14:30	21:30	6:00		
9	Freitag, 25. Mai 2007	8:30	16:30	7:00		
10	Samstag, 26. Mai 2007	8:30	12:30		4:00	
11	Sonntag, 27. Mai 2007	8:30	10:30			2:00
12				34:50	4:00	2:00
13	* eine Stunde Pause					

Abbildung 4.6 Das Tabellenblatt mit korrekter Zeitsummenanzeige

4.5 Tabellen für die Erfassung der Arbeitszeit

Das folgende Beispiel, in dem ganz unterschiedliche Funktionen für die Berechnung von Arbeitszeiten genutzt werden, zeigt eine komplexere Lösung zum Thema Arbeitszeitberechnung. Dabei geht es um eine Aufgabe, die in der Regel ein genaues Ermitteln von Über- und zu wenig geleisteten Stunden erfordert. Hier hilft ein geeignetes Tabellenblatt.

Damit auch mit negativen Zeitwerten gerechnet werden kann, wählen Sie die Registerkarte **Erweitert** unter **Excel-Optionen**. Aktivieren Sie das Kontrollkästchen **1904-Datumswerte verwenden**. Sie sollten dies möglichst vor der Eingabe von Datumswerten tun, weil bei nachträglicher Umstellung auf ein schon eingetragenes Datum immer 1492 Tage aufaddiert werden. Das Datumssystem, das von 1.1.1900 rechnet, kann zwar auch negative Zeitwerte berechnen, ist aber nicht in der Lage, das Ergebnis in einem Zeitformat anzuzeigen, die Zelle wird mit #-Zeichen gefüllt. Dies ist auch in Excel 2013 noch so.

4.5.1 Erfassen der Stammdaten

Beginnen Sie mit der Erstellung eines Stammdatenblatts. Dort werden folgende Informationen erfasst:

- Name, Personalnummer und Abteilung des Mitarbeiters
- Stundenübertrag aus dem Vorjahr
- die tägliche Arbeitszeit
- die jährlichen Feiertage

	A	B
1	**Stammdaten der Arbeitszeitberechnung**	
2		
3	Personal-Nr.:	111
4	Name:	Wilfried Meyer
5	Abteilung:	Buchhaltung
6	Positiver Stundensaldo aus dem Vorjahr:	5:00
7	Negativer Stundensaldo aus dem Vorjahr:	
8	Tägliche Arbeitszeit	8:00
9	Land	NRW
10	Datum	Feiertag
11	01.01.2013	Neujahr
12	29.03.2013	Karfreitag
13	01.04.2013	Ostermontag
14	01.05.2012	Tag der Arbeit
15	09.05.2013	Christi Himmelfahrt
16	20.05.2013	Pfingstmontag
17	30.05.2013	Fronleichnam
18	03.10.2013	Tag der Einheit
19	01.11.2013	Allerheiligen
20	25.12.2013	1. Weihnachtsfeiertag
21	26.12.2013	2. Weihnachtsfeiertag

Abbildung 4.7 Stammdatenblatt für die Arbeitszeitberechnung

1. Legen Sie ein Arbeitsblatt an, wie in Abbildung 4.7 zu sehen ist.
2. Falls ein positiver Überstundensaldo aus dem vergangenen Jahr vorhanden ist, können Sie diesen in die Zelle B6 eintragen. Sind noch Minusstunden aus dem Vorjahr zu berücksichtigen, erfassen Sie diese in Zelle B7.
3. In B8 geben Sie die tägliche Arbeitszeit (Wochenarbeitszeit / 5) ein.
4. Damit die Feiertage später bei der Ermittlung der geleisteten Arbeitsstunden automatisch berücksichtigt werden, können auch sie in das Arbeitsblatt »Stammdaten« eingetragen werden. Geben Sie die Feiertage und die zugehörige Bezeichnung in den Bereich A10 bis B25 ein. Da in den einzelnen Bundesländern unterschiedliche Feiertagsregelungen gelten, können Ihre Eingaben unter Umständen von denen der Abbildung 4.7 abweichen.

Markieren Sie den Bereich A11 bis A21, und weisen Sie diesem über **Formeln • Definierte Namen • Namen definieren** die Bezeichnung `Feiertage` zu. Vergeben Sie für die Zelle B8 den Namen `tägliche_Arbeitszeit`. In den älteren Versionen verwenden Sie hier **Einfügen • Namen • Definieren**.

4.5.2 Monatskalender erstellen

Auf den einzelnen Monatsblättern werden die eigentlichen Berechnungen durchgeführt. Hierfür wird für jeden Monat ein Kalenderblatt angelegt, in dem für jeden Tag die Arbeitszeit festgelegt wird. Beginnen Sie mit dem Januar.

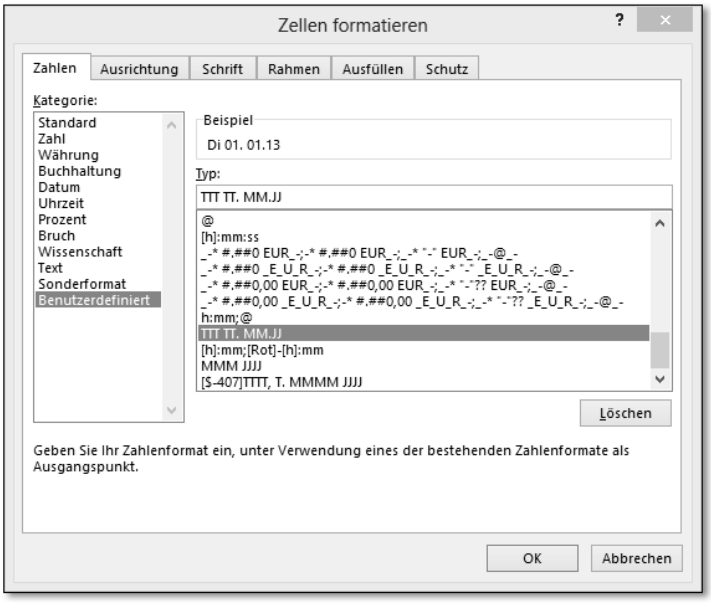
1. Erfassen Sie zunächst für den Monat Januar die Überschriften wie in der folgenden Abbildung:

	A	B	C	D	E	F	G	H	I
1	**Arbeitszeitermittlung**			**Pers.-Nr.**	111				
2				**Name**	Wilfried Meyer				
3				**Abteilung**	Buchhaltung				
4	**Datum**	**Arbeitstag**	**Sollstunden**	**Arbeitsbeginn**	**Arbeitsende**	**Pause**	**Iststunden**	**Mehr-/Minderstunden**	**Monatssaldo**

2. Sie benötigen für jeden Tag des Monats Januar eine eigene Zeile. Der erste Eintrag soll das Datum enthalten. Tippen Sie das Datum des ersten Januars in die Zelle A5 ein.
3. Markieren Sie den Bereich A5 bis A35.

4 Benutzen Sie dann **Start • Bearbeiten • Füllbereich • Reihe**. In den älteren Versionen ist das der Befehl **Bearbeiten • Ausfüllen • Reihe**.

5 Im abgebildeten Dialogfeld sehen Sie, dass Excel von sich aus die korrekten Optionen wie beispielsweise den Typ **Datum** ausgewählt hat. Wenn Sie die Auswahl bestätigen, trägt Excel alle Datumsangaben für den Januar ein.



6 Um ein handlicheres Format für den Wochentag zu erhalten, richten Sie ein benutzerdefiniertes Zahlenformat im Dialog **Zellen formatieren** ein. Rufen Sie unter **Kategorie** den Eintrag **Benutzerdefiniert** auf, und tippen Sie das Format `TTTTT.MM.JJ` ein.

Dieses Format kürzt den Wochentag in der Anzeige auf zwei Buchstaben ab und den Jahreswert ebenfalls auf zwei Stellen.

4.5.3 Arbeitstage und Arbeitszeiten berechnen

In Spalte B untersuchen Sie nun zuerst, ob es sich bei dem Datum in Spalte A um einen Arbeitstag oder einen freien Tag handelt. Das kann auf einfache Weise mit Hilfe der Funktion NETTOARBEITSTAGE() ermittelt werden. Für die Berechnung der Nettoarbeitstage werden die Feiertage des Abrechnungsjahres benötigt, die Sie bereits im Arbeitsblatt *Stammdaten* erfasst haben.

Die Funktion NETTOARBEITSTAGE() liefert die Anzahl ganzer Arbeitstage zwischen einem Anfangs- und Enddatum. Wochenenden werden automatisch abgezogen, ebenso die Tage, die als Feiertage beziehungsweise Ferien angegeben werden.

4.5.4 Nettoarbeitstage ermitteln

1. Wählen Sie die Zelle B5 aus, und benutzen Sie das Symbol **Funktion einfügen**.
2. Nehmen Sie als Kategorie die Option **Datum & Zeit**, und suchen Sie in der Liste die Funktion NETTOARBEITSTAGE().
3. Wenn Sie die Auswahl bestätigen, gelangen Sie zur Auswahl der Funktionsargumente.
4. Geben Sie sowohl unter **Ausgangsdatum** als auch unter **Enddatum** die Zelle A5 an. In das Textfeld **Freie_Tage** übernehmen Sie den vorhin vergebenen Bereichsnamen Feiertage. Damit geben Sie Excel alle zuvor erfassten Feiertage an. Bestätigen Sie die Eingaben.

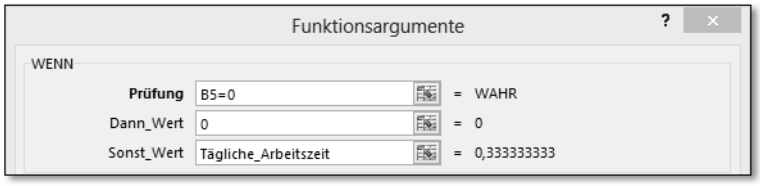
Wenn es sich um einen Arbeitstag handelt, gibt Excel die Ziffer 1 aus, ansonsten erscheint in der Zelle eine Null. Die Null steht also für Samstage sowie Sonn- und Feiertage.

4.5.5 Ermitteln der Sollarbeitsstunden

Durch Auswertung der Ergebnisse in Spalte B können Sie jetzt leicht die Sollstunden der täglichen Arbeitszeit ermitteln. Wenn ein Arbeitstag vorliegt, soll die tägliche Arbeitszeit erscheinen, ist dies nicht der Fall, beträgt die Anzahl der Sollstunden 0.

1. Wählen Sie die Zelle C5, und rufen Sie erneut **Funktion einfügen** auf.
2. Markieren Sie unter **Logik** die Funktion **Wenn**.

3 Geben Sie unter **Prüfung** die Bedingung `B5=0` ein.



4 Unter **Dann_Wert** geben Sie `0` ein. Das ist der Wert, der angezeigt werden soll, wenn die Bedingung erfüllt ist.

5 Unter **Sonst_Wert** geben Sie den Bereichsnamen `Tägliche_Arbeitszeit` an und bestätigen das Dialogfeld.

6 Markieren Sie die Zelle C5, und öffnen Sie das Dialogfeld **Zellen formatieren**.

7 Aktivieren Sie die Registerkarte **Benutzerdefiniert**, und weisen Sie der Zelle das Format `hh:mm` zu.

4.5.6 Stundenermittlung

In den folgenden Spalten werden die Angaben zu Arbeitsbeginn, Arbeitsende und Pause eingetragen und die Arbeitszeit sowie die Mehr- und Minderstunden ausgerechnet. Dazu müssen Sie die Anwesenheitszeit – Iststunden – und anschließend die täglichen Mehr- bzw. Minderstunden berechnen. Danach wird der fortlaufende Gesamtsaldo aller Stundensalden ausgerechnet.

Als Nächstes müssen die Formeln, die zur Ermittlung der Iststunden und der Stundensalden führen, eingegeben werden. Die Istzeit in der Zelle G5 ergibt sich durch die Formel `=WENN(ISTLEER(E5);0;E5-D5-F5)`.

Mehr- und Minderstunden in Zelle H5 erhalten Sie durch die Formel `=WENN(NICHT(ISTLEER(E5));G5-C5;0)`. Mit der Funktion `ISTLEER()` prüfen Sie, ob in E5 ein Eintrag vorhanden ist. Eine Berechnung erfolgt demnach nur dann, wenn das Arbeitsende in E5 erfasst worden ist. Nur dann wird der Saldo zwischen Anfangs-, Endzeit und Pausen errechnet.

Der aktuelle Überstundensaldo ergibt sich aus dem Saldo des ersten Tages sowie den Mehr- oder Minderstunden aus dem vergangenen Jahr. Dabei greifen Sie auf Zelladressen des Stammdatenblatts zu. Excel setzt dabei vor die Zelladresse den Blattnamen und trennt beide durch ein Ausrufezeichen.

4.5.7 Formeln für den Gesamtsaldo

Der Stundensaldo in Zelle I5 errechnet sich im Januar für den ersten Arbeitstag durch `=H5+Stammdaten!B6-Stammdaten!B7`. Der Überstundensaldo in I6 ergibt sich durch `=+H6+I5`, also den Stand des Vortages plus den aktuellen Tag.

Um das Blatt fertigzustellen, weisen Sie den Zellen G5, H5, I5 und I6 das benutzerdefinierte Zahlenformat `[h]:mm;[Rot]-[h]:mm` zu, um negative Zahlen in roter Farbe mit einem Minuszeichen darzustellen. Kopieren Sie die Formeln der Spalten B bis H bis in die Zeile 35. Auch die Zelle I6 kopieren Sie bis in Zeile 35.

4.5.8 Name, Personal-Nr. und Abteilung anzeigen

Damit Name, Personal-Nr. und Abteilung aus den Stammdaten auf den einzelnen Monatsblättern erscheinen, geben Sie noch folgende Formeln ein:

```
D1: =Stammdaten!B3
D2: =Stammdaten!B4
D3: =Stammdaten!B5
```

Da die Spalte B mit der Angabe der Nettoarbeitstage nur hilfsweise verwendet wurde, kann sie nachträglich auch ausgeblendet werden. Klicken oder tippen Sie in den Spaltenkopf der Spalte B, um diese zu markieren. Wählen Sie aus dem Kontextmenü des Spaltenkopfes den Befehl **Ausblenden**.

4.5.9 Anlegen der anderen Monatsblätter

Jetzt können Sie die weiteren Monate anlegen. Damit Sie den Wust von Formeln und Formatierungen nicht mehrfach erfassen müssen, kopieren Sie das Arbeitsblatt *Januar*.

1 Benutzen Sie im Kontextmenü des Blattregisters den Eintrag **Verschieben oder kopieren**.

2 Im Dialogfeld aktivieren Sie **Kopie erstellen** und wählen unter **Einfügen vor** die Option **(ans Ende stellen)**. Sie erhalten ein Blatt mit dem Namen »Januar(2)«. Benennen Sie es in »Februar« um.

3 Ersetzen Sie die Januar-Datumswerte durch Februar-Werte. Denken Sie daran, dass der Februar in der Regel nur 28 Tage hat.

4 Der erste Überstundensaldo im Februar ergibt sich durch `=Januar!I35+H5`. Für die übrigen Monate sind die Formeln entsprechend anzupassen.

4.5.10 Arbeiten im fertigen Arbeitsblatt

In den fertigen Tabellenarbeitsblättern können Sie oder Ihre Mitarbeiter jetzt Ihre täglichen Arbeitszeiten zuzüglich Pausen eintragen. Sollten Sie einmal so im Stress sein, dass Sie bis Mitternacht arbeiten, tragen Sie `24:00` und nicht `0:00` ein, da Sie ansonsten Fehlerwerte erhalten. Falls Sie einmal an Wochenenden oder Feiertagen arbeiten müssen, werden Ihnen diese Sonderschichten automatisch als positive Salden gutgeschrieben.

	A	B	C	D	E	F	G	H	I
1	**Arbeitszeitermittlung**			**Pers.-Nr.**	111				
2				**Name**	Wilfried Meyer				
3				**Abteilung**	Buchhaltung				
4	**Datum**	**Arbeitstag**	**Sollstunden**	**Arbeitsbeginn**	**Arbeitsende**	**Pause**	**Iststunden**	**Mehr-/Minderstunden**	**Monatssaldo**
5	Di 01. 01.13	0	00:00				0:00	0:00	5:00
6	Mi 02. 01.13	1	08:00	08:00	18:00	0:30	9:30	1:30	6:30
7	Do 03. 01.13	1	08:00	10:00	20:00	0:30	9:30	1:30	8:00
8	Fr 04. 01.13	1	08:00	09:00	15:00	1:00	5:00	-3:00	5:00
9	Sa 05. 01.13	0	00:00	09:00	18:15	1:00	8:15	8:15	13:15
10	So 06. 01.13	0	00:00				0:00	0:00	13:15
11	Mo 07. 01.13	1	08:00	09:15	19:00	1:00	8:45	0:45	14:00

Abbildung 4.8 Auszug aus einer ausgefüllten Monatstabelle

4.6 Referenz der Datums- und Zeitfunktionen

ARBEITSTAG()

WORKDAY()

Syntax: `ARBEITSTAG(`**`Ausgangsdatum; Tage;`** `Freie_Tage)`

Beispiel: `=ARBEITSTAG(DATWERT("01.01.2013");30)`

Ergebnis: `12.02.2013`

Die Funktion ARBEITSTAG() liefert von einem angegebenen Ausgangsdatum aus gerechnet ein neues Datum, das um die gegebene Zahl von Arbeitstagen vor oder nach dem Ausgangsdatum liegt. Dabei können zusätzlich zu den Wochenenden noch die angegebenen freien Tage berücksichtigt werden.

Ausgangsdatum kann als serieller Wert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann das Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist. Ist das nicht der Fall, liefert die Funktion den Fehler #WERT!. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler #ZAHL!.

Zu Ausgangsdatum rechnet die Funktion die mit Tage angegebene Anzahl von Arbeitstagen sowie die als Freie_Tage aufgeführten Tage hinzu, berücksichtigt die in den Zeitraum fallenden Samstage und Sonntage und gibt als Ergebnis das neue Datum aus. Freie_Tage kann ein Bezug auf einen Zellbereich sein, in dem die Daten der freien Tage abgelegt sind, oder auch eine Matrixkonstante mit seriellen Datumswerten oder Zeichenfolgen für Datumswerte. Die Werte der Matrixkonstanten müssen in geschweiften Klammern eingefasst werden, beispielsweise:

```
=ARBEITSTAG("20.12.2013";10;{"25.12.2013";"26.12.2013"})
```

Wird ein negativer Wert für Tage angegeben, rechnet die Funktion rückwärts. Hat der Wert Nachkommastellen, werden diese ohne Rundung einfach abgeschnitten.

Ist einer der Datumswerte kein gültiger Wert, liefert die Funktion den Fehler #WERT!. Ergibt die Addition von Ausgangsdatum + Tage ein ungültiges Datum, liefert die Funktion den Fehler #ZAHL!.

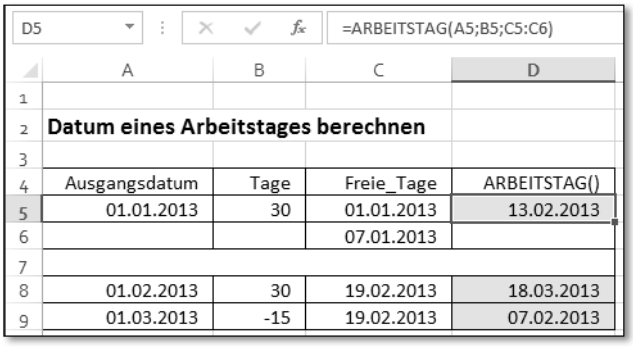


Abbildung 4.9 Kommende oder zurückliegende Arbeitstage berechnen

Die Funktion gibt Ihnen eine einfache Möglichkeit, beispielsweise zu berechnen, wann eine Zahlung fällig ist oder wann eine Lieferung zu erwarten ist, wenn eine entsprechende Anzahl von Arbeitstagen angegeben ist.

ARBEITSTAG.INTL()

WORKDAY.INTL()

Syntax:	ARBEITSTAG.INTL(**Ausgangsdatum**; **Tage**; Wochenende; Freie_Tage)
Beispiel:	=ARBEITSTAG.INTL(DATUM(2013;1;1); 30; 17; D5:D6)
Ergebnis:	07.02.2013, wenn in D5:D6 zwei Datumswerte für freie Tage eingetragen sind

Die Funktion ARBEITSTAG.INTL() liefert von einem mit Ausgangsdatum angegebenen Datum aus gerechnet ein neues Datum, das sich um die mit Tage angegebene Zahl von Arbeitstagen, die Wochenendtage und die mit Freie_Tage angegebene Zahl von freien Tagen vom Ausgangsdatum unterscheidet.

Ausgangsdatum kann als serieller Wert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann das Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist. Ist das nicht der Fall, liefert die Funktion den Fehler #WERT!. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler #ZAHL!.

Hat der Wert für Tage Nachkommastellen, werden diese ohne Rundung einfach abgeschnitten. Ist einer der Datumswerte kein gültiger Wert, liefert die Funktion den Fehler #WERT!. Ergibt die Addition von Ausgangsdatum + Tage ein ungültiges Datum, liefert die Funktion den Fehler #ZAHL!.

Freie_Tage kann ein Bezug auf einen Zellbereich sein, in dem die Daten der freien Tage abgelegt sind, oder auch eine Matrixkonstante mit seriellen Datumswerten oder Zeichenfolgen für Datumswerte. Die Werte der Matrixkonstanten müssen in geschweiften Klammern eingefasst werden.

Die Funktion erweitert die Funktion ARBEITSTAG() um das Argument Wochenende, wobei beliebige Wochentage als Wochenende behandelt werden können. Mit den Wochenendnummern wird angegeben, welche Tage als Wochenende genommen werden.

Nr.	Wochenendtage
1 (oder nicht angegeben)	Samstag, Sonntag
2	Sonntag, Montag
3	Montag, Dienstag
4	Dienstag, Mittwoch
5	Mittwoch, Donnerstag
6	Donnerstag, Freitag
7	Freitag, Samstag
11	nur Sonntag
12	nur Montag
13	nur Dienstag
14	nur Mittwoch
15	nur Donnerstag
16	nur Freitag
17	nur Samstag

Tabelle 4.1 Die möglichen Wochenendnummern und ihre Bedeutung

Statt der Nummern kann auch eine Zeichenfolge angegeben werden, bei der beginnend mit dem Montag `1` für Wochenendtage steht, `0` für Arbeitstage, etwa `0000011` für eine normale Woche mit dem Wochenende am Samstag und Sonntag.

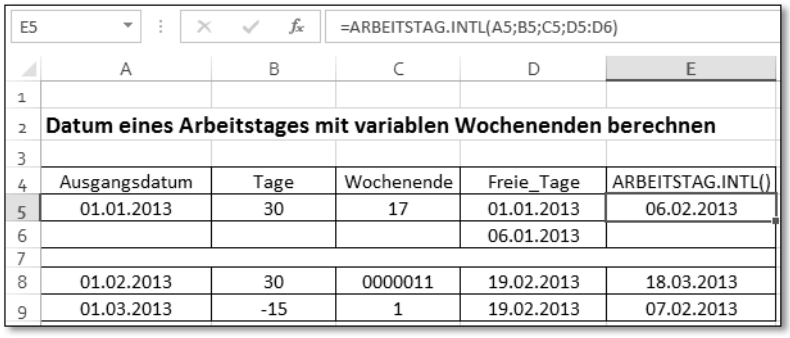


Abbildung 4.10 Kommende oder zurückliegende Arbeitstage mit variablem Wochenende berechnen

Siehe dazu auch die Funktion `NETTOARBEITSTAGE.INTL()`.

BRTEILJAHRE()

YEARFRAC()

Syntax: `BRTEILJAHRE(`**`Ausgangsdatum`**`; `**`Enddatum`**`; Basis)`

Beispiel: `=BRTEILJAHRE(DATWERT("01.02.2013"); DATWERT("12.11.2013");4)`

Ergebnis: `0,78`

Die Funktion `BRTEILJAHRE()` liefert die Zeitspanne zwischen einem `Ausgangsdatum` und einem `Enddatum` als Bruchteil von Jahren in Form einer Dezimalzahl. Das Ergebnis kann beispielsweise für Zinsberechnungen im unterjährigen Bereich verwendet werden.

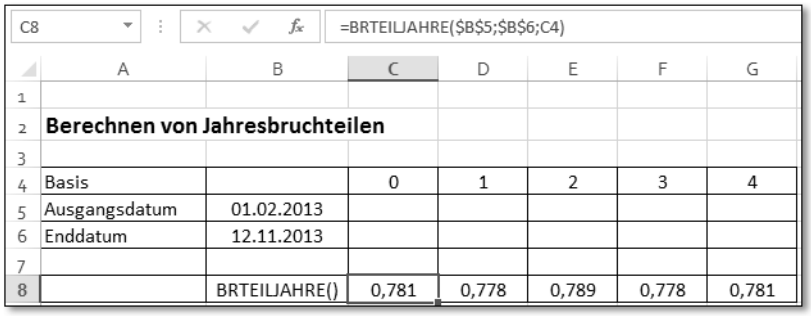
Mit `Basis` wird angegeben, wie das Jahr berechnet werden soll:

Basis	Bedeutung
0	US-amerikanisches System: Die Monate werden mit 30 Tagen, das Jahr wird mit 360 Tagen gerechnet. Ein Enddatum zum 31. eines Monats wird, falls das Startdatum vor dem 30. eines Monats liegt, als 1. Tag des Folgemonats behandelt, sonst als 30. des Monats.
1	Die tatsächlichen Tage der Monate und die tatsächlichen Tage der Jahre werden gerechnet.
2	Es wird mit den tatsächlichen Tagen des Monats gerechnet, für das Jahr werden aber 360 Tage angesetzt.
3	Es wird mit den tatsächlichen Tagen des Monats gerechnet, für das Jahr werden aber 365 Tage angesetzt.
4	Europäisches System: Die Monate werden mit 30 Tagen, das Jahr wird mit 360 Tagen gerechnet. Fällt ein Start- oder Enddatum auf den 31. eines Monats, wird es als 30. Tag behandelt.

Tabelle 4.2 Mögliche Werte für `Basis`

Die Unterschiede, die sich aus der Wahl von `Basis` ergeben, zeigt Abbildung 4.11.

Ist `Ausgangsdatum` oder `Enddatum` kein gültiger Datumswert, liefert die Funktion den Fehler `#WERT!`, ein ungültiger Wert für `Basis` führt zu dem Fehler `#ZAHL!`.



C8 | =BRTEILJAHRE($B$5;$B$6;C4)

	A	B	C	D	E	F	G
1							
2	**Berechnen von Jahresbruchteilen**						
3							
4	Basis		0	1	2	3	4
5	Ausgangsdatum	01.02.2013					
6	Enddatum	12.11.2013					
7							
8		BRTEILJAHRE()	0,781	0,778	0,789	0,778	0,781

Abbildung 4.11 Umrechnung in Jahresbruchteile

DATEDIF()

DATEDIF()

Syntax: `DATEDIF(`**`Ausgangsdatum; Enddatum; Einheit`**`)`

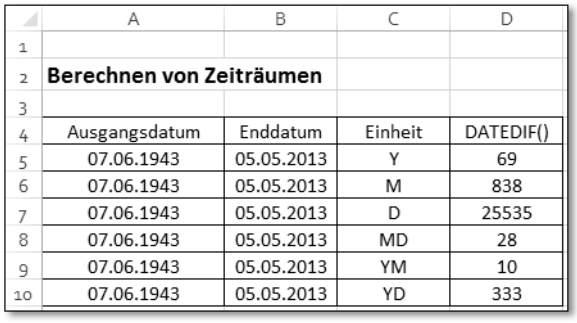
Beispiel: `=DATEDIF(DATUM(1943;6;7); DATUM(2013;6;23);"Y")`

Ergebnis: `70`

Die Funktion `DATEDIF()` berechnet den Zeitraum zwischen dem Argument `Ausgangsdatum` und dem Argument `Enddatum` in der gewählten `Einheit`. Diese wird mit Hilfe der in Abbildung 4.12 sichtbaren Abkürzungen in Spalte C angegeben.

Ist `Ausgangsdatum` oder `Enddatum` kein gültiger Datumswert, liefert die Funktion den Fehler `#WERT!`, ein ungültiger Wert für `Einheit` führt zu dem Fehler `#ZAHL!`.

Die Funktion wird von Microsoft seit mehreren Versionen »versteckt«, also weder im Funktions-Assistenten noch in der Hilfe angeboten, obwohl sie durchaus nützlich ist.



	A	B	C	D
1				
2	**Berechnen von Zeiträumen**			
3				
4	Ausgangsdatum	Enddatum	Einheit	DATEDIF()
5	07.06.1943	05.05.2013	Y	69
6	07.06.1943	05.05.2013	M	838
7	07.06.1943	05.05.2013	D	25535
8	07.06.1943	05.05.2013	MD	28
9	07.06.1943	05.05.2013	YM	10
10	07.06.1943	05.05.2013	YD	333

Abbildung 4.12 Eine seit Langem versteckte Funktion in Excel berechnet Zeiträume.

DATUM()

DATE()

Syntax: `DATUM(Jahr; Monat; Tag)`
Beispiel: `=DATUM(2013;1;1)`
Ergebnis: `01.01.2013`

Die Funktion `DATUM()` berechnet eine serielle Zahl für das mit Hilfe der einzelnen Argumente eingegebene Datum, das im Zeitraum vom 1.1.1900 bis zum 31.12.9999 liegen darf, wenn das vorgegebene 1900-Datumssystem verwendet wird. Die wiedergegebenen Werte liegen dann zwischen `1` und `2958465`, wenn die Zelle im Standardformat angezeigt wird. Excel formatiert das Ergebnis aber als Vorgabe mit dem über die Systemsteuerung als Standard gewählten Datumsformat.

Der Einsatz dieser Funktion ist besonders dann sinnvoll, wenn die einzelnen Argumente über Zellbezüge angegeben sind, so dass mit variablen Werten gearbeitet werden kann. Alle Argumente sind Zahlen oder Bezüge auf Zellen, die Zahlenwerte ergeben, oder Funktionen oder Formeln, die entsprechende Werte liefern.

```
=DATUM(JAHR(HEUTE());MONAT(HEUTE());TAG(HEUTE()+30))
```

ergibt beispielsweise ein Datum, das 30 Tage in der Zukunft des aktuellen Tages liegt. Allerdings ist die Formel unnötig verkompliziert.

```
=DATUM(HEUTE())+30
```

würde das gleiche Ergebnis liefern, weil die serielle Zahl, die die Funktion liefert, ja immer die Anzahl von Tagen angibt, auf die einfach addiert werden kann.

Die Werte für `Monat` und `Tag` dürfen auch größer als 12 bzw. 31 sein, die »überschüssigen« Monate oder Tage werden intern verrechnet. Auch Werte ≤ `0` sind erlaubt, Excel rechnet dann entsprechend zurück. Diese Möglichkeit lässt sich beispielsweise nutzen, um den letzten Tag eines Monats auszugeben.

```
=DATUM(2013;3;0)
```

liefert den letzten Februartag in 2013, weil Tag 0 einen Tag vor dem 1.3. liegt.

Ist der Wert für `Jahr` kleiner `0` oder größer `9999`, liefert die Funktion den Fehler `#ZAHL!`. Nicht numerische Argumente führen zu dem Fehler `#ZAHL!`.

	A	B	C	D
1				
2	**Ausgabe eines Datums**			
3				
4	Jahr	Monat	Tag	DATUM()
5	0	1	2	02.01.1900
6	43	6	7	07.06.1943
7	23	8	12	12.08.1923
8	2006	11	13	13.11.2006
9	2013	5	38	07.06.2013
10	2013	15	28	28.03.2014
11	9999	12	31	31.12.9999
12	10000	12	31	#ZAHL!

Abbildung 4.13 Die Funktion `DATUM()` kommt auch mit dem Monatswert `15` zurecht.

4.6.1 Besonderheiten der Funktion DATUM()

Anders als bei der direkten Eingabe eines Datums in eine Zelle verhält sich Excel, wenn das Datum durch eine Datumsfunktion erzeugt wird. Wird das Jahr bei `=DATUM(Jahr; Monat; Tag)` nur mit zwei Stellen eingegeben, ergänzt Excel nicht nach der in Abschnitt 1.5, »Die Rolle der Zahlenformate«, beschriebenen Zeitfensterregel, sondern addiert immer 1900 dazu. Das gilt für alle Eingaben für das Jahr, die von 0 bis 1899 reichen. Das Datum 10.10.2000 kann also auch angegeben werden durch `=DATUM(100;10;10)`.

Ein Datum für 2000 und darüber kann aber auch mit einer vierstelligen Jahreszahl eingegeben werden. Alle vierstelligen Eingaben von 1900 bis 9999 werden so verarbeitet. Wird das 1904-Datumssystem verwendet, gilt Entsprechendes für Zahlen von 4 bis 9999.

Für den Anwender ist dieses abweichende Verhalten sicher gewöhnungsbedürftig, die Funktion gibt Ihnen aber eine größere Flexibilität für Datumsberechnungen über größere Zeiträume hinweg.

Beachten Sie, dass es noch eine ganze Reihe von Funktionen in Excel gibt, die ein Datum als Argument verlangt. Ist dieses Argument als Datum in einer Zelle abgelegt, gilt auch hier die Fensterregel. Wird das Argument mit der Datumsfunktion erzeugt, gilt das zuletzt beschriebene Verhalten.

4.6.2 Datumsberechnungen in Makros

Ein besonderes Problem stellen selbstgestrickte Datumsberechnungen in Makros dar. Werden dabei Datumsfunktionen von Excel benutzt, lassen sich die betreffenden Stellen

durch Suchoperationen nach den Funktionsnamen noch leicht aufspüren. Schwieriger ist es, wenn Berechnungen unabhängig von diesen Funktionen vorgenommen wurden. Wenn z. B. die beiden letzten Stellen der Jahreszahl einfach als Zahl in einer Zelle abgelegt werden, kann das Aufaddieren auf diese Zahl zu unerwünschten Ergebnissen führen, falls die Jahrtausendwende dabei überschritten wird. Besondere Vorsicht ist geboten, wenn die Jahreszahl nur als Zeichenfolge vorliegt.

Während Excel bei Variablen vom Typ `Date` Jahreszahlen auch dann vierstellig speichert, wenn nur die beiden letzten Stellen eingegeben werden, sind Jahreszahlen, die als Textvariable eingegeben oder durch Zeichenfolgenfunktionen aus einer Zeichenfolge herausgezogen werden, unsichere Kandidaten für eine eindeutige Interpretation, falls nur zwei Stellen vorhanden sind. Oft wird z. B. in einer Bestellnummer das Datum als Teil einer Zeichenfolge verwendet. Setzen Makros hier mit Berechnungen auf, etwa um zu prüfen, wann ein Kunde zum letzten Mal bestellt hat, tun Sie gut daran, das Ergebnis zu kontrollieren.

Schwierigkeiten bereiten auch selbstprogrammierte Schaltjahrberechnungen in Makros, die beispielsweise nicht die Ausnahmeregelung für das Jahr 2000 – ein Schaltjahr – berücksichtigen. Es kann also durchaus notwendig sein, alle Berechnungen in vorhandenen Makros, die mit Jahreszahlen zu tun haben, durchzuchecken.

DATWERT()

DATEVALUE()

Syntax:	`DATWERT(`**`Datumstext`**`)`
Beispiel:	`=DATWERT("12. Dezember 13")`
Ergebnis:	`12.12.2013`

Die Funktion `DATWERT()` wandelt ein als Zeichenfolge angegebenes Datum in eine serielle Zahl um.

Die Zeichenfolge muss in einer Schreibweise eingegeben werden, die als gültiger Datumswert ausgewertet werden kann. Ist das nicht der Fall, liefert die Funktion den Fehler `#WERT!`. Die zeitlichen Begrenzungen (Jahreswerte von 1900 bis 9999) sind dieselben wie bei der Funktion `DATUM()`.

Diese Funktion eignet sich speziell auch für Importe von tabellarischen Werten, z. B. aus Textverarbeitungsprogrammen oder aus Programmen, die ein anderes Datumsformat verwenden. Wenn Sie daraus eine Spalte mit Zeichenfolgen gewinnen können, lassen sich diese Werte sehr einfach in gültige Datumswerte umwandeln.

	A	B	C
1			
2	**Ausgabe eines als Text eingegebenen Datums**		
3			
4	Datumstext	DATWERT()	
5	12. Dezember 07	12.12.2007	
6	12. Dez	12.12.2013	
7	12. Dez 43	12.12.1943	
8	11/10/2007	11.10.2007	
9	11-10-2007	11.10.2007	
10	10.2007	01.10.2007	
11	2007-10-11	11.10.2007	
12	2007/07/06	06.07.2007	
13	2013/02/30	#WERT!	
14			

Abbildung 4.14 Umwandlung von Texteingaben in Datumswerte

Für die Behandlung von Zeitwerten, die als Zeichenfolge vorliegen, finden Sie eine entsprechende Funktion: `ZEITWERT()`.

EDATUM()

EDATE()

Syntax: `EDATUM(`**`Ausgangsdatum; Monate`**`)`

Beispiel: `=EDATUM(DATWERT("30.05.2013");10)`

Ergebnis: `30.03.2014`

Die Funktion `EDATUM()` liefert ein um eine bestimmte Anzahl von Monaten gegenüber dem angegebenen `Ausgangsdatum` verschobenes Datum. Der Wert für `Monate` kann positiv oder negativ sein.

`Ausgangsdatum` kann als serieller Wert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann das Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Lässt sich ein Argument nicht als Datumswert behandeln, liefert die Funktion den Fehler `#WERT!`. Hat der Wert für `Monate` Nachkommastellen, werden diese ohne Rundung einfach abgeschnitten. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999). Ergibt die Addition von `Ausgangsdatum + Monate` ein ungültiges Datum, liefert die Funktion den Fehler `#ZAHL!`.

Die Funktion kann beispielsweise verwendet werden, um Fälligkeitstermine für Zinszahlungen einfacher zu berechnen, die jeweils auf denselben Tag fallen wie der Emissionstermin. Die Alternative wäre, ein Datum erst in seine Bestandteile zu zerlegen, den Monatswert zu ändern und das Datum dann neu zusammenzusetzen.

	A	B	C
1			
2	**Zieldatum berechnen**		
3			
4	Ausgangsdatum	Monate	EDATUM()
5	05.05.2013	10	Mittwoch, 5. März 2014
6	05.05.2013	-10	Donnerstag, 5. Juli 2012
7	05.05.2013	0	Sonntag, 5. Mai 2013

Abbildung 4.15 Errechnen eines um eine Anzahl von Monaten entfernten Datums

Gegenüber der Funktion DATUM() hat diese Funktion folgenden Vorteil: Bei der Berechnung des Zieldatums wird berücksichtigt, dass die Monate unterschiedlich viele Tage haben. In Abschnitt 4.2, »Periodische Datumsreihen berechnen«, ist bereits gezeigt worden, dass die Funktion immer den korrekten Schlusstag eines Monats ausgibt, wenn das Ausgangsdatum ein solcher Schlusstag ist.

HEUTE()

TODAY()

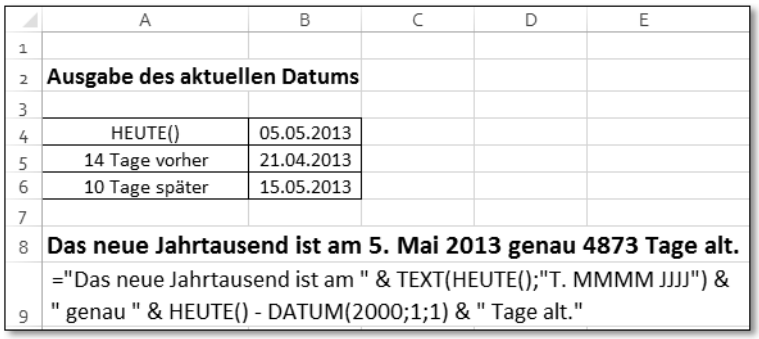
Syntax: HEUTE()
Beispiel: =HEUTE()
Ergebnis: z.B. 24.06.2013

Beim Aufruf der Funktion HEUTE() wird das Systemdatum aus der Computeruhr als Ergebnis geliefert. (Die Uhr sollte also möglichst korrekt eingestellt sein.)

Die Funktion aktualisiert den ausgegebenen Wert jeweils bei der Neuberechnung eines Tabellenblatts. Das kann ein durchaus unerwünschter Effekt sein. Vermeiden lässt sich die Neuberechnung, wenn nach Eingabe der Funktion F9 (Umwandeln in den Wert) gedrückt wird oder indem mit **Kopieren** und anschließend **Einfügen • Werte** die Funktion durch ihr Ergebnis ersetzt wird.

Wird die Funktion innerhalb einer Formel verwendet, bewirkt die Addition einer Zahl, dass ein Datum ausgegeben wird, das eine entsprechende Zahl von Tagen später liegt.

Wird vom Ergebnis der Funktion der Wert einer anderen Datumsfunktion abgezogen, die ein früheres Datum liefert, kann die Differenz in Tagen berechnet werden, wie das Beispiel in Abbildung 4.16 zeigt:



	A	B	C	D	E
1					
2	**Ausgabe des aktuellen Datums**				
3					
4	HEUTE()	05.05.2013			
5	14 Tage vorher	21.04.2013			
6	10 Tage später	15.05.2013			
7					
8	**Das neue Jahrtausend ist am 5. Mai 2013 genau 4873 Tage alt.**				
9	="Das neue Jahrtausend ist am " & TEXT(HEUTE();"T. MMMM JJJJ") & " genau " & HEUTE() - DATUM(2000;1;1) & " Tage alt."				

Abbildung 4.16 Zeitrechnungen mit der Funktion `HEUTE()`

ISOKALENDERWOCHE()

ISOWEEKNUM()

Syntax: `ISOKALENDERWOCHE (`**`Datum`**`)`

Beispiel: `=ISOKALENDERWOCHE (DATWERT("19.12.2014"))`

Ergebnis: `51`

Die in Excel 2013 neue Funktion `ISOKALENDERWOCHE()` liefert die Zahl der Kalenderwoche für den mit dem Argument `Datum` angegebenen Tag nach dem internationalen ISO-Standard 8601. Dieser Standard ist in Deutschland seit 2006 gültig. Er bestimmt als letzten Tag der Woche den Sonntag. Als erste Woche eines Jahres gilt die Woche, die mindestens 4 Tage im neuen Jahr enthält.

`Datum` kann als serieller Wert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann das Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Lässt sich das Argument nicht als Datumswert behandeln, liefert die Funktion den Fehler `#WERT!`. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.

	A	B
1		
2	**Angabe der Kalenderwoche nach dem ISO-Standard**	
3		
4	Datum	ISOKALENDERWOCHE()
5	Mittwoch, 2. Januar 2013	1
6	Sonntag, 5. Mai 2013	18
7	Mittwoch, 3. Juli 2013	27
8	Dienstag, 31. Dezember 2013	1

Abbildung 4.17 Berechnung der Kalenderwoche nach dem ISO-Standard

In anderen Ländern wie den USA wird der Sonntag als erster Tag der Woche behandelt. Die erste Woche ist immer die Woche, die den 1. Januar enthält. Die bisher in Excel zur Verfügung stehende Funktion `KALENDERWOCHE()` berechnet mit dem vorgegebenen Wert für das Argument `Zahl_Typ` die Wochennummer nach dem in den USA geltenden Verfahren. Diese Abweichungen müssen natürlich beachtet werden, wenn Kalenderdaten im internationalen Verkehr ausgetauscht werden.

```
16 =KALENDERWOCHE(DATWERT("14.04.2013"))
15 =ISOKALENDERWOCHE(DATWERT("14.04.2013"))
```

JAHR()

YEAR()

Syntax: `JAHR(`**`Zahl`**`)`
Beispiel: `=JAHR(34249)`
Ergebnis: `1993`

Die Funktion `JAHR()` extrahiert aus einem mit `Zahl` angegebenen Datum die Jahreszahl. `Zahl` kann als serieller Datumswert oder mit Hilfe einer Datumsfunktion (z.B. `=JAHR(HEUTE())`) angegeben werden. Ebenso kann als Argument auch eine Zeichenfolge verwendet werden, wenn sie als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Lässt sich das Argument nicht als Datumswert behandeln, liefert die Funktion den Fehler `#WERT!`. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.

	A	B
1		
2	**Die Jahreszahl eines Datums ausgeben**	
3		
4	Zahl	JAHR()
5		1900
6	30.05.2013	2013
7	12. Okt. 43	1943
8	10. Mrz. 12	2012
9	01.01.1877	#WERT!
10	01.01.2999	2999
11	99999999	#ZAHL!

Abbildung 4.18 Das Zeitfenster der Funktion JAHR() ist eingegrenzt.

JETZT()

NOW()

Syntax: JETZT()
Beispiel: =JETZT()
Ergebnis: z.B. 24.06.2013 8:40

Die Funktion JETZT() liefert Datum und Uhrzeit aus der Systemuhr Ihres Computers. Die Funktion aktualisiert den ausgegebenen Wert jeweils bei Neuberechnung. Das kann ein durchaus unerwünschter Effekt sein. Vermeiden lässt sich die Neuberechnung, wenn nach Eingabe der Funktion F9 (Umwandeln in den Wert) gedrückt wird oder indem mit **Kopieren** und anschließend **Einfügen • Werte** die Funktion durch ihr Ergebnis ersetzt wird.

Wie bei der Funktion HEUTE() führt die Addition einer ganzen Zahl zum Ergebnis der Funktion zu einem Datum, das um eine entsprechende Anzahl von Tagen verschoben ist. Wird eine gebrochene Zahl addiert, werden die entsprechenden Zeitanteile des Datums verändert.

```
=JETZT()+0,5
```

liefert beispielsweise einen Termin, der 12 Stunden später liegt.

Verwenden können Sie JETZT() für zeitliche Dokumentationen oder als Stoppuhr für Zeitabläufe. Obwohl die Funktion kein Argument verlangt, müssen die leeren Klammern mit angegeben werden.

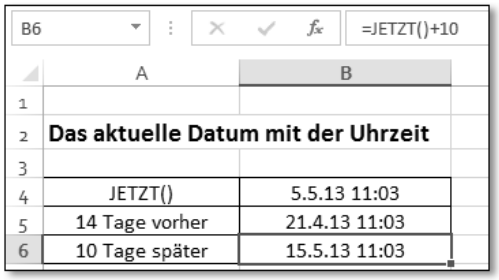


Abbildung 4.19 Berechnungen mit dem aktuellen Datum

KALENDERWOCHE()

WEEKNUM()

Syntax: `KALENDERWOCHE(`**`Fortlaufende_Zahl`**`; Zahl_Typ)`
Beispiel: `=KALENDERWOCHE(HEUTE();1)`
Ergebnis: `22` (am 26.05.2013)

Die Funktion `KALENDERWOCHE()` liefert für das mit `Fortlaufende_Zahl` angegebene Datum die entsprechende Kalenderwochenzahl. Ein Wert `1` für `Zahl_Typ` gibt an, dass die Woche am Sonntag beginnt, dies ist auch die Vorgabe; `Zahl_Typ 2` bedeutet, dass die Woche am Montag beginnt.

`Fortlaufende_Zahl` kann als serieller Wert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann das Argument auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

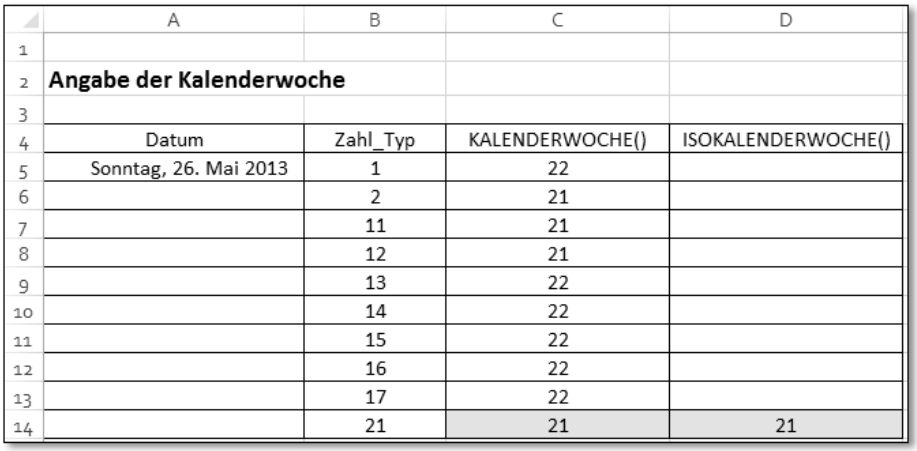
Lässt sich das Argument nicht als Datumswert behandeln, liefert die Funktion den Fehler `#WERT!`. Gleiches gilt, wenn ein ungültiger Wert für `Zahl_Typ` angegeben wird. Der Datumswert muss sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.

Seit Excel 2010 sind die Werte für `Zahl_Typ` erweitert worden, um gleich zwei Berechnungssysteme abzudecken:

- System 1: Die Woche, in die der 1. Januar eines Jahres fällt, ist auch die Woche Nr. 1.
- System 2: Die erste Kalenderwoche ist die Woche, in die der erste Donnerstag des Jahres fällt. Das entspricht dem Standard ISO-8601, der in Europa bei der Wochennummernberechnung gültig ist.

Zahl_Typ	Wochenanfang	System
1 oder nicht angegeben	Sonntag	1
2	Montag	1
11	Montag	1
12	Dienstag	1
13	Mittwoch	1
14	Donnerstag	1
15	Freitag	1
16	Samstag	1
17	Sonntag	1
21	Montag	2

Tabelle 4.3 Werte für `Zahl_Typ` und ihre Bedeutung



	A	B	C	D
1				
2	**Angabe der Kalenderwoche**			
3				
4	Datum	Zahl_Typ	KALENDERWOCHE()	ISOKALENDERWOCHE()
5	Sonntag, 26. Mai 2013	1	22	
6		2	21	
7		11	21	
8		12	21	
9		13	22	
10		14	22	
11		15	22	
12		16	22	
13		17	22	
14		21	21	21

Abbildung 4.20 Die Funktion `KALENDERWOCHE()` rechnet als Vorgabe nach den amerikanischen Regeln.

Das Ergebnis der Funktion entspricht bei dem vorgegebenen `Zahl_Typ 1` nicht den für die Berechnung der Kalenderwoche in Deutschland geltenden Normen – DIN 1355/ ISO-Norm 8601. Diese Normen verlangen nicht nur, dass die Woche mit einem Montag beginnt, was mit dem Rückgabewert `2` zu erreichen ist, sondern zusätzlich, dass in der ersten Woche eines Jahres mindestens 4 Tage enthalten sein müssen, ansonsten wird

dieser Wochenrest als 53. Woche des Vorjahres gerechnet. Daraus folgt, dass ein 29. Dezember schon zur ersten Woche des Folgejahres, ein 3. Januar noch zur letzten Woche des Vorjahres gehören kann.

Die erweiterte Werteliste für das Argument `Zahl_Typ` erlaubt seit Excel 2010 eine der Norm entsprechende Berechnung mit dem Wert `21`, wie Abbildung 4.20 zeigt. In Excel 2013 ist jetzt eine normgerechte Berechnung auf einfachere Weise mit der neuen Funktion `ISOKALENDERWOCHE()` möglich, siehe dort.

In anderen Ländern, beispielsweise in den USA, gelten teilweise andere Kalenderwochenberechnungen, die etwa den Sonntag als ersten Wochentag betrachten. Wenn Sie Dokumente erstellen, die für den amerikanischen Markt gedacht sind, können Sie also die Funktion `KALENDERWOCHE()` mit dem vorgegebenen Wert für `Zahl_Typ` benutzen.

Wenn Sie mit einer älteren Excel-Version arbeiten, haben Sie die Möglichkeit, mit folgender Formel zu einer ISO-konformen Kalenderwochenangabe zu kommen:

```
=KÜRZEN((A5-DATUM(JAHR(A5-REST(A5-2;7)+3);1;REST(A5-2;7)-9))/7)
```

Dabei wird hier angenommen, dass das zu untersuchende Datum in Zelle A5 abgelegt ist.

Bequemer ist es allerdings, für diese Fälle eine benutzerdefinierte Funktion einzurichten. Dies wird in Kapitel 18, »Entwicklung eigener Funktionen«, beschrieben.

	A	B	C	D
1				
2	**Korrigierte Kalenderwochenberechnung**			
3				
4		ISO-konforme Berechnung	KALENDERWOCHE()	
5	01.01.2000	52	1	
6	01.01.2001	1	1	
7	01.01.2002	1	1	
8	01.01.2003	1	1	
9	01.01.2004	1	1	
10	01.01.2005	53	1	
11	01.01.2006	52	1	
12	01.01.2007	1	1	
13	01.01.2008	1	1	
14		**=KÜRZEN((A5-DATUM(JAHR(A5-REST(A5-2;7)+3);1;REST(A5-2;7)-9))/7)**		
15				

Abbildung 4.21 Kalenderwochenberechnung mit einer komplexen Formel

MINUTE()

MINUTE()

Syntax: `MINUTE(Zahl)`
Beispiel: `=MINUTE(0,001)`
Ergebnis: `1`

Die Funktion `MINUTE()` extrahiert die Anzahl der Minuten aus dem mit dem Argument `Zahl` angegebenen Zeit- oder Datums- und Zeitwert. Der Minutenanteil wird als eine Zahl von 0 bis 59 ausgegeben.

Der Wert für `Zahl` kann über eine Zeit- oder Datumsfunktion geliefert werden oder über einen Verweis darauf. Er kann aber auch direkt als serielle Zahl angegeben werden oder als Zeichenfolge in Anführungszeichen, die als gültiger Datums- bzw. Zeitwert ausgewertet werden kann, beispielsweise `=MINUTE("12:12")`.

Bei einem ungültigen Wert für `Zahl` liefert die Funktion den Fehler `#WERT!`.

Abbildung 4.22 zeigt einige Beispiele. Die Funktion berechnet dort die Minutenanteile von Werten, die durch Datums- oder Zeitfunktionen geliefert werden. `=MINUTE(JETZT())` zeigt jeweils die aktuelle Minute. Liegt `Zahl` als serielle Zahl vor, wird von der Funktion nur der Zeitanteil rechts vom Komma für die Minutenausgabe ausgewertet.

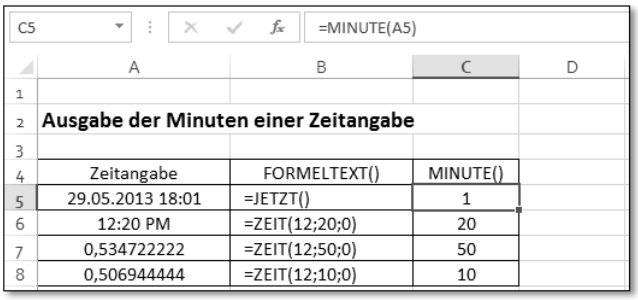


Abbildung 4.22 Extrahieren des Minutenanteils

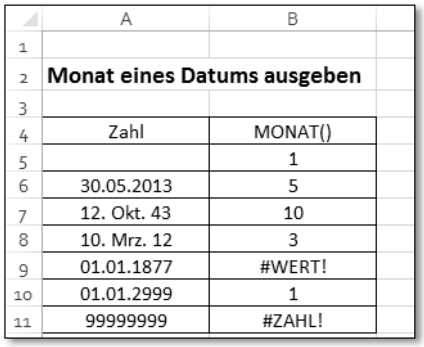
MONAT()

MONTH()

Syntax: `MONAT(Zahl)`
Beispiel: `=Monat(34406)`
Ergebnis: `3`

Die Funktion MONAT() extrahiert aus einem mit dem Argument Zahl angegebenen Datum die entsprechende Monatszahl, also einen Wert von 1 bis 12. Zahl kann als serieller Datumswert oder mit Hilfe einer Datumsfunktion angegeben werden. =MONAT(HEUTE()) liefert immer die Zahl des aktuellen Monats. Ebenso kann als Argument auch Text übergeben werden, wenn er sich als gültiger Datumswert auswerten lässt und in Anführungszeichen gesetzt ist.

Bei einem ungültigen Wert für Zahl liefert die Funktion den Fehler #WERT!. Die Werte für Zahl müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999); verweist ein Wert über diese Begrenzung hinaus, wie in dem Beispiel in Abbildung 4.23, liefert die Funktion den Fehler #ZAHL!.



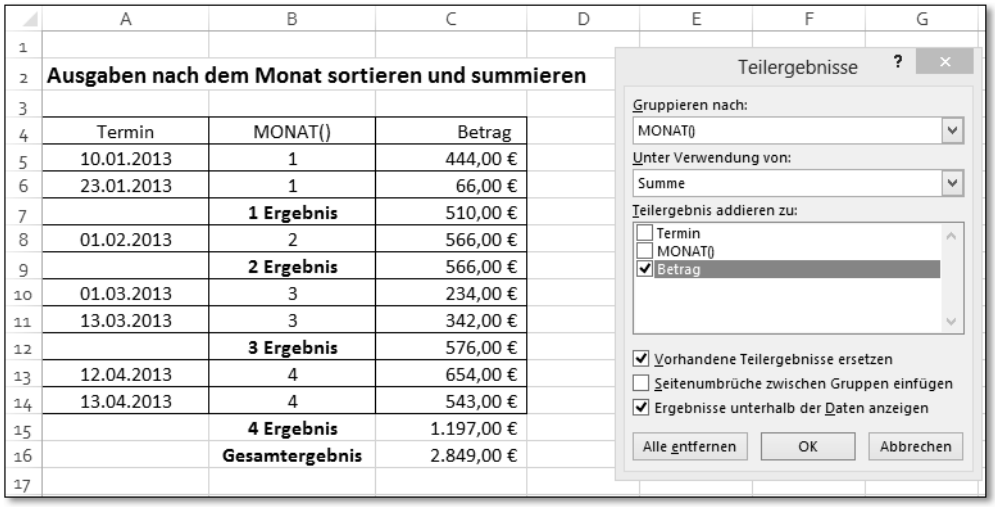
	A	B
1		
2	**Monat eines Datums ausgeben**	
3		
4	Zahl	MONAT()
5		1
6	30.05.2013	5
7	12. Okt. 43	10
8	10. Mrz. 12	3
9	01.01.1877	#WERT!
10	01.01.2999	1
11	99999999	#ZAHL!

Abbildung 4.23 Extrahieren des Monats aus einem Datum

4.6.3 Nach dem Monat sortieren und summieren

An dieser Stelle sei auf eine Möglichkeit hingewiesen, die alle Funktionen anbieten, die einen Datums- oder auch Zeitwert in seine einzelnen Bestandteile zerlegen. In allen Fällen wird es möglich, Tabellen nach den mit Hilfe der jeweiligen Funktion isolierten Bestandteilen zu sortieren. Das folgende Beispiel deutet dies an. Eine Liste von Ausgaben ist jeweils mit einem Datum gekennzeichnet. Aus diesem Datum wird der Monat in einer eigenen Spalte herausgezogen. Diese Spalte wird dann als Sortierschlüssel für die ganze Tabelle benutzt.

Anschließend kann die Funktion **Daten • Gliederung • Teilergebnisse** verwendet werden, um Zwischensummen für die einzelnen Monate zu erzeugen. In den älteren Versionen ist dies der Befehl **Daten • Teilergebnisse**. Dabei wird die Monatsspalte als Gruppierungsmerkmal ausgewählt.



	A	B	C	D	E	F	G
1							
2	**Ausgaben nach dem Monat sortieren und summieren**						
3							
4	Termin	MONAT()	Betrag				
5	10.01.2013	1	444,00 €				
6	23.01.2013	1	66,00 €				
7		**1 Ergebnis**	510,00 €				
8	01.02.2013	2	566,00 €				
9		**2 Ergebnis**	566,00 €				
10	01.03.2013	3	234,00 €				
11	13.03.2013	3	342,00 €				
12		**3 Ergebnis**	576,00 €				
13	12.04.2013	4	654,00 €				
14	13.04.2013	4	543,00 €				
15		**4 Ergebnis**	1.197,00 €				
16		**Gesamtergebnis**	2.849,00 €				
17							

Abbildung 4.24 Sortierte Monatsausgaben

MONATSENDE()

EOMONTH()

Syntax: `MONATSENDE(`**`Ausgangsdatum; Monate`**`)`

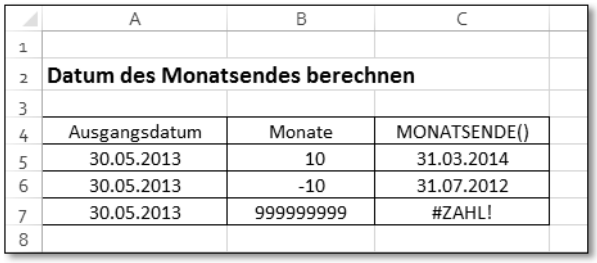
Beispiel: `=MONATSENDE(A5;10)`

Ergebnis: `31.03.2014`, wenn in A5 das Datum 30.05.2013 steht

Die Funktion `MONATSENDE()` liefert den letzten Tag des Monats, in den das um die mit dem Argument `Monate` angegebene Zahl von Monaten gegenüber `Ausgangsdatum` verschobene Datum fällt. Ist `Monate` eine positive Zahl, liefert die Funktion ein Datum in der Zukunft, bei einem negativen Wert rechnet die Funktion zurück. Nachkommastellen werden abgeschnitten.

`Ausgangsdatum` kann als serieller Datumswert oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso kann als Argument auch Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Bei einem ungültigen Datumswert oder einem nicht numerischen Wert für `Monate` liefert die Funktion den Fehler `#WERT!`. Der Datumswert und die Summe aus `Ausgangsdatum` + `Monate` müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.



	A	B	C
1			
2	**Datum des Monatsendes berechnen**		
3			
4	Ausgangsdatum	Monate	MONATSENDE()
5	30.05.2013	10	31.03.2014
6	30.05.2013	-10	31.07.2012
7	30.05.2013	999999999	#ZAHL!
8			

Abbildung 4.25 Ein zukünftiges oder vergangenes Monatsende finden

Mit Hilfe dieser Funktion lassen sich beispielsweise Fälligkeitstermine leicht ermitteln, die auf ein Monatsende fallen. Soll ein Termin dagegen am Monatsanfang liegen, reicht es, zu dem Ergebnis der Funktion `1` zu addieren, um den Anfang des nächsten Monats zu erreichen.

Wenn Sie wissen wollen, auf welchen Wochentag das gesuchte Monatsende fällt, benutzen Sie die Funktion `WOCHENTAG()`:

```
=WOCHENTAG(MONATSENDE(DATUM(2013;05;30);10))
```

NETTOARBEITSTAGE()

NETWORKDAYS()

Syntax:	`NETTOARBEITSTAGE(`**`Ausgangsdatum; Enddatum;`** `Freie_Tage)`
Beispiel:	`=NETTOARBEITSTAGE(DATWERT("01.05.2013");` `DATWERT("31.05.2013");{41395;41411;41422})`
Ergebnis:	`20`

Die Funktion `NETTOARBEITSTAGE()` liefert die Zahl der Arbeitstage zwischen den angegebenen Datumswerten `Ausgangsdatum` und `Enddatum`. Als Arbeitstage werden alle Werktage (nicht Samstag und Sonntag) gezählt; mit `Freie_Tage` lassen sich zusätzlich Feiertage und Urlaubstage angeben, die abgezogen werden.

`Ausgangsdatum` und `Enddatum` können als serielle Datumswerte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso können die Argumente auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

`Freie_Tage` kann ein Bezug auf einen Zellbereich sein, in dem die Daten der freien Tage abgelegt sind, oder auch eine Matrixkonstante mit seriellen Datumswerten oder Zei-

chenfolgen für Datumswerte. Die Werte der Matrixkonstanten müssen in geschweifte Klammern eingefasst werden.

Bei ungültigen Datumswerten liefert die Funktion den Fehler `#WERT!`. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.

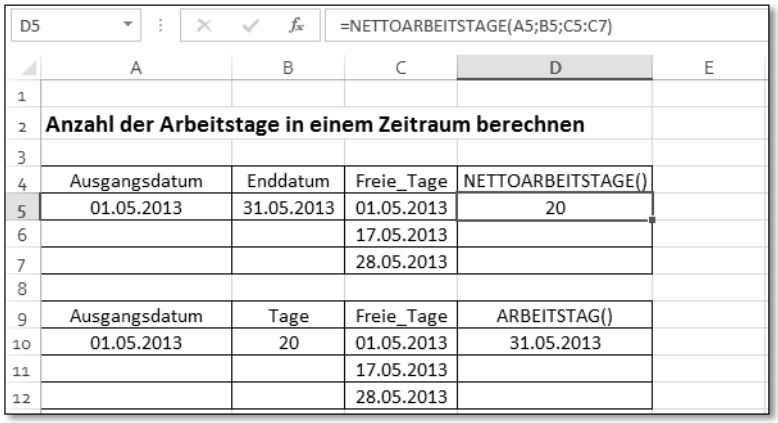


Abbildung 4.26 Arbeitstage nach Abzug der freien Tage

Während die Funktion `ARBEITSTAG()` ein Datum liefert, das ein Arbeitstag in der Zukunft oder in der Vergangenheit ist, berechnet `NETTOARBEITSTAGE()` die Anzahl der Arbeitstage zwischen zwei Terminen.

NETTOARBEITSTAGE.INTL()

NETWORKDAYS.INTL()

Syntax:	`NETTOARBEITSTAGE.INTL(`**Ausgangsdatum**`;` **Enddatum**`; Wochenende; Freie_Tage)`
Beispiel:	`=NETTOARBEITSTAGE.INTL(DATUM(2013;1;1); DATUM(2013;1;31); 11)`
Ergebnis:	`27`, da nur die Sonntage als Wochenendtage gezählt werden

Die Funktion `NETTOARBEITSTAG.INTL()` liefert die Zahl der Arbeitstage zwischen den beiden Datumswerten `Ausgangsdatum` und `Enddatum`. Als Arbeitstage werden alle Werktage (nicht Samstag und Sonntag) gezählt; mit `Freie_Tage` lassen sich zusätzlich Feiertage und Urlaubstage angeben, die abgezogen werden.

Ausgangsdatum und Enddatum können als serielle Datumswerte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso können die Argumente auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Freie_Tage kann ein Bezug auf einen Zellbereich sein, in dem die Daten der freien Tage abgelegt sind, oder auch eine Matrixkonstante mit seriellen Datumswerten oder Zeichenfolgen für Datumswerte. Die Werte der Matrixkonstanten müssen in geschweifte Klammern eingefasst werden.

Die Funktion erweitert die Funktion NETTOARBEITSTAGE() um das Argument Wochenende, wobei beliebige Wochentage als Wochenende behandelt werden können. Mit Wochenendnummern wird angegeben, welche Tage als Wochenende gewertet werden.

Nr.	Wochenendtage
1 (oder nicht angegeben)	Samstag, Sonntag
2	Sonntag, Montag
3	Montag, Dienstag
4	Dienstag, Mittwoch
5	Mittwoch, Donnerstag
6	Donnerstag, Freitag
7	Freitag, Samstag
11	nur Sonntag
12	nur Montag
13	nur Dienstag
14	nur Mittwoch
15	nur Donnerstag
16	nur Freitag
17	nur Samstag

Tabelle 4.4 Die möglichen Wochenendnummern und ihre Bedeutung

Statt der Nummern kann auch eine Zeichenfolge angegeben werden, bei der beginnend mit dem Montag 1 für Wochenendtage steht, 0 für Arbeitstage, etwa 0000011 für eine normale Woche mit dem Wochenende am Samstag und Sonntag.

Bei ungültigen Datumswerten oder einem ungültigen Wert für `Wochenende` liefert die Funktion den Fehler `#WERT!`. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999), sonst liefert die Funktion den Fehler `#ZAHL!`.

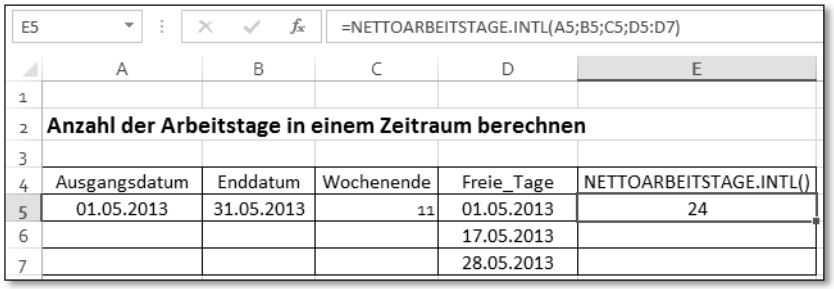


Abbildung 4.27 Arbeitstagberechnung mit variablen Wochenenden

Siehe dazu auch die Funktion `ARBEITSTAG.INTL()`.

SEKUNDE()

SECOND()

Syntax: `SEKUNDE(`**Zahl**`)`
Beispiel: `=SEKUNDE(41424,60785)`
Ergebnis: `18`

Die Funktion `SEKUNDE()` extrahiert die Anzahl der Sekunden aus dem mit dem Argument `Zahl` angegebenen Zeit- oder Datums- und Zeitwert. Der Sekundenanteil einer Zeitseriennummer wird als eine Zahl von 0 bis 59 ausgegeben.

Der Wert für `Zahl` kann über eine Zeit- oder Datumsfunktion geliefert werden oder über einen Verweis darauf. Er kann aber auch direkt als serielle Zahl angegeben werden oder als Zeichenfolge in Anführungszeichen, die als gültiger Datums- bzw. Zeitwert ausgewertet werden kann, beispielsweise `=SEKUNDE("12:12:20")`.

Bei ungültigen Datums- bzw. Zeitwerten liefert die Funktion den Fehler `#WERT!`.

Im Beispiel wird von der Funktion nur der Zeitanteil rechts vom Komma für die Sekundenausgabe verwendet. Die Funktion liefert auch die Sekundenanteile von Werten, die durch eine der Datums- oder Zeitfunktionen geliefert werden, wie das Beispiel in Abbildung 4.28 zeigt. `=SEKUNDE(JETZT())` zeigt jeweils die aktuelle Sekunde.

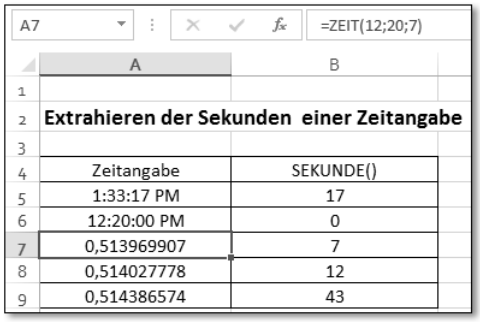


Abbildung 4.28 Ausgabe der Sekunden einer Uhrzeit

STUNDE()

HOUR()

Syntax: `STUNDE(`**`Zahl`**`)`

Beispiel: `=STUNDE(41424,61692)`

Ergebnis: `14`

Die Funktion `STUNDE()` extrahiert die Anzahl der Stunden aus dem mit dem Argument `Zahl` angegebenen Zeit- oder Datums- und Zeitwert. Der Stundenanteil eines Zeitwerts wird als eine Zahl von 0 bis 23 ausgegeben.

Der Wert für `Zahl` kann über eine Zeit- oder Datumsfunktion geliefert werden oder über einen Verweis darauf. Er kann aber auch direkt als serielle Zahl angegeben werden oder als Zeichenfolge in Anführungszeichen, die als gültiger Datums- bzw. Zeitwert ausgewertet werden kann, beispielsweise `=STUNDE("12:12")`.

Bei ungültigen Datums- bzw. Zeitwerten liefert die Funktion den Fehler `#WERT!`.

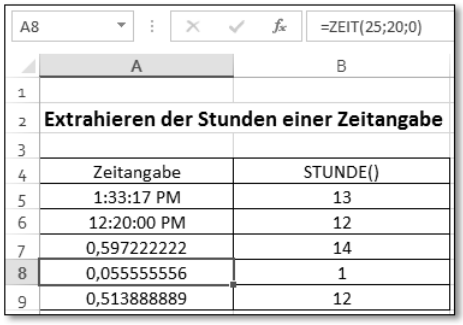


Abbildung 4.29 Ausgabe der Stunden der Uhrzeit

TAG()

DAY()

Syntax: TAG(**Zahl**)
Beispiel: =TAG(34240)
Ergebnis: 28

Die Funktion TAG() extrahiert aus einem mit dem Argument Zahl angegebenen Datum die entsprechende Tageszahl, also einen Wert von 1 bis 31.

Der Wert für Zahl kann über eine Datumsfunktion geliefert werden oder über einen Verweis darauf. Er kann aber auch direkt als serielle Zahl angegeben werden oder als Zeichenfolge in Anführungszeichen, die als gültiger Datumswert ausgewertet werden kann, beispielsweise =TAG("12.10.2013").

Bei ungültigen Datumswerten liefert die Funktion den Fehler #WERT!. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999); verweist ein Wert über diese Begrenzung hinaus, wie in dem Beispiel in Abbildung 4.30, liefert die Funktion den Fehler #ZAHL!.

	A	B
1		
2	**Tag eines Datums ausgeben**	
3		
4	Zahl	TAG()
5		0
6	30.05.2013	30
7	12. Okt. 43	12
8	10. Mrz. 12	10
9	01.01.1877	#WERT!
10	01.01.2999	1
11	99999999	#ZAHL!

Abbildung 4.30 Extrahieren des Tags aus einem Datum

TAGE()

DAYS()

Syntax: TAGE(**Zieldatum**; **Ausgangsdatum**)
Beispiel: =TAGE(DATWERT("31.01.2013");DATWERT("01.01.2013"))
Ergebnis: 30

Die in Excel 2013 neue Funktion TAGE()gibt die Anzahl der Tage an, die zwischen den mit Zieldatum und Ausgangsdatum angegebenen Terminen liegen. Liegt Zieldatum kalendarisch vor dem Ausgangsdatum, so wird das Ergebnis als negativer Wert ausgegeben.

Beide Argumente können als serielle Datumswerte oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso können die Argumente auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Bei ungültigen Datumswerten liefert die Funktion den Fehler #WERT!. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999); verweist ein Wert über diese Begrenzung hinaus, liefert die Funktion den Fehler #ZAHL!.

	A	B	C
1			
2	**Tage zwischen zwei Terminen berechnen**		
3			
4	Zieldatum	Ausgangsdatum	TAGE()
5	31.12.2013	30.05.2013	215
6	31.05.2014	31.05.2013	365
7	07.06.1943	07.06.2013	-25568
8	02.06.2013	10.03.2012	449
9	03.06.2013	01.01.1877	#WERT!
10	4000000	41424	#ZAHL!

Abbildung 4.31 Tagesberechnungen

TAGE360()

DAYS360()

Syntax: TAGE360(**Ausgangsdatum; Enddatum;** Methode)

Beispiel: =TAGE360(DATWERT("01.01.2013");DATWERT("31.01.2013";FALSCH))

Ergebnis: 30

Die Funktion TAGE360() wird im Zusammenhang mit der tagesgenauen Zinsberechnung eingesetzt. Sie gibt die Anzahl der Tage an, die zwischen den mit Ausgangsdatum und Enddatum angegebenen Terminen liegen. Grundlage für die Berechnung ist dabei das Zinsjahr, die Einteilung des Jahres in 12 Monate zu 30 Tagen, die der noch heute gebräuchlichen kaufmännischen Zinsrechnung zugrunde liegt und zur Vereinfachung der

Zinsrechnung eingeführt wurde. Liegt `Enddatum` kalendarisch vor `Ausgangsdatum`, so wird das Ergebnis in negativer Form ausgegeben.

Beide Datumswerte können als serielle Zahlen oder mit Hilfe einer Datumsfunktion angegeben werden. Ebenso können die Argumente auch als Text übergeben werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Bei ungültigen Datumswerten liefert die Funktion den Fehler `#WERT!`. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999); verweist ein Wert über diese Begrenzung hinaus, liefert die Funktion den Fehler `#ZAHL!`.

Das optionale Argument `Methode` schaltet einen Wahrheitswert. Mit `FALSCH` oder nicht angegeben wird die amerikanische Methode (auch *NASD*-Methode, so benannt nach der Regulierungsbehörde National Association of Securities Dealers, kurz *NASD*) für die Berechnung verwendet. Ist `Ausgangsdatum` der 31. Tag des Monats, wird dieser zum 30. Tag des Monats. Ist `Enddatum` der 31. Tag des Monats und gleichzeitig `Ausgangsdatum` ein Tag vor dem 30. eines Monats, wird `Enddatum` zum 1. des Folgemonats.

Das Argument `WAHR` schaltet die europäische Methode ein. Dabei wird jedes Argument `Ausgangsdatum` oder `Enddatum`, das auf den 31. eines Monats fällt, zum 30. desselben Monats.

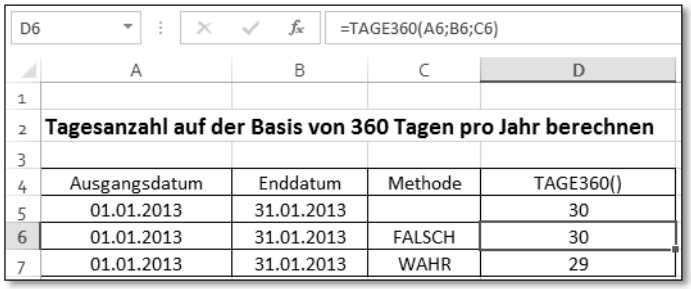


Abbildung 4.32 Tagesberechnung auf der Basis der amerikanischen oder der europäischen Methode

4.6.4 Berechnung von Tageszinsen

Die Funktion ist hilfreich bei der Zinsberechnung, wenn das Jahr mit 360 Tagen und jeder Monat mit 30 Tagen angesetzt wird, wie es bei den Banken in Deutschland üblich ist. Abbildung 4.33 zeigt eine Anwendung der Funktion bei der tagesgenauen Zinsberechnung. In der Zelle B16 wird mit Hilfe der `TAGE360()`-Funktion die Anzahl der Zinstage ermittelt. In Zelle B17 kann dann mit der Formel für die einfache Zinsberechnung gear-

beitet werden, die in Abschnitt 3.1, »Einsatzbereiche der finanzmathematischen Funktionen«, beschrieben ist.

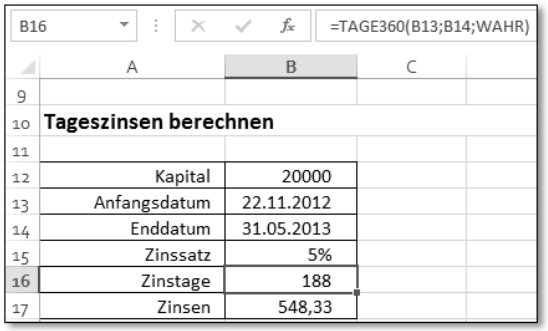


Abbildung 4.33 Die tageweise Berechnung von Zinsen

WOCHENTAG()

WEEKDAY()

Syntax: `WOCHENTAG(Zahl; Typ)`

Beispiel: `=WOCHENTAG(DATWERT("01.04.2013");2)`

Ergebnis: `1`

Die Funktion `WOCHENTAG()` liefert den Wochentag für das mit dem Argument `Zahl` angegebene Datum. Die Reihenfolge, in der die Wochentage gezählt werden, kann über das Argument `Typ` gesteuert werden.

`Zahl` kann als serieller Datumswert oder mit Hilfe einer Datumsfunktion angegeben werden. Die Formel `=WOCHENTAG(HEUTE();2)` liefert immer die Nummer des aktuellen Tages. Ebenso kann als Argument auch Text verwendet werden, wenn er als gültiger Datumswert ausgewertet werden kann und in Anführungszeichen gesetzt ist.

Bei ungültigen Datumswerten oder ungültigen Werten für `Typ` liefert die Funktion den Fehler `#WERT!`. Die Datumswerte müssen sich in dem Zeitfenster bewegen, das Excel für die Datumsberechnung verwendet (1900 bis 9999); verweist ein Wert über diese Begrenzung hinaus, liefert die Funktion den Fehler `#ZAHL!`.

Das Ergebnis dieser Funktion ist eine Zahl von 1 bis 7 oder von 0 bis 6. Jede Zahl stellt dabei die Nummer eines Wochentags dar. Ist für das optionale Argument `Typ` `1` angegeben oder fehlt das Argument, beginnt die Zählung am Sonntag mit `1`. Das entspricht der Zählweise in den USA.

Mit Typ = 2 beginnt die Zählung am Montag, das entspricht der in Deutschland üblichen Zählweise, die seit 2006 der ISO-Norm 8601 entspricht. Mit Typ = 3 beginnt die Zählung ebenfalls am Montag, aber mit dem Wert 0 und endet am Sonntag mit 6.

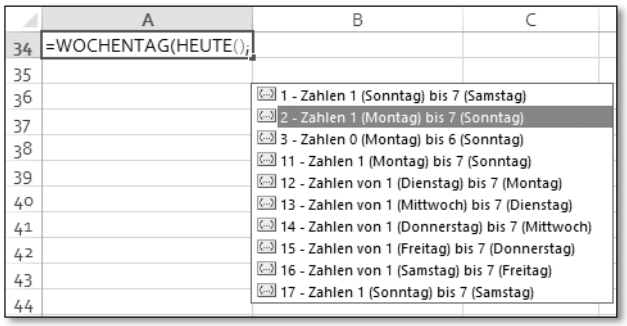


Abbildung 4.34 Die Liste der Werte für das Argument Typ

Die verschiedenen Werte für Typ werden Ihnen bei der manuellen Eingabe der Funktion angeboten, wie Abbildung 4.35 zeigt.

Ermitteln der Wochentagsnummer

Tag	Typ	WOCHENTAG()
Sonntag, 24. März 2013	1	1
Montag, 25. März 2013	1	2
Dienstag, 26. März 2013	1	3
Mittwoch, 27. März 2013	1	4
Donnerstag, 28. März 2013	1	5
Freitag, 29. März 2013	1	6
Samstag, 30. März 2013	1	7
Sonntag, 31. März 2013	2	7
Montag, 1. April 2013	2	1
Dienstag, 2. April 2013	2	2
Mittwoch, 3. April 2013	2	3
Donnerstag, 4. April 2013	2	4
Freitag, 5. April 2013	2	5
Samstag, 6. April 2013	2	6
Sonntag, 7. April 2013	3	6
Montag, 8. April 2013	3	0
Dienstag, 9. April 2013	3	1
Mittwoch, 10. April 2013	3	2
Donnerstag, 11. April 2013	3	3
Freitag, 12. April 2013	3	4
Samstag, 13. April 2013	3	5

Abbildung 4.35 Unterschiedliche Nummerierung der Wochentage

ZEIT()

TIME()

Syntax: `ZEIT(Stunde; Minute; Sekunde)`
Beispiel: `=ZEIT(1;1;1)`
Ergebnis: `01:01:01`

Die Funktion `ZEIT()` liefert die Zeit, die durch die Werte für `Stunde`, `Minute` und `Sekunde` festgelegt wird. Die Werte können als Konstanten oder über Zellbezüge oder Formeln angegeben werden, über die variable Zeitangaben erzeugt werden können.

Für `Stunde` kann eine Zahl zwischen 0 und 32767 angegeben werden, jede Zahl über 23 wird aber durch 24 geteilt und der Rest als Wert für die Stunde genommen.

Für `Minute` kann eine Zahl zwischen 0 und 32767 angegeben werden, jede Zahl über 59 wird aber durch 24 * 60 geteilt und der Rest als Wert für Stunden und Minuten umgerechnet.

Für `Sekunde` kann eine Zahl zwischen 0 und 32767 angegeben werden, jede Zahl über 59 wird aber durch 24 * 60 * 60 geteilt und der Rest als Wert für die Stunden, Minuten und Sekunden umgerechnet.

Wird das Ergebnis der Funktion mit dem Standardformat belegt, wird die Dezimalzahl sichtbar, die die Funktion liefert. Es ist eine Zahl zwischen 0 und 0,99999999. Das entspricht den Zeitangaben 00:00:00 und 23:59:59. 0,5 entspricht 12:00:00.

Bei nicht numerischen Argumenten liefert die Funktion den Fehler `#WERT!`.

	A	B	C	D
1				
2	**Ausgabe einer Zeitangabe**			
3				
4	Stunde	Minute	Sekunde	ZEIT()
5	0	1	2	00:01:02
6	13	6	7	13:06:07
7	23	8	12	23:08:12
8	23	70	13	00:10:13
9	23	40	70	23:41:10
10	25	40	70	01:41:10

Abbildung 4.36 Die Funktion `ZEIT()` kommt mit »unmöglichen« Werten zurecht.

ZEITWERT()

TIMEVALUE()

Syntax: `ZEITWERT(`**`Zeit`**`)`
Beispiel: `=ZEITWERT("12:00:00")`
Ergebnis: `0,5`

Die Funktion `ZEITWERT()` wandelt eine Zeichenfolge in eine serielle Zahl um, die sich dann wiederum über ein Zeitformat als Uhrzeit anzeigen lässt. Eine Zeitangabe lässt sich als Zeichenfolge eingeben, wenn sie einem der gültigen Zeitformate entspricht.

Bei ungültigen Werten für `Zeit` liefert die Funktion den Fehler `#WERT!`.

Diese Funktion kann hilfreich sein, wenn Daten aus anderen Anwendungen importiert werden und es darum geht, sicherzustellen, dass gültige Zeitwerte entstehen.

	A	B	C	D
1				
2	**Umwandlung einer als Text eingegebenen Zeitangabe**			
3				
4	Zeitangabe	ZEITWERT()	unformatiert	
5	12:20	12:20:00	0,513888889	
6	6:20:40	06:20:40	0,264351852	
7	25:20	01:20:00	0,055555556	
8	23:70	00:10:00	0,006944444	
9				

Abbildung 4.37 Interpretation von Zeitangaben

5 Mathematische und trigonometrische Funktionen

Funktion	Seite
ABRUNDEN()	311
ABS()	312
AGGREGAT()	313
ARABISCH()	316
ARCCOS()	317
ARCCOSHYP()	319
ARCCOT()	320
ARCCOTHYP()	320
ARCSIN()	321
ARCSINHYP()	323
ARCTAN()	324
ARCTAN2()	325
ARCTANHYP()	326
AUFRUNDEN()	327
BASIS()	328
BOGENMASS()	329
COS()	330
COSEC()	332
COSECHYP()	333
COSHYP()	334
COT()	335
COTHYP()	337
DEZIMAL()	337
EXP()	339
FAKULTÄT()	340
GANZZAHL()	341
GERADE()	341
GGT()	342
GRAD()	343
KGV()	344
KOMBINATIONEN()	345
KOMBINATIONEN2()	346
KÜRZEN()	347
LN()	348
LOG()	348
LOG10()	350
MDET()	351
MEINHEIT()	352
MINV()	353
MMULT()	354
OBERGRENZE.MATHEMATIK()	355
PI()	356
POLYNOMIAL()	357
POTENZ()	358
POTENZREIHE()	359
PRODUKT()	360
QUADRATESUMME()	361
QUOTIENT()	362
REST()	363
RÖMISCH()	365
RUNDEN()	366
SEC()	367
SECHYP()	368
SIN()	369
SINHYP()	371
SUMME()	372
SUMMENPRODUKT()	374
SUMMEWENN()	374
SUMMEWENNS()	376
SUMMEX2MY2()	377
SUMMEX2PY2()	378
SUMMEXMY2()	379

Funktion	Seite	Funktion	Seite
TAN()	380	VRUNDEN()	387
TANHYP()	382	WURZEL()	388
TEILERGEBNIS()	383	WURZELPI()	389
UNGERADE()	384	ZUFALLSBEREICH()	390
UNTERGRENZE.MATHEMATIK()	385	ZUFALLSZAHL()	391
VORZEICHEN()	386	ZWEIFAKULTÄT()	392

5.1 Einsatzbereiche

Es wäre vermessen, an dieser Stelle die Einsatzbereiche der im Folgenden behandelten mathematischen und trigonometrischen Funktionen auflisten zu wollen. Mathematische Verfahren sind heute in allen Gebieten einsetzbar, in denen ausreichend formalisierbare Strukturen vorkommen. Stattdessen soll nur kurz angedeutet werden, wie die in diesem Kapitel beschriebenen Funktionen gruppiert werden können.

5.1.1 Mathematische Basisoperationen

Eine große Gruppe betrifft allgemeine mathematische Aufgaben. Dazu gehören Funktionen für das Addieren, Multiplizieren, Potenzieren, Radizieren und Logarithmieren wie `SUMME()`, `PRODUKT()`, `POTENZ()`, `WURZEL()` oder `LOG()`, wobei für einige Grundfunktionen noch mehrere Varianten angeboten werden, etwa `SUMMEWENN()`, `SUMMEWENNS()`, `LN()`, `LOG10()`.

5.1.2 Werte runden

Wenn Sie in Excel einen Wert, der drei Dezimalstellen hat, durch ein entsprechendes Format mit zwei Dezimalstellen anzeigen lassen, wird Excel den Wert nach der üblichen Regel runden, also beispielsweise für `3,345` den Wert `3,35` ausgeben, für `3,344` aber nur `3,34`. Obwohl der Wert in der Zelle gerundet erscheint, bleibt der tatsächliche Wert weiter gespeichert, und Rechenoperationen, die diesen Wert verwenden, arbeiten mit dem ungerundeten Wert. Um Werte so zu runden, dass das Rundungsergebnis auch tatsächlich den Inhalt der Zelle bestimmt, benutzen Sie die Rundungsfunktionen.

Gleich zehn Funktionen haben mit dem Runden von Werten zu tun. Es ist nicht immer ganz einfach zu entscheiden, in welcher Situation welche Funktion angemessen ist. Dabei geht es häufig darum, Werte mit vielen Dezimalstellen in eine für die weiteren Be-

rechnungen leichter zu handhabende Form zu bringen. Der andere Anwendungsbereich betrifft Verfahren, bei denen es darum geht, Werte an bestimmte Intervallgrenzen anzupassen, etwa bei der Preisgestaltung oder bei der Aufstellung von Zeitplänen, bei denen bestimmte Intervalle benötigt werden. Zu dieser Gruppe gehören die Funktionen `ABRUNDEN()`, `AUFRUNDEN()`, `RUNDEN()`, `GERADE()`, `UNGERADE()`, `VRUNDEN()`, `GANZZAHL()`, `KÜRZEN()`, `OBERGRENZE.MATHEMATIK()` oder `UNTERGRENZE.MATHEMATIK()`.

Runden von Zahlen

Zahl	Stellen	RUNDEN()	AUFRUNDEN()	ABRUNDEN()
12347,85	0	12348	12348	12347
12347,23	0	12347	12348	12347
12347,85	1	12347,9	12347,9	12347,8
12347,23	1	12347,2	12347,3	12347,2
12347,85	2	12347,85	12347,85	12347,85
12347,23	2	12347,23	12347,23	12347,23
12347,85	-1	12350	12350	12340
12344,23	-1	12340	12350	12340
12357,85	-2	12400	12400	12300
12344,23	-2	12300	12400	12300

Abbildung 5.1 Vergleich der Rundungsfunktionen

5.1.3 Rechnen mit Matrizen

Ein spezielles Gebiet, für das Excel eine größere Gruppe von Funktionen zur Verfügung stellt, ist das Rechnen mit Matrizen. Dazu gehören die Funktionen `MDET()`, `MEINHEIT()`, `MINV()`, `MMULT()` und `MTRANS()`, aber auch Funktionen, mit denen die Quadrate von Matrizen addiert oder die Differenzen der Matrizen potenziert werden können, wie etwa `SUMMEX2PY2()` oder `SUMMEXMY2()`. Matrixfunktionen helfen beispielsweise bei der Lösung von Gleichungen mit mehreren Unbekannten.

5.1.4 Trigonometrische und hyperbolische Funktionen

Neben den allgemeinen mathematischen Funktionen gibt es noch zahlreiche Funktionen für zwei Spezialgebiete der Mathematik. Das eine ist die Trigonometrie, das Teilgebiet der Geometrie, das mit Berechnungen zu Dreiecken zu tun hat, das andere ist der Bereich der hyperbolischen Funktionen.

5.1.5 Neue Funktionen in Excel 2010 und Excel 2013

Excel 2010 brachte in der Kategorie der mathematischen Funktionen nur einige wenige Neuerungen: AGGREGAT(), OBERGRENZE.GENAU(), UNTERGRENZE.GENAU(). Verbessert wurden folgende mathematische Funktionen: ARCSINHYP(), MOD(), ZUFALLSZAHL() und ZWEIFAKULTÄT().

Excel 2013 bringt eine ganze Reihe von neuen Funktionen in dieser Kategorie. Dabei geht es hauptsächlich um Funktionen, die gleichwertigen Funktionen in Dokumenten entsprechen, die mit Office-Paketen wie *OpenOffice* oder *LibreOffice* erstellt wurden. Dazu gehören die trigonometrischen Funktionen ARCCOT(), ARCCOTHYP(), COSEC(), COSECHYP(), COT(), COTHYP(), SEC(), SECHYP(). Neu sind ebenfalls die Funktionen ARABISCH(), BASIS() und DEZIMAL(), die mit Umwandlungen zwischen Zeichenfolgen und Zahlen zu tun haben. Für kombinatorische Berechnungen gibt es in Excel 2013 eine neue Variante: KOMBINATIONEN2(). Für Berechnungen mit Matrizen steht nun mit MEINHEIT() eine Funktion zur Berechnung der Einheitsmatrix zur Verfügung.

Die beiden Funktionen OBERGRENZE.MATHEMATIK() und UNTERGRENZE.MATHEMATIK() ersetzen die in der Version 2010 eingeführten Funktionen OBERGRENZE.GENAU() sowie UNTERGRENZE.GENAU(). Dafür wurden die beiden älteren Funktionen OBERGRENZE() und UNTERGRENZE() in die Kategorie **Kompatibilität** verschoben, sollten also in Zukunft möglichst nicht mehr verwendet werden.

5.2 Zu den trigonometrischen Funktionen

Zu den zahlreichen trigonometrischen Funktionen, die Excel bereitstellt, vorweg einige Erläuterungen. Die Winkelfunktionen, die nicht nur in der Geometrie, sondern auch in all jenen wissenschaftlichen Bereichen benötigt werden, die sich mit Schwingungen im weitesten Sinne (Schall, Licht, Elektrizität, Mechanik) befassen, verdanken sich ursprünglich Berechnungen am rechtwinkligen Dreieck.

Abbildung 5.2 ruft die am rechtwinkligen Dreieck verwendeten Bezeichnungen in Erinnerung. Die Seiten, die den rechten Winkel bilden, werden Katheten genannt, wobei die Seite, die dem Winkel α gegenüberliegt, als Gegenkathete, die am Winkel anliegende Seite dagegen als Ankathete bezeichnet wird. Die Seite gegenüber dem rechten Winkel, die zwangsläufig zugleich die längste Seite ist, wird Hypotenuse genannt.

Die Summe der Winkel im Dreieck ist bekanntlich immer 180 Grad. Die Verhältnisse der verschiedenen Seiten und der durch die Seiten gebildeten Winkel lassen sich mit Hilfe der trigonometrischen Funktionen berechnen.

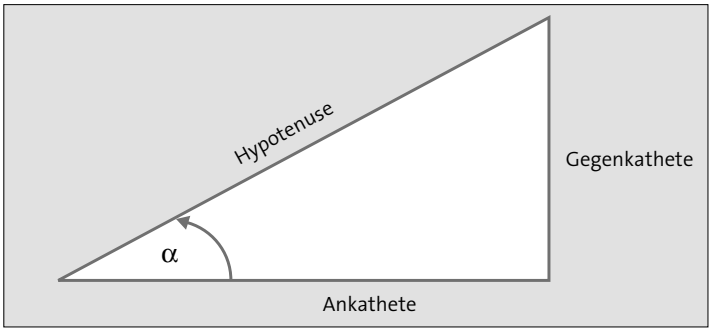


Abbildung 5.2 Bezeichnungen in einem rechtwinkligen Dreieck

Beispielsweise wird das Verhältnis der Länge der Gegenkathete zur Länge der Hypotenuse beim rechtwinkligen Dreieck als Sinus des Winkels bezeichnet, der der Gegenkathete gegenüberliegt. Der Kosinus des gleichen Winkels bezeichnet das Verhältnis von Ankathete zu Hypotenuse. Die Winkelfunktionen sind zwar in Bezug auf rechtwinklige Dreiecke definiert, da sich aber jedes nicht rechtwinklige Dreieck durch Einzeichnen einer Höhe in zwei rechtwinklige Dreiecke zerlegen lässt, sind die Funktionen universell anwendbar, wo es um Dreiecke geht.

5.2.1 Darstellung am Einheitskreis

Die Beziehungen zwischen Winkel und Seiten lassen sich gut an dem sogenannten Einheitskreis verdeutlichen, dessen Radius als 1 gesetzt ist. Abbildung 5.3 zeigt diejenigen Strecken, die die Werte für den Sinus und Kosinus des Winkels α darstellen. Wächst der Winkel auf 90 Grad, ist der Sinuswert 1, der Kosinuswert 0.

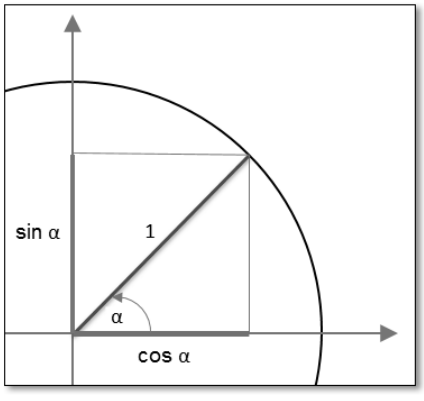


Abbildung 5.3 Geometrische Darstellung von Sinus und Kosinus am Einheitskreis

Mit Hilfe des Einheitskreises im Koordinatensystem lässt sich zudem der Definitionsbereich der Winkelfunktionen über den Bereich von 0 bis 360 Grad erweitern. Der Einheitskreis wird dabei in vier jeweils 90 Grad große Quadranten unterteilt. Der Quadrant bestimmt dann auch die Verwendung von Vorzeichen für die Werte der Winkelfunktionen.

Gearbeitet wird in Excel allerdings immer mit Winkelgrößen, die nicht in Grad, sondern im Bogenmaß angegeben sind. Als Bogenmaß eines Winkels wird die Länge des Kreisbogens bezeichnet, den der Winkel aus dem Einheitskreis (Kreis mit dem Radius 1) ausschneidet. Da der Umfang eines Kreises `2 * r * π` ist, beträgt der Umfang des Einheitskreises `2 * π`, das Bogenmaß des Winkels 360 Grad ist also `2 * π`.

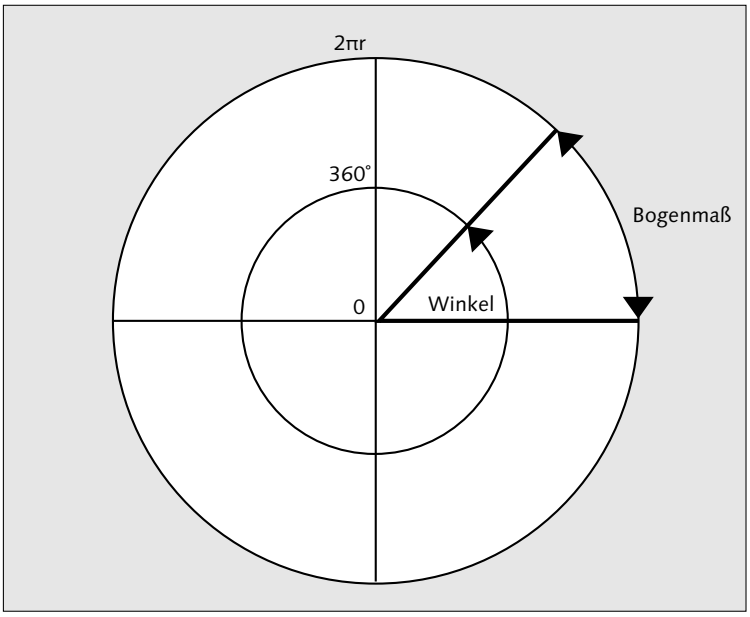


Abbildung 5.4 Die Beziehung von Grad zu Bogenmaß

Die Umrechnung von Grad in Bogenmaß ist demnach:

```
Grad = Bogenmaß * 180 / π
Bogenmaß = Grad * π / 180
```

Die vier gebräuchlichsten Winkelfunktionen geben die Verhältnisse bestimmter Seiten eines rechtwinkligen Dreiecks in Abhängigkeit von einem Winkel an:

```
SIN(x) = Gegenkathete/Hypotenuse
COS(x) = Ankathete/Hypotenuse
```

```
TAN(x) = Gegenkathete/Ankathete
COT(x) = Ankathete/Gegenkathete
```

Der Kotangens war bisher als eigene Funktion nicht in Excel enthalten, ließ sich aber leicht als Kehrwert des Tangens berechnen:

```
COT(x) = 1 / TAN(x)
```

Mit Excel 2013 steht Ihnen die Funktion `COT()` aber direkt zur Verfügung.

Der `TAN(x)` ist gleichzeitig als der Quotient `SIN(x) / COS(x)` definiert, `COT(x)` ist entsprechend als Quotient `COS(x) / SIN(x)` definiert.

Abbildung 5.5 zeigt die geometrische Interpretation von Tangens und Kotangens.

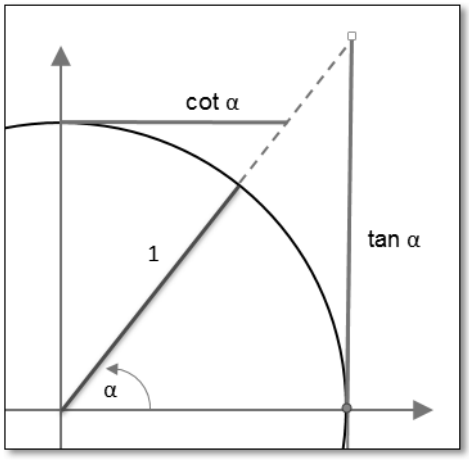


Abbildung 5.5 Geometrische Darstellung von Tangens und Kotangens am Einheitskreis

Außerdem gilt:

```
SIN(BOGENMASS(90)-x)=COS(x)
COS(BOGENMASS(90)-x)=SIN(x)
TAN(BOGENMASS(90)-x)=COT(x)
COT(BOGENMASS(90)-x)=TAN(x)
```

Die Vorsilbe »Ko« nennt also in allen Fällen den Komplimentärwinkel.

Neu in Excel 2013 sind die Funktionen für Sekans und Kosekans:

```
SEC(x) = Hypotenuse/Ankathete
COSEC(x) = Hypotenuse/Gegenkathete
```

Dabei gilt:

```
SEC(x) = 1 / COS(x)
COSEC(x) = 1 / SIN(x)
```

Abbildung 5.6 zeigt die geometrische Interpretation am Einheitskreis: Der Funktionswert des Sekans wird durch die Strecke OT, der des Kosekans durch OK wiedergegeben.

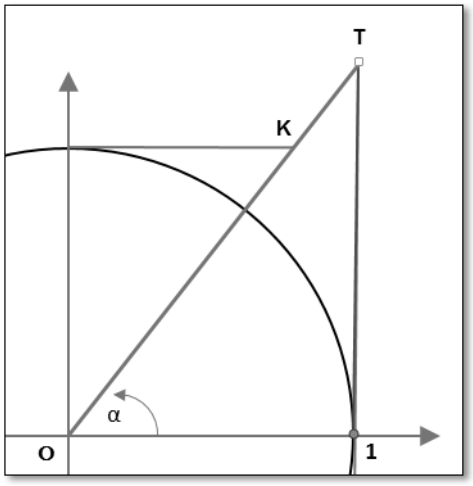


Abbildung 5.6 Geometrische Darstellung von Sekans und Kosekans am Einheitskreis

5.2.2 Umkehrfunktionen

Für die bisher beschriebenen Winkelfunktionen sind entsprechende Umkehrfunktionen definiert. Sie werden eingesetzt, wenn der Wert der Winkelfunktion bekannt ist, um daraus den Winkel selbst zu berechnen. Wissen Sie beispielsweise, dass `SIN(x) = 0,2` ist, können Sie mit der Umkehrfunktion `ARCSIN(0,2)` den entsprechenden Winkel finden, es ist 0,2014 im Bogenmaß, also etwa 11.537 Grad.

Das Präfix `Arc` steht für `Arcus` (Bogenmaß). Mit den passenden Arkusfunktionen wird also ausgehend vom Ergebnis einer Winkelfunktion der zugehörige Winkel im Bogenmaß ermittelt. Es gilt dabei für alle Winkelfunktionen eine Beziehung nach dem Muster:

Wenn `y = SIN(x)`, dann gilt `ARCSIN(y) = x`.

In Tabelle 5.1 sind die Winkelfunktionen ihren in Excel 2013 verfügbaren Umkehrungen gegenübergestellt. Gleichzeitig wird jedes Mal der Wertebereich angegeben.

Funktion	Umkehrfunktion	Name	Wertebereich
SIN()	ARCSIN()	Arkussinus	–90 Grad ≤ α ≤ 90 Grad
COS()	ARCCOS()	Arkuskosinus	0 Grad ≤ α ≤ 180 Grad
TAN()	ARCTAN()	Arkustangens	–90 Grad ≤ α ≤ 90 Grad
	ARCTAN2()	Arkustangens	–180 Grad < α ≤ 180 Grad
COT()	ARCCOT()	Arkuskotangens	–180 Grad < α ≤ 180 Grad

Tabelle 5.1 Winkelfunktionen und ihre Umkehrfunktionen

Für die neu eingeführten Funktionen SEC() und COSEC() fehlen die entsprechenden Umkehrfunktionen für Arkussekans und Arkuskosekans. Sie lassen sich mit Hilfe anderer Arkusfunktionen berechnen:

```
arcsec(x) = arccos(1 / x)
arccsc(x) = arcsin(1 / x)
```

5.3 Zu den hyperbolischen Funktionen

Die hyperbolischen Funktionen sind, trotz ihrer an Winkelfunktionen erinnernden Namen, keine eigentlichen Winkelfunktionen. Das wird schon daran deutlich, dass sie im Gegensatz zu den Winkelfunktionen nicht periodisch verlaufen. Die Namensgebung rechtfertigt sich aber aufgrund einer großen formalen Übereinstimmung bei den Beziehungen zwischen den einzelnen Funktionen sowie aus mathematischen Zusammenhängen zwischen Winkelfunktionen und hyperbolischen Funktionen.

Dass die Funktionen als hyperbolisch bezeichnet werden, rührt daher, dass sie geometrisch anhand einer gleichseitigen Hyperbel mit der Gleichung x^2 – y^2 = 1 gedeutet werden, also einer Kurve, die aus zwei zueinander symmetrischen Ästen besteht, die sich ins Unendliche erstrecken.

Die Funktionen stellen eine Verbindung zwischen der Fläche her, die von einer vom Nullpunkt ausgehenden Geraden, ihrem Spiegelbild an der x-Achse und der genannten Hyperbel begrenzt wird, sowie der Länge verschiedener Strecken, die sich dabei ergeben. So ist sinh(α) die positive y-Koordinate des Schnittpunkts der Geraden mit der Hyperbel, cosh(α) die dazugehörige x-Koordinate, tanh(α) ist die y-Koordinate bei x = 1.

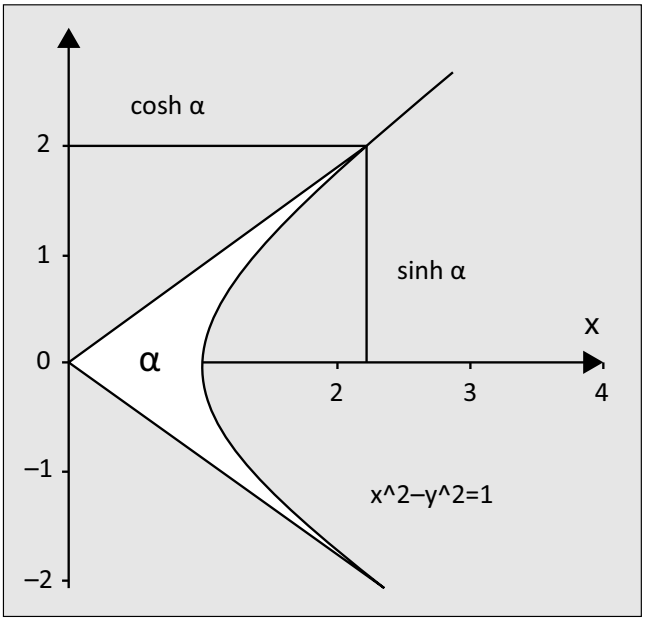


Abbildung 5.7 Deutung der Funktionen an der gleichseitigen Hyperbel

Anwendung finden derartige Funktionen z. B. in einigen statistischen Näherungsverfahren, bei statischen Berechnungen und in der Analysis.

Die Kennzeichnung der inversen hyperbolischen Funktionen, die als Areafunktionen bezeichnet werden, mit dem Präfix `ARC` (analog zu den Arkusfunktionen der Winkelfunktionen) ist allerdings trotz der formalen Ähnlichkeit zu den Arkusfunktionen eher unglücklich: Die Areafunktionen liefern keinen Winkel im Bogenmaß (lat.: Arcus), sondern in der geometrischen Deutung eine Fläche (lat.: Area).

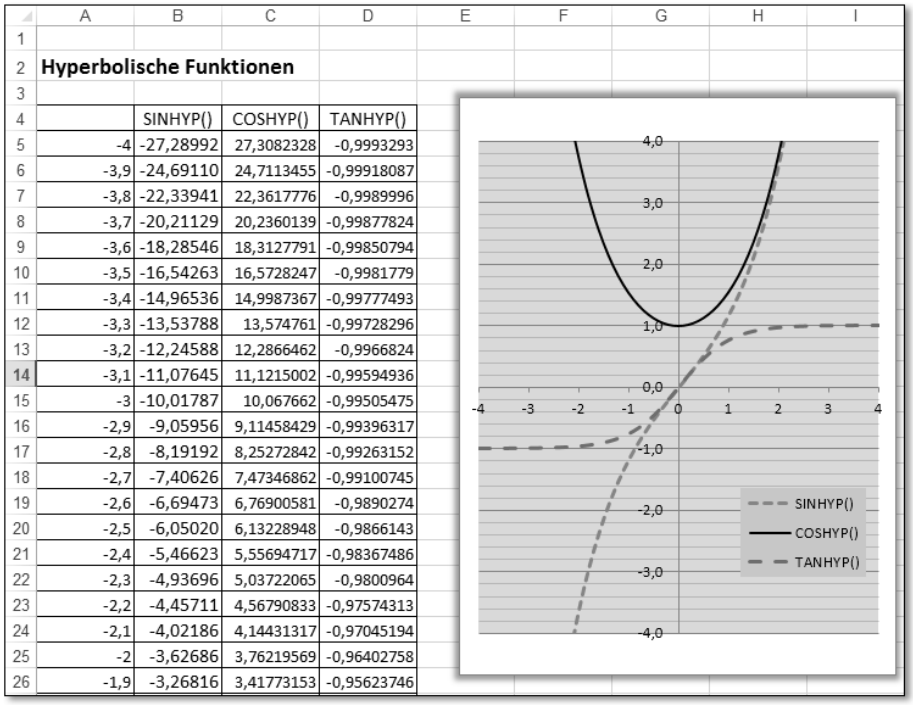
Der Kotangens Hyperbolicus war bisher in Excel nicht eigens als Funktion enthalten, da er der Kehrwert des Tangens Hyperbolicus ist:

```
coth(x) = 1/tanh(x)
```

Um die Dinge zu vereinfachen, steht in Excel 2013 nun eine eigene Funktion zur Verfügung: `COTHYP()`. Neu sind auch die beiden hyperbolischen Funktionen `ARCCOTHYP()` und `COSECHYP()`.

Für die Areafunktionen gilt (analog zu den Winkelfunktionen) in Bezug auf die hyperbolischen Funktionen eine Beziehung nach dem Muster:

Wenn `y = SINHYP(x)`, dann gilt `ARCSINHYP(y) = x`.



Hyperbolische Funktionen

	SINHYP()	COSHYP()	TANHYP()
-4	-27,28992	27,3082328	-0,9993293
-3,9	-24,69110	24,7113455	-0,99918087
-3,8	-22,33941	22,3617776	-0,9989996
-3,7	-20,21129	20,2360139	-0,99877824
-3,6	-18,28546	18,3127791	-0,99850794
-3,5	-16,54263	16,5728247	-0,9981779
-3,4	-14,96536	14,9987367	-0,99777493
-3,3	-13,53788	13,574761	-0,99728296
-3,2	-12,24588	12,2866462	-0,9966824
-3,1	-11,07645	11,1215002	-0,99594936
-3	-10,01787	10,067662	-0,99505475
-2,9	-9,05956	9,11458429	-0,99396317
-2,8	-8,19192	8,25272842	-0,99263152
-2,7	-7,40626	7,47346862	-0,99100745
-2,6	-6,69473	6,76900581	-0,9890274
-2,5	-6,05020	6,13228948	-0,9866143
-2,4	-5,46623	5,55694717	-0,98367486
-2,3	-4,93696	5,03722065	-0,9800964
-2,2	-4,45711	4,56790833	-0,97574313
-2,1	-4,02186	4,14431317	-0,97045194
-2	-3,62686	3,76219569	-0,96402758
-1,9	-3,26816	3,41773153	-0,95623746

Abbildung 5.8 Hyperbolische Funktionen

Sowohl die Winkelfunktionen als auch die hyperbolischen Funktionen sind mathematisch sowohl für reelle als auch für komplexe Zahlen definiert. Die in der Kategorie **Mathematik und Trigonometrie** angebotenen Funktionen von Excel arbeiten aber ausschließlich mit reellen Werten. Für entsprechende Berechnungen mit komplexen Zahlen stellt Excel parallel zahlreiche Funktionen in der Kategorie **Technisch** zur Verfügung, die alle mit dem Präfix `IM` (für imaginär) anfangen, beispielsweise `IMSIN()`, `IMSINHYP()`, `IMCOS()`, `IMTAN()` etc.

5.4 Referenz der mathematischen Funktionen

ABRUNDEN()

ROUNDDOWN()

Syntax: `ABRUNDEN(`**`Zahl; Anzahl_Stellen`**`)`

Beispiel: `=ABRUNDEN(-657,65;-1)`

Ergebnis: `-650`

Die Funktion ABRUNDEN() rundet eine mit dem Argument Zahl angegebene reelle Zahl in Richtung 0 auf dem Zahlenstrahl. Positive Zahlen werden also abgerundet, negative Zahlen aufgerundet. Anders als bei der Funktion RUNDEN() ist die Richtung der Rundung also festgelegt und hängt nicht von dem jeweiligen Wert ab.

Mit dem Argument Anzahl_Stellen wird die Anzahl der Stellen angegeben, die beim Runden berücksichtigt wird. Ist Anzahl_Stellen gleich 1 oder größer, dann wird auf die entsprechende Anzahl Dezimalstellen abgeschnitten, ist es 0, werden die Dezimalstellen abgeschnitten, so dass die Ganzzahl übrig bleibt. Ist das Argument negativ, wird um die angegebenen Stellen nach links gerundet, und zwar so, dass sich die gerundete Zahl in Richtung 0 bewegt.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist Anzahl_Stellen größer als die Anzahl der Stellen, die der Wert von Zahl hat, werden nur die vorhandenen Stellen ausgegeben.

	A	B	C
1			
2	**Abrunden von Zahlen**		
3			
4	Zahl	Stellen	ABRUNDEN()
5	12347,85	0	12347
6	12347,85	1	12347,8
7	12347,85	2	12347,85
8	12347,85	3	12347,85
9	12347,85	-1	12340
10	12357,85	-2	12300
11	12347,85	-3	12000
12	-12347,85	0	-12347
13	-12347,85	1	-12347,8
14	-12347,85	2	-12347,85
15	-12347,85	3	-12347,85
16	-12347,85	-1	-12340
17	-12347,85	-2	-12300
18	-12347,85	-3	-12000

Abbildung 5.9 Abrunden mit frei wählbarer Stellenanzahl

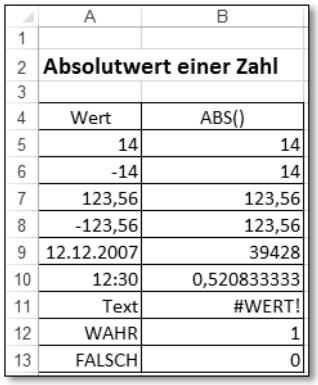
ABS()

ABS()

Syntax: ABS(**Zahl**)
Beispiel: =ABS(-8)
Ergebnis: 8

Die Funktion ABS() eliminiert die Vorzeichen von Zahlenwerten. Ganz gleich, ob ABS() auf Zahlen mit negativem oder positivem Vorzeichen angewendet wird, als Ergebnis wird der nicht negative Betrag dieser Zahlen geliefert.

In der mathematischen Schreibweise werden die absoluten Werte mit zwei senkrechten Strichen gekennzeichnet: |x|. Der Begriff »absolut« bezieht sich darauf, dass eine Zahl x und ihre negative Entsprechung –x auf dem Zahlenstrahl denselben Abstand vom Nullpunkt haben.



	A	B
1		
2	**Absolutwert einer Zahl**	
3		
4	Wert	ABS()
5	14	14
6	-14	14
7	123,56	123,56
8	-123,56	123,56
9	12.12.2007	39428
10	12:30	0,520833333
11	Text	#WERT!
12	WAHR	1
13	FALSCH	0

Abbildung 5.10 Absolutwert verschiedener Werte

Nicht numerische Werte ergeben den Fehler #WERT!.

AGGREGAT()

AGGREGATE()

Syntax1:	AGGREGAT(**Funktion**; **Optionen**; **Bezug1**; Bezug2 ...)
Beispiel:	=AGGREGAT(1; 6; Wertebereich)
Ergebnis:	liefert den Mittelwert der Werte in Wertebereich und ignoriert dabei die Fehlerwerte (siehe Abbildung 5.11)
Syntax2:	AGGREGAT(**Funktion**; **Optionen**; **Array**; k)
Beispiel:	=AGGREGAT(14; 6; {120.290.120.450.290};2)
Ergebnis:	290

Die Funktion AGGREGAT() erlaubt es, Daten in Zellbereichen oder Matrizen auf unterschiedliche Weise auszuwerten und dabei die normalerweise stattfindende Reaktion auf fehlerhafte Werte zu vermeiden, die bei Funktionen wie SUMME(), MITTELWERT() etc.

auftritt. Die Bildung der Summe oder des Mittelwerts eines Bereichs scheitert immer dann, wenn Fehlerwerte in diesem Bereich vorkommen, wie in Abbildung 5.11 in Zeile 16 zu sehen ist.

Die Art der Auswertung wird über den Wert für das Argument `Funktion` bestimmt. Werte von `1` bis `19` sind erlaubt.

Nr.	Funktion
1	MITTELWERT
2	ANZAHL
3	ANZAHL2
4	MAX
5	MIN
6	PRODUKT
7	STABW.S
8	STABW.N
9	SUMME
10	VAR.S
11	VAR.P
12	MEDIAN
13	MODUS.EINF
14	KGRÖSSTE
15	KKLEINSTE
16	QUANTIL.INKL
17	QUARTILE.INKL
18	QUANTIL.EXKL
19	QUARTILE.EXKL

Tabelle 5.2 Werte für das Argument `Funktion`

Das zweite Argument `Optionen` bestimmt, welche Werte die Auswertung ignorieren soll. Dabei ist es auch möglich, die Ergebnisse von `TEILERGEBNIS()`- oder `AGGREGAT()`-Funktionen, die sich innerhalb des auszuwertenden Bereichs befinden, aus der Berechnung herauszunehmen, damit die Werte nicht doppelt berücksichtigt werden.

Nr.	Option
0	verschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
1	ausgeblendete Zellen, verschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
2	Fehlerwerte, verschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
3	Fehlerwerte, ausgeblendete Zellen, verschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
4	Leerwerte ignorieren
5	ausgeblendete Zellen ignorieren
6	Fehlerwerte ignorieren
7	ausgeblendete Zellen und Fehlerwerte ignorieren

Tabelle 5.3 Werte für das Argument Optionen

In Syntax1 wird mit dem dritten Argument Bezug1 ein Bezug auf einen Zellbereich mit auszuwertenden Daten angegeben. Weitere Bezüge – bis zu 253 insgesamt – können folgen.

In Syntax2 werden die auszuwertenden Daten über das Argument Array bestimmt. Array ist eine Matrix, eine Matrixformel oder ein Bezug auf einen Bereich mit Werten. Benötigt die mit Funktion gewählte Auswertung noch ein weiteres Element, wird dieses als Argument k angegeben. In dem in Abbildung 5.11 gezeigten Beispiel wird in Zelle D7 die Funktion KKLEINSTE() verwendet, und mit k wird angegeben, dass der zweitkleinste Wert gesucht wird.

	A	B	C	D	E
1					
2	**AGGREGAT()**				
3					
4	Wertebereich			Ergebnis	Formel
5	120	120	Summe:	4.172	=AGGREGAT(9;6; Wertebereich)
6	290	290	Mittelwert:	219,58	=AGGREGAT(1;6; Wertebereich)
7	120	#DIV/0!	2. kleinster Wert	390,00	=AGGREGAT(14;6;{120.290.170.450.390};2)
8	#NAME?	450			
9	290	290			
10	33	120			
11	140	130			
12	220	#NV			
13	290	290			
14	219	300			
15	120	340			
16	#NAME?	#DIV/0!			
17	=SUMME(A5:A15)	=MITTELWERT(B5:B15)			

Abbildung 5.11 Beispiele für die AGGREGAT()-Funktion

Wird ein zweites Bezugsargument benötigt, wie bei den Funktionen QUARTILE.INKL(Matrix; Quartil) und QUARTILE.EXKL(Matrix; Quartil), aber nicht angegeben, oder ist ein Bezug ein 3D-Bezug, liefert die Funktion den Fehler #WERT!. Das gilt auch, wenn ungültige Argumente für Funktion oder Optionen angegeben werden.

Freundlicherweise unterstützt Sie Excel bei der Angabe der Werte für Funktion und Optionen, indem die möglichen Werte als Liste angeboten werden, wie Abbildung 5.12 zeigt.

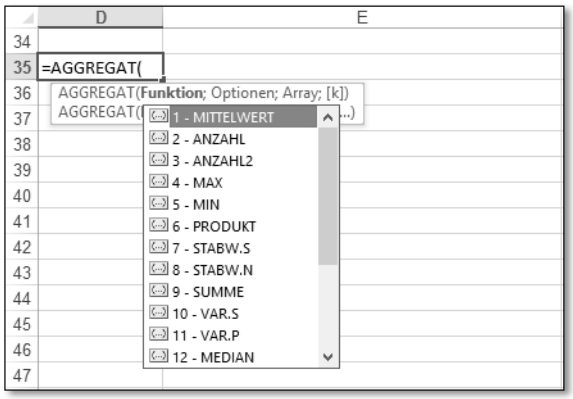


Abbildung 5.12 Angebot der Funktionsliste bei der Formeleingabe

Zu beachten ist, dass die Funktion nur dort sinnvoll angewendet werden kann, wo die Werte, die zusammengefasst werden sollen, spaltenweise angeordnet sind. Dagegen wird beispielsweise bei zeilenweisen Quersummen nicht berücksichtigt, ob eine Spalte ausgeblendet ist.

ARABISCH()

ARABIC()

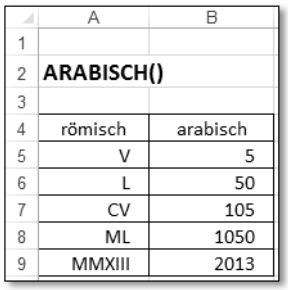
Syntax: ARABISCH(**Text**)
Beispiel: =ARABISCH("MMXIII")
Ergebnis: 2013

Die in Excel 2013 neue Funktion ARABISCH() konvertiert die mit dem Argument Text angegebenen Zeichenfolge in arabische Zahlen. Sie dient der Umwandlung von römischen Zahlen in arabische Zahlen.

Text darf maximal 255 Zeichen enthalten. Die größtmögliche Zahl ist 255000. Wird eine leere Zeichenfolge angegeben, ist das Ergebnis 0. Groß bzw. Kleinschreibung wird

ignoriert. Führende oder nachfolgende Leerzeichen werden ebenfalls ignoriert. Negative Zahlen werden unterstützt, wenn ein Minuszeichen vorgesetzt wird, obwohl die Römer negative Zahlen noch gar nicht kannten.

Ist `Text` kein gültiger Wert, liefert die Funktion den Fehlerwert `#WERT!`. Dazu gehören alle Zahlen, Datums- und Zeitangaben oder sonstige Zeichenfolgen, für die es keine gültige römische Zahl gibt.



	A	B
1		
2	**ARABISCH()**	
3		
4	römisch	arabisch
5	V	5
6	L	50
7	CV	105
8	ML	1050
9	MMXIII	2013

Abbildung 5.13 Umwandlung römischer Zahlen in arabische

Die Funktion ist die Umkehrung der Funktion `RÖMISCH()`, die arabische Zahlen in römische übersetzt. Dort finden Sie auch noch einige Informationen zum römischen Zahlensystem.

ARCCOS()

ACOS()

Syntax: `ARCCOS(`**`Zahl`**`)`
Beispiel: `=ARCCOS(0,9)`
Ergebnis: `0,451026`

Die Funktion `ARCCOS()` berechnet zu dem mit dem Argument `Zahl` gegebenen Kosinuswert, der das Verhältnis von Ankathete zu Hypotenuse im rechtwinkligen Dreieck angibt, den anliegenden Winkel im Bogenmaß, also den Arkuskosinus, siehe dazu auch Abbildung 5.26.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Da jeder Winkel im Vollkreis zweimal vorkommt, wird der Wertebereich eingeschränkt. Für Kosinuswerte zwischen –1 und 1 liegen die Winkelwerte im Bogenmaß zwischen 0 und π, in Grad angegeben zwischen 0 und 180 Grad.

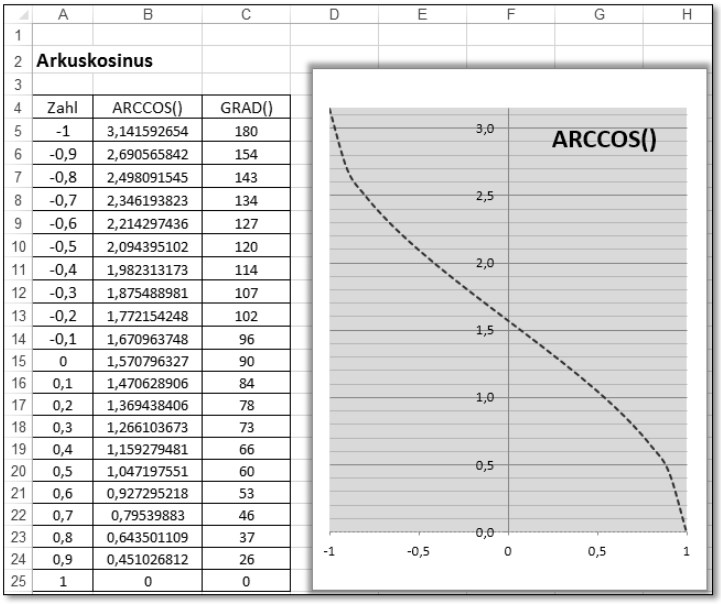
Soll der Winkel in Grad angegeben werden, müsste die Formel im Beispiel lauten:

```
=ARCCOS(0,9)*180/PI()
```

Das Ergebnis ist dann 25,84 Grad. Dazu kann aber auch die Funktion `GRAD()` verwendet werden:

```
=GRAD(ARCCOS(0,9))
```

Die Funktion ist, wie die Abbildung 5.14 zeigt, monoton fallend.



	A	B	C
1			
2	**Arkuskosinus**		
3			
4	Zahl	ARCCOS()	GRAD()
5	-1	3,141592654	180
6	-0,9	2,690565842	154
7	-0,8	2,498091545	143
8	-0,7	2,346193823	134
9	-0,6	2,214297436	127
10	-0,5	2,094395102	120
11	-0,4	1,982313173	114
12	-0,3	1,875488981	107
13	-0,2	1,772154248	102
14	-0,1	1,670963748	96
15	0	1,570796327	90
16	0,1	1,470628906	84
17	0,2	1,369438406	78
18	0,3	1,266103673	73
19	0,4	1,159279481	66
20	0,5	1,047197551	60
21	0,6	0,927295218	53
22	0,7	0,79539883	46
23	0,8	0,643501109	37
24	0,9	0,451026812	26
25	1	0	0

Abbildung 5.14 Die Arkuskosinus-Funktion und ihre grafische Darstellung

Die Funktion kann verwendet werden, um den Neigungswinkel zu berechnen, wenn das Verhältnis von Ankathete und Hypotenuse in einem spitzen Winkel bekannt ist. Wenn beispielsweise eine Feuerleiter von 10 m Länge so an eine Hauswand gelehnt ist, dass der Fuß der Leiter 3 m Abstand von der Hauswand hält, hat der Kosinus den Wert `=3/10`. Daraus lässt sich der Arkuskosinus berechnen:

```
=ARCCOS(3/10)
```

ergibt `1,266`, umgewandelt in Grad:

```
=GRAD(ARCCOS(3/10))
```

Daraus ergibt sich ein Neigungswinkel von `72,54` Grad.

ARCCOSHYP()

ACOSH()

Syntax: ARCCOSHYP(**Zahl**)
Beispiel: =ARCCOSHYP(1)
Ergebnis: 0

Die Funktion ARCCOSHYP() ist die Umkehrfunktion der Funktion COSHYP(). Entgegen dem Präfix ARC ist die korrekte Bezeichnung »Areakosinus Hyperbolicus«, siehe dazu auch den Abschnitt 5.3, »Zu den hyperbolischen Funktionen«.

Für das Argument Zahl kann jede beliebige reelle Zahl verwendet werden, die größer oder gleich 1 ist. Der Wertebereich reicht von 0 bis +unendlich. Werte < 1 ergeben den Fehler #ZAHL!, nicht numerische Werte den Fehler #WERT!.

Die Funktion rechnet mit folgender Formel:

$$\text{arcosh}(x) = \ln(x + \sqrt{x^2 - 1})$$

Der Graph der Funktion ist streng monoton steigend, die Nullstelle liegt bei x = 1.

Positiver umgekehrter hyperbolischer Kosinus

Zahl	ARCCOSHYP()
1	0
1,2	0,622362504
1,4	0,867014726
1,6	1,046967915
1,8	1,192910731
2	1,316957897
2,2	1,425416943
2,4	1,522079367
2,6	1,609437912
2,8	1,68923555
3	1,762747174
3,2	1,830938007
3,4	1,894559013
3,6	1,954207529
3,8	2,010367491
4	2,063437069
4,2	2,113748231
4,4	2,161580941
4,6	2,207173683
4,8	2,250731414
5	2,29243167

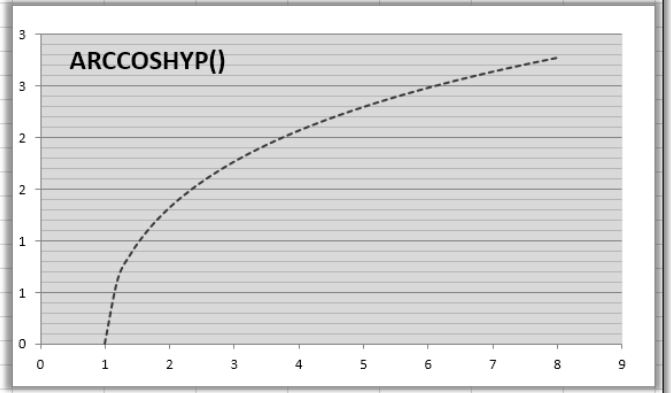


Abbildung 5.15 Die ARCCOSHYP()-Funktion und ihre grafische Darstellung

ARCCOT()

ACOT()

Syntax: ARCCOT(**Zahl**)
Beispiel: =ARCCOT(2)
Ergebnis: 0,463647609

Die in Excel 2013 neue Funktion ARCCOT() berechnet den Arkuskotangens einer Zahl, die den Kotangens eines Winkels angibt, also das Verhältnis von Ankathete zu Gegenkathete im rechtwinkligen Dreieck, siehe dazu auch Abbildung 5.32. Sie ist die Umkehrfunktion des Kotangens.

Nicht numerische Werte ergeben den Fehler #WERT!.

Das Ergebnis liefert den gesuchten Winkel im Bogenmaß mit dem Wertebereich 0 bis PI(). Soll der Winkel in Grad angegeben werden, multiplizieren Sie das Ergebnis mit 180/PI() oder benutzen die Funktion GRAD().

```
=ARCCOT(2) * 180/PI()
=GRAD(ARCCOT(2))
```

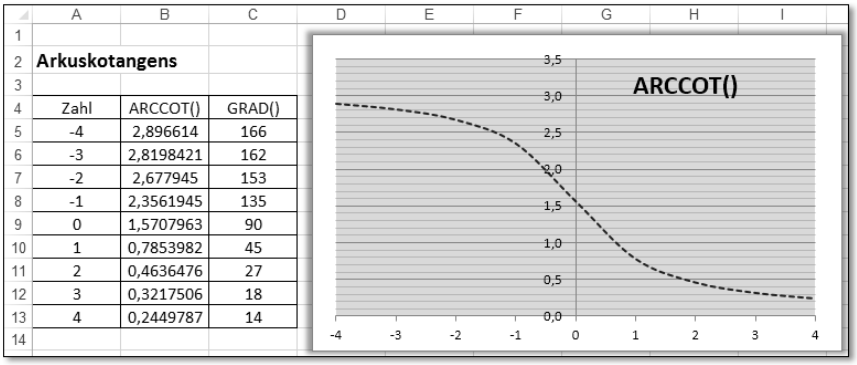


Abbildung 5.16 Die ARCCOT()-Funktion und ihre grafische Darstellung

ARCCOTHYP()

ACOTH()

Syntax: ARCCOTHYP(**Zahl**)
Beispiel: =ARCCOTHYP(2)
Ergebnis: 0,549306144

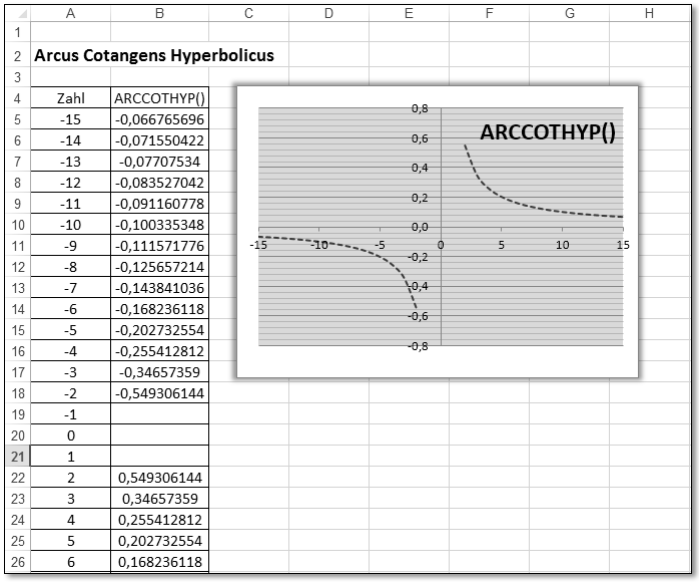
Die in Excel 2013 neue Funktion ARCCOTHYP() berechnet den umgekehrten hyperbolischen Kotangens einer Zahl. Entgegen dem Präfix ARC ist die korrekte Bezeichnung »Areakotangens Hyperbolicus«, siehe dazu auch den Abschnitt 5.3, »Zu den hyperbolischen Funktionen«.

Der absolute Wert von Zahl muss dabei größer 1 sein, ansonsten liefert die Funktion den Fehler #ZAHL!. Bei einem nicht numerischen Wert liefert die Funktion den Fehler #WERT!.

Die Funktion rechnet mit folgender Formel:

$$\text{arcoth}(x) = \frac{1}{2}\ln\left(\frac{x+1}{x-1}\right)$$

Der Graph der Funktion ist streng monoton steigend, es gibt keine Nullstellen.



	A	B	C	D	E	F	G	H
1								
2	**Arcus Cotangens Hyperbolicus**							
3								
4	Zahl	ARCCOTHYP()						
5	-15	-0,066765696						
6	-14	-0,071550422						
7	-13	-0,07707534						
8	-12	-0,083527042						
9	-11	-0,091160778						
10	-10	-0,100335348						
11	-9	-0,111571776						
12	-8	-0,125657214						
13	-7	-0,143841036						
14	-6	-0,168236118						
15	-5	-0,202732554						
16	-4	-0,255412812						
17	-3	-0,34657359						
18	-2	-0,549306144						
19	-1							
20	0							
21	1							
22	2	0,549306144						
23	3	0,34657359						
24	4	0,255412812						
25	5	0,202732554						
26	6	0,168236118						

Abbildung 5.17 Die ARCCOTHYP()-Funktion und ihre grafische Darstellung

ARCSIN()

ASIN()

Syntax: ARCSIN(**Zahl**)
Beispiel: =ARCSIN(0,5)
Ergebnis: 0,523599

Die Funktion ARCSIN() berechnet zu dem mit dem Argument Zahl angegebenen Sinuswert, der im rechtwinkligen Dreieck das Verhältnis von Gegenkathete zu Hypotenuse darstellt, den anliegenden Winkel im Bogenmaß, also den Arkussinus. Da jeder Winkel im Vollkreis zweimal vorkommt, wird der Wertebereich eingeschränkt. Für Sinuswerte zwischen –1 und 1 liegen die Winkelwerte im Bogenmaß zwischen $-\pi/2$ und $\pi/2$, d.h. zwischen –90 und 90 Grad.

Das Ergebnis liefert den gesuchten Winkel im Bogenmaß mit dem Wertebereich 0 bis PI(). Soll der Winkel in Grad angegeben werden, multiplizieren Sie das Ergebnis mit 180/PI() oder benutzen die Funktion GRAD().

```
=ARCSIN(0,5) * 180/PI()
=GRAD(ARCSIN(0,5))
```

ergibt 30 Grad.

Nicht numerische Werte ergeben den Fehler #WERT!.

Die Funktion ist, wie Abbildung 5.18 zeigt, monoton steigend.

	A	B	C
1			
2	**Arkussinus**		
3			
4	Zahl	ARCSIN()	GRAD()
5	-1	-1,570796327	-90
6	-0,9	-1,119769515	-64
7	-0,8	-0,927295218	-53
8	-0,7	-0,775397497	-44
9	-0,6	-0,643501109	-37
10	-0,5	-0,523598776	-30
11	-0,4	-0,411516846	-24
12	-0,3	-0,304692654	-17
13	-0,2	-0,201357921	-12
14	-0,1	-0,100167421	-6
15	0	0	0
16	0,1	0,100167421	6
17	0,2	0,201357921	12
18	0,3	0,304692654	17
19	0,4	0,411516846	24
20	0,5	0,523598776	30
21	0,6	0,643501109	37
22	0,7	0,775397497	44
23	0,8	0,927295218	53
24	0,9	1,119769515	64
25	1	1,570796327	90

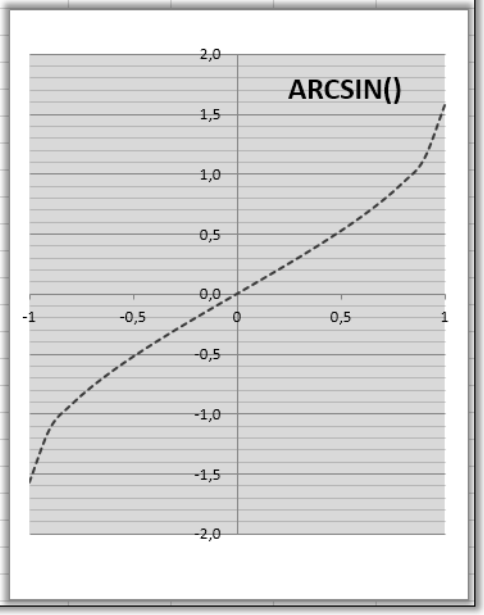


Abbildung 5.18 Die ARCSIN()-Funktion und ihre grafische Darstellung

ARCSINHYP()

ASINH()

Syntax: ARCSINHYP(**Zahl**)
Beispiel: =ARCSINHYP(2)
Ergebnis: 1,443635

Die Funktion ARCSINHYP() ist die Umkehrfunktion der Funktion SINHYP(). Entgegen dem Präfix ARC ist die korrekte Bezeichnung »Areasinus Hyperbolicus«, siehe dazu auch den Abschnitt 5.3, »Zu den hyperbolischen Funktionen«.

Für das Argument Zahl kann jede beliebige reelle Zahl verwendet werden. Der Wertebereich liegt zwischen –unendlich bis +unendlich.

Nicht numerische Werte ergeben den Fehler #WERT!.

Die Funktion rechnet mit folgender Formel:

$$\text{arsinh}(x) = \ln(x + \sqrt{x^2 + 1})$$

Der Graph der Funktion ist streng monoton steigend, die Nullstelle liegt bei x = 0.

Umgekehrter hyperbolischer Sinus

Zahl	ARCSINHYP()
-8	-2,776472281
-7,5	-2,712465305
-7	-2,644120761
-6,5	-2,570814678
-6	-2,491779853
-5,5	-2,406059125
-5	-2,312438341
-4,5	-2,209347709
-4	-2,094712547
-3,5	-1,965720472
-3	-1,818446459
-2,5	-1,647231146
-2	-1,443635475
-1,5	-1,194763217
-1	-0,881373587
-0,5	-0,481211825
0	0
0,5	0,481211825
1	0,881373587
1,5	1,194763217
2	1,443635475

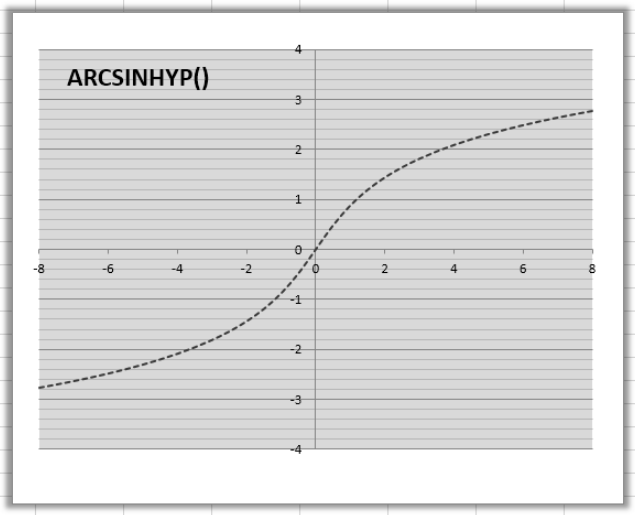


Abbildung 5.19 Der umgekehrte hyperbolische Sinus und der zugehörige Graph

ARCTAN()

ATAN()

Syntax: `ARCTAN(Zahl)`
Beispiel: `=ARCTAN(0,3)`
Ergebnis: `0,291457`

Die Funktion `ARCTAN()` berechnet zu dem mit dem Argument `Zahl` angegebenen Tangenswert, der im rechtwinkligen Dreieck das Verhältnis von Gegenkathete zu Ankathete darstellt, den anliegenden Winkel im Bogenmaß, also den Arkustangens.

Das Ergebnis liefert den gesuchten Winkel im Bogenmaß mit dem Wertebereich $-\pi/2$ und $\pi/2$. Soll der Winkel in Grad angegeben werden, multiplizieren Sie das Ergebnis mit `180/PI()` oder benutzen die Funktion `GRAD()`.

```
=ARCTAN(0,3) * 180/PI()
=GRAD(ARCTAN(0,3))
```

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B	C
1			
2	**Arkustangens**		
3			
4	Zahl	ARCTAN()	GRAD()
5			
6	-57,29	-1,553	-89
7	-28,64	-1,536	-88
8	-19,08	-1,518	-87
9	-14,30	-1,501	-86
10	-11,43	-1,484	-85
11	-9,51	-1,466	-84
12	-8,14	-1,449	-83
13	-7,12	-1,431	-82
14	-6,31	-1,414	-81
15	-5,67	-1,396	-80
16	-5,14	-1,379	-79
17	-4,70	-1,361	-78
18	-4,33	-1,344	-77
19	-4,01	-1,326	-76
20	-3,73	-1,309	-75
21	-3,49	-1,292	-74
22	-3,27	-1,274	-73

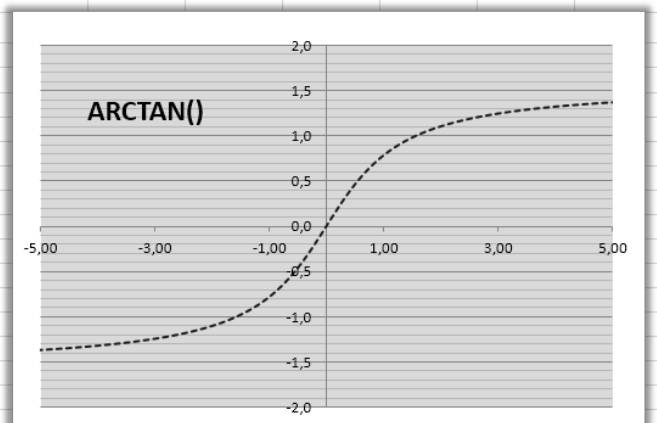


Abbildung 5.20 Der Arkustangens und seine grafische Darstellung

ARCTAN2()

ATAN2()

Syntax: ARCTAN2(**x_Koordinate; y_Koordinate**)
Beispiel: =ARCTAN2(4;-3)
Ergebnis: -0,643501

Die Funktion ARCTAN2() ist eine spezielle Variante der Berechnung des Arkustangens. Sie gestattet es, bei einer Geraden, die durch den Nullpunkt eines Koordinatenkreuzes geht, direkt durch Angabe der x-Koordinate und der y-Koordinate eines Punkts den Steigungswinkel der Geraden zu ermitteln.

Das Ergebnis liefert den gesuchten Winkel im Bogenmaß mit dem Wertebereich π bis –π (ausgenommen –π selbst). Ein positiver Wert entspricht einem Winkel, der von der x-Achse aus entgegen dem Uhrzeigersinn abgetragen wird, ein negativer Wert entspricht einem im Uhrzeigersinn abgetragenen Winkel.

Wenn beide Koordinaten den Wert 0 haben, liefert die Funktion den Fehler #DIV/0!, sind die Werte nicht numerisch, erscheint der Fehler #WERT!.

ARCTAN2

x_Koordinate	y_Koordinate	ARCTAN2()	GRAD()
12	0	0	0,00
11	1	0,090659887	5,19
10	2	0,19739556	11,31
9	3	0,321750554	18,43
8	4	0,463647609	26,57
7	5	0,620249486	35,54
6	6	0,785398163	45,00
5	7	0,950546841	54,46
4	8	1,107148718	63,43
3	9	1,249045772	71,57
2	10	1,373400767	78,69
1	11	1,48013644	84,81
0	12	1,570796327	90,00

x_Koordinate	y_Koordinate		
0	0		
5	6	0,876058051	50,19

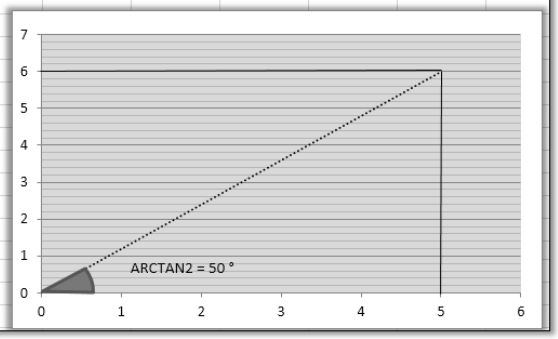


Abbildung 5.21 ARCTAN2() und die entsprechenden GRAD()-Werte

Soll der Winkel in Grad angegeben werden, multiplizieren Sie das Ergebnis mit 180/PI() oder benutzen die Funktion GRAD().

```
=ARCTAN2(4;4) * 180/PI()
=GRAD(ARCTAN2(4;4))
```

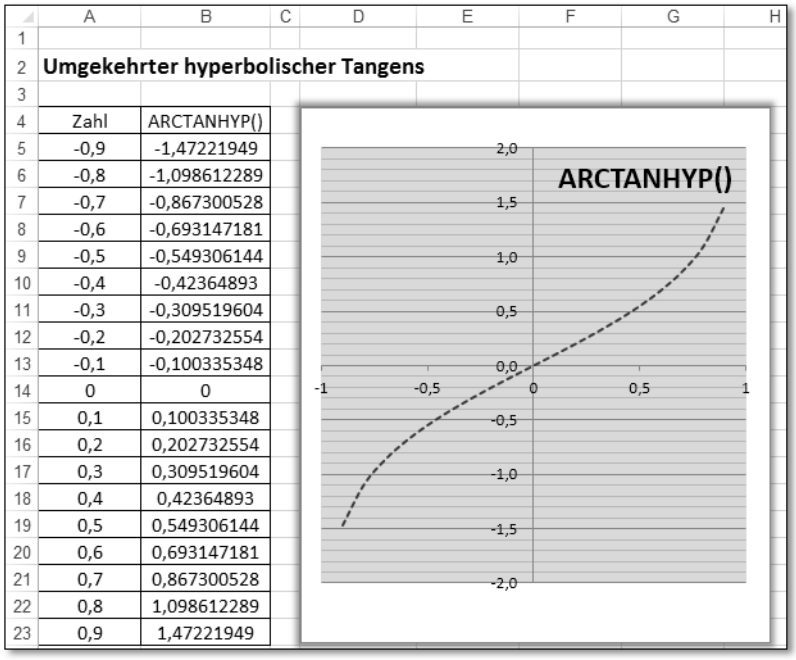
liefern beide, wie nicht anders zu erwarten, 45 Grad, da ja beide Koordinaten in diesem Fall gleich groß sind. Abbildung 5.21 zeigt einige Beispiele der Funktion und ein Diagramm für den Punkt x = 5, y = 6.

ARCTANHYP()

ATANH()

Syntax:	ARCTANHYP(**Zahl**)
Beispiel:	=ARCTANHYP(0,5)
Ergebnis:	0,549306

Die Funktion ARCTANHYP() ist die Umkehrfunktion der Funktion TANHYP(). Entgegen dem Präfix ARC ist die korrekte Bezeichnung »Areatangens Hyperbolicus«, siehe dazu auch den Abschnitt 5.3, »Zu den hyperbolischen Funktionen«.



Zahl	ARCTANHYP()
-0,9	-1,47221949
-0,8	-1,098612289
-0,7	-0,867300528
-0,6	-0,693147181
-0,5	-0,549306144
-0,4	-0,42364893
-0,3	-0,309519604
-0,2	-0,202732554
-0,1	-0,100335348
0	0
0,1	0,100335348
0,2	0,202732554
0,3	0,309519604
0,4	0,42364893
0,5	0,549306144
0,6	0,693147181
0,7	0,867300528
0,8	1,098612289
0,9	1,47221949

Abbildung 5.22 Der umgekehrte hyperbolische Tangens und seine grafische Darstellung

Für das Argument Zahl können nur Werte zwischen -1 und 1 verwendet werden (ausgenommen –1 und 1). Der Wertebereich liegt zwischen –unendlich bis +unendlich.

Werte außerhalb des erlaubten Bereichs ergeben den Fehler #ZAHL!, nicht numerische Werte den Fehler #WERT!.

Die Funktion rechnet mit folgender Formel:

$$\text{artanh}(x) = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right)$$

Der Graph der Funktion ist streng monoton steigend, die Nullstelle liegt bei x = 0.

AUFRUNDEN()

ROUNDUP()

Syntax:	AUFRUNDEN(**Zahl; Anzahl_Stellen**)
Beispiel:	=AUFRUNDEN(2,356;2)
Ergebnis:	2,36

Die Funktion AUFRUNDEN() rundet eine Zahl vom Nullwert weg. Mit dem Argument Anzahl_Stellen wird die Anzahl der Stellen angegeben, die beim Runden berücksichtigt wird. Ist Anzahl_Stellen gleich 1 oder größer, dann wird auf die entsprechende Anzahl Dezimalstellen aufgerundet. Ist es 0, wird auf die nächste Ganzzahl aufgerundet; ist es negativ, wird die entsprechende Ziffer links vom Dezimalzeichen aufgerundet. Anders als bei der Funktion RUNDEN() ist die Richtung der Rundung also festgelegt und hängt nicht von dem jeweiligen Wert ab.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler #WERT!. Falls Anzahl_Stellen größer ist als die Anzahl der Stellen, die der Wert von Zahl hat, werden nur die vorhandenen Stellen ausgegeben.

Das Aufrunden ist beispielsweise hilfreich, wenn bestimmte Stückelungen vorgegeben sind. Soll etwa eine Strecke von 100 cm mit Kacheln ausgelegt werden, die 15 cm breit sind, ergibt =AUFRUNDEN(100/15;0) eine Stückzahl von 7. Von der 7. Kachel muss dann ein Stück abgeschnitten werden.

	A	B	C
1			
2	**Aufrunden von Zahlen**		
3			
4	Zahl	Stellen	AUFRUNDEN()
5	12347,85	0	12348
6	12347,85	1	12347,9
7	12347,85	2	12347,85
8	12347,85	3	12347,85
9	12347,85	-1	12350
10	12357,85	-2	12400
11	12347,85	-3	13000
12	-12347,85	0	-12348
13	-12347,85	1	-12347,9
14	-12347,85	2	-12347,85
15	-12347,85	3	-12347,85
16	-12347,85	-1	-12350
17	-12347,85	-2	-12400
18	-12347,85	-3	-13000

Abbildung 5.23 Aufrunden mit unterschiedlicher Stellenanzahl

BASIS()

BASE()

Syntax: `BASIS(`**`Zahl`**`; `**`Basis`**`; Mindestlänge)`

Beispiel: `=BASIS(10000; 2; 15)`

Ergebnis: `010011100010000`

Die in Excel 2013 neue Funktion `BASIS()` wandelt den mit `Zahl` angegebenen Wert in eine Zeichenfolge entsprechend der angegebenen `Basis` um. Mit dem optionalen Wert `Mindestlänge` kann die Länge der zurückgegebenen Zeichenfolge angegeben werden. Der Wert muss eine ganze Zahl sein, die größer `0` ist. Das Maximum ist `255`.

Als Wert für `Zahl` sind nur ganze Zahlen erlaubt. Der Wertebereich ist ≥ `0` und < `2^53`. `Basis` muss eine ganze Zahl sein, ≥ `2` und ≤ `36`. Tabelle 5.4 zeigt die möglichen Werte für `Basis`.

Basis	Bedeutung
2	binär
8	oktal
16	hexadezimal
36	0 bis 9 und A bis Z (oder a bis z)

Tabelle 5.4 Werte für `Basis`

Liegen die Werte für eines der Argumente außerhalb des erlaubten Wertebereichs, liefert die Funktion den Wert `#ZAHL!`. Ist einer der Werte nicht numerisch, liefert die Funktion den Fehlerwert `#WERT!`.

Wird eine Zahl mit Dezimalstellen angegeben, werden diese – ohne Rundung – abgeschnitten. Abbildung 5.24 zeigt einige Beispiele.

	A	B	C	D
1				
2	**BASIS()**			
3				
4	1234567890	01001001100101100000001011010010	**=BASIS(A4;2;32)**	binäre Darstellung
5	1234567890	011145401322	**=BASIS(A5;8;12)**	oktale Darstellung
6	1234567890	00499602D2	**=BASIS(A6;16;10)**	hexadezimale Darstellung
7	1234567890	1234567890	**=BASIS(A7;10;10)**	dezimale Darstellung
8	1234567890	0KF12OI	**=BASIS(A8;36;7)**	Basis 36

Abbildung 5.24 Beispiele für Zahlenumwandlungen mit verschiedenen Werten für `Basis`

Beachten Sie, dass die Funktionen immer Zeichenfolgen liefern. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion `DEZIMAL()` ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. Die Formel `=DEZIMAL(E10;16)` wandelt beispielsweise die hexadezimale Zeichenfolge 000F in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine hexadezimale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: `0006 + 10 = 16`. Kommen in der hexadezimalen Zeichenfolge Buchstaben vor, liefert die Berechnung den Fehler `#WERT!`.

BOGENMASS()

RADIANS()

Syntax: `BOGENMASS(`**`Winkel`**`)`
Beispiel: `=BOGENMASS(180)`
Ergebnis: `3,14`

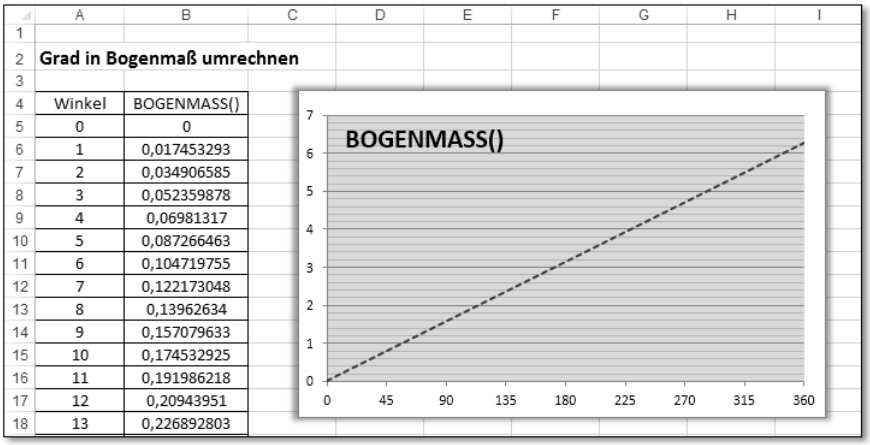
Die Funktion `BOGENMASS()` wandelt den Gradwert des angegebenen Winkels in Bogenmaß um. Die Einheit für das Bogenmaß ist *rad* für Radiant. 1 Radiant entspricht auf der Umfangslinie eines Kreises mit 1 Meter Radius einem Kreisbogen von 1 Meter. Der Vollwinkel, also 360 Grad, entspricht 2 π rad. Über das Verhältnis Grad/Bogenmaß siehe auch Abbildung 5.4. Für das Argument `Winkel` sind beliebige positive oder negative Werte zugelassen.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Da die Winkelfunktionen in Excel Winkelangaben im Bogenmaß erwarten, müssen Gradangaben mit dieser Funktion vorher umgerechnet werden, beispielsweise:

```
=SIN(BOGENMASS(45))
```

liefert den Sinus für einen Winkel von 45 Grad.



Grad in Bogenmaß umrechnen

Winkel	BOGENMASS()
0	0
1	0,017453293
2	0,034906585
3	0,052359878
4	0,06981317
5	0,087266463
6	0,104719755
7	0,122173048
8	0,13962634
9	0,157079633
10	0,174532925
11	0,191986218
12	0,20943951
13	0,226892803

Abbildung 5.25 Umrechnung von Winkelangaben

COS()

COS()

Syntax: `COS(`**Zahl**`)`
Beispiel: `=COS(30*PI()/180)`
Ergebnis: `0,866025`

Die Funktion `COS()` berechnet den Kosinus des mit dem Argument `Zahl` angegebenen Winkels. Dabei wird der Winkel nicht in Grad, sondern im Bogenmaß angegeben. Für die Umrechnung können Sie entweder den Gradwert mit `PI()/180` multiplizieren oder die Funktion `BOGENMASS()` verwenden:

```
=COS(30 * PI()/180)
=COS(BOGENMASS(30))
```

Der Kosinus eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis der Ankathete zur Hypotenuse.

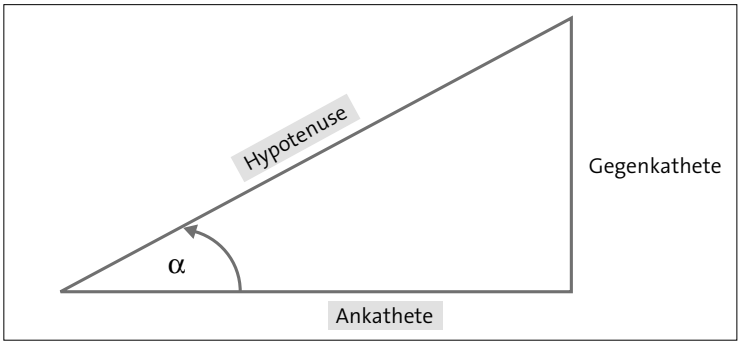


Abbildung 5.26 Der Kosinus liefert das Verhältnis der Ankathete zur Hypotenuse.

Die Definition kann aber auf der Basis des Einheitskreises erweitert werden. Einem Winkel von 90 Grad oder 270 Grad entspricht dann genau ein Kosinuswert von 0, einem von 180 Grad entspricht –1, von 360 Grad der Wert 1. Das Ergebnis der Funktion liegt also immer zwischen –1 und 1.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Das Diagramm in Abbildung 5.27 zeigt den periodischen Verlauf der Kosinusfunktion für den Bereich 0 bis 360 Grad. Vergleichen Sie die Abbildung mit der der Sinusfunktion, wird deutlich, dass der Kosinus ein um 90 Grad phasenverschobener Sinus ist. Es gilt also `cos(x) = sin(x + 90 Grad)`.

Kosinus eines Winkels

Winkel	BOGENMASS()	COS()
0	0	1
1	0,017453293	0,9998477
2	0,034906585	0,99939083
3	0,052359878	0,99862953
4	0,06981317	0,99756405
5	0,087266463	0,9961947
6	0,104719755	0,9945219
7	0,122173048	0,99254615
8	0,13962634	0,99026807
9	0,157079633	0,98768834
10	0,174532925	0,98480775
11	0,191986218	0,98162718
12	0,20943951	0,9781476
13	0,226892803	0,97437006
14	0,244346095	0,97029573
15	0,261799388	0,96592583
16	0,27925268	0,9612617

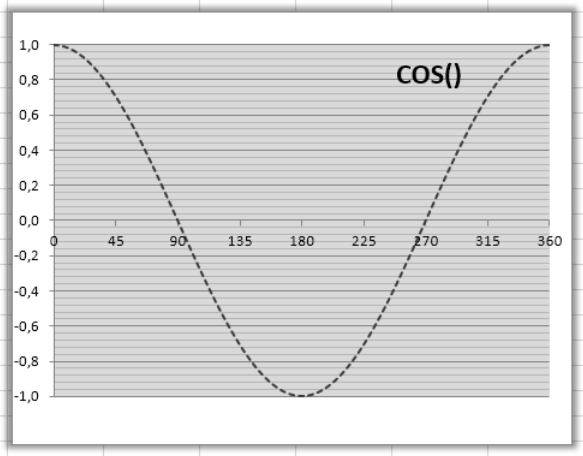


Abbildung 5.27 Die Kosinusfunktion und ihre grafische Darstellung

Die COS()-Funktion kann verwendet werden, um die Länge eines Anstiegs zu berechnen. Angenommen, es ist bekannt, dass eine gerade Straße einen Steigungswinkel von 8 Grad hat. Nun soll berechnet werden, wie lang die Strecke pro 1 km Luftlinie sein wird. Die Formel

```
=1/COS(BOGENMASS(8))
```

ergibt 1,010. Die Straße ist also pro Kilometer 10 m länger als die Luftlinienentfernung.

COSEC()

CSC()

Syntax:	COSEC(**Zahl**)
Beispiel:	=COSEC(1,10)
Ergebnis:	1,122

Die in Excel 2013 neue Funktion COSEC() liefert den Kosekans eines im Bogenmaß angegebenen Winkels. Für die Umrechnung können Sie entweder den Gradwert mit PI()/180 multiplizieren oder die Funktion BOGENMASS() verwenden:

```
=COSEC(30 * PI()/180)
=COSEC(BOGENMASS(30))
```

Der Kosekans eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis der Hypotenuse zur Gegenkathete.

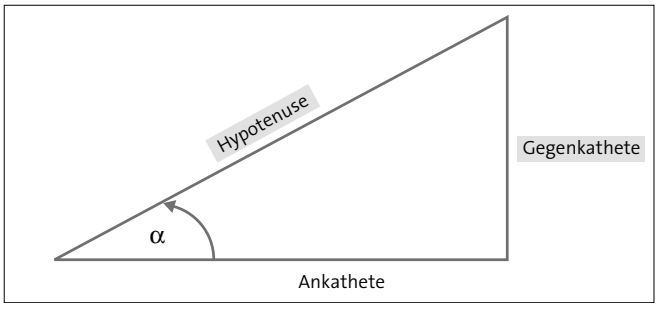


Abbildung 5.28 Der Kosekans liefert das Verhältnis der Hypotenuse zur Gegenkathete.

Der absolute Wert von Zahl muss kleiner sein als 2^27. Liegt der Wert von Zahl außerhalb dieses Wertebereichs, zeigt die Funktion den Fehlerwert #ZAHL!. Wird für Zahl ein nicht numerischer Wert angegeben, ist der Fehlerwert #WERT!.

	A	B	C
1			
2	**Kosekans**		
3			
4	Winkel	BOGENMASS()	COSEC()
5	0	0	
6	1	0,017453293	57,29868850
7	2	0,034906585	28,65370835
8	3	0,052359878	19,10732261
9	4	0,06981317	14,33558703
10	5	0,087266463	11,47371325
11	6	0,104719755	9,56677223
12	7	0,122173048	8,20550905
13	8	0,13962634	7,18529653
14	9	0,157079633	6,39245322
15	10	0,174532925	5,75877048
16	11	0,191986218	5,24084306
17	12	0,20943951	4,80973434
18	13	0,226892803	4,44541148
19	14	0,244346095	4,13356549
20	15	0,261799388	3,86370331
21	16	0,27925268	3,62795528
22	17	0,296705973	3,42030362
23	18	0,314159265	3,23606798

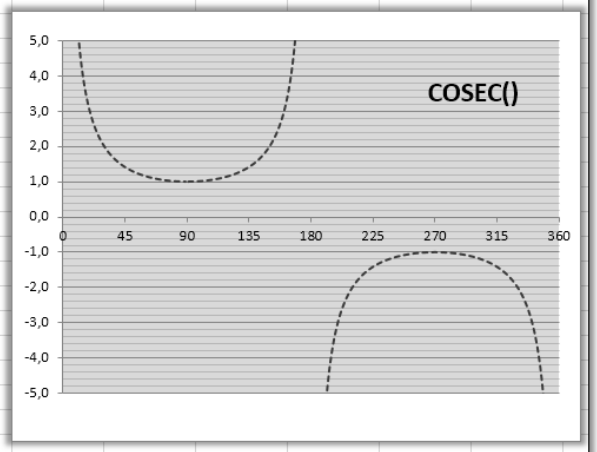


Abbildung 5.29 Die COSEC()-Funktion und ihre grafische Darstellung

COSECHYP()

CSCH()

Syntax: COSECHYP(**Zahl**)
Beispiel: =COSECHYP(1,5)
Ergebnis: 0,4696

Die in Excel 2013 neue Funktion COSECHYP() liefert den hyperbolischen Kosekans eines im Bogenmaß angegebenen Winkels.

Der absolute Wert von Zahl muss kleiner sein als 2^27. Liegt der Wert von Zahl außerhalb dieses Wertebereichs, zeigt die Funktion den Fehlerwert #ZAHL!. Wird für Zahl ein nicht numerischer Wert angegeben, ist der Fehlerwert #WERT!.

Die Funktion rechnet mit folgender Formel:

$$\mathrm{csch}(x) = \frac{2}{e^{x} - e^{-x}}$$

Der Graph der Funktion ist streng monoton fallend für $x < 0$ und $x > 0$.

	A	B	C
1			
2	**Hyperbolischer Kosekans**		
3			
4	Winkel	BOGENMASS()	COSECHYP()
5			
6	-360	-6,283185307	-0,003734898
7	-359	-6,265732015	-0,003800657
8	-358	-6,248278722	-0,003867574
9	-357	-6,23082543	-0,003935669
10	-356	-6,213372137	-0,004004963
11	-355	-6,195918845	-0,004075477
12	-354	-6,178465552	-0,004147232
13	-353	-6,16101226	-0,004220251
14	-352	-6,143558967	-0,004294556
15	-351	-6,126105675	-0,004370168
16	-350	-6,108652382	-0,004447113
17	-349	-6,091199089	-0,004525411
18	-348	-6,073745797	-0,004605089
19	-347	-6,056292504	-0,004686169
20	-346	-6,038839212	-0,004768677
21	-345	-6,021385919	-0,004852638
22	-344	-6,003932627	-0,004938077
23	-343	-5,986479334	-0,00502502

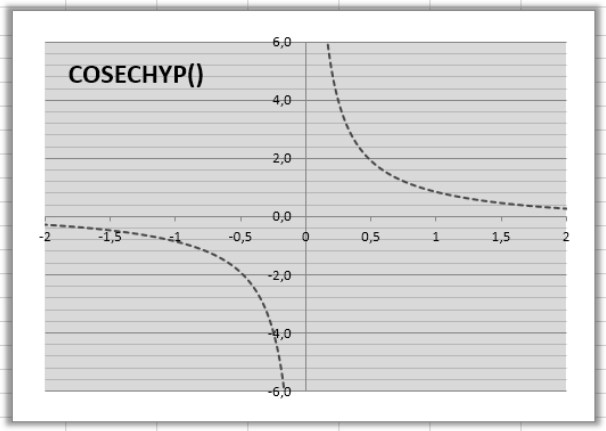


Abbildung 5.30 Die `COSECHYP()`-Funktion und ihre grafische Darstellung

COSHYP()

COSH()

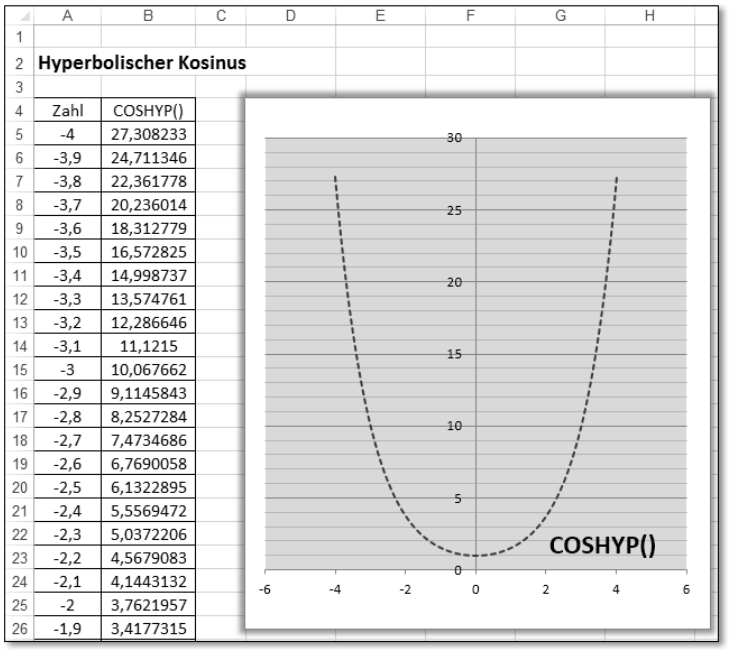
Syntax: `COSHYP(`**Zahl**`)`
Beispiel: `=COSHYP(1,5)`
Ergebnis: `2,352410`

Die Funktion `COSHYP()` liefert den hyperbolischen Kosinus einer beliebigen mit `Zahl` angegebenen reellen Zahl zwischen –unendlich und +unendlich. Das Ergebnis liegt zwischen 1 und unendlich. Die dabei verwendete Formel ist:

```
COSHYP(x) = (e^x + e^-x)/2
```

Der Graph der Funktion ähnelt einem hängenden Seil und wird deshalb auch als Katenoide oder Kettenlinie bezeichnet. In der Praxis wird die Funktion auch bei entsprechenden Berechnungen genutzt.

Nicht numerische Werte ergeben den Fehler `#WERT!`.



Zahl	COSHYP()
-4	27,308233
-3,9	24,711346
-3,8	22,361778
-3,7	20,236014
-3,6	18,312779
-3,5	16,572825
-3,4	14,998737
-3,3	13,574761
-3,2	12,286646
-3,1	11,1215
-3	10,067662
-2,9	9,1145843
-2,8	8,2527284
-2,7	7,4734686
-2,6	6,7690058
-2,5	6,1322895
-2,4	5,5569472
-2,3	5,0372206
-2,2	4,5679083
-2,1	4,1443132
-2	3,7621957
-1,9	3,4177315

Abbildung 5.31 Der hyperbolische Kosinus und sein Graph

COT()

COT()

Syntax:	`COT(`**Zahl**`)`
Beispiel:	`=COT(0,785)`
Ergebnis:	`1`

Die in Excel 2013 neue Funktion `COT()` liefert den Kotangens eines im Bogenmaß angegebenen Winkels. Für die Umrechnung können Sie entweder den Gradwert mit `PI()/180` multiplizieren oder die Funktion `BOGENMASS()` verwenden:

```
=COT(30 * PI()/180)
=COT(BOGENMASS(30))
```

Der Kotangens eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis von Ankathete zu Gegenkathete. Der Kotangens ist die Kehrwertfunktion des Tangens, es gilt also:

```
cot(x) = 1 / tan(x) = cos(x) / sin(x)
```

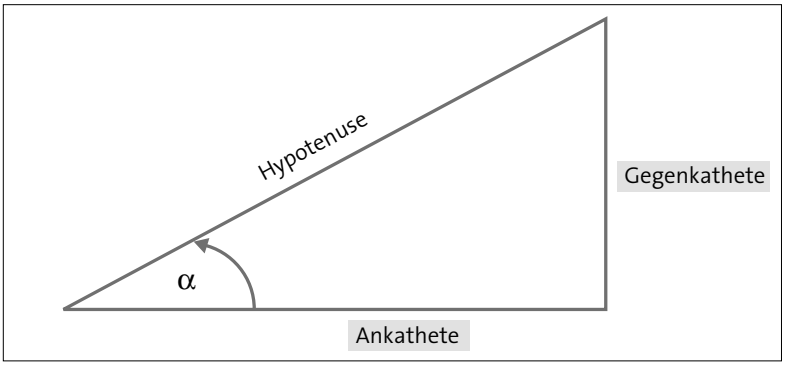


Abbildung 5.32 Der Kotangens liefert das Verhältnis der Ankathete zur Gegenkathete.

Der absolute Wert von `Zahl` muss kleiner sein als `2^27`. Liegt der Wert von `Zahl` außerhalb dieses Wertebereichs, zeigt die Funktion den Fehlerwert `#ZAHL!`. Wird für `Zahl` ein nicht numerischer Wert angegeben, ist der Fehlerwert `#WERT!`. Die Funktion `=COT(0)` liefert den Fehlerwert `#DIV/0!`.

Die Funktion ist für ganzzahlige Vielfache von π nicht definiert, `COT()` gibt an diesen Stellen sehr große oder sehr kleine Werte aus, die fehlerhafte Berechnungen verursachen können, wenn damit weitergerechnet wird.

Abbildung 5.33 zeigt den Graphen der Funktion für den Bereich 0 bis 360 Grad.

Kotangens

Winkel	BOGENMASS()	COT()
1	0,017453293	57,2899616
2	0,034906585	28,6362533
3	0,052359878	19,0811367
4	0,06981317	14,3006663
5	0,087266463	11,4300523
6	0,104719755	9,51436445
7	0,122173048	8,14434643
8	0,13962634	7,11536972
9	0,157079633	6,31375151
10	0,174532925	5,67128182
11	0,191986218	5,14455402
12	0,20943951	4,70463011
13	0,226892803	4,33147587
14	0,244346095	4,01078093
15	0,261799388	3,73205081
16	0,27925268	3,48741444
17	0,296705973	3,27085262

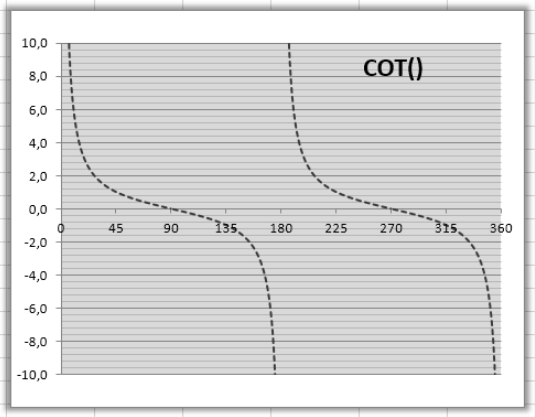


Abbildung 5.33 Der Kotangens und sein Graph

COTHYP()

COTH()

Syntax: `COTHYP(`**`Zahl`**`)`

Beispiel: `=COTHYP(0,2269)`

Ergebnis: `4,48`

Die in Excel 2013 neue Funktion `COTHYP()` liefert den hyperbolischen Kotangens eines im Bogenmaß angegebenen Winkels. Der hyperbolische Kotangens ist die Kehrwertfunktion des hyperbolischen Tangens.

Der absolute Wert von `Zahl` muss kleiner sein als `2^27` und `<> 0`. Liegt der Wert von `Zahl` außerhalb dieses Wertebereichs, zeigt die Funktion den Fehlerwert `#ZAHL!`. Wird für `Zahl` ein nicht numerischer Wert angegeben, ist der Fehlerwert `#WERT!`.

Bei der Berechnung der Funktion wird folgende Gleichung verwendet:

```
coth(x) = 1 / tanh(x) = cosh(x) / sinh(x) = (e^x - e^-x)/(e^x + e^-x)
```

Hyperbolischer Kotangens

Winkel	BOGENMASS()	COTHYP()
-360	-6,283185307	
-359	-6,265732015	-1,000007222
-358	-6,248278722	-1,000007479
-357	-6,23082543	-1,000007745
-356	-6,213372137	-1,00000802
-355	-6,195918845	-1,000008305
-354	-6,178465552	-1,0000086
-353	-6,16101226	-1,000008905
-352	-6,143558967	-1,000009222
-351	-6,126105675	-1,000009549
-350	-6,108652382	-1,000009888
-349	-6,091199089	-1,00001024
-348	-6,073745797	-1,000010603
-347	-6,056292504	-1,00001098
-346	-6,038839212	-1,00001137
-345	-6,021385919	-1,000011774
-344	-6,003932627	-1,000012192

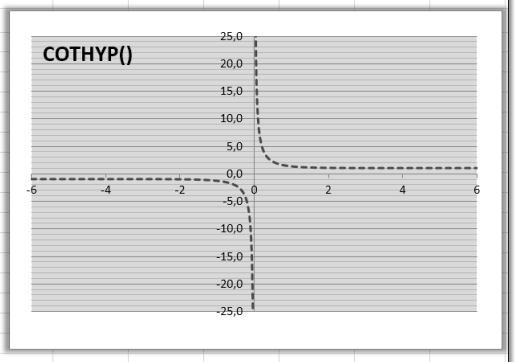


Abbildung 5.34 Der hyperbolische Kotangens und sein Graph

DEZIMAL()

DECIMAL()

Syntax: `DEZIMAL(`**`Zahl`**`; `**`Basis`**`)`

Beispiel: `=DEZIMAL(10101;2)`

Ergebnis: `21`

Die in Excel 2013 neue Funktion DEZIMAL() bewirkt, dass die mit Zahl angegebene Zeichenfolge in eine Dezimalzahl verwandelt wird. Der Wert muss ≥ 0 und < 2^53 sein.

Das Argument Basis gibt das Zahlensystem an, zu dem die als Zeichenfolge angegebene Zahl gehört. Tabelle 5.5 zeigt die möglichen Werte für Basis.

Basis	Bedeutung
2	binär
8	oktal
16	hexadezimal
36	0 bis 9 und A bis Z (oder a bis z)

Tabelle 5.5 Werte für Basis

Liegen die Werte für eines der Argumente außerhalb des erlaubten Wertebereichs, liefert die Funktion den Wert #ZAHL!. Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehlerwert #WERT!.

	A	B	C
1			
2	**Umwandlung in einen Dezimalwert**		
3			
4	Zahl	DEZIMAL()	
5	101010	42	=DEZIMAL(A5;2)
6	1750	1000	=DEZIMAL(A6;8)
7	3E8	1000	=DEZIMAL(A7;16)
8	abcdefghijklm	48.876.887.013.087.300.000,00	=DEZIMAL(A8;36)

Abbildung 5.35 Umwandeln in einen Dezimalwert

Die Funktion kann unter anderem dazu verwendet werden, die Ergebnisse von Funktionen, die Zahlen als Zeichenfolge ausgeben, wieder in dezimale Zahlen zu verwandeln, so dass damit korrekt weitergerechnet werden kann. Dazu gehören die Funktion BASIS() und die Umwandlungsfunktionen aus der Kategorie der technischen Funktionen wie BININHEX(), BININOKT(), DEZINBIN(), DEZINOKT(), DEZINHEX(), HEXINBIN(), HEXINOKT(), OKTINBIN() und OKTINHEX().

EXP()

EXP()

Syntax: `EXP(`**`Zahl`**`)`
Beispiel: `=EXP(2)`
Ergebnis: `7,3890561`

Die Funktion `EXP()` bewirkt, dass die Eulersche Zahl `e` mit dem im Argument `Zahl` angegebenen Wert potenziert wird. Da `e` die Basis der natürlichen Logarithmen ist, ergibt `=EXP(1)` den Wert von `e`, also 2,71828182845905.

Die nach dem Mathematiker Euler benannte Zahl ist wie die Zahl π eine irrationale, transzendente reelle Zahl, die in der Infinitesimalrechnung eine wichtige Rolle spielt. Sie lässt sich also weder als Bruch zweier natürlicher Zahlen noch als Lösung einer algebraischen Gleichung endlichen Grades darstellen. Eine mögliche Darstellung ist die folgende endlose Reihenbildung:

`e = 1 + 1/1 + 1/(1 * 2) + 1/ (1 * 2 * 3) + 1/(1 * 2 * 3 * 4) + ???`

`EXP(LN(x))` liefert den Wert `x`. `EXP()` ist die Umkehrfunktion von `LN()`. Die Funktion `EXP()` wird in erster Linie zur Beschreibung von Wachstumsfunktionen verwendet.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B
1		
2	**Potenzieren auf der Basis der Eulerschen Zahl *e***	
3		
4	Zahl	EXP()
5	-6	0,00247875
6	-5	0,00673795
7	-4	0,01831564
8	-3	0,04978707
9	-2	0,13533528
10	-1	0,36787944
11	0	1,00000000
12	1	2,71828183
13	2	7,38905610
14	3	20,08553692
15	4	54,59815003
16	5	148,41315910
17	6	403,42879349

Abbildung 5.36 Potenzieren der Basis `e` mit der angegebenen Zahl

FAKULTÄT()

FACT()

Syntax: `FAKULTÄT(`**Zahl**`)`
Beispiel: `=FAKULTÄT(4)`
Ergebnis: `24 (=1*2*3*4)`

Die Funktion `FAKULTÄT()` liefert die Fakultät der als Argument angegebenen `Zahl`. Die Fakultät einer Zahl ist das Produkt aller natürlichen Zahlen von 1 bis zur angegebenen Zahl, also: `1*2*3*4*...*Zahl`. In der Mathematik wird *n!* für *Fakultät von n* geschrieben. Die Zahl darf nicht negativ sein. Dezimalstellen werden ohne Rundung abgeschnitten.

Anwendungsbereich ist insbesondere die Kombinatorik, in der im weitesten Sinne die Frage gestellt wird, wie viele Möglichkeiten es gibt, Elemente aus einer Menge zu kombinieren. Hier nur ein Beispiel: Wie viele Möglichkeiten gibt es, 6 verschiedene Zahlen von 49 im Lotto auszuwählen?

```
= FAKULTÄT(49)/(FAKULTÄT(6)*FAKULTÄT(49-6))
```

Ergebnis: 13.983.816, d.h., es gibt über 13 Mio. Möglichkeiten, 6 Zahlen auf einem Lottoschein anzukreuzen, und so viele Kombinationen müssten durchgespielt werden, um ganz sicher »6 Richtige« im Lotto zu haben.

	A	B
1		
2	**Fakultät einer Zahl**	
3		
4	Zahl	FAKULTÄT()
5	0	1
6	1	1
7	2	2
8	3	6
9	4	24
10	5	120
11	6	720
12	7	5.040
13	8	40.320
14	9	362.880
15	10	3.628.800
16	11	39.916.800
17	12	479.001.600
18	13	6.227.020.800
19	14	87.178.291.200
20	15	1.307.674.368.000
21	16	20.922.789.888.000
22	17	355.687.428.096.000
23	18	6.402.373.705.728.000

Abbildung 5.37 Berechnen der Fakultät

Nicht numerische Werte ergeben den Fehler #WERT!.

GANZZAHL()

INT()

Syntax: GANZZAHL(**Zahl**)
Beispiel: =GANZZAHL(5,7)
Ergebnis: 5

Mit Hilfe der Funktion GANZZAHL() wird ein numerischer Ausdruck auf die nächstkleinere ganze Zahl reduziert, die Dezimalstellen werden gleichsam abgeschnitten. Negative Zahlen werden von 0 weg, also zur nächsten kleineren Zahl, gerundet. Demnach ergibt =GANZZAHL(-5,7) also -6.

Die Funktion GANZZAHL() hat eine ähnliche Wirkung wie die Funktion KÜRZEN(), wenn Anzahl_Stellen nicht angegeben oder 0 ist. Der Unterschied ist allerdings, dass diese Funktion bei negativen Werten nur die Nachkommastellen abschneidet. =KÜRZEN(-5,7) ergibt also -5.

	A	B
1		
2	**Runden auf die nächste Ganzzahl**	
3		
4	Zahl	GANZZAHL()
5	123,456	123
6	-12,78	-13
7	2,99	2

Abbildung 5.38 Abrunden auf die nächstkleinere ganze Zahl

Nicht numerische Werte ergeben den Fehler #WERT!.

GERADE()

EVEN()

Syntax: GERADE(**Zahl**)
Beispiel: =GERADE(-3)
Ergebnis: -4

Die Funktion GERADE() rundet Zahlen zur nächsten geraden Ganzzahl weg von 0 und ist damit komplementär zur Funktion UNGERADE(). Wenn Zahl schon eine gerade ganze Zahl ist, dann wird dieser Wert wiedergegeben. Die Funktion kann beispielsweise zur paarweisen Verarbeitung von Daten verwendet werden.

Ist Zahl kein numerischer Wert, gibt die Funktion den Fehler #WERT! zurück. Ob ein Wert eine gerade Ganzzahl ist, kann mit der Funktion ISTGERADE() geprüft werden.

	A	B	C
1			
2	**Aufrunden auf die nächste gerade Zahl**		
3			
4	Zahl	GERADE()	
5	0	0	
6	1	2	
7	2	2	
8	3	4	
9	4	4	
10	5	6	
11	6,2	8	
12	8,45	10	
13	-3	-4	

Abbildung 5.39 Aufrunden auf die nächste gerade Ganzzahl

GGT()

GCD()

Syntax: GGT(**Zahl1**; Zahl2; ...)
Beispiel: =GGT(56;98)
Ergebnis: 14

Die Funktion GGT() liefert den größten gemeinsamen Teiler von zwei oder mehreren ganzen Zahlen. Ein gemeinsamer Teiler ist eine Zahl, durch die alle angegebenen Zahlen geteilt werden können, ohne dass ein Rest bleibt. Der größte gemeinsame Teiler ist die größte dieser Zahlen. Seit Excel 2007 sind bis zu 255 Werte möglich, in den älteren Versionen nur 29 Werte. Wird eine gebrochene Zahl angegeben, werden die Dezimalstellen einfach abgeschnitten.

Negative Werte sind nicht erlaubt und führen zu dem Fehler #ZAHL!, nicht numerische Werte zu dem Fehler #WERT!.

	A	B	C	D	E
1					
2	**Größter gemeinsamer Teiler**				
3					
4	Zahl1	Zahl2	Zahl3	Zahl4	GGT()
5	9	27	6	18	3
6	4	8	9	10	1
7	25	50	75	0	25
8	25,4	50,7	75,6	0	25
9	90	252	180		18
10	-5	-10	-15	-20	#ZAHL!
11	h	b	5	3	#WERT!

Abbildung 5.40 Größter gemeinsamer Teiler einer Reihe von Werten

Der größte gemeinsame Teiler ist in der Mathematik insbesondere beim Kürzen von Brüchen gefragt.

In der Praxis kann die Funktion bei Aufteilungsfragen helfen. Ein kleines Beispiel: In einem Saal sollen drei Seminargruppen so auf Tische verteilt werden, dass möglichst niemand allein sitzen muss:

```
=GGT(36;24;18)
```

ergibt `6`. Die drei Gruppen lassen sich also auf Sechsertische verteilen.

GRAD()

DEGREES()

Syntax:	`GRAD(`**`Winkel`**`)`
Beispiel:	`=GRAD(PI()/3)`
Ergebnis:	`60`

Mit Hilfe der Funktion `GRAD()` wird der mit dem Argument `Winkel` angegebene Wert aus dem Bogenmaß (Radiant oder als Einheit rad) in Grad übertragen. Für die Konversion gilt die Formel:

```
Grad = rad*360/(2*PI)
```

Da zahlreiche Winkelfunktionen in Excel ihr Ergebnis im Bogenmaß ausgeben – siehe die Anmerkungen am Anfang des Kapitels –, wird die Funktion häufig benötigt, um die Winkelangaben in die geläufigere Maßeinheit zu übersetzen.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B	C
1			
2	**Umrechnen von Bogenmaß in Grad**		
3			
4	BOGENMASS()	GRAD()	
5	0	0	
6	0,087266463	5	
7	0,174532925	10	
8	0,261799388	15	
9	0,34906585	20	
10	0,436332313	25	
11	0,523598776	30	
12	0,610865238	35	
13	0,698131701	40	
14	0,785398163	45	
15	0,872664626	50	
16	0,959931089	55	
17	1,047197551	60	
18	1,134464014	65	
19	1,221730476	70	
20	1,308996939	75	
21	1,396263402	80	
22	1,483529864	85	
23	1,570796327	90	

Abbildung 5.41 Gradberechnung aus Winkelangaben im Bogenmaß

KGV()

LCM()

Syntax: `KGV(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=KGV(3;4;5;10)`

Ergebnis: `60`

Die Funktion `KGV()` liefert für die als Argumente angegebenen Zahlen das kleinste gemeinsame Vielfache. Seit Excel 2007 sind 255 Werte erlaubt, in den älteren Versionen nur 29.

Werden gebrochene Zahlen angegeben, werden die Dezimalstellen abgeschnitten. Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler `#WERT!`. Negative Werte führen zu dem Fehler `#ZAHL!`.

Das kleinste gemeinsame Vielfache ist die kleinste Zahl, die durch alle angegebenen Zahlen ohne Rest teilbar ist. Die Funktion kann beim Addieren von Brüchen, die unterschiedliche Nenner haben, hilfreich sein. Sollen beispielsweise die folgenden Brüche addiert werden:

```
1/3 + 3/4 + 1/2
```

kann zunächst ein kleinster gemeinsamer Nenner gesucht werden:

```
=KGV(3;4;2)
```

ergibt 12. Nun können die drei Brüche auf den Nenner 12 erweitert werden:

```
1*4/3*4 + 3*3/4*3 + 1*6/2*6 = (4 + 9 + 6)/12 = 1 7/12
```


	A	B	C	D	E
1					
2	**Kleinstes gemeinsames Vielfaches**				
3					
4	Zahl1	Zahl2	Zahl3	Zahl4	KGV()
5	9	27	6	18	54
6	4	8	9	10	360
7	25	50	75	0	0
8	25,4	50,7	75,6	0	0
9	90	252	180		1260
10	-5	-10	-15	-20	#ZAHL!
11	h	b	5	3	#WERT!

Abbildung 5.42 Kleinstes gemeinsames Vielfaches einer Reihe von Werten

KOMBINATIONEN()

COMBIN()

Syntax: KOMBINATIONEN(**n**; **k**)
Beispiel: =KOMBINATIONEN(7;3)
Ergebnis: 35

Die Funktion KOMBINATIONEN() gibt den Wert des Binomialkoeffizienten aus. KOMBINATIONEN() rechnet nach der Formel:

```
n! / (k! * (n-k)!)
```

Die Funktion beantwortet die Frage, wie viele Gruppen der Größe k aus n Elementen gebildet werden können, wenn die Reihenfolge keine Rolle spielt.

Werden keine ganzen Zahlen eingegeben, so werden die Dezimalstellen abgeschnitten. Wenn die Argumente kleiner 0 eingegeben werden oder n kleiner als k sein sollte, wird der Fehler #ZAHL! ausgegeben. Werden nicht numerische Werte angegeben, ist der Fehler #WERT!.

	A	B	C
1			
2	**Anzahl möglicher Kombinationen**		
3			
4	n	k	KOMBINATIONEN()
5	7	3	35
6	24	3	2024
7	32	4	35960
8	49	6	13983816

Abbildung 5.43 Kombinationen aus k Elementen aus der Menge n

Wollen Sie etwa berechnen, wie viele Möglichkeiten es gibt, 6 Zahlen von 49 anzukreuzen, dann liefert `=KOMBINATIONEN(49;6)` das Ergebnis `13.983.816`.

Siehe auch `KOMBINATIONEN2()` und `FAKULTÄT()`.

KOMBINATIONEN2()

COMBINA()

Syntax: `KOMBINATIONEN2(`**`Zahl; gewählte_Zahl`**`)`

Beispiel: `=KOMBINATIONEN2(36;6)`

Ergebnis: `4496388`

Die in Excel 2013 neue Funktion `KOMBINATIONEN2()` berechnet die Anzahl der möglichen Kombinationen der mit `gewählte_Zahl` angegebenen Menge von Elementen aus der mit `Zahl` angegebenen Gesamtmenge, wenn Wiederholungen zugelassen sind. `Zahl` muss ≥ `0` und ≥ `gewählte_Zahl` sein. Der Wert für `gewählte_Zahl` muss > `0` sein. Werden Werte mit Dezimalstellen angegeben, werden diese abgeschnitten.

Liegt ein Wert außerhalb des zugelassenen Wertebereichs, liefert die Funktion den Fehlerwert `#ZAHL!`, ist ein nicht numerischer Wert angegeben, ist der Fehlerwert `#WERT!`.

	A	B	C	D
1				
2	**Vergleich der Kombinationen ohne und mit Wiederholung**			
3				
4	n	k	KOMBINATIONEN()	KOMBINATIONEN2()
5	7	3	35	84
6	24	3	2.024	2.600
7	32	4	35.960	52.360
8	49	6	13.983.816	25.827.165

Abbildung 5.44 Vergleich der möglichen Kombinationen ohne und mit Wiederholung

Die Funktion arbeitet mit der Formel:

```
(n + k - 1)! / k!(n - 1)!
```

Die Anzahl der Kombinationen mit Wiederholung ist naturgemäß größer als die Anzahl von Kombinationen, bei denen die Wiederholung nicht zugelassen ist, wie es beispielsweise beim Lotto der Fall ist, wie Abbildung 5.44 zeigt.

Siehe auch KOMBINATIONEN() und FAKULTÄT().

KÜRZEN()

TRUNC()

Syntax: KÜRZEN(**Zahl**; Anzahl_Stellen)
Beispiel: =KÜRZEN(5,7)
Ergebnis: 5

Mit Hilfe der Funktion KÜRZEN() wird ein numerischer Ausdruck Zahl auf die mit Anzahl_Stellen angegebenen Stellen verkürzt, d.h., die restlichen Stellen hinter dem Komma werden abgeschnitten. Es findet keine Rundung statt. Wird Anzahl_Stellen nicht angegeben, so wird das Argument als 0 angenommen und nur der ganzzahlige Rest wiedergegeben. Dies gilt auch (im Gegensatz zu GANZZAHL()) bei negativen Zahlen: =KÜRZEN(-5,7) ergibt also -5.

Mit negativen Werten für Anzahl_Stellen kann auch auf ganze Zehner, Hunderter oder Tausender abgerundet werden:

```
=KÜRZEN(1234;-2)
```

ergibt also 1.200.

Nicht numerische Werte ergeben den Fehler #WERT!.

	A	B	C	D	E
1					
2	**Kürzen einer Zahl**				
3					
4	Zahl	Stellen	KÜRZEN()	RUNDEN()	
5	100,85714	0	100	101	
6	100,85714		100	101	
7	100,85714	1	100,8	100,9	
8	100,85714	2	100,85	100,86	
9	100,85714	3	100,857	100,857	
10					

Abbildung 5.45 KÜRZEN() im Vergleich zu RUNDEN()

LN()

LN()

Syntax: `LN(`**Zahl**`)`
Beispiel: `=LN(10)`
Ergebnis: `2,30258509`

Die Funktion `LN()` liefert den natürlichen Logarithmus des mit `Zahl` angegebenen Werts, d.h. des Werts, mit dem die Eulersche Zahl (2,718281828...) potenziert werden müsste, damit `Zahl` als Ergebnis herauskommt.

`Zahl` muss eine positive reelle Zahl > `0` sein, sonst liefert die Funktion den Wert `#ZAHL!`. Nicht numerische Werte ergeben den Fehler `#WERT!`.

Anwendungen der Funktion finden Sie ähnlich wie bei der Funktion `EXP()` im Bereich von Wachstumsprozessen in Naturwissenschaft und Technik. `LN()` ist die Umkehrfunktion von `EXP()`, denn es gilt:

`y = LN(x)` und `x = EXP(y)`

	A	B
1		
2	**Natürlicher Logarithmus**	
3		
4	Zahl	LN()
5	10	2,30259
6	100	4,60517
7	120	4,78749
8	150	5,01064
9	200	5,29832
10	2000	7,60090
11	4000	8,29405
12	10000	9,21034

Abbildung 5.46 Berechnen des natürlichen Logarithmus

LOG()

LOG()

Syntax: `LOG(`**Zahl**`; Basis)`
Beispiel: `=LOG(4;2)`
Ergebnis: `2`

Die Funktion LOG() ist von allen Logarithmusfunktionen, die Excel bereitstellt, die allgemeinste. Sie gestattet es, den Logarithmus eines mit Zahl angegebenen Werts zu jeder beliebigen Basis zu ermitteln, wobei das Ergebnis immer der Wert ist, mit dem die Basis potenziert werden müsste, um Zahl zu erhalten.

Beide Argumente müssen > 0 sein, sonst liefert die Funktion den Fehler #ZAHL!. Nicht numerische Werte ergeben den Fehler #WERT!.

Das Logarithmieren einer Basis ist bekanntlich die Umkehr des Potenzierens einer Basis und damit die Suche nach dem Exponenten. Sind a und b zwei reelle Zahlen und gilt:

```
a = b^x
```

dann ist x der Logarithmus von b zur Basis a.

Drei Basis-Werte spielen eine besondere Rolle:

```
=LOG(Zahl;2)
```

ergibt den binären Logarithmus;

```
=LOG(Zahl;EXP(1))
```

ergibt den natürlichen Logarithmus;

```
=LOG(Zahl;10)
```

ergibt den dekadischen Logarithmus.

Wird auf die Angabe von Basis verzichtet, dann nimmt Excel als Basis 10 an (das entspricht der folgenden Funktion LOG10()).

Zur Erinnerung hier einige der Rechenregeln, die für alle Logarithmen gelten:

```
LOG(a;a) = 1
```

Der Logarithmus der Basis ist 1.

```
LOG(1) = 0
```

Unabhängig von der Basis ist der Logarithmus von 1 gleich 0.

```
LOG(a*b) = LOG(a)+LOG(b)
LOG(a/b) = LOG(a)-LOG(b)
```

Das heißt, die Multiplikation und Division werden auf logarithmischer Ebene auf die Addition und Subtraktion zurückgeführt.

```
LOG(a^n) = n*LOG(a)
```

Potenzen werden auf logarithmischer Ebene auf Multiplikationen zurückgeführt.

	A	B	C
1			
2	**Logarithmus einer Zahl zur angegebenen Basis**		
3			
4	Zahl	Basis	LOG()
5	144	12	2,00000
6	1000		3,00000
7	1000	2	9,96578
8	1000	3	6,28771
9	1000	4	4,98289
10	1000	5	4,29203
11	1000	6	3,85529

Abbildung 5.47 Logarithmen mit unterschiedlichen Basen

LOG10()

LOG10()

Syntax: `LOG10(`**Zahl**`)`
Beispiel: `=LOG10(4)`
Ergebnis: `0,60205999`

Die Funktion `LOG10()` liefert den dekadischen Logarithmus von `Zahl`, d.h. dem Wert, mit dem 10 potenziert werden müsste, um diese Zahl zu erhalten. So ist:

```
=LOG10(100) = 2
```

Das heißt, 10 muss zum Quadrat genommen werden, um 100 zu erhalten. Der dekadische Logarithmus ist gleichzeitig ein Maß für die Größe einer Zahl.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B
1		
2	**Dekadischer Logarithmus**	
3		
4	Zahl	LOG10()
5	10	1,00000
6	100	2,00000
7	1000	3,00000
8	10000	4,00000
9	100000	5,00000

Abbildung 5.48 Logarithmus zur Basis 10

MDET()

MDETERM()

Syntax: `MDET(`**`Matrix`**`)`
Beispiel: `=MDET({2.3;4.5})`
Ergebnis: `-2`

Die Funktion `MDET()` liefert die Determinante der angegebenen Matrix. Das Argument `Matrix` kann dabei über einen Zellbezug, einen Bereichsnamen oder als Matrixkonstante angegeben werden.

Die Matrix muss quadratisch sein, also ebenso viele Zeilen wie Spalten enthalten. Alle Werte müssen numerisch sein. Ist eine dieser Bedingungen nicht erfüllt, liefert die Funktion den Fehler `#WERT!`.

Die Determinante im Beispiel wird berechnet nach der Formel `=(2*5)-(3*4)`. Die Ermittlung der Determinante einer Matrix wird verwendet bei der Lösung von linearen Gleichungen mit mehreren Unbekannten. Hat eine Matrix, die die Koeffizienten der Gleichungen enthält, eine Determinante ungleich 0, ist sicher, dass das entsprechende lineare Gleichungssystem eine eindeutige Lösung hat. Ein Beispiel, wie Gleichungssysteme mit Hilfe von Matrixfunktionen gelöst werden können, wurde bereits im Abschnitt »Matrizenrechnung und lineare Gleichungssysteme« im ersten Kapitel, »Einstieg in Berechnungen mit Excel«, vorgestellt.

Gleichungen dieser Art kommen unter anderem im naturwissenschaftlich-technischen Bereich (z. B. Berechnung von Widerstandsnetzwerken) und in der Ökonomie (z. B. Optimierungsaufgaben, die von mehreren Faktoren abhängen) vor.

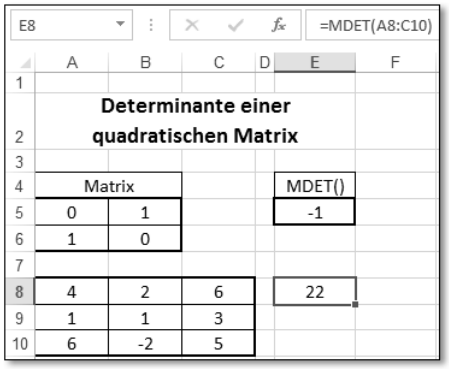


Abbildung 5.49 Bestimmung der Determinante von quadratischen Matrizen

Die Lösung derartiger Gleichungssysteme ist »von Hand« eine äußerst zeitraubende Angelegenheit, über die Bildung von Determinanten dagegen einfacher erledigt. In Excel lassen sich derartige Aufgaben allerdings wesentlich schneller mit dem Solver bearbeiten.

MEINHEIT()

MUNIT()

Syntax: `MEINHEIT(`**Größe**`)`
Beispiel: `=MEINHEIT(3)`
Ergebnis: `{1,0,0;0,1,0;0,0,1}`

Die in Excel 2013 neue Funktion `MEINHEIT()` liefert die Einheitsmatrix für die angegebene `Größe`. Der Wert für `Größe` ist eine ganze Zahl, die die Anzahl der Zeilen und Spalten der Matrix angibt.

`Größe` muss größer `0` sein, andernfalls liefert die Funktion den Fehlerwert `#WERT!`.

Die Einheitsmatrix ist eine quadratische Matrix, die Anzahl der Spalten und Zeilen ist also gleich. Ihre Hauptdiagonale – von oben links nach unten rechts – besteht aus lauter Einsen. Alle anderen Werte sind 0. Die Matrix wird auch als Identitätsmatrix bezeichnet. Die Multiplikation einer Matrix A mit der Einheitsmatrix ergibt wieder die Matrix A.

Die Formel muss als Matrixformel eingegeben werden. Um beispielsweise eine Matrix in der Größe 6 x 6 anzulegen, markieren Sie zunächst einen entsprechend großen Zellbereich, im Beispiel in Abbildung 5.50 A4 bis F9, geben die Formel `=MEINHEIT(6)` ein und schließen die Eingabe mit Strg + ⇧ + ↵ ab.

	A	B	C	D	E	F	G
1							
2	**Einheitsmatrix**						
3							
4	1	0	0	0	0	0	
5	0	1	0	0	0	0	
6	0	0	1	0	0	0	
7	0	0	0	1	0	0	
8	0	0	0	0	1	0	
9	0	0	0	0	0	1	
10							

Abbildung 5.50 Einheitsmatrix der Größe 6

MINV()

MINVERSE()

Syntax: `MINV(`**Matrix**`)`
Beispiel: `=MINV({2.3;4.5})`
Ergebnis: `{-2,5.1,5;2.-1}`

Die Funktion `MINV()` bildet die Inverse zu einer Matrix, liefert also die sogenannte Kehrmatrix. Das Argument `Matrix` kann dabei ein Zellbezug, ein Bereichsname oder eine Matrixkonstante sein.

Die Formel wird als Matrixformel für einen vorher markierten Zellbereich eingetragen, dessen Größe dem Quadrat der Matrix entspricht. Sie muss also mit der Tastenkombination Strg + ⇧ + ↵ abgeschlossen werden.

Die Funktion findet Anwendung bei der Lösung von Gleichungssystemen mit mehreren Unbekannten. Voraussetzung für die Lösbarkeit ist, dass die mit der Funktion `MDET()` berechenbare Determinante ungleich `0` ist.

Die Matrix muss quadratisch sein, also ebenso viele Zeilen wie Spalten enthalten. Alle Werte müssen numerisch sein. Ist eine der beiden Bedingungen nicht erfüllt, gibt die Funktion den Fehler `#WERT!` aus. Wenn die Determinante `0` ist, kann auch keine Kehrmatrix gebildet werden, die Funktion `MINV()` liefert dann den Fehler `#ZAHL!`.

Ein Beispiel, wie Gleichungssysteme mit Hilfe von Matrixfunktionen gelöst werden können, wurde bereits im Abschnitt »Matrizenrechnung und lineare Gleichungssysteme« im ersten Kapitel, »Einstieg in Berechnungen mit Excel«, vorgestellt.

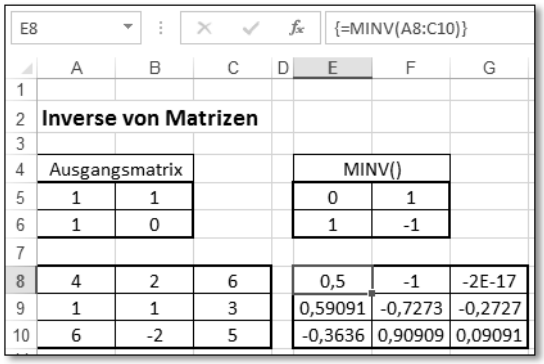


Abbildung 5.51 Die Kehrmatrix zu einer Ausgangsmatrix bilden

Das Produkt aus Ausgangsmatrix und Kehrmatrix, das Sie mit der Funktion MMULT() bilden können, ist die Einheitsmatrix, bei der die Werte auf der Hauptdiagonalen 1, alle anderen Werte 0 sind, siehe dazu die Beschreibung zu MEINHEIT(). Da Excel die Werte für MINV() aber nur mit maximal 15 Stellen genau berechnen kann, kommt es zu kleinen Abweichungen, wenn Sie in dem Beispiel in Abbildung 5.51 die Matrizen in A8:C10 und E8:G10 multiplizieren.

MMULT()

MMULT()

Syntax: MMULT(**Array1**; **Array2**)
Beispiel: =MMULT({2.3;4.5};{6.7;8.9})
Ergebnis: {36.41;64.73}

Die Funktion MMULT() liefert das Produkt zweier Matrizen. Die Argumente Array1 und Array2 können dabei über einen Zellbezug, einen Bereichsnamen oder als Matrixkonstante angegeben werden.

Die Spalten von Array1 müssen in der Anzahl mit den Zeilen von Array2 übereinstimmen. Alle Zellen müssen numerische Werte enthalten. Ist eine der Bedingungen nicht erfüllt, liefert die Funktion den Fehler #WERT!.

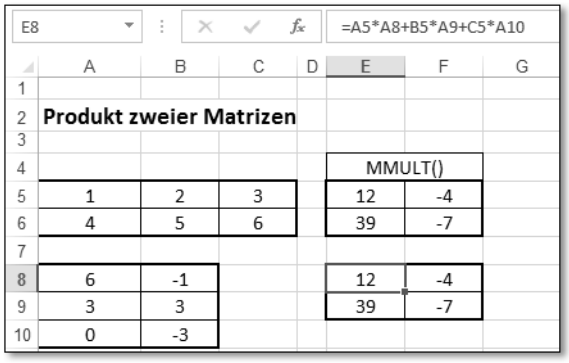


Abbildung 5.52 Matrizenprodukt und Kontrollberechnung

Da die Funktion als Ergebnis wieder eine Matrix liefert, müssen Sie vor der Eingabe der Funktion einen entsprechend großen Bereich markieren und die Funktion mit der Tastenkombination Strg + ⇧ + ↵ abschließen. Das Ergebnis ist eine Matrix, die die

Zahl der Zeilen von der ersten, die Zahl der Spalten von der zweiten Matrix übernimmt. Die Kontrollformel in der Zelle E8 in Abbildung 5.52 zeigt, wie die Multiplikation ausgeführt wird. Es werden jeweils die Zeilenelemente der ersten Matrix mit den entsprechenden Spaltenelementen der zweiten Matrix multipliziert.

Ein Beispiel, wie Gleichungssysteme mit Hilfe von Matrixfunktionen gelöst werden können, wurde bereits im Abschnitt »Matrizenrechnung und lineare Gleichungssysteme« im ersten Kapitel, »Einstieg in Berechnungen mit Excel«, vorgestellt.

OBERGRENZE.MATHEMATIK()

CEILING.PRECISE()

Syntax: `OBERGRENZE.MATHEMATIK(`**`Zahl`**`; Schritt; Modus)`

Beispiel: `=OBERGRENZE.MATHEMATIK(-5,5; 2;-1)`

Ergebnis: `-6`

Die in Excel 2013 umbenannte Funktion – in Excel 2010 hieß sie `OBERGRENZE.GENAU()` – rundet den mit `Zahl` angegebenen Wert auf die nächste ganze Zahl oder das nächste Vielfache von `Schritt` auf und ist damit komplementär zu `UNTERGRENZE.MATHEMATIK()`.

`Zahl` muss kleiner als `9,99E+307` und größer als `-2,229E-308` sein. Fehlt der Wert für `Schritt`, wird `1` oder `-1` vorgegeben, abhängig davon, ob `Zahl` ein positiver oder ein negativer Wert ist. Mit einem optionalen Wert `Modus` von < `0` kann erreicht werden, dass bei negativen Werten auf den nächstkleineren Wert – weg von der Null – gerundet werden soll. Bei positiven Werten wird der Wert für `Modus` ignoriert.

Mit dem Wert `0,05` für `Schritt` kann beispielsweise bestimmt werden, dass die Hundertstelstelle beim Aufrunden immer nur eine 5 oder eine 0 sein kann. Auf diese Weise können Sie bei der Berechnung von Werten für eine Preisliste dafür sorgen, dass keine Preise vorkommen, die einzelne Cents verlangen, sondern minimal 5-Cent-Stücke.

Bei dieser Funktion sind auch unterschiedliche Vorzeichen für `Zahl` und `Schritt` möglich, weil die Funktion immer mit dem Absolutwert von `Schritt` arbeitet, `OBERGRENZE.MATHEMATIK(-4,2546;-0,5)` liefert deshalb ebenso wie `OBERGRENZE.MATHEMATIK(-4,2546;0,5)` als Ergebnis `-4`.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B	C	D
1				
2	**OBERGRENZE.MATHEMATIK**			
3				
4	Wert	Schritt	Modus	OBERGRENZE.MATHEMATIK()
5	14,25	0,5		14,5
6	14,33	1		15
7	123,56	2		124
8	124,33	5		125
9	126,33	5		130
10	124,33	10		130
11	-6,3	2		-6
12	-6,3	2	-1	-8

Abbildung 5.53 Aufrunden auf Obergrenzen mit absoluten Schritten

PI()

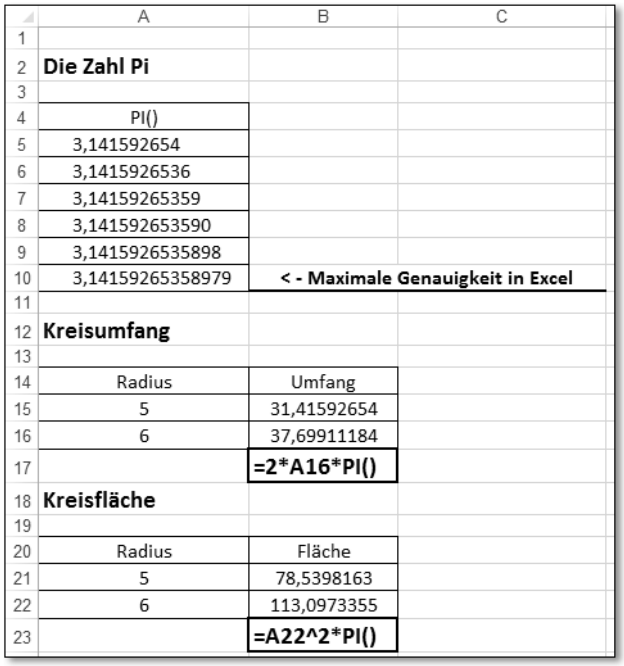
PI()

Syntax: `PI()`
Beispiel: `=PI()`
Ergebnis: `3,14159265358979`

Die Funktion `PI()` liefert den numerischen Wert von π mit der für Excel möglichen maximalen Genauigkeit von 15 Stellen. Die sogenannte Kreiszahl ist der Faktor, mit dem der Radius eines Kreises multipliziert werden muss, um den Umfang zu erhalten. Diese Zahl ist irrational und transzendent, d.h., sie lässt sich nicht durch eine algebraische Gleichung mit ganzzahligen Koeffizienten gewinnen.

Neben der Umrechnung von Winkeln von Grad in Bogenmaß wird π insbesondere für Kreis- und Kugelberechnungen benötigt. Obwohl für π als Verhältniszahl kein Argument angegeben wird, benötigt Excel die beiden Klammern hinter dem Funktionsnamen. Zur Erinnerung einige der Formeln, die π verwenden:

```
Kreisumfang: U=2*r*PI()
Kreisfläche: A=r^2*PI()
Kugeloberfläche: O=4*r^2*PI()
Kugelvolumen: V=4/3*r^3*PI()
```



	A	B	C
1			
2	**Die Zahl Pi**		
3			
4	PI()		
5	3,141592654		
6	3,1415926536		
7	3,14159265359		
8	3,141592653590		
9	3,1415926535898		
10	3,14159265358979	**< - Maximale Genauigkeit in Excel**	
11			
12	**Kreisumfang**		
13			
14	Radius	Umfang	
15	5	31,41592654	
16	6	37,69911184	
17		**=2*A16*PI()**	
18	**Kreisfläche**		
19			
20	Radius	Fläche	
21	5	78,5398163	
22	6	113,0973355	
23		**=A22^2*PI()**	

Abbildung 5.54 Berechnungen mit der Funktion `PI()`

POLYNOMIAL()

MULTINOMIAL()

Syntax: `POLYNOMIAL(`**Zahl1**`; Zahl2; ...)`
Beispiel: `=POLYNOMIAL(3;4;5)`
Ergebnis: `27720`

Die Funktion `POLYNOMIAL()` liefert den Polynomialkoeffizienten einer Gruppe von Zahlen, also die Fakultät der Summe der Argumente geteilt durch das Produkt der Fakultäten:

`(Zahl1+Zahl2...)! / Zahl1!*Zahl2!...`

Während seit Excel 2007 bis zu 255 Argumente erlaubt sind, ist die Zahl in den Vorgängerversionen auf 29 begrenzt.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler `#WERT!`, ist ein Argument kleiner als `0`, erscheint der Fehler `#ZAHL!`.

	A	B	C	D
1				
2	**Polynomialkoeffizient**			
3				
4	Zahl1	Zahl2	Zahl3	POLYNOMIAL()
5	2	2	1	30
6	7	5	2	72072

Abbildung 5.55 Berechnen des Polynomialkoeffizienten einer Gruppe von Zahlen

Die Funktion wird in der Kombinatorik verwendet, um zu bestimmen, wie viele verschiedene Gruppen sich aus Elementen bilden lassen, wenn die Reihenfolge zu beachten ist. Sie kann beispielsweise die Frage beantworten, wie viele Möglichkeiten es gibt, die 32 Karten eines Skatspiels mit je 10 Karten für drei Spieler und 2 Restkarten zu verteilen. Die Lösung kann mit der Formel

```
=32! / (10! * 10! * 10! * 2!)
```

berechnet werden. Die Funktion

```
=POLYNOMIAL(10;10;10;2)
```

liefert die Antwort: `2.753.294.408.504.620`.

POTENZ()

POWER()

Syntax: `POTENZ(`**`Zahl`**`; `**`Potenz`**`)`
Beispiel: `=POTENZ(2;10)`
Ergebnis: `1024`

Die Funktion `POTENZ()` liefert die Potenzierung des mit `Zahl` angegebenen Werts, der auch als Basis oder Grundzahl bezeichnet wird, und zwar mit Hilfe des mit `Potenz` angegebenen Exponenten, der auch Hochzahl genannt wird. Für `Zahl` sind alle reellen Zahlen möglich. Das Potenzieren ist bekanntlich die wiederholte Multiplikation einer Zahl mit sich selbst.

Hat `Potenz` den Wert `0`, ist das Ergebnis immer 1. Negative Werte für `Potenz` werden nach der Formel

$x^{-n} = 1 / x^{n}$

berechnet, bei der die Multiplikation in eine Division umgewandelt wird.

Ansonsten verwendet die Funktion dasselbe Rechenverfahren, das auch mit dem Operatorzeichen ^ zur Anwendung kommt. Statt `=4^2` kann also auch `=POTENZ(4;2)` eingegeben werden.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B	C
1			
2	**Potenzieren einer Zahl**		
3			
4	Zahl	Potenz	POTENZ()
5	2	0	1
6	4	2	16
7	6	1,8	25
8	8	5	32.768
9	10	6	1.000.000
10	12	7	35.831.808
11	14	8	1.475.789.056
12	16	9	68.719.476.736
13	18	-10	0,000000000000280075389726

Abbildung 5.56 Potenzieren mit verschiedenen Potenzen

POTENZREIHE()

SERIESSUM()

Syntax: `POTENZREIHE(x; n; m; Koeffizienten)`
Beispiel: `=POTENZREIHE(3;2;1;{2;3;4})`
Ergebnis: `423`

Die Funktion `POTENZREIHE()` liefert eine Summe von Potenzen der Zahl `x`. Berechnet wird nach der Formel:

```
a1*x^n + a2*x^(n+m) + a3*x^(n+2m) ...
```

wobei `a1`, `a2` ... mit dem Argument `Koeffizienten` angegeben wird. Dies kann ein Zellbereich, ein Bereichsname oder eine Matrixkonstante sein. Die Zahl der hier eingetragenen Werte liefert zugleich die Zahl der Summanden in der Formel und damit die Zahl der Glieder in der Reihe. Das Argument `n` ist die Potenz von `x` im ersten Term der Potenzreihe, das Argument `m` benennt das Inkrement, mit dem `n` in jedem Glied der Reihe vergrößert werden soll.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler `#WERT!`.

	A	B	C	D	E	F	G	H
1								
2	**Summe von Potenzen**							
3								
4	x	n	m	k1	k2	k3	k4	POTENZREIHE()
5	0,523598776	1	2	1	-0,166666667	0,008333333	-0,000198413	0,49999999
6	**=BOGENMASS(30)**				**=-1/FAKULTÄT(3)**	**=1/FAKULTÄT(5)**	**=-1/(FAKULTÄT(7))**	**=POTENZREIHE(A5;B5;C5;D5:G5)**
7								
8	**=A5/FAKULTÄT(1)-A5^3/FAKULTÄT(3)+A5^5/FAKULTÄT(5)-A5^7/FAKULTÄT(7)...**							
9								
10								SIN()
11								0,50000000

Abbildung 5.57 Bilden einer Potenzreihe zur Annäherung an den Sinuswert

Mit dieser Funktion lassen sich Näherungsberechnungen für Funktionen durchführen. Abbildung 5.57 zeigt eine Potenzreihe, die einen Wert berechnet, der mit dem Ergebnis der Funktion `SIN()` fast genau übereinstimmt. Die Reihe sieht so aus:

$$\sin x = x / 1! - x^3 / 3! + x^5 / 5! - x^7 / 7! + \ldots$$

Das Vorzeichen der Koeffizienten wechselt jedes Mal.

Die Annäherung einer Potenzreihe an den Wert für `COS()` ist ähnlich, nur der Wert für `n` ist hier `0`. Die Reihe sieht dann so aus:

$$\cos x = 1 / 0! - x^2 / 2! + x^4 / 4! - x^6 / 6! + \ldots$$

	A	B	C	D	E	F	G	H
1								
2	**Potenzreihe für den Kosinus**							
3								
4	x	n	m	k1	k2	k3	k4	POTENZREIHE()
5	0,523598776	0	2	1	-0,5	0,041666667	-0,001388889	0,86602526
6	**=BOGENMASS(30)**				**=-1/FAKULTÄT(2)**	**=1/FAKULTÄT(4)**	**=-1/(FAKULTÄT(6))**	**=POTENZREIHE(A5;B5;C5;D5:G5)**
7								
8								COS()
9								0,86602540

Abbildung 5.58 Bilden einer Potenzreihe zur Annäherung an den Kosinuswert

PRODUKT()

PRODUCT()

Syntax: `PRODUKT(Zahl1; Zahl2; ...)`

Beispiel: `=PRODUKT(4;5;6)`

Ergebnis: `120`

Die Funktion PRODUKT() multipliziert alle angegebenen Argumente (Zahl1, Zahl2; ...) miteinander. Seit Excel 2007 sind 255 Argumente erlaubt, 30 ist das Maximum bei den älteren Versionen. Jedes Argument kann ein numerischer Wert sein, ein Bezug auf einen Bereich, der solche Werte enthält, oder eine Matrixkonstante. Wenn ein solcher Bereich oder eine Matrixkonstante selbst wiederum zahlreiche Werte enthalten, können auch wesentlich mehr als 255 Werte in einem Zug miteinander multipliziert werden. Wenn Sie eine Funktion wie


```
=PRODUKT(A10:A12;B10:B12)
```

verwenden, entspricht das Ergebnis der Multiplikation aller Zellen in beiden Bereichen, ist also gleich:

```
=A10 * A11 * A12 * B10 * B11 * B12
```

Obwohl die Funktion PRODUKT() wie auch der *-Operator beide Multiplikationen durchführen, gibt es kleine Unterschiede, was die Behandlung unterschiedlicher Datentypen bei den Operanden angeht. Dies zeigt Abbildung 5.59. Wird als Argument der Funktion ein Bereich oder eine Matrixkonstante angegeben, werden nur die Elemente berücksichtigt, die numerisch sind, Zeichenfolgen oder Wahrheitswerte werden ignoriert, wie die Ergebnisse in den Zellen D7 bis E10 zeigen. Die Funktion verhält sich also anders als eine Multiplikation mit dem *-Operator.

	A	B	C	D	E
1					
2	**Datentypen bei der Funktion PRODUKT()**				
3					
4	Bereich1	Bereich2	A*B	PRODUKT(A;B)	PRODUKT(A:B)
5	2	5	10	10	10
6	4	4	16	16	16
7	3	apfel	#WERT!	3	3
8	abend	morgen	#WERT!	0	0
9	WAHR	WAHR	1	0	0
10	WAHR	FALSCH	0	0	0

Abbildung 5.59 Behandlung verschiedener Datentypen bei der Funktion PRODUKT()

QUADRATESUMME()

SUMSQ()

Syntax: QUADRATESUMME(**Zahl1**; Zahl2; ...)

Beispiel: =QUADRATESUMME(2;3;4)

Ergebnis: 29

Die Funktion QUADRATESUMME() berechnet die Summe der Quadrate der als Argument angegebenen Werte. Werden Zahlen in Textform direkt angegeben, werden sie ebenso berücksichtigt wie Wahrheitswerte.

```
=QUADRATESUMME(WAHR;"3";2)
```

ergibt 14.

Nur Zeichenfolgen, die sich nicht als Zahlen interpretieren lassen, führen zu dem Fehler #WERT!.

Haben die Zellen B3 bis B5 dagegen die Werte WAHR, ="3" und 2, ist das Ergebnis für die Formel =QUADRATESUMME(B3:B5) = 4. In diesem Fall werden also nur die Zellen eines Bezugs oder einer Matrix berücksichtigt, die numerisch sind, alle anderen werden ignoriert. Seit Excel 2007 können 255 Argumente verwendet werden, in älteren Versionen nur 30. Statt einzelner Werte kann auch eine einzeilige Matrix oder ein Bezug darauf angegeben werden.

	A	B	C
1			
2	**Quadratesumme**		
3			
4	Werte	=QUADRATESUMME()	Werte ^ 2
5	2	54	4
6	3		9
7	4		16
8	5		25
9		Summe	54

Abbildung 5.60 Summe der Quadrate einer Reihe von Werten

QUOTIENT()

QUOTIENT()

Syntax: QUOTIENT(**Zähler; Nenner**)
Beispiel: =QUOTIENT(15;6)
Ergebnis: 2

Die Funktion QUOTIENT() liefert das ganzzahlige Ergebnis einer Division; der Rest wird weggelassen. Der Wert für Zähler bestimmt den Dividenden, der Wert für Nenner den Divisor.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist der Nenner gleich 0, gibt die Funktion den Fehler #DIV/0! aus.

	A	B	C	D
1				
2	**Ganzzahlergebnis einer Divison**			
3				
4	Zähler	Nenner	Zähler/Nenner	QUOTIENT()
5	5	2	2,5	2
6	-23	5	-4,6	-4
7	120,33	12,25	9,822857143	9
8	240	22	10,90909091	10

Abbildung 5.61 Abtrennen der Dezimalstellen beim Ergebnis einer Division

REST()

MOD()

Syntax: `REST(Zahl;Divisor)`
Beispiel: `=REST(25;6)`
Ergebnis: `1`

Die Funktion `REST()` liefert den Restbetrag (Modulus) bei einer Division. Das erste Argument, `Zahl`, gibt den Dividenden an, der geteilt werden soll, das zweite Argument ist der `Divisor`.

Der Divisor muss ein anderer Wert als `0` sein, da ja die Division durch 0 nicht erlaubt ist und den Fehler `#DIV/0` liefern würde. Das Vorzeichen des Ergebnisses ist immer das des Divisors. Ist `Zahl` mit `0` angegeben, so ist das Ergebnis auch 0.

Eine der möglichen Verwendungen der Funktion sind bedingte Formate, um Tabellenzeilen paarweise mit unterschiedlichen Hintergründen zu belegen. Abbildung 5.62 zeigt ein Beispiel.

	A	B	C
1			
2	**Rest einer Divison**		
3			
4	Zahl	Divisor	REST()
5	5	2	1
6	-23	5	2
7	120,33	12,25	10,08
8	240	22	20
9			
10	**Bedingtes Format mit der Formel:**		
11	**=REST(ZEILE(A5);2)=1**		

Abbildung 5.62 Modulo-Rechnen mit der Funktion `REST()`

Dabei wird im Dialog für die Formatierungsregel eine Formel verwendet, die prüft, ob die Zeilennummer ohne Rest durch 2 teilbar ist.

```
=REST(ZEILE(A5);2)=1
```

Ist das der Fall, wird der Zellhintergrund anders eingefärbt.

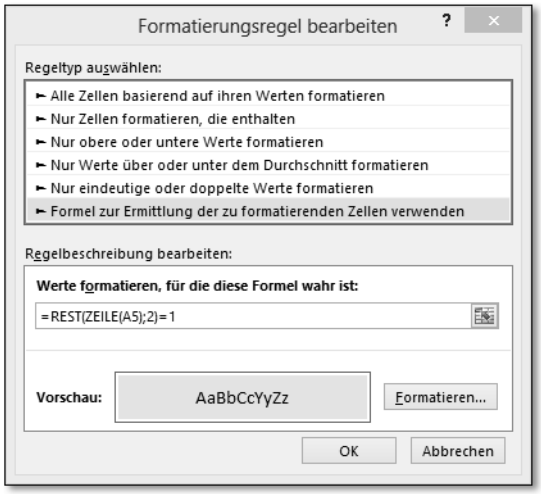


Abbildung 5.63 Der Dialog für ein bedingtes Format

5.4.1 Zyklische Wertreihen

Wenn Sie wissen wollen, ob ein bestimmtes Jahr ein Schaltjahr ist, kann die Funktion `REST()` helfen. In den meisten Fällen reicht schon die einfache Formel:

```
=REST(Jahr;4)
```

Liefert sie das Ergebnis `0`, ist das Jahr ein Schaltjahr. Bis auf wenige Ausnahmen trifft diese Regel auf alle Jahreszahlen zwischen 1900 und 2100 zu. Die Ausnahmen sind das Jahr 1900 und das Jahr 2100, die, obwohl durch 4 teilbar, keine Schaltjahre sind. Abfangen lässt sich das mit einer Erweiterung der Formel, die die beiden Ausnahmeregeln berücksichtigt: Das Jahr darf nicht durch 100 teilbar sein, wenn es nicht gleichzeitig durch 400 teilbar ist.

```
=(REST(JAHR;4)=0)-(REST(JAHR;100)=0)+(REST(JAHR;400)=0)
```

ist die Formel, die für alle Schaltjahre korrekt den Wert `0` ausgibt, für alle anderen Jahre den Wert `1`.

RÖMISCH()

ROMAN()

Syntax: RÖMISCH(**Zahl**; Typ)
Beispiel: =RÖMISCH(88)
Ergebnis: LXXXVIII

Die Funktion RÖMISCH() wandelt (arabische) Zahlen in Zeichenfolgen um, die verschiedenen Varianten der römischen Zahlendarstellung entsprechen. Mit Typ kann festgelegt werden, nach welchem Schema umgewandelt wird: 0 oder nicht festgelegt ergibt den klassischen Typus, 1, 2 und 3 ergeben eine verkürzte Form, 4 die vereinfachte. Ist für Typ WAHR gesetzt, erhalten Sie den klassischen Typ, FALSCH ergibt den vereinfachten.

Ist Zahl negativ oder > 3999, liefert die Funktion den Fehler #WERT!. Für 0 gibt die Funktion eine leere Zeichenfolge zurück.

Das römische Zahlensystem ist von Kerbzeichen, die in Kerbhölzer geritzt wurden, abgeleitet. Verwendet werden folgende Zeichen:

I für 1, V für 5, X für 10, L für 50, C für 100, D für 500, M für 1000.

Sie werden der Größe nach von links nach rechts jeweils in der benötigten Menge angeordnet, wobei die entsprechenden Werte addiert werden. Nur wenn ein Zeichen mit einem kleineren Wert vor einem Zeichen mit einem größeren Wert erscheint, wird es abgezogen, wie etwa bei XIX für 19. Dieses System ist bis ins Mittelalter verwendet worden, das Zeichen M für 1000 oder die Schreibweise IV für 4 oder IX für 9 hat sich sogar erst im Mittelalter eingebürgert.

	A	B	C	D	E	F	G	H
1								
2	**Römische Zahlen**							
3								
4		klassisch	kürzer	kürzer	kürzer	vereinfacht	klassisch	vereinfacht
5	Typ	0	1	2	3	4	WAHR	FALSCH
6	499	CDXCIX	LDVLIV	XDIX	VDIV	ID	CDXCIX	ID
7	500	D	D	D	D	D	D	D
8	999	CMXCIX	LMVLIV	XMIX	VMIV	IM	CMXCIX	IM
9	1000	M	M	M	M	M	M	M
10	1999	MCMXCIX	MLMVLIV	MXMIX	MVMIV	MIM	MCMXCIX	MIM
11	2000	MM	MM	MM	MM	MM	MM	MM

Abbildung 5.64 Beispiele für die Umwandlung in römische Zahlen

Mit Excel 2013 steht auch die Umkehrfunktion von RÖMISCH() zur Verfügung: Die Funktion ARABISCH() wandelt römische Zahlen in arabische um.

RUNDEN()

ROUND()

Syntax: RUNDEN(**Zahl**; **Anzahl_Stellen**)
Beispiel: =RUNDEN(3,45678;2)
Ergebnis: 3,46

Die Funktion RUNDEN() rundet den mit Zahl angegebenen Wert auf die mit Anzahl_Stellen angegebene Stellenzahl auf oder ab. Ist die zu rundende Dezimalstelle größer oder gleich 5, dann wird von 0 weg, also aufgerundet, im anderen Fall zur 0 hin abgerundet. =RUNDEN(3,45;1) ergibt also 3,5. =RUNDEN(3,44;1) ergibt dagegen 3,4. Negative Werte werden von der 0 weg aufgerundet: =RUNDEN(-3,256;2) ergibt -3,26, =RUNDEN(-3,254;2) dagegen -3,25. Anders als bei den Funktionen AUFRUNDEN() und ABRUNDEN() ist die Richtung der Rundung also nicht festgelegt, sondern hängt vom jeweiligen Wert ab.

	A	B	C	D
1				
2	**Runden von Zahlen**			
3				
4	Zahl	Stellen	RUNDEN()	
5	12347,85	0	12348	
6	12347,23	0	12347	
7	12347,85	1	12347,9	
8	12347,23	1	12347,2	
9	12347,85	2	12347,85	
10	12347,23	2	12347,23	
11	12347,85	3	12347,85	
12	12347,23	3	12347,23	
13	12347,85	-1	12350	
14	12344,23	-1	12340	
15	12357,85	-2	12400	
16	12344,23	-2	12300	
17	12347,85	-3	12000	
18	12747,23	-3	13000	

Abbildung 5.65 Runden mit variabler Stellenanzahl

Ist Anzahl_Stellen = 0, wird Zahl auf eine Ganzzahl gerundet, im Unterschied zur Funktion GANZZAHL(), die die Dezimalstellen einfach abschneidet. Mit einem negativen Wert für Anzahl_Stellen kann auch der links vom Komma stehende Teil von Zahl gerundet werden. =RUNDEN(1234;-3) rundet beispielsweise auf 1.000. Abbildung 5.65 zeigt einige Beispiele.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist Anzahl_Stellen größer als die Anzahl der Stellen, die der Wert von Zahl hat, werden nur die vorhandenen Stellen ausgegeben.

SEC()

SEC()

Syntax:	SEC(**Zahl**)
Beispiel:	=SEC(0,7854)
Ergebnis:	1,414

Die Funktion SEC() berechnet den Sekans des mit dem Argument Zahl angegebenen Winkels. Dabei wird der Winkel nicht in Grad, sondern im Bogenmaß angegeben. Für die Umrechnung können Sie entweder den Gradwert mit PI()/180 multiplizieren oder die Funktion BOGENMASS() verwenden:

```
=SEC(30 * PI()/180)
=SEC(BOGENMASS(30))
```

Der Sekans eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis der Hypotenuse zur Ankathete. Der Sekans ist die Umkehrfunktion des Kosinus. Es gilt also:

```
sec(x) = 1 / cos(x)
```

Der absolute Wert von Zahl muss dabei kleiner sein als 2^27. Liegt der Wert außerhalb des erlaubten Wertebereichs, liefert die Funktion den Fehlerwert #ZAHL!. Ist das Argument nicht numerisch, ergibt sich der Fehlerwert #WERT!.

Mit der neuen Funktion kann die umständliche Berechnung mit dem Kehrwert 1/COS() vereinfacht werden.

	A	B	C
1			
2	**Sekans**		
3			
4	Winkel	BOGENMASS()	SEC()
5	0	0	1,00000000
6	1	0,017453293	1,00015233
7	2	0,034906585	1,00060954
8	3	0,052359878	1,00137235
9	4	0,06981317	1,00244190
10	5	0,087266463	1,00381984
11	6	0,104719755	1,00550828
12	7	0,122173048	1,00750983
13	8	0,13962634	1,00982757
14	9	0,157079633	1,01246513
15	10	0,174532925	1,01542661
16	11	0,191986218	1,01871669
17	12	0,20943951	1,02234059
18	13	0,226892803	1,02630411
19	14	0,244346095	1,03061363
20	15	0,261799388	1,03527618
21	16	0,27925268	1,04029944
22	17	0,296705973	1,04569176
23	18	0,314159265	1,05146222
24	19	0,331612558	1,05762068
25	20	0,34906585	1,06417777

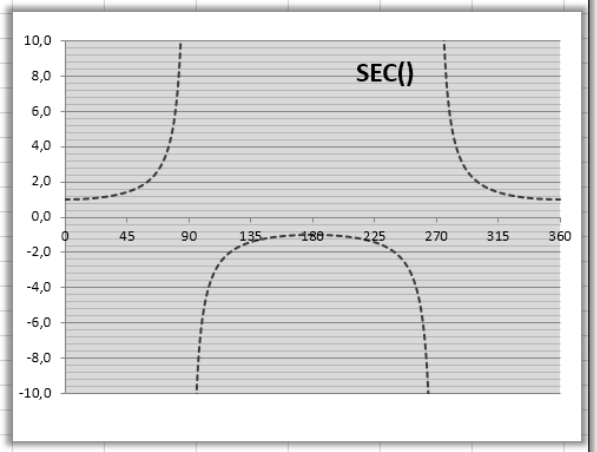


Abbildung 5.66 Die Funktion SEC() und ihr Graph

SECHYP()

SECH()

Syntax: SECHYP(**Zahl**)
Beispiel: =SECHYP(0,7854)
Ergebnis: 0,7549

Die Funktion SECHYP() berechnet den hyperbolischen Sekans des mit dem Argument Zahl angegebenen Winkels. Dabei wird der Winkel nicht in Grad, sondern im Bogenmaß angegeben. Ist das Argument nicht numerisch, ergibt sich der Fehlerwert #WERT!.

Die Funktion arbeitet nach der Formel:

$$\text{sech}(x) = \frac{2}{e^{x} + e^{-x}}$$

Der Graph der Funktion ist im Bereich $x > 0$ streng monoton fallend, im Bereich $x < 0$ dagegen monoton steigend.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

	A	B	C
1			
2	**Hyperbolischer Sekans**		
3			
4	Winkel	BOGENMASS()	SECHYP()
5	-360	-6,283185307	0,003734872
6	-359	-6,265732015	0,00380063
7	-358	-6,248278722	0,003867545
8	-357	-6,23082543	0,003935639
9	-356	-6,213372137	0,004004931
10	-355	-6,195918845	0,004075443
11	-354	-6,178465552	0,004147197
12	-353	-6,16101226	0,004220214
13	-352	-6,143558967	0,004294516
14	-351	-6,126105675	0,004370127
15	-350	-6,108652382	0,004447069
16	-349	-6,091199089	0,004525365
17	-348	-6,073745797	0,00460504
18	-347	-6,056292504	0,004686118
19	-346	-6,038839212	0,004768623
20	-345	-6,021385919	0,004852581
21	-344	-6,003932627	0,004938017
22	-343	-5,986479334	0,005024957
23	-342	-5,969026042	0,005113427
24	-341	-5,951572749	0,005203456
25	-340	-5,934119457	0,005295069

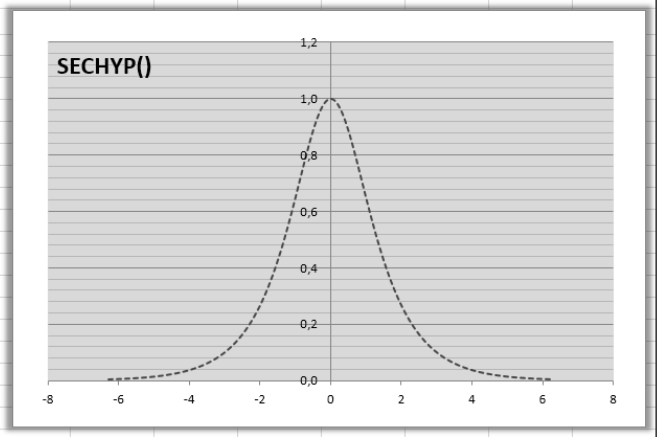


Abbildung 5.67 Die Funktion `SECHYP()` und ihr Graph

SIN()

SIN()

Syntax: `SIN(`**Zahl**`)`
Beispiel: `=SIN(30*PI()/180)`
Ergebnis: `0,5`

Die Funktion `SIN()` berechnet den Sinus des mit dem Argument `Zahl` angegebenen Winkels. Dabei wird der Winkel nicht in Grad, sondern im Bogenmaß angegeben. Für die Umrechnung können Sie entweder den Gradwert mit `PI()/180` multiplizieren oder die Funktion `BOGENMASS()` verwenden:

```
=SIN(30 * PI()/180)
=SIN(BOGENMASS(30))
```

Der Sinus eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis der Gegenkathete zur Hypotenuse.

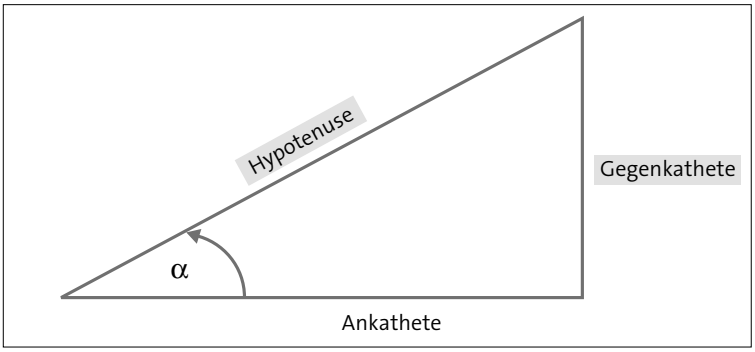


Abbildung 5.68 Der Sinus liefert das Verhältnis der Gegenkathete zur Hypotenuse.

Die Definition kann auf der Basis des Einheitskreises erweitert werden. Einem Winkel von 90 Grad entspricht dann genau ein Sinuswert von 1, 270 Grad entspricht –1, 0 und 360 Grad entsprechen 0. Das Ergebnis der Funktion liegt also immer zwischen –1 und 1.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Das Diagramm in Abbildung 5.69 zeigt den periodischen Verlauf der Sinusfunktion für den Bereich von 0 bis 360 Grad.

Sinus eines Winkels

Winkel	BOGENMASS()	SIN()
0	0	0
1	0,017453293	0,017452406
2	0,034906585	0,034899497
3	0,052359878	0,052335956
4	0,06981317	0,069756474
5	0,087266463	0,087155743
6	0,104719755	0,104528463
7	0,122173048	0,121869343
8	0,13962634	0,139173101
9	0,157079633	0,156434465
10	0,174532925	0,173648178
11	0,191986218	0,190808995
12	0,20943951	0,207911691
13	0,226892803	0,224951054
14	0,244346095	0,241921896
15	0,261799388	0,258819045
16	0,27925268	0,275637356

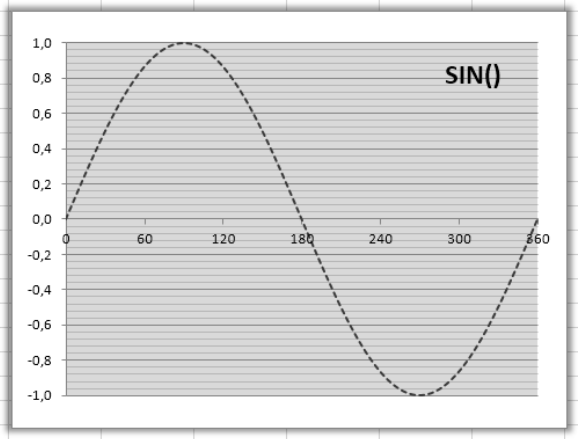


Abbildung 5.69 Berechnen des Sinus aus dem Bogenmaß

Die SIN()-Funktion kann beispielsweise bei Vermessungen verwendet werden, um Höhenunterschiede zu berechnen. Wenn die Entfernung zu einer Kirchturmspitze 100 m beträgt und der Anpeilwinkel 17 Grad ist, kann die Höhe des Kirchturms wie folgt berechnet werden:

```
=SIN(BOGENMASS(17))*100
```

ergibt 29,23 m.

SINHYP()

SINH()

Syntax: SINHYP(**Zahl**)
Beispiel: =SINHYP(1,5)
Ergebnis: 2,129279

	A	B
1		
2	**Hyperbolischer Sinus**	
3		
4	Zahl	SINHYP()
5	-4	-27,28992
6	-3,9	-24,69110
7	-3,8	-22,33941
8	-3,7	-20,21129
9	-3,6	-18,28546
10	-3,5	-16,54263
11	-3,4	-14,96536
12	-3,3	-13,53788
13	-3,2	-12,24588
14	-3,1	-11,07645
15	-3	-10,01787
16	-2,9	-9,05956
17	-2,8	-8,19192
18	-2,7	-7,40626
19	-2,6	-6,69473
20	-2,5	-6,05020
21	-2,4	-5,46623
22	-2,3	-4,93696
23	-2,2	-4,45711
24	-2,1	-4,02186
25	-2	-3,62686
26	-1,9	-3,26816
27	-1,8	-2,94217

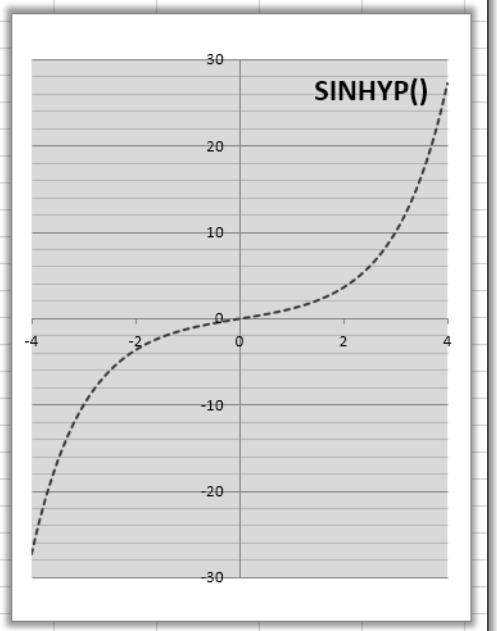


Abbildung 5.70 Berechnen des hyperbolischen Sinus und seine grafische Darstellung

Die Funktion SINHYP() liefert den hyperbolischen Sinus des mit Zahl angegebenen Werts. Dabei muss es sich um eine reelle Zahl handeln, die zwischen –unendlich und unendlich liegen kann. Gerechnet wird mit der Formel:

```
SINHYP(x) = (e^x - e^-x)/2
```

Das Ergebnis von =SINHYP(x) liegt ebenfalls zwischen –unendlich und unendlich. Die Funktion ist monoton steigend.

Nicht numerische Werte ergeben den Fehler #WERT!.

SUMME()

SUM()

Syntax: SUMME(**Zahl1**; Zahl2; ...)
Beispiel: =SUMME(4;4,5;5)
Ergebnis: 13,5

Die Funktion SUMME() berechnet die Summe der angegebenen Argumente. Insgesamt sind seit Excel 2007 255, in den älteren Versionen 30 Argumente erlaubt. Die Argumente können als Konstanten, als Matrixkonstanten, als Zellbezüge oder als Formelausdrücke, die Werte liefern, eingegeben werden. Am effektivsten kann die Funktion zum Summieren von Zellbereichen eingesetzt werden.

Enthalten die Argumente direkt angegebene Wahrheitswerte oder als Text eingegebene Zahlen, werden diese wie Zahlen verarbeitet.

```
=SUMME(WAHR;WAHR;"3")
```

ergibt also 5, da WAHR als 1 berechnet wird.

Texte, die nicht in Zahlen umgewandelt werden können, führen zu Fehlern:

```
=SUMME("www";3)
```

führt zu dem Fehler #WERT!. Bei Fehlerwerten liefert die Funktion als Ergebnis den ersten Fehlerwert. Ist als Argument eine Matrixkonstante oder ein Zellbezug angegeben, werden dagegen nur numerische Werte verarbeitet; leere Zellen, Texte und Wahrheitswerte werden ignoriert.

Wie sich unterschiedliche Datentypen auswirken, zeigt die Tabelle in Abbildung 5.71. Auch hier ergeben sich wieder Unterschiede (analog zu PRODUKT()) zu den Ergebnissen von Formeln, die mit dem +-Operator aufgebaut sind.

	A	B	C	D	E
1					
2	**Datentypen bei der Funktion SUMME()**				
3					
4	Bereich1	Bereich2	A+B	SUMME(A;B)	SUMME(A:B)
5	2	5	7	7	7
6	4	4	8	8	8
7	abend	morgen	#WERT!	0	0
8	WAHR	FALSCH	1	0	0
9	3	apfel	#WERT!	3	3
10	WAHR	WAHR	2	0	0
11	SUMME(Bereich1;Bereich2):		18		

Abbildung 5.71 Datentypbehandlung bei der Funktion `SUMME()`

5.4.2 Aufsummierungen

Durch eine geschickte Mischung von relativen und absoluten Bereichsbezügen können Sie mit Hilfe der Funktion `SUMME()` auch Aufsummierungen vornehmen. Hier das einfache Beispiel einer Wochenauswertung mit einer Spalte von Quersummen, die jeweils die Vortagsergebnisse mit enthalten. Die erste Quersumme in der Zelle G5 enthält die Formel:

```
=SUMME($B$5:F5)
```

G6 =SUMME($B$5:F6)

	A	B	C	D	E	F	G
1							
2	**Wochenergebnis**						
3							
4		**Sorte 1**	**Sorte 2**	**Sorte 3**	**Sorte 4**	**Sorte 5**	**aufgelaufenes Gesamtergebnis**
5	**Montag**	200	300	250	340	230	**1090**
6	**Dienstag**	200	300	250	340	230	**2640**
7	**Mittwoch**	200	300	250	340	230	**3960**
8	**Donnerstag**	200	300	250	340	230	**5280**
9	**Freitag**	200	300	250	340	230	**6600**

Abbildung 5.72 Durch Mischbezüge lassen sich leicht Aufsummierungen vornehmen.

Im ersten Teil des Bereichsarguments wird mit der absolut gesetzten Adresse `$B$5` ein fixer Ausgangspunkt angegeben. Sie können diese Formeln anschließend nach unten kopieren. In der Zelle G6 steht dann die Formel:

```
=SUMME($B$5:F6)
```

die bereits das aufgelaufene Ergebnis für beide Zeilen liefert.

SUMMENPRODUKT()

SUMPRODUCT()

Syntax: `SUMMENPRODUKT(`**`Array1`**`; Array2; Array3; ...)`
Beispiel: `=SUMMENPRODUKT({4.5.6};{1.2.3})`
Ergebnis: `32 für (4*1+5*2+6*3)`

Die Funktion `SUMMENPRODUKT()` bewirkt, dass die einzelnen Elemente der als Argumente angegebenen Matrizen der Reihe nach miteinander multipliziert und anschließend die Multiplikationsergebnisse summiert werden.

Die Matrizen müssen von der Zeilenanzahl und der Spaltenanzahl her gleich sein, sonst liefert die Funktion den Fehler `#WERT!`.

Seit Excel 2007 sind 2 bis 255 Matrizen als Argumente erlaubt, in den älteren Versionen nur 2 bis 30. Nicht numerische Elemente in einer Matrix werden als `0` gewertet.

Es handelt sich also hier nicht, wie der Name der Funktion nahelegt, um ein Summenprodukt, sondern um eine Produktensumme, weil erst multipliziert wird und dann die Ergebnisse addiert werden. Die Funktion ist immer dann nützlich, wenn die einzelnen Produkte nicht angezeigt werden müssen oder sollen, sondern nur die Endsumme. Das in Abbildung 5.73 gezeigte Beispiel zeigt eine typische Anwendung der Funktion, bei der es darum geht, das Gesamtgewicht einer Lieferung zu bestimmen.

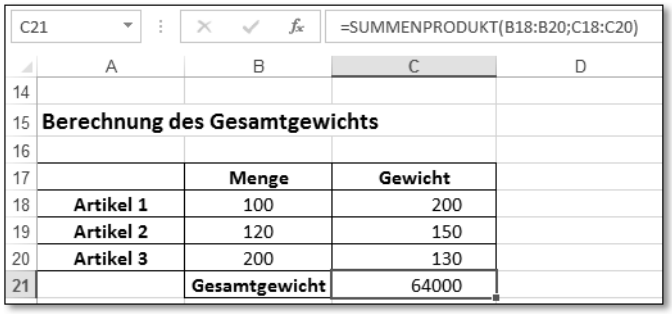


Abbildung 5.73 Die Funktion `SUMMENPRODUKT()` liefert das Gesamtgewicht in einem Zug.

SUMMEWENN()

SUMIF()

Syntax: `SUMMEWENN(`**`Bereich`**`; `**`Suchkriterien`**`; Summe_Bereich)`
Beispiel: `=SUMMEWENN(A6:A14;C5;B6:B14)`, siehe Abbildung 5.74

Die Funktion SUMMEWENN() vergleicht die Werte aus dem mit Bereich angegebenen Zellbereich mit dem Suchkriterium, das über das Argument Suchkriterien angegeben ist. Dies kann eine Zahl, eine Zeichenfolge oder ein Ausdruck mit einem Vergleichsoperator wie "<10" oder ">Karl" sein oder der Bezug auf eine Zelle, die ein Kriterium enthält.

Bei der Formulierung der Kriterien dürfen auch Jokerzeichen verwendet werden:

- * steht für eine beliebige Anzahl von Zeichen einschließlich 0.
- ? steht für ein Zeichen.
- ~ wird vor * oder ? gesetzt, wenn diese Zeichen selbst gesucht werden sollen.

Werden Werte in Bereich gefunden, die dem Kriterium entsprechen, sucht die Funktion nach korrespondierenden Werten in einem zweiten Bereich Summe_Bereich und summiert diese auf. Wird kein Wert gefunden, der dem Kriterium entspricht, wird 0 ausgegeben. Wird Summe_Bereich nicht angegeben, wird der Teil von Bereich aufsummiert, der das Suchkriterium erfüllt.

Die angegebene Größe von Summe_Bereich muss nicht mit der von Bereich übereinstimmen. Excel benutzt die Angabe der linken oberen Eckzelle von Summe_Bereich und versucht, von da aus einen Bereich auszuwerten, der genauso groß ist wie Bereich.

Abbildung 5.74 zeigt eine Tabelle mit den Besuchszahlen von Vertretern. Zelle C5 enthält eine Gültigkeitsregel, die die Vertreternamen als Dropdown-Liste anbietet. Die Zelle D5 liefert dann jeweils die Summe der Besuche des in C5 ausgewählten Vertreters.

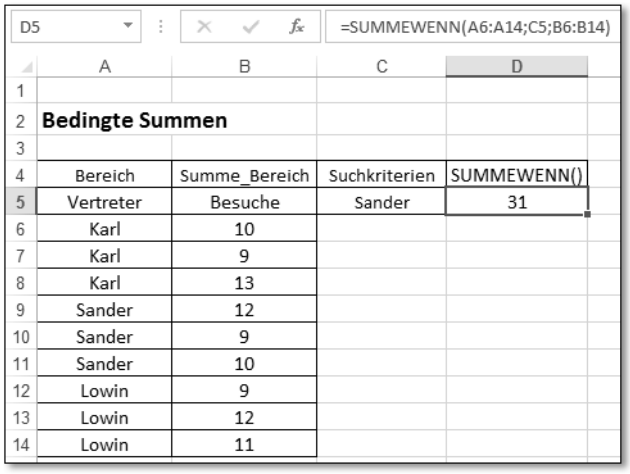


Abbildung 5.74 Bedingte Summenbildung

Bedingte Zusammenfassungen von Werten stellt Excel auch über andere Funktionen zur Verfügung: SUMMEWENNS(), ZÄHLENWENN(), MITTELWERTWENN(), MITTELWERTWENNS().

SUMMEWENNS()

SUMIFS()

Syntax: `SUMMEWENNS(Summe_Bereich; Kriterium_Bereich1; Kriterien1; Kriterium_Bereich2; Kriterien2 ...)`

Beispiel: siehe Abbildung 5.75

Die Funktion `SUMMEWENNS()` erweitert die Möglichkeiten der Funktion `SUMMEWENN()` um zusätzliche Auswahlkriterien. Sie steht erst seit Excel 2007 zur Verfügung. Das Argument `Summe_Bereich` gibt den Zellbereich an, in dem sich die zu summierenden Werte befinden.

Es folgt eine Liste von Bereichspaaren, die jeweils zunächst den Bereich angeben, für den das anschließend angegebene Kriterium geprüft werden soll. Es sind bis zu 127 solcher Bereichspaare möglich. Die Kriterien sind additiv zu verstehen, eine Summierung findet also nur statt, wenn alle Kriterien erfüllt sind. Anders als bei `SUMMEWENN()` gilt dabei: Alle Argumente für `Kriterium_Bereich(n)` müssen Bereiche angeben, die genauso groß sind wie `Summe_Bereich`.

Bei der Formulierung der Kriterien dürfen auch Jokerzeichen verwendet werden:

- * steht für eine beliebige Anzahl von Zeichen einschließlich 0.
- ? steht für ein Zeichen.
- ~ wird vor * oder ? gesetzt, wenn diese Zeichen selbst gesucht werden sollen.

In dem Beispiel in Abbildung 5.75 werden die Teilnehmerzahlen aus Zeile 5 summiert, wenn sie zwei Kriterien erfüllen.

E9 =SUMMEWENNS(B5:E5;B6:E6;"<30%";B8:E8;">=20%")

	A	B	C	D	E
1					
2	**Mehrfach bedingte Summen**				
3					
4		Kurs 1	Kurs 2	Kurs 3	Kurs 4
5	Teilnehmer	80	60	70	110
6	unter 20	30%	25%	15%	25%
7	20 - 50	60%	65%	45%	55%
8	über 50	10%	10%	40%	20%
9	Summe der Teilnehmer der Kurse mit einem Anteil der unter Zwanzigjährigen von unter 30 % und einem Anteil der über Fünfzigjährigen von mindestens 20 %.				180

Abbildung 5.75 Kombination von Kriterien für eine bedingte Summenbildung

Bedingte Zusammenfassungen von Werten stellt Excel auch über folgende andere Funktionen zur Verfügung: SUMMEWENN(), ZÄHLENWENN(), ZÄHLENWENNS(), MITTELWERTWENN(), MITTELWERTWENNS().

SUMMEX2MY2()

SUMX2MY2()

Syntax: SUMMEX2MY2(**Matrix_x**; **Matrix_y**)
Beispiel: siehe Abbildung 5.76

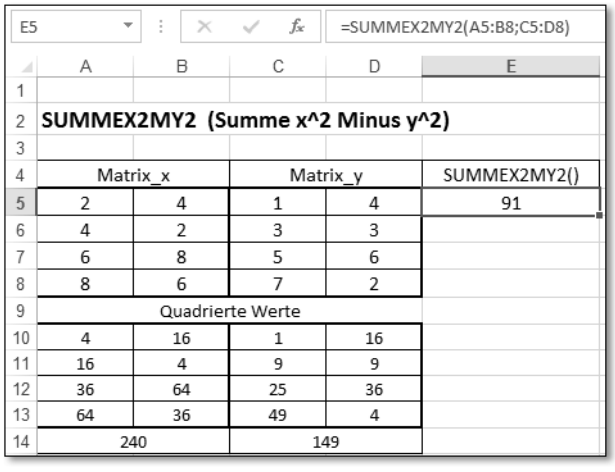
Die Funktion SUMMEX2MY2() summiert die Differenzen der quadrierten x-Werte und der quadrierten y-Werte, die über die Matrizen Matrix_x und Matrix_y angegeben sind. (Der Name der Funktion bedeutet also: Summe x^2 minus y^2).

Die Funktion arbeitet mit der Formel:

$$\text{SUMMEX2MY2} = \Sigma(x^2 - y^2)$$

Um die Arbeitsweise der Funktion zu erläutern, wurde in der Beispieltabelle eine Spalte für x^2-Werte und eine für y^2-Werte angelegt. Die Summen der Spalten voneinander abgezogen geben das Resultat der Funktion wieder.

Die beiden Matrizen müssen dieselbe Anzahl von Werten haben, sonst wird der Fehler #NV ausgegeben. Zellen innerhalb der Matrizen, die keine numerischen Werte enthalten, werden bei der Berechnung ignoriert.



	A	B	C	D	E
1					
2	**SUMMEX2MY2 (Summe x^2 Minus y^2)**				
3					
4	Matrix_x		Matrix_y		SUMMEX2MY2()
5	2	4	1	4	91
6	4	2	3	3	
7	6	8	5	6	
8	8	6	7	2	
9	Quadrierte Werte				
10	4	16	1	16	
11	16	4	9	9	
12	36	64	25	36	
13	64	36	49	4	
14	240		149		

Abbildung 5.76 Die Summe der Differenzen zwischen quadrierten Werten

Die Funktion SUMMEX2MY2() und die verwandten Funktionen SUMMEX2PY2() und SUMMEXMY2() werden bei statistischen Anwendungen eingesetzt, beispielsweise wenn es um die Vergleiche der Mittelwerte von mehreren Stichproben geht.

SUMMEX2PY2()

SUMX2PY2()

Syntax: SUMMEX2PY2(**Matrix_x**; **Matrix_y**)

Beispiel: siehe Abbildung 5.77

Die Funktion SUMMEX2PY2() addiert die Summen der quadrierten x-Werte und der quadrierten y-Werte, die über die Matrizen Matrix_x und Matrix_y angegeben sind. (Der Name der Funktion bedeutet also: Summe x^2 plus y^2).

Die Funktion arbeitet mit der Formel:

$$SUMMEX2PY2 = \Sigma(x^2 + y^2)$$

Die beiden Matrizen müssen dieselbe Anzahl von Werten haben, sonst wird der Fehler #NV ausgegeben. Zellen innerhalb der Matrizen, die keine numerischen Werte enthalten, werden bei der Berechnung ignoriert.

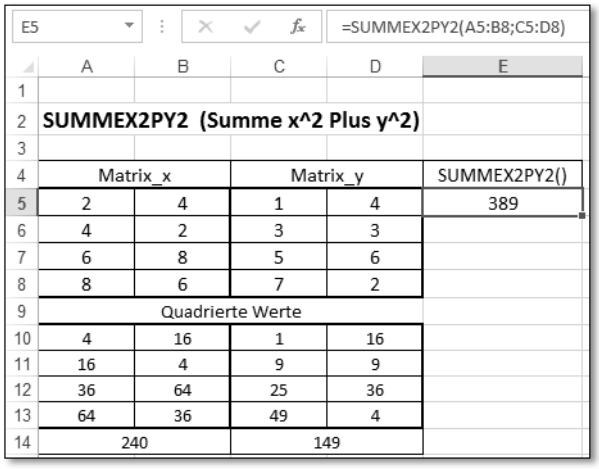


Abbildung 5.77 Die Summe der Addition quadrierter Werte

Die Funktion SUMMEX2PY2() und die verwandten Funktionen SUMMEX2MY2() sowie SUMMEXMY2() werden bei statistischen Anwendungen eingesetzt, beispielsweise wenn es um die Vergleiche der Mittelwerte von mehreren Stichproben geht.

SUMMEXMY2()

SUMXMY2()

Syntax: SUMMEXMY2(**Matrix_x**; **Matrix_y**)
Beispiel: siehe Abbildung 5.78

Die Funktion SUMMEXMY2() bewirkt, dass die Differenz zwischen x-Werten und y-Werten, die über die Matrizen Matrix_x und Matrix_y angegeben sind, zunächst ausquadriert wird und dann die einzelnen Werte aufsummiert werden. (Der Name der Funktion bedeutet also: Summe (x minus y)^2). Die Funktion arbeitet mit der Formel:

$$\text{SUMMEXMY2} = \Sigma(x - y)^2$$

Die beiden Matrizen müssen dieselbe Anzahl von Werten haben, sonst wird der Fehler #NV ausgegeben. Zellen innerhalb der Matrizen, die keine numerischen Werte enthalten, werden bei der Berechnung ignoriert.

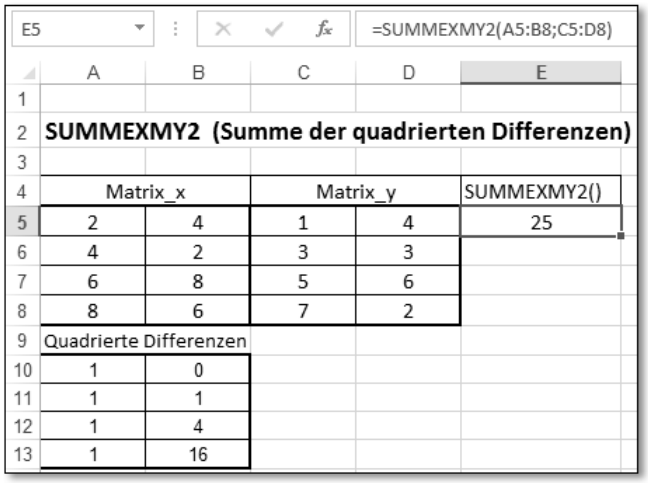


Abbildung 5.78 Summe der quadrierten Differenzen zwischen Matrixwerten

Die Funktion SUMMEXMY2() und die verwandten Funktionen SUMMEX2MY2() sowie SUMMEX2PY2() werden bei statistischen Anwendungen eingesetzt, beispielsweise wenn es um die Vergleiche der Mittelwerte von mehreren Stichproben geht.

TAN()

TAN()

Syntax: `TAN(`**Zahl**`)`

Beispiel: `=TAN(30*PI()/180)`

Ergebnis: `0,577350`

Die Funktion `TAN()` berechnet den Tangens des mit dem Argument `Zahl` angegebenen Winkels. Dabei wird der Winkel nicht in Grad, sondern im Bogenmaß angegeben. Für die Umrechnung können Sie entweder den Gradwert mit `PI()/180` multiplizieren oder die Funktion `BOGENMASS()` verwenden:

```
=TAN(30 * PI()/180)
=TAN(BOGENMASS(30))
```

Der Tangens eines Winkels wird am rechtwinkligen Dreieck definiert als das Verhältnis von Gegenkathete zu Ankathete. Folglich muss, wenn beide Katheten gleich lang sind, also bei einem Winkel von 45 Grad, der Wert 1 sein.

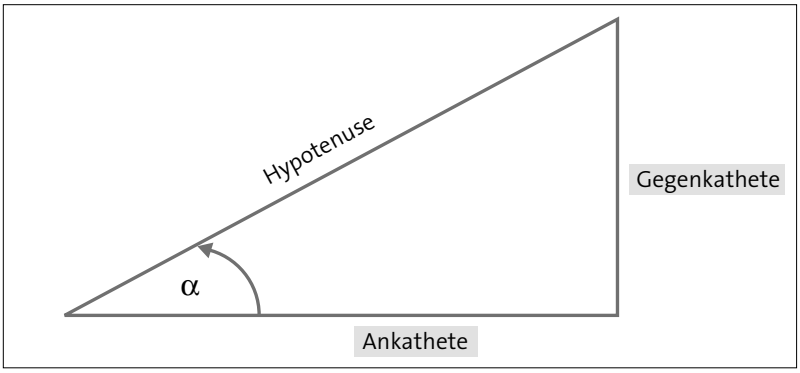


Abbildung 5.79 Der Tangens liefert das Verhältnis von Gegenkathete zu Ankathete.

Die Definition des Tangens wird auf der Basis des Einheitskreises erweitert, siehe dazu auch Abbildung 5.5. Der Wertebereich der Funktion liegt dann zwischen $+\infty$ und $-\infty$. Der Tangens für 90 Grad und –90 Grad ist nicht definiert. Das ist leicht zu erklären, weil in beiden Fällen die Ankathete mit dem Wert 0 angenommen werden müsste, was eine Division durch 0 bedeuten würde. Die Funktion `TAN()` versucht, diesen Fehler zu vermeiden, indem sie einen gegen unendlich laufenden Wert ausgibt.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Das Diagramm in Abbildung 5.80 zeigt, dass die Funktion in den jeweiligen Intervallen monoton steigend ist.

Für die `TAN()`-Funktion gilt außerdem die folgende Beziehung zu der `SIN()`- und der `COS()`-Funktion:

```
TAN(x) = SIN(x) / COS(x)
```


Tangens eines Winkels

Winkel	BOGENMASS()	TAN()
0	0	0
1	0,017453293	0,017455065
2	0,034906585	0,034920769
3	0,052359878	0,052407779
4	0,06981317	0,069926812
5	0,087266463	0,087488664
6	0,104719755	0,105104235
7	0,122173048	0,122784561
8	0,13962634	0,140540835
9	0,157079633	0,15838444
10	0,174532925	0,176326981
11	0,191986218	0,194380309
12	0,20943951	0,212556562
13	0,226892803	0,230868191
14	0,244346095	0,249328003
15	0,261799388	0,267949192
16	0,27925268	0,286745386
17	0,296705973	0,305730681
18	0,314159265	0,324919696

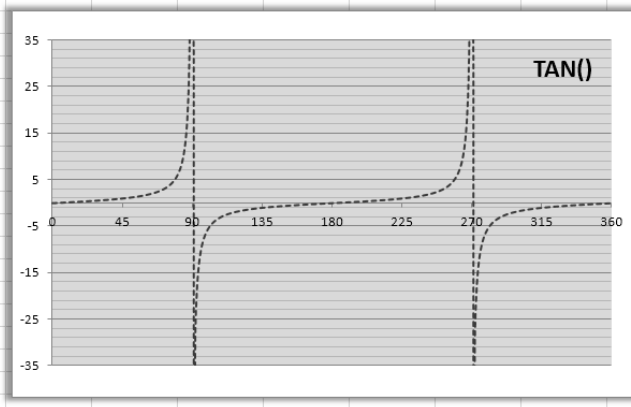


Abbildung 5.80 Die Tangensfunktion und ihr Graph

In der Praxis wird die `TAN()`-Funktion beispielsweise zur Beschreibung von Steigungen benutzt. So werden etwa die Angaben zum Anstieg einer Straße in %-Werten ausgedrückt, die näherungsweise dem Tangens des Steigungswinkels entsprechen. Einem Anstieg von 10 Grad entspricht ein Tangens von 0,17. Auf dem Straßenschild wird dann 17% angegeben, was 17 m Höhenunterschied pro 100 m Entfernung bedeutet.

Bis zur Version 2013 bot Excel keine eigene Funktion für den Kotangens an. Da aber der Kotangens der Kehrwert des Tangens ist, lässt er sich leicht gemäß der Formel

```
COT(x) = 1/TAN(x)
```

berechnen. In Excel 2013 benutzen Sie jetzt dafür die neue Funktion `COT()`.

TANHYP()

TANH()

Syntax: `TANHYP(`**`Zahl`**`)`

Beispiel: `=TANHYP(1,5)`

Ergebnis: `0,905148`

Die Funktion `TANHYP()` liefert den hyperbolischen Tangens des mit `Zahl` angegebenen Werts. Dabei muss es sich um eine reelle Zahl handeln, die zwischen –unendlich und +unendlich liegen kann. Gerechnet wird mit der Formel:

```
TANHYP(x) = (e^x - e^-x)/(e^x + e^-x) = SINHYP(x) / COSHYP(x)
```

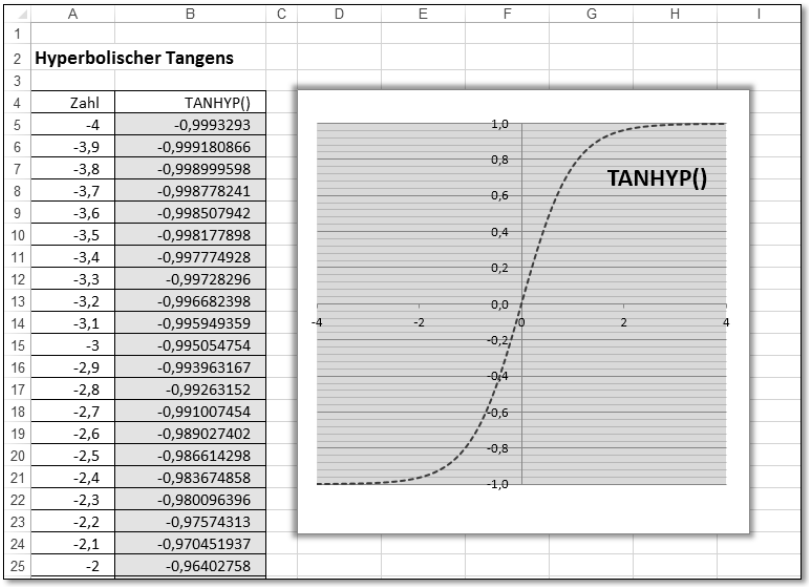
Das Ergebnis von `=TANHYP(x)` liegt zwischen –1 und 1. Die Funktion ist monoton steigend.

Nicht numerische Werte ergeben den Fehler `#WERT!`.

Der hyperbolische Kotangens, für den Excel bis zur Version 2010 keine eigene Funktion angeboten hat, errechnet sich analog zu den Winkelfunktionen als:

```
coth(x) = 1/tanh(x)
```

In Excel 2013 steht hier die neue Funktion `COTHYP()` zur Verfügung.



Hyperbolischer Tangens

Zahl	TANHYP()
-4	-0,9993293
-3,9	-0,999180866
-3,8	-0,998999598
-3,7	-0,998778241
-3,6	-0,998507942
-3,5	-0,998177898
-3,4	-0,997774928
-3,3	-0,99728296
-3,2	-0,996682398
-3,1	-0,995949359
-3	-0,995054754
-2,9	-0,993963167
-2,8	-0,99263152
-2,7	-0,991007454
-2,6	-0,989027402
-2,5	-0,986614298
-2,4	-0,983674858
-2,3	-0,980096396
-2,2	-0,97574313
-2,1	-0,970451937
-2	-0,96402758

Abbildung 5.81 Der hyperbolische Tangens und seine grafische Darstellung

TEILERGEBNIS()

SUBTOTAL()

Syntax: `TEILERGEBNIS(`**`Funktion`**`; `**`Bezug1`**`; Bezug2; ...)`
Beispiel: siehe Abbildung 5.82

Die Funktion `TEILERGEBNIS()` gibt ein Teilergebnis aus einer Tabelle oder Datentabelle zurück. Gleiche Ergebnisse erzielen Sie seit Excel 2007 auch mit **Daten • Gliederung • Teilergebnis**, in den früheren Versionen mit **Daten • Teilergebnisse**. `Funktion` gibt mittels einer Nummer von `1` bis `11` bzw. `101` bis `110` eine Funktion an, mit der die Teilergebnisse berechnet werden sollen.

Andere Werte für `Funktion` führen zu dem Fehler `#WERT!`.

Nr.	Funktion
1	MITTELWERT
2	ANZAHL
3	ANZAHL2
4	MAX
5	MIN
6	PRODUKT
7	STABW.S
8	STABW.N
9	SUMME
10	VAR.S
11	VAR.P

Tabelle 5.6 Werte für das Argument `Funktion`

Wird auf die Funktionsnummer jeweils 100 addiert – beispielsweise 109 für `Summe` –, wird die Funktion so verwendet, dass ausgeblendete Werte ignoriert werden. Alle möglichen Werte werden angeboten, sobald die öffnende Klammer eingegeben ist.

`Bezug(n)` gibt die Bereiche an, für die Teilergebnisse berechnet werden sollen. Bis zu 254 Bereiche sind möglich. Bereits vorhandene Teilergebnisse innerhalb der angegebenen Bereiche werden ignoriert, weil sie sonst mehrfach in die Berechnung eingehen würden.

Abbildung 5.82 zeigt ein Beispiel für die Auswertung von Testergebnissen für verschiedene Altersstufen. Der Datenbereich der oberen Tabelle ohne die Beschriftungszeile ist mit dem Namen `Testwerte` belegt. Im unteren Teil werden die verschiedenen Auswertungsmöglichkeiten gezeigt, wobei die Werte für das Argument `Funktion` aus der ersten Spalte übernommen werden. Werden aus der oberen Tabelle bestimmte Altersgruppen über die Filterschaltflächen herausgefiltert, werden die Werte in der unteren Tabelle sofort aktualisiert.

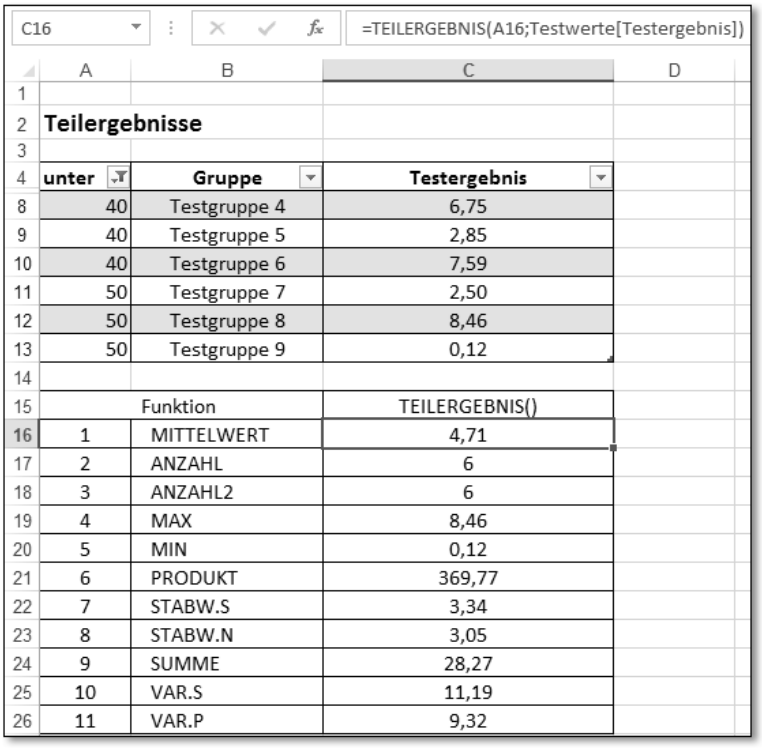


Abbildung 5.82 Gruppenauswertungen mit der Funktion `TEILERGEBNIS()`

UNGERADE()

ODD()

Syntax: `UNGERADE(`**`Zahl`**`)`

Beispiel: `=UNGERADE(-5,3)`

Ergebnis: `-7`

Die Funktion UNGERADE() rundet Zahlen zur nächsten ungeraden Ganzzahl von 0 weg und ist damit komplementär zur Funktion GERADE(). Wenn Zahl schon eine ungerade ganze Zahl ist, dann wird dieser Wert wiedergegeben.

Ist Zahl kein numerischer Wert, gibt die Funktion den Fehler #WERT! zurück. Ob ein Wert eine gerade Ganzzahl ist, kann mit der Funktion ISTUNGERADE() geprüft werden.

	A	B
1		
2	**Runden auf die nächste ungerade Zahl**	
3		
4	Zahl	UNGERADE()
5	0	1
6	1	1
7	1,5	3
8	2,9	3
9	4	5
10	5,2	7
11	6,2	7
12	8,45	9
13	-3,4	-5

Abbildung 5.83 Aufrunden auf die nächstliegende ungerade Zahl

UNTERGRENZE.MATHEMATIK()

FLOOR.PRECISE()

Syntax: UNTERGRENZE.MATHEMATIK(**Zahl**; Schritt: Modus)

Beispiel: =UNTERGRENZE.MATHEMATIK(2,2434; -0,05)

Ergebnis: 2,2

Die in Excel 2013 umbenannte Funktion – in Excel 2010 hieß sie UNTERGRENZE.GENAU() – rundet den mit Zahl angegebenen Wert auf die nächste Ganzzahl oder das nächste Vielfache von Schritt ab und ist damit komplementär zu OBERGRENZE.MATHEMATIK().

Zahl muss kleiner als 9,99E+307 und größer als -2,229E-308 sein. Fehlt der Wert für Schritt, wird 1 oder -1 vorgegeben, abhängig davon, ob Zahl ein positiver oder ein negativer Wert ist. Mit einem optionalen Wert Modus von < 0 kann erreicht werden, dass bei negativen Werten auf den nächstgrößeren Wert – hin zur 0 – gerundet werden soll. Bei positiven Werten wird der Wert für Modus ignoriert.

Nicht numerische Werte ergeben den Fehler #WERT!.

	A	B	C	D
1				
2	**UNTERGRENZE.MATHEMATIK**			
3				
4	Wert	Schritt	Modus	UNTERGRENZE.MATHEMATIK()
5	14,25	0,5		14
6	14,33	1		14
7	123,56	2		122
8	124,33	5		120
9	126,33	5		125
10	124,33	10		120
11	-6,3	2		-8
12	-6,3	2	-1	-6

Abbildung 5.84 Abrunden auf das kleinste Vielfache des als `Schritt` angegebenen Werts

Bei dieser Variante sind auch unterschiedliche Vorzeichen für `Zahl` und `Schritt` möglich, weil die Funktion immer mit dem Absolutwert von `Schritt` arbeitet. `UNTERGRENZE.MATHEMATIK(-4,2546;-0,5)` liefert deshalb ebenso wie `UNTERGRENZE.MATHEMATIK(-4,2546;0,5)` als Ergebnis `-4,5`.

VORZEICHEN()

SIGN()

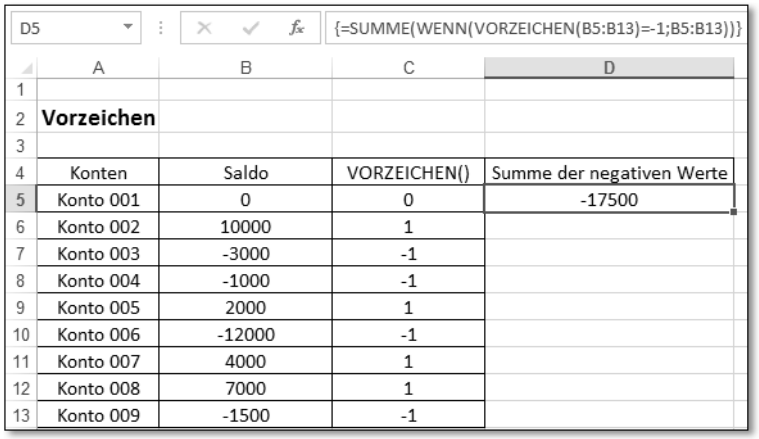
Syntax: `VORZEICHEN(`**Zahl**`)`
Beispiel: `=VORZEICHEN(-6)`
Ergebnis: `-1`

Die Funktion `VORZEICHEN()` ergibt bei positiven Zahlen 1, bei negativen Zahlen –1 und bei null 0. Sie lässt sich immer dann anwenden, wenn Operationen vom Vorzeichen abhängig sein sollen. Wenn etwa in der Zelle B22 die Gewinne errechnet werden, dann liefert (eingetragen z.B. in C22) die Formel:

```
WENN(VORZEICHEN(B22)= 1;"OK";"Kein Gewinn!")
```

ein `OK`, solange ein Gewinn > `0` vorhanden ist. Bei Verlusten erscheint dagegen der Hinweis `Kein Gewinn!`.

Nicht numerische Werte ergeben den Fehler `#WERT!`.



D5 | {=SUMME(WENN(VORZEICHEN(B5:B13)=-1;B5:B13))}

	A	B	C	D
1				
2	**Vorzeichen**			
3				
4	Konten	Saldo	VORZEICHEN()	Summe der negativen Werte
5	Konto 001	0	0	-17500
6	Konto 002	10000	1	
7	Konto 003	-3000	-1	
8	Konto 004	-1000	-1	
9	Konto 005	2000	1	
10	Konto 006	-12000	-1	
11	Konto 007	4000	1	
12	Konto 008	7000	1	
13	Konto 009	-1500	-1	

Abbildung 5.85 Summenbildung nach Prüfung des Vorzeichens

Die Funktion kann auch in Kombination mit der Summenfunktion verwendet werden, um etwa die negativen Werte getrennt zu addieren, wie Abbildung 5.85 zeigt. Dabei wird in C5 eine Matrixformel verwendet, die nur dann zu einer Summenbildung führt, wenn in dem Bereich B5:B13 negative Werte vorkommen.

VRUNDEN()

MROUND()

Syntax: `VRUNDEN(Zahl; Vielfaches)`

Beispiel: `=VRUNDEN(12;5)`

Ergebnis: `10`

Die Funktion `VRUNDEN()` rundet den mit `Zahl` angegebenen Wert auf das nächste erreichbare Vielfache einer mit `Vielfaches` angegebenen Zahl. Wenn die Division von `Zahl` durch `Vielfaches` einen Rest liefert, der unter der Hälfte von `Vielfaches` liegt, rundet die Funktion in Richtung 0 ab, ansonsten wird von der 0 weg aufgerundet.

Beide Argumente müssen das gleiche Vorzeichen haben, sonst liefert die Funktion den Fehler `#ZAHL!`. Sind die Argumente nicht numerisch, erscheint der Fehler `#WERT!`.

Ähnlich wie die Funktionen `OBERGRENZE.MATHEMATIK()` und `UNTERGRENZE.MATHEMATIK()` kann auch diese Funktion verwendet werden, um Werte an bestimmten Intervallgrenzen auszurichten, etwa um Preise zu generieren, die sich immer durch 0,05 teilen lassen, also ohne 1- oder 2-Cent-Stücke bezahlt werden können.

	A	B	C	D
1				
2	**Runden auf Vielfaches einer Zahl**			
3				
4	Zahl	Vielfaches	VRUNDEN()	
5	12,4	0,5	12,5	
6	13	5	15	
7	25	3	24	
8	-107	-10	-110	
9	-123,56	5	#ZAHL!	
10				

Abbildung 5.86 Auf- oder Abrunden auf das nächstliegende Vielfache

WURZEL()

SQRT()

Syntax: `WURZEL(`**`Zahl`**`)`

Beispiel: `=WURZEL(36)`

Ergebnis: `6`

Die Funktion `WURZEL()` ermittelt die Quadratwurzel einer Zahl. Das Wurzelziehen ist die Umkehrfunktion des Potenzierens, der allgemein übliche Operator √ stammt von dem Buchstaben *r* ab und steht für »Radizieren«.

In der Gleichung

$x = \sqrt[n]{a}$

wird `x` als Wurzel oder Radix bezeichnet, `n` ist der Wurzelexponent und `a` der Radikand. Bei der Quadratwurzel ist der Wurzelexponent 2. Für Wurzeln mit einem höheren Wurzelexponenten kann die Tatsache genutzt werden, dass die Gleichung auch so geschrieben werden kann:

```
x = a^(1/n)
```

Die dritte Wurzel – Kubikwurzel – der Zahl 27 kann in Excel also mit der Formel

```
=27^(1/3)
```

errechnet werden.

Zahl darf keine negative Zahl sein, sonst liefert die Funktion den Fehler #ZAHL!. Soll verhindert werden, dass Zahl einen negativen Wert annimmt, kann mit der ABS()-Funktion gearbeitet werden:

```
=WURZEL(ABS(A10))
```

sorgt dafür, dass eine negative Zahl auf jeden Fall vermieden wird.

	A	B
1		
2	**Wurzel einer Zahl**	
3		
4	Zahl	WURZEL()
5	12	3,464101615
6	25	5
7	107	10,34408043
8	-123,56	#ZAHL!

Abbildung 5.87 Wurzelberechnung

WURZELPI()

SQRTPI()

Syntax: WURZELPI(**Zahl**)
Beispiel: =WURZELPI(2)
Ergebnis: 2,5066...

Eine spezielle Variante der Wurzelfunktion ist WURZELPI(). Sie liefert die Quadratwurzel aus (Zahl * π).

Das Argument Zahl darf nicht < 0 sein, sonst liefert die Funktion den Fehler #ZAHL!. Nicht numerische Werte ergeben den Fehler #WERT!.

	A	B
1		
2	**Quadratwurzel aus Zahl * Pi**	
3		
4	Zahl	WURZELPI()
5	12	6,139960248
6	25	8,862269255
7	107	18,3344052

Abbildung 5.88 Wurzel einer mit π multiplizierten Zahl

ZUFALLSBEREICH()

RANDBETWEEN()

Syntax: `ZUFALLSBEREICH(Untere_Zahl; Obere_Zahl)`
Beispiel: `=ZUFALLSBEREICH(100,200)`
Ergebnis: `133`

Mit der Funktion `ZUFALLSBEREICH()` erzeugen Sie eine ganzzahlige Zufallszahl zwischen `Untere_Zahl` und `Obere_Zahl` (beide eingeschlossen). Zufallszahlen können beispielsweise hilfreich sein, um Testdaten zu generieren, die die schnelle Überprüfung eines Kalkulationsmodells erlauben.

Als Werte für die beiden Argumente können nicht nur Zahlen, sondern auch Datumswerte angegeben werden.

```
=ZUFALLSBEREICH("1.1.2008";"31.12.2013")
```

liefert einen zufälligen Tag zwischen den beiden Terminen.

Nicht numerische Werte ergeben den Fehler `#WERT!`. Ist `Untere_Zahl` > `Obere_Zahl`, erscheint der Fehler `#ZAHL!`.

Während die verwandte Funktion `ZUFALLSZAHL()` nur Werte zwischen 0 und 1 erzeugt, können Sie bei dieser Funktion beliebige Grenzwerte nach oben und unten angeben, zwischen denen sich die generierten Zahlen bewegen sollen.

Eine weitere Alternative für die Erzeugung von Zufallszahlen finden Sie unter den Analysetools, die in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, in Abschnitt 17.13, »Zufallszahlengenerierung«, beschrieben sind.

	A	B	C
1			
2	**Zufallszahl innerhalb eines Zahlenbereichs**		
3			
4	Min_Wert	Max_Wert	ZUFALLSBEREICH()
5	10	100	69
6	0	1	0
7	400	1000	560
8	0	360	39
9	01.01.2007	31.12.2008	12.12.2007
10	WAHR	FALSCH	#WERT!

Abbildung 5.89 Zufallszahlen innerhalb eines Wertebereichs

Wenn Sie mit Zufallszahlen arbeiten wollen, die sich im Nachhinein nicht mehr verändern, also nicht bei jeder Neuberechnung der Tabelle ebenfalls neu berechnet werden,

dann drücken Sie die Taste F9, wenn Sie die Funktion in die Bearbeitungszeile eingegeben haben (und zwar bevor Sie die ↵-Taste drücken). Zellbereiche mit der Funktion ZUFALLSZAHL() können Sie durch **Kopieren** und **Einfügen** mit der Option **Werte** in Zahlenwerte umwandeln.

ZUFALLSZAHL()

RAND()

Syntax: ZUFALLSZAHL()
Beispiel: =ZUFALLSZAHL()
Ergebnis: 0,75419672

Die Funktion ZUFALLSZAHL() produziert Zufallszahlen zwischen 0 und 1 mit bis zu 16 Dezimalstellen. Zufallszahlen können beispielsweise hilfreich sein, um Testdaten zu generieren, die die schnelle Überprüfung eines Kalkulationsmodells erlauben.

Zufallszahlen anderer Größenordnungen können Sie durch entsprechende Rechenoperationen mit der generierten Zufallszahl oder mit der Funktion ZUFALLSBEREICH() erreichen:

=ZUFALLSZAHL * 100 ergibt eine Zufallszahl zwischen 0 und 100,

=ZUFALLSZAHL * 100 + 50 ergibt eine Zufallszahl zwischen 50 und 150.

	A	B	C
1			
2	**Zufallszahlen erzeugen**		
3			
4	ZUFALLSZAHL()	ZUFALLSZAHL() * 100	ZUFALLSZAHL() * 100 + 50
5	0,59	59,02	109,02
6	0,93	93,39	143,39
7	0,69	68,88	118,88
8	0,77	76,95	126,95
9	0,11	10,96	60,96
10	0,98	98,03	148,03
11	0,34	34,46	84,46
12	0,52	51,82	101,82
13	1,00	99,68	149,68
14	0,18	18,21	68,21
15	0,15	14,89	64,89
16	0,96	95,78	145,78
17	0,93	93,47	143,47
18	0,38	37,82	87,82
19	0,88	88,48	138,48

Abbildung 5.90 Zufallszahlen

Wenn Sie mit Zufallszahlen arbeiten wollen, die sich im Nachhinein nicht mehr verändern, also nicht bei jeder Neuberechnung der Tabelle ebenfalls neu berechnet werden, dann drücken Sie die Taste F9, wenn Sie die Funktion in die Bearbeitungszeile eingegeben haben (und zwar bevor Sie die ↵-Taste drücken). Zellbereiche mit der Funktion ZUFALLSZAHL() können Sie durch **Kopieren** und **Einfügen** mit der Option **Werte** in Zahlenwerte umwandeln.

ZWEIFAKULTÄT()

FACTDOUBLE()

Syntax: ZWEIFAKULTÄT(**Zahl**)
Beispiel: =ZWEIFAKULTÄT(5)
Ergebnis: 15

Die Funktion ZWEIFAKULTÄT() liefert die Doppelfakultät einer positiven Zahl, bei der mit einer Schrittlänge von 2 gearbeitet wird. Sie wird berechnet als:

```
1 * 3 * 5 * ... * Zahl (bei ungerader Zahl)
2 * 4 * 6 * ... * Zahl (bei gerader Zahl)
```

Ist Zahl keine Ganzzahl, werden die Dezimalstellen abgeschnitten. Nicht numerische Werte ergeben den Fehler #WERT!. Negative Werte liefern den Fehler #ZAHL!.

	A	B	C
1			
2	**Doppelfakultät einer Zahl**		
3			
4	Zahl	ZWEIFAKULTÄT()	
5	1	1	
6	2	2	
7	3	3	
8	4	8	**= 2 * 4**
9	5	15	**= 1 * 3 * 5**
10	6	48	**= 2 * 4 * 6**
11	7	105	**= 1 * 3 * 5 * 7**
12	8	384	
13	9	945	
14	10	3840	

Abbildung 5.91 Beispiele für ZWEIFAKULTÄT()

6 Technische Funktionen

Funktion	Seite
BESSELI()	401
BESSELJ()	402
BESSELK()	403
BESSELY()	404
BININDEZ()	404
BININHEX()	405
BININOKT()	406
BITLVERSCHIEB()	407
BITODER()	408
BITRVERSCHIEB()	408
BITUND()	409
BITXODER()	410
DELTA()	410
DEZINBIN()	411
DEZINHEX()	412
DEZINOKT()	413
GAUSSF.GENAU()	414
GAUSSFEHLER()	414
GAUSSFKOMPL()	415
GAUSSFKOMPL.GENAU()	416
GGANZZAHL()	416
HEXINBIN()	417
HEXINDEZ()	419
HEXINOKT()	419
IMABS()	420
IMAGINÄRTEIL()	421
IMAPOTENZ()	421
IMARGUMENT()	422
IMCOS()	422
IMCOSEC()	423
IMCOSECHYP()	423
IMCOSHYP()	424
IMCOT()	424
IMDIV()	425
IMEXP()	425
IMKONJUGIERTE()	426
IMLN()	427
IMLOG10()	427
IMLOG2()	428
IMPRODUKT()	428
IMREALTEIL()	429
IMSEC()	429
IMSECHYP()	430
IMSIN()	430
IMSINHYP()	431
IMSUB()	431
IMSUMME()	431
IMTAN()	432
IMWURZEL()	432
KOMPLEXE()	433
OKTINBIN()	434
OKTINDEZ()	435
OKTINHEX()	435
UMWANDELN()	436

6.1 Einsatzbereiche der technischen Funktionen

In dieser Gruppe finden Sie verschiedene Varianten der Besselfunktion, die insbesondere im technischen Bereich, etwa für Schwingungsberechnungen, benutzt werden, daher wohl auch die allgemeine Kennzeichnung dieser Gruppe als technische Funktionen. Hier geht es um Lösungen im Rahmen der Differenzialgleichungen und der Wahrscheinlichkeitsrechnung.

Dazu kommt eine ganze Reihe von Umwandlungsfunktionen, um Werte zwischen den verschiedenen Zahlensystemen zu übersetzen. Besonders praktisch ist auch eine allgemeine Funktion zur Umrechnung zwischen zahlreichen Maßeinheiten.

Eine große Gruppe von Funktionen hat mit den komplexen Zahlen zu tun. Während die übrigen Excel-Funktionen mit reellen Zahlen hantieren, erlauben diese Funktionen, die mit dem Präfix `IM` für »imaginär« gekennzeichnet sind, auch Rechenoperationen mit nicht reellen Werten.

Neu in Excel 2013 sind nachfolgende Funktionen, die mit Bitoperationen zu tun haben: `BITLVERSCHIEB()`, `BITODER()`, `BITRVERSCHIEB()`, `BITUND()`, `BITXODER()`.

Außerdem gibt es eine Reihe zusätzlicher Funktionen für den Umgang mit imaginären Zahlen: `IMCOSEC()`, `IMCOSECHYP()`, `IMCOSHYP()`, `IMCOT()`, `IMSEC()`, `IMSECHYP()`, `IMSINHYP()`, `IMTAN()`.

6.2 Besselfunktionen

Bei den besselschen Funktionen (auch unter dem Namen »Zylinderfunktionen« bekannt) handelt es sich um mehrere miteinander verwandte Funktionen, die in Physik und Technik besonders bei Schwingungsberechnungen benutzt werden. Die Funktionen sind Lösungen der Besselschen Differenzialgleichung:

```
x^2 (d^2y/dx^2)+x(dy)/(dx)+(x^2-n^2)y=0
```

Der Name geht auf den Astronomen und Mathematiker Bessel zurück, einem Zeitgenossen von Gauß.

Die Besselschen Funktionen finden Sie in mathematischen Tabellenwerken unter:

```
JO(x), J1(x).. entspricht BESSELJ()
IO(x), I1(x).. entspricht BESSELI()
KO(x), K1(x).. entspricht BESSELK()
YO(x), Y1(x).. entspricht BESSELY()
```

Dabei werden aber nur Werte für die nullte und erste Ordnung angegeben. Die Funktionen, die Excel bereitstellt, gestatten auch eine Berechnung für andere Ordnungen (mit `n` angegeben). Hinweise auf die Auswirkung von `n` bei den verschiedenen Funktionen bietet die Abbildung 6.1.

	A	B	C	D	E	F	G	H
1								
2	**Besselfunktionen**							
3								
4	x	I0	I1	J0	K0	K1	Y0	Y1
5	0,1	1,002501561	0,0500625	0,997502	2,427069	9,8538448	-1,534239	-6,458951
6	0,2	1,010025022	0,1005008	0,990025	1,7527038	4,7759725	-1,081105	-3,323825
7	0,3	1,022626867	0,1516938	0,977626	1,37246	3,055992	-0,807274	-2,293105
8	0,4	1,040401763	0,2040268	0,960398	1,1145291	2,1843544	-0,606025	-1,780872
9	0,5	1,063483344	0,2578943	0,93847	0,924419	1,6564411	-0,444519	-1,471472
10	0,6	1,092045331	0,313704	0,912005	0,7775221	1,3028349	-0,30851	-1,260391
11	0,7	1,126302982	0,3718797	0,881201	0,6605198	1,0502835	-0,190665	-1,10325
12	0,8	1,166514885	0,4328648	0,846287	0,5653471	0,8617816	-0,086802	-0,978144
13	0,9	1,21298513	0,4971264	0,807524	0,4867303	0,7165336	0,0056283	-0,873127
14	1	1,266065848	0,5651591	0,765198	0,4210244	0,6019072	0,088257	-0,781213
15	1,1	1,326160162	0,6374889	0,719622	0,3656024	0,50976	0,1621632	-0,69812
16	1,2	1,393725572	0,7146779	0,671133	0,3185082	0,4345924	0,2280835	-0,621136
17	1,3	1,469277796	0,7973293	0,620086	0,2782476	0,3725475	0,2865354	-0,54852

Abbildung 6.1 Die in Excel verfügbaren Besselfunktionen

In Abbildung 6.2 ist für die Funktion `BESSELJ()` für mehrere Ordnungen der grafische Verlauf dargestellt, der sehr schön zeigt, dass die Funktion zur Berechnung gedämpfter Schwingungen geeignet ist.

	A	B	C	D	E	F
1						
2	**BESSELJ() mit verschiedenen Ordnungen**					
3						
4	x	J0	J1	J2	J3	J4
5	0	1,000000003	0	0	0	0
6	1	0,765197684	0,4400506	0,114903	0,0195634	0,0024766
7	2	0,223890782	0,5767248	0,352834	0,1289432	0,0339957
8	3	-0,260051958	0,339059	0,486091	0,3090629	0,1320342
9	4	-0,397149807	-0,0660433	0,364128	0,4301715	0,2811291
10	5	-0,177596774	-0,3275791	0,046565	0,3648312	0,3912324
11	6	0,150645259	-0,2766839	-0,242873	0,1147684	0,3576416
12	7	0,300079274	-0,0046828	-0,301417	-0,167556	0,1577981
13	8	0,171650807	0,2346284	-0,112994	-0,291125	-0,10535
14	9	-0,090333611	0,2453074	0,144846	-0,180931	-0,265467
15	10	-0,245935764	0,0434722	0,25463	0,0583798	-0,219602
16	11	-0,171190301	-0,1767844	0,139048	0,2273472	-0,01504
17	12	0,047689311	-0,2234464	-0,08493	0,1951363	0,1824985

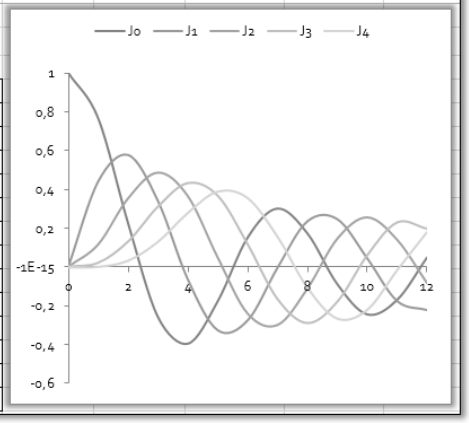


Abbildung 6.2 Verlauf der Funktion `BESSELJ()`

6.3 Umwandlungen zwischen Zahlensystemen

Zwölf Funktionen in dieser Kategorie dienen der Umwandlung von Zahlen aus einem Zahlensystem in ein anderes System. Neben dem Dezimalsystem werden das Dualsystem, das Oktalsystem und das Hexadezimalsystem berücksichtigt. Für jedes System sind jeweils drei Funktionen vorhanden, die für die Umwandlung in die anderen Systeme sorgen.

Die Zahlensysteme unterscheiden sich durch die Zahl der verwendeten Ziffern, die als Basis bezeichnet wird. Das Dezimalsystem verfügt bekanntlich über 10 Ziffernzeichen (0 bis 9). Kommen Sie beim Zählen einer Zahl nach der 9 an, dann wird die erste Ziffer genommen und eine 0 angehängt.

Das Dualsystem, das auch als Binärsystem bezeichnet wird, verwendet nur die Ziffern 0 und 1. Sie zählen also 0, 1, 10, 11, 100 etc.

Im Oktalsystem werden 8 Ziffern verwendet, Sie zählen 0, 1, 2, ..., 7, 10, 11, ..., 17, 20.

Das Hexadezimalsystem verwendet zusätzlich zu den bekannten Ziffern von 0 bis 9 noch die Buchstaben von A bis F, um auf 16 Zeichen zu kommen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1																		
2	**Zahlensysteme**																	
3																		
4	**Dezimalsystem**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	**Dualsystem**	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
6	**Oktalsystem**	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
7	**Hexadezimalsystem**	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

Abbildung 6.3 Übersicht über die unterstützten Zahlensysteme

Bei der Darstellung binärer Werte kann es schnell zu Zahlen mit vielen Stellen kommen, da ja nur zwei Ziffern zur Verfügung stehen. Das Oktal- und das Hexadezimalsystem werden deshalb auch benutzt, um binäre Werte kürzer darstellen zu können. Das ist leicht möglich, weil es sich bei einer Basis von 8 bzw. 16 Zeichen jeweils um Potenzen von 2 handelt.

Der Zusammenhang zwischen dem Wertebereich – `m` – und der notwendigen Anzahl von Ziffern – `n` – für die Darstellung jeder Zahl in diesem Bereich und der Basis des verwendeten Zahlensystems – `b` – ist allgemein mit der Formel

```
m = bn
```

bestimmbar. Also sind mit 8 Ziffern im Dualsystem maximal 256 Zahlen darstellbar. Wird ein Vorzeichen für eine negative Zahl benötigt, wird die am weitesten links stehende Ziffer dafür verwendet, `1` für negativ, `0` für positiv. Der Wertebereich ist dann `-128` bis `+128`.

Beachtet werden muss, dass Excel die Ergebnisse der Umwandlungsfunktionen als Zeichenfolge, also nicht als Zahl, ausgibt, es sei denn, das Zielsystem ist das Dezimalsystem. Wird beispielsweise eine Zelle mit einem binären Wert zu einem anderen Wert addiert, interpretiert Excel die Zeichenfolge als Dezimalzahl, was zu falschen Ergebnissen führt.

6.4 Umwandeln von Maßeinheiten

Ganz praktisch ist die Funktion `UMWANDELN()`, mit der zwischen zahlreichen Maßeinheiten umgerechnet werden kann. Sie können sich damit leicht eine Tabelle aufbauen, die die gewünschten Umwandlungen für Maßeinheiten liefert. In Excel 2013 ist die Liste der verfügbaren Maßeinheiten noch einmal beträchtlich erweitert worden. Sie finden die entsprechenden Tabellen in der Beschreibung der Funktion.

E4 =UMWANDELN(B4;C5;F5)

	A	B	C	D	E	F
1						
2	**Umwandeln von Maßeinheiten**					
3						
4		1000	**Gramm**	entsprechen:	2,204622622	**Pfund (Handelsgewicht)**
5			**g**			**lbm**
6		1000	**Meter**	entsprechen:	0,621371192	**Britische Meilen**
7			**m**			**mi**
8					39370,07874	**Zoll**
9						**in**
10					3280,839895	**Fuß**
11						**ft**
12					1093,613298	**Yard**
13						**yd**
14		32	**Grad Celsius**	entsprechen:	89,6	**Grad Fahrenheit**
15			**C**			**F**
16		1000	**Joule**	entsprechen:	238,8458966	**Kalorie**
17			**J**			**cal**
18		10	**Pferdestärke (PS)**	entsprechen:	7456,998716	**Watt**
19			**HP**			**W**

Abbildung 6.4 Beispiele für Umrechnungen von Maßeinheiten

Die Bezeichnungen der Maßeinheiten müssen der Funktion jeweils als Zeichenfolge übergeben werden, wobei auf die Groß-/Kleinschreibung zu achten ist. Abbildung 6.4 zeigt einige häufig benötigte Umrechnungen. Sie brauchen in den Zellen in Spalte B nur die Werte einzugeben, die Sie umwandeln wollen. Die Formeln beziehen sich dabei jeweils auf die Bezeichnungen der Maßeinheiten.

6.5 Rechenoperationen mit komplexen Zahlen

Der Zahlenbereich ist ausgehend von der Menge der natürlichen Zahlen in mehreren Stufen erweitert worden. Aus der paarweisen Kombination von ganzen Zahlen in Brüchen ergab sich die Menge der rationalen Zahlen. Wo auch die rationalen Zahlen nicht ausreichten, wurden die irrationalen Zahlen eingeführt, die sich nicht als Bruch zweier ganzer Zahlen darstellen lassen. Neben algebraischen Zahlen wie √2 sind das transzendente Zahlen wie die Kreiszahl π oder die Eulersche Zahl e. Die Gesamtmenge der rationalen und irrationalen Zahlen ist die Menge der reellen Zahlen. Die meisten Funktionen in Excel, soweit sie mit Rechenoperationen zu tun haben, arbeiten im Bereich der reellen Zahlen.

An ihre Grenze stoßen Lösungen mit reellen Zahlen, wenn es darum geht, Wurzeln aus negativen Zahlen zu berechnen. Dieses Problem bildete den Anstoß zur Einführung der komplexen Zahlen, die aus einem reellen und einem imaginären Anteil zusammengesetzt sind. Dazu wurde als neue Zahl die imaginäre Einheit `i` eingeführt. Sie ist definiert als:

`i = √–1`

Wird in den Excel-Funktionen als Argument eine komplexe Zahl erwartet, muss sie als Zeichenfolge in der Form

`x+yi oder x+yj`

eingegeben werden, `j` kann bei Anwendungen im Bereich der Elektrotechnik verwendet werden, um Verwechslungen mit der Stromstärke zu vermeiden. Die Kleinschreibung von `i` oder `j` ist erforderlich.

Dabei können `x` und `y` zwei beliebige reelle Zahlen sein. `yi` oder `yj` ist die Kurzform für `y*i` oder `y*j`. Es wird also eine reelle Zahl mit der imaginären Zahl multipliziert.

Wird eine komplexe Zahl direkt als Funktionsargument angegeben, müssen Anführungszeichen verwendet werden: `=IMABS("3+3i")`; wird ein Zellbezug als Argument angegeben, steht die Zeichenfolge ohne Anführungszeichen in der entsprechenden Zelle.

Mit imaginären Zahlen lässt sich ganz normal rechnen, wenn dabei beachtet wird, dass:

`i^2 = –1; i^3 = –i; i^4 = 1; i^7 = –i` usw.:

```
(3 + 4i) + (6 + 2i) = 9 + 7i
(9 + 7i) – (3 + 4i) = 6 + 2i
(3 + 5i) * (6 + 2i) = 3 * 6 + 3 *2i + 5*6i + 10i^2
                    = 18 + 36i + 10i^2
```

```
= 18 + 36i -10
= 8 + 36i
```

```
10i/2i = 5
```

Komplexe Zahlen lassen sich geometrisch in der Gaußschen Zahlenebene als Punkte in einem rechtwinkligen Koordinatensystem darstellen, bei dem die waagerechte Achse die reellen Zahlen und die senkrechte Koordinate den imaginären Anteil repräsentiert.

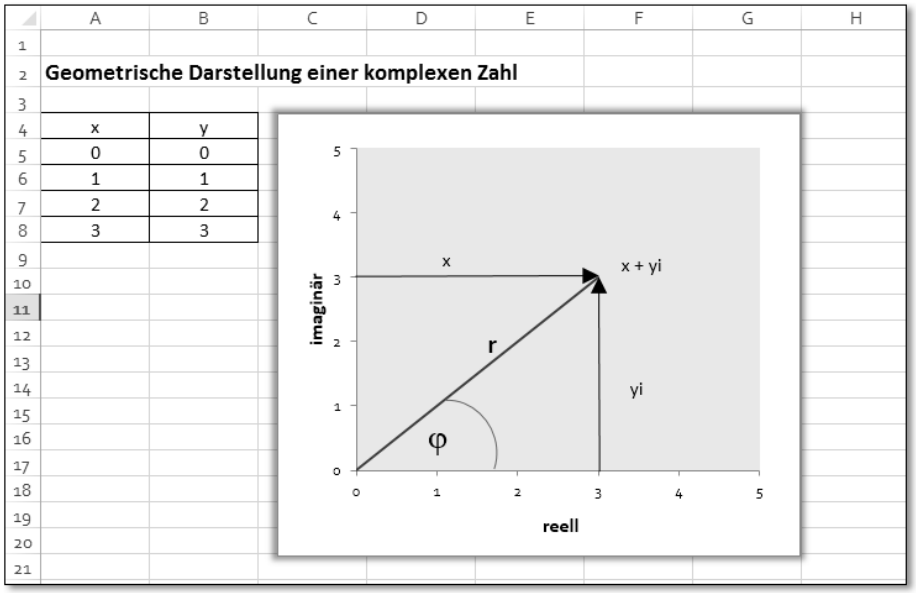


Abbildung 6.5 Die x-Achse repräsentiert den reellen, die y-Achse den imaginären Anteil der komplexen Zahl.

Neben der Darstellung im kartesischen Koordinatensystem ist auch eine mit Polarkoordinaten möglich, die Polarform oder goniometrische Form genannt wird. Statt:

```
z = x + yi
```

gilt dann:

```
z = r * e^(iφ) = r * (cosφ + i*sinφ)
```

Dabei wird `r` der Betrag von `z` genannt und entspricht dem Absolutwert von `z`, der mit der Funktion `=IMABS(z)` ermittelt werden kann. φ wird der Polarwinkel oder das Argument von `z` genannt. Es kann mit Hilfe der Funktion `=IMARGUMENT(z)` ermittelt werden und gibt den Winkel φ an. Der Absolutwert wird nach der Formel berechnet:

$$\texttt{IMABS(z)} = |z| = \sqrt{x^2 + y^2}$$

Die abgebildeten Koordinaten ergeben mit `=IMABS("3+3i")` einen Betrag von `4,2426...`, was, wenn die komplexe Zahl als Vektor vorgestellt wird, dessen Länge angibt. Der angezeigte Winkel ergibt mit `=GRAD(IMARGUMENT("3+3i"))` wie erwartet den Wert `45°`.

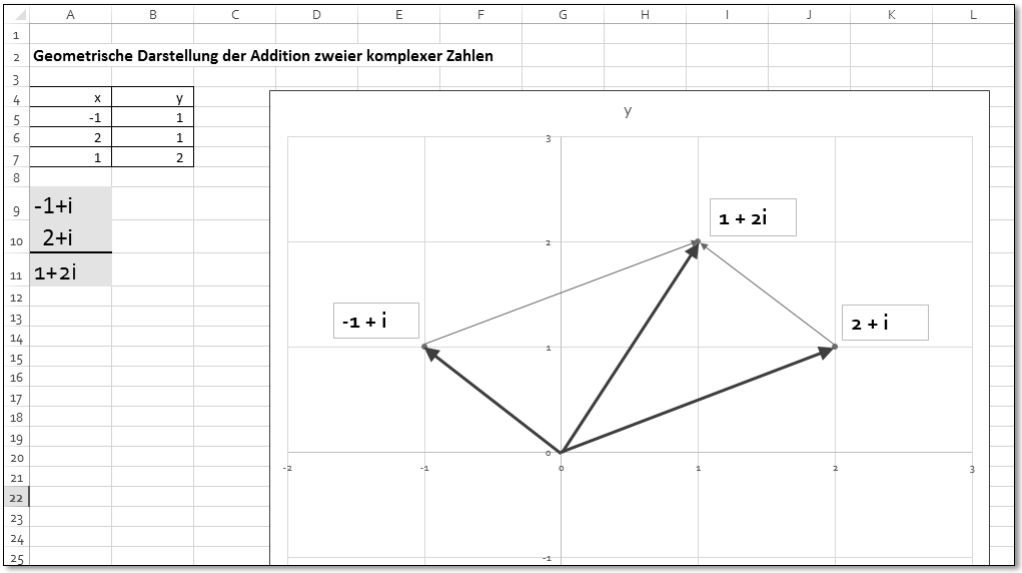


Abbildung 6.6 Geometrische Deutung der Addition zweier komplexer Zahlen

Die Addition zweier komplexer Zahlen lässt sich in diesem Zahlenraum mit Hilfe von Vektoren darstellen. Abbildung 6.7 zeigt ein einfaches Beispiel. Die Summe ist die Diagonale des durch die beiden Vektoren (0,0; –1+i) (0,0; 2+i) aufgespannten Parallelogramms.

In Abbildung 6.7 sind die Funktionen für komplexe Zahlen mit einfachen Beispielen zusammengestellt. Die meisten Funktionen liefern als Ergebnis selbst wieder komplexe Zahlen, die in Excel als Zeichenfolgen dargestellt werden.

Komplexe Zahlen spielen in der Physik und in der Technik eine große Rolle, insbesondere weil sie die Behandlung von Differenzialgleichungen zu Schwingungsvorgängen vereinfachen, unter anderem auch bei den modifizierten Besselfunktionen. In der Elektrotechnik tauchen komplexe Zahlen etwa bei der Berechnung des Blindwiderstands in der komplexen Wechselstromrechnung auf. In der reinen Mathematik werden komplexe Zahlen in der Funktionentheorie verwendet.

	A	B	C	D
1				
2	**Berechnungen mit imaginären Zahlen**			
3				
10	Komplexe_Zahl1	Komplexe_Zahl2		
11	25-10i	1+2i	26-8i	=IMSUMME(A11;B11)
12	25-10i	1+2i	24-12i	=IMSUB(A12;B12)
13	25-10i	1+2i	45+40i	=IMPRODUKT(A13;B13)
14	25-10i	1+2i	1-12i	=IMDIV(A14;B14)
15				
16	1+2i	2,236067977		=IMABS(A16)
17	1+2i	2		=IMAGINÄRTEIL(A17)
18	1+2i	-3+4i		=IMAPOTENZ(A18;2)
19	1+2i	1,107148718		=IMARGUMENT(A19)
20	1+2i	-1,13120438375681+2,47172667200482i		=IMEXP(A20)
21	1+2i	1-2i		=IMKONJUGIERTE(A21)
22	1+2i	0,80471895621705+1,07148717794091i		=IMLN(A22)
23	1+2i	0,349485002168009+0,480828578784234i		=IMLOG10(A23)
24	1+2i	1,16096404744368+1,59727796468811i		=IMLOG2(A24)
25	1+2i	1		=IMREALTEIL(A25)
26	1+2i	1,27201964951407+0,786151377757423i		=IMWURZEL(A26)
27				
28	1+2i	2,03272300701967-3,0518977991518i		=IMCOS(A28)
29	1+2i	0,228375065599687-0,141363021612408i		=IMCOSEC(A29)
30	1+2i	-0,221500930850509-0,6354937992539i		=IMCOSECHYP(A30)
31	1+2i	-0,64214812471552+1,06860742138278i		=IMCOSHYP(A31)
32	1+2i	0,0327977555337526-0,984329226458191i		=IMCOT(A32)
33	1+2i	0,151176298265577+0,226973675393722i		=IMSEC(A33)
34	1+2i	-0,41314934426694-0,687527438655479i		=IMSECHYP(A34)
35	1+2i	3,16577851321617+1,95960104142161i		=IMSIN(A35)
36	1+2i	-0,489056259041294+1,40311925062204i		=IMSINHYP(A36)
37	1+2i	0,0338128260798967+1,01479361614663i		=IMTAN(A37)

Abbildung 6.7 Übersicht über die `IM*()`-Funktionen

6.6 Referenz der technischen Funktionen

BESSELI()

BESSELI()

Syntax: `BESSELI(x; n)`
Beispiel: `=BESSELI(0,5;1)`
Ergebnis: `0,25789`

Die Funktion `BESSELI()` liefert Werte der modifizierten Besselfunktion `In(x)`. Sie entspricht der Besselfunktion `Jn(x)` für den Fall, dass nur imaginäre Argumente verwendet werden. Das Argument `x` ist der Wert, für den die Funktion ausgewertet werden soll.

x muss eine reelle Zahl sein. Mit n wird die Ordnung angegeben. n muss eine positive ganze Zahl sein, Nachkommastellen werden abgeschnitten. Die verwendete Ordnung hat Einfluss auf den Wertebereich.

Ist n < 0, liefert die Funktion den Fehler #ZAHL!, ist eines der Argumente nicht numerisch, erscheint der Fehler #WERT!.

BESSELI() mit verschiedenen Ordnungen

x	I0	I1	I2
0	1	0	0
0,1	1,002501561	0,0500625	0,001251
0,2	1,010025022	0,1005008	0,005017
0,3	1,022626867	0,1516938	0,011335
0,4	1,040401763	0,2040268	0,020268
0,5	1,063483344	0,2578943	0,031906
0,6	1,092045331	0,313704	0,046365
0,7	1,126302982	0,3718797	0,06379
0,8	1,166514885	0,4328648	0,084353
0,9	1,21298513	0,4971264	0,10826
1	1,266065848	0,5651591	0,135748
1,1	1,326160162	0,6374889	0,167089
1,2	1,393725572	0,7146779	0,202596
1,3	1,469277796	0,7973293	0,242617
1,4	1,553395106	0,886092	0,287549
1,5	1,646723202	0,9816664	0,337835

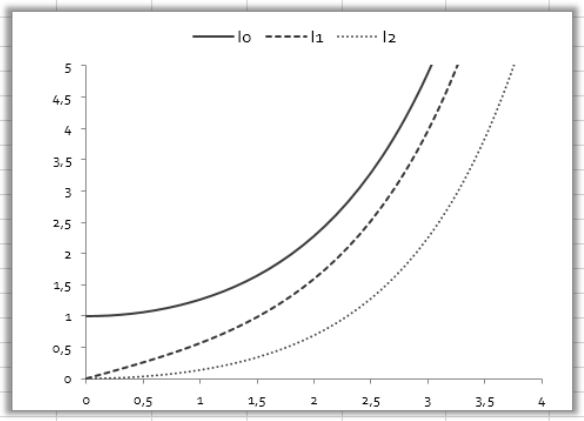


Abbildung 6.8 Werte der Funktion BESSELI() für verschiedene Ordnungen und ihr Graph

Siehe auch Abschnitt 6.2, »Besselfunktionen«.

BESSELJ()

BESSELJ()

Syntax: BESSELJ(**x**; **n**)
Beispiel: =BESSELJ(0,5;2)
Ergebnis: 0,0306

Die Funktion BESSELJ() liefert Werte der Besselfunktion Jn(x). Das Argument x ist der Wert, für den die Funktion ausgewertet werden soll. Mit n wird die Ordnung angegeben. n muss eine positive ganze Zahl sein, Nachkommastellen werden abgeschnitten. Die verwendete Ordnung hat Einfluss auf den Wertebereich. x muss eine reelle Zahl sein.

Ist n < 0, liefert die Funktion den Fehler #ZAHL!, ist eines der Argumente nicht numerisch, erscheint der Fehler #WERT!.

Siehe auch Abbildung 6.2 und den Abschnitt 6.2, »Besselfunktionen«.

BESSELK()

BESSELK()

Syntax: BESSELK(**x**; **n**)
Beispiel: =BESSELK(0,5;1)
Ergebnis: 1,65644

Die Funktion BESSELK() liefert Werte der modifizierten Besselfunktion Kn(x). Das Argument x ist der Wert, für den die Funktion ausgewertet werden soll. Mit n wird die Ordnung angegeben. n muss eine positive ganze Zahl sein, Nachkommastellen werden abgeschnitten. Die verwendete Ordnung hat Einfluss auf den Wertebereich. x muss eine reelle Zahl sein.

Ist n < 0, liefert die Funktion den Fehler #ZAHL!, ist eines der Argumente nicht numerisch, erscheint der Fehler #WERT!.

BESSELK() mit verschiedenen Ordnungen

x	K0	K1	K2
0			
0,5	0,924419035	1,6564411	7,550184
1	0,421024421	0,6019072	1,624839
1,5	0,213805569	0,2773878	0,583656
2	0,11389388	0,1398659	0,25376
2,5	0,062347554	0,0738908	0,12146
3	0,034739504	0,0401564	0,06151
3,5	0,019598897	0,0222394	0,032307
4	0,011159676	0,0124835	0,017401
4,5	0,006399857	0,0070781	0,009546
5	0,003691098	0,0040446	0,005309
5,5	0,002138709	0,0023256	0,002984
6	0,001243994	0,0013439	0,001692

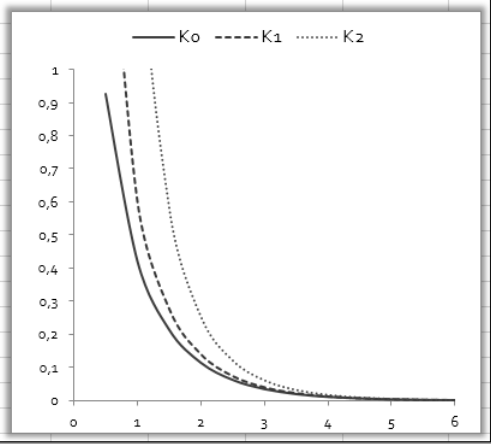


Abbildung 6.9 Werte der Funktion BESSELK() für verschiedene Ordnungen und ihr Graph

Siehe auch Abschnitt 6.2, »Besselfunktionen«.

BESSELY()

BESSELY()

Syntax: `BESSELY(x; n)`
Beispiel: `=BESSELY(0,5;1)`
Ergebnis: `1,47147`

Die Funktion `BESSELY()` gibt Werte der Besselfunktion `Yn(x)` zurück, die auch als Webersche oder Neumannsche Funktion bezeichnet wird. Das Argument `x` ist der Wert, für den die Funktion ausgewertet werden soll. Mit `n` wird die Ordnung angegeben. `n` muss eine positive ganze Zahl sein, Nachkommastellen werden abgeschnitten. Die verwendete Ordnung hat Einfluss auf den Wertebereich. `x` muss eine reelle Zahl sein.

Ist `n < 0`, liefert die Funktion den Fehler `#ZAHL!`, ist eines der Argumente nicht numerisch, erscheint der Fehler `#WERT!`.

BESSELY() mit verschiedenen Ordnungen

x	J0	J1	J2
0			
1	0,088256971	-0,7812128	-1,650683
2	0,510375667	-0,1070324	-0,617408
3	0,376850015	0,3246744	-0,1604
4	-0,016940745	0,3979257	0,215904
5	-0,308517623	0,1478631	0,367663
6	-0,288194691	-0,1750103	0,229858
7	-0,025949756	-0,3026672	-0,060527
8	0,223521489	-0,1580605	-0,263037
9	0,249936698	0,1043146	-0,226756
10	0,055671167	0,2490154	-0,005868
11	-0,168847324	0,1637055	0,198612
12	-0,225237313	-0,0570992	0,215721

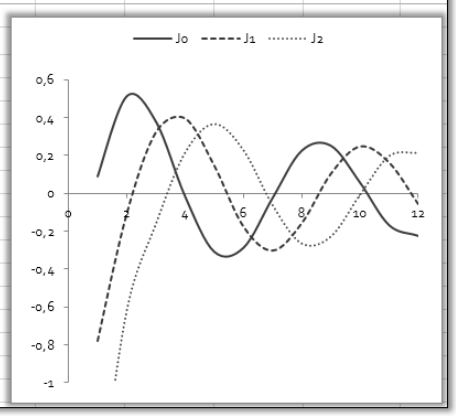


Abbildung 6.10 Werte der Funktion `BESSELY()` für verschiedene Ordnungen und ihr Graph

Siehe auch Abschnitt 6.2, »Besselfunktionen«.

BININDEZ()

BIN2DEC()

Syntax: `BININDEZ(Zahl)`
Beispiel: `=BININDEZ(11101)`
Ergebnis: `29`

Die Funktion BININDEZ() wandelt den mit Zahl angegebenen binären Wert in eine Dezimalzahl um. Der binäre Wert darf höchstens zehn Zeichen (0 oder 1) lang sein; bei zehn Zeichen ist das erste Zeichen das Vorzeichenbit (1 = negativ). Diese Darstellungsform, bei der für das Vorzeichen kein spezielles Zeichen verwendet wird, wird auch als Zweierkomplement bezeichnet.

Wenn Zahl keine zulässige binäre Zahl ist oder mehr als 10 Zeichen enthält, gibt die Funktion den Fehler #ZAHL! aus. Ist Zahl nicht numerisch, erscheint der Fehler #WERT!.

	A	B	C	D	E	F	G
1							
2	**Umwandlungen von Binärzahlen**						
3							
4	Zahl	Stellen	BININDEZ()	BININHEX(Zahl)	BININHEX(Zahl;Stellen)	BININOKT(Zahl)	BININOKT(Zahl;Stellen)
5	0	4	0	0	0000	0	0000
6	1	4	1	1	0001	1	0001
7	10	4	2	2	0002	2	0002
8	11	4	3	3	0003	3	0003
9	100	4	4	4	0004	4	0004
10	101	4	5	5	0005	5	0005
11	110	4	6	6	0006	6	0006
12	111	4	7	7	0007	7	0007
13	1000	4	8	8	0008	10	0010
14	1001	4	9	9	0009	11	0011
15	1010	4	10	A	000A	12	0012

Abbildung 6.11 Funktionen für die Umwandlung von binären Zahlen

Anders als bei den anderen Umwandlungsfunktionen zwischen den Zahlensystemen liefert diese Funktion ihr Ergebnis als dezimale Zahl ab, mit der also ohne Umstände weitergerechnet werden kann.

BININHEX()

BIN2HEX()

Syntax: BININHEX(**Zahl**; Stellen)
Beispiel: =BININHEX(1100)
Ergebnis: C

Die Funktion BININHEX() wandelt den mit Zahl angegebenen binären Wert in eine Hexadezimalzahl um. Der binäre Wert darf höchstens zehn Zeichen (0 oder 1) lang sein; bei zehn Zeichen ist das erste Zeichen das Vorzeichenbit (1 = negativ). Das Argument Stellen gibt an, wie viele Stellen angezeigt werden. Ohne Angabe werden nur die not-

wendigen Stellen angezeigt; ist Stellen größer als deren Anzahl, werden führende Nullen ausgegeben.

Wenn Zahl keine zulässige binäre Zahl ist oder mehr als 10 Zeichen enthält, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!.

Ist Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist Zahl eine negative binäre Zahl, ignoriert die Funktion das Argument Stellen und gibt eine Zahl mit 10 Zeichen aus.

Im Unterschied zu der Funktion BININDEZ() liefert BININHEX() ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Textfunktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;16) wandelt die hexadezimale Zeichenfolge 000F in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine hexadezimale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0006 + 10 = 16. Kommen in der hexadezimalen Zeichenfolge Buchstaben vor, liefert die Berechnung den Fehler #WERT!.

Siehe auch Abbildung 6.12 zu der Funktion BININDEZ().

BININOKT()

BIN2OCT()

Syntax: BININOKT(**Zahl**; Stellen)
Beispiel: =BININOKT(1100)
Ergebnis: 14

Die Funktion BININOKT() wandelt den mit Zahl angegebenen binären Wert in eine Oktalzahl um. Der binäre Wert darf höchstens zehn Zeichen (0 oder 1) lang sein; bei zehn Zeichen ist das erste Zeichen das Vorzeichenbit (1 = negativ). Das Argument Stellen gibt an, wie viele Stellen angezeigt werden. Ohne Angabe werden nur die notwendigen Stellen angezeigt; ist Stellen größer als deren Anzahl, werden führende Nullen ausgegeben.

Wenn Zahl keine zulässige binäre Zahl ist oder mehr als 10 Zeichen enthält, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!. Ist Zahl

oder Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist Zahl eine negative binäre Zahl, ignoriert die Funktion das Argument Stellen und gibt eine Zahl mit 10 Zeichen aus.

Im Unterschied zu der Funktion BININDEZ() liefert BININOKT() ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;8) wandelt die oktale Zeichenfolge 0017 in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine oktale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0073 + 10 = 83.

Siehe auch Abbildung 6.12 zu der Funktion BININDEZ().

BITLVERSCHIEB()

BITLSHIFT()

Syntax: BITLVERSCHIEB(**Zahl**; **Verschiebebetrag**)
Beispiel: =BITLVERSCHIEB(4;2)
Ergebnis: 16

Die in Excel 2013 neu eingeführte Funktion BITLVERSCHIEB() führt in der binären Repräsentation einer dezimalen Zahl eine Linksverschiebung um die Anzahl von Stellen durch, die mit Verschiebebetrag angegeben wird, und liefert den sich daraus ergebenden Betrag als dezimale Zahl. Zahl muss eine ganze Zahl sein, die ≥ 0 und < (2^48)-1 ist.

Die Linksverschiebung entspricht dem Hinzufügen von Nullen auf der rechten Seite der binären Darstellung der Zahl. Der Verschiebebetrag muss eine ganze Zahl und < 53 sein.

Liegt ein Wert außerhalb des gültigen Wertebereichs, liefert die Funktion den Wert #ZAHL!. Ist eines der Argumente nicht numerisch, wird der Fehlerwert #WERT! zurückgegeben.

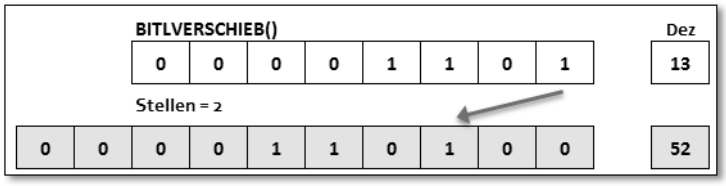


Abbildung 6.12 Linksverschiebung eines Binärwerts um 2 Stellen

BITODER()

BITOR()

Syntax: `BITODER(`**`Zahl1`**`; `**`Zahl2`**`)`

Beispiel: `=BITODER(3;5)`

Ergebnis: `7`

Die in Excel 2013 neu eingeführte Funktion `BITODER()` gibt ein bitweises ODER zweier Zahlen zurück. `Zahl1` und `Zahl2` müssen als dezimale Zahlen angegeben werden und jeweils ≥ `0` und < `(2^48)-1` sein. Die Funktion vergleicht die Bits der beiden Zahlen positionsweise. Ist eines dieser Bits 1, ist das Ergebnis für diese Bitposition 1. Die für die Bitpositionen zurückgegebenen Werte wachsen von rechts nach links als Potenzen von 2. Das äußerste rechte Bit entspricht 1 (2^0), das Bit daneben 2 (2^1) usw.

Liegt ein Wert außerhalb des gültigen Wertebereichs, liefert die Funktion den Wert `#ZAHL!`. Ist eines der Argumente nicht numerisch, wird der Fehlerwert `#WERT!` zurückgegeben.

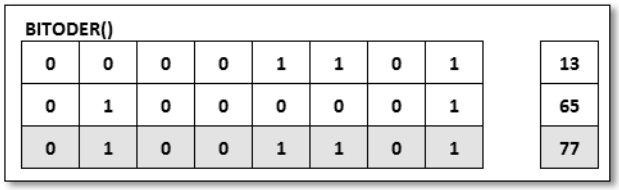


Abbildung 6.13 Oder zweier Binärwerte

BITRVERSCHIEB()

BITRSHIFT()

Syntax: `BITRVERSCHIEB(`**`Zahl`**`; `**`Verschiebebetrag`**`)`

Beispiel: `=BITRVERSCHIEB(16;2)`

Ergebnis: `4`

Die in Excel 2013 neu eingeführte Funktion `BITRVERSCHIEB()` führt in der binären Repräsentation einer dezimalen Zahl eine Rechtsverschiebung um die AnzahlB von Stellen durch, die mit `Verschiebebetrag` angegeben wird, und liefert den sich daraus ergebenden Betrag als dezimale Zahl. `Zahl` muss eine ganze Zahl sein, die ≥ `0` und < `(2^48)-1` ist. Die Rechtsverschiebung entspricht dem Entfernen von Stellen auf der rechten Seite der binären Darstellung der Zahl. Der `Verschiebebetrag` muss eine ganze Zahl und < `53` sein.

Liegt ein Wert außerhalb des gültigen Wertebereichs, liefert die Funktion den Wert #ZAHL!. Ist eines der Argumente nicht numerisch, wird der Fehlerwert #WERT! zurückgegeben.

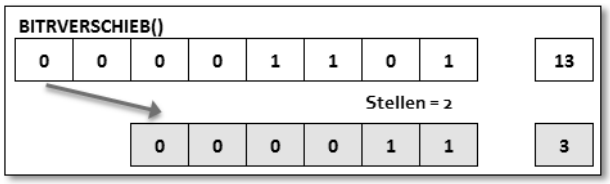


Abbildung 6.14 Rechtsverschiebung eines Binärwerts um 2 Stellen

BITUND()

BITAND()

Syntax: BITUND(**Zahl1**; **Zahl2**)
Beispiel: =BITUND(3;5)
Ergebnis: 1

Die in Excel 2013 neu eingeführte Funktion BITUND() gibt ein bitweises UND zweier Zahlen zurück. Zahl1 und Zahl2 müssen als dezimale Zahlen angegeben werden und jeweils ≥ 0 und < (2^48)-1 sein. Die Funktion vergleicht die Bits der beiden Zahlen positionsweise. Nur wenn beide Bits 1 sind, ist das Ergebnis für diese Bitposition 1. In allen anderen Fällen ergibt der Vergleich eine 0.

Die für die Bitpositionen zurückgegebenen Werte wachsen von rechts nach links als Potenzen von 2. Das äußerste rechte Bit entspricht 1 (2^0), das Bit daneben 2 (2^1) usw.

Liegt ein Wert außerhalb des gültigen Wertebereichs, liefert die Funktion den Wert #ZAHL!. Ist eines der Argumente nicht numerisch, wird der Fehlerwert #WERT! zurückgegeben.

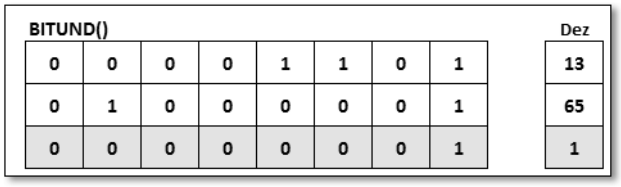


Abbildung 6.15 Und zweier Binärwerte

BITXODER()

BITXOR()

Syntax: `BITXODER(`**`Zahl1`**`; `**`Zahl2`**`)`
Beispiel: `=BITXODER(3;5)`
Ergebnis: `6`

Die in Excel 2013 neu eingeführte Funktion `BITXODER()` gibt ein bitweises XODER zweier Zahlen zurück. `Zahl1` und `Zahl2` müssen als dezimale Zahlen angegeben werden und jeweils ≥ `0` und < `(2^48)-1` sein. Die Funktion vergleicht die Bits der beiden Zahlen positionsweise. Ist nur eines dieser Bits 1, ist das Ergebnis für diese Bitposition 1. In allen anderen Fällen ergibt der Vergleich eine 0.

Die für die Bitpositionen zurückgegebenen Werte wachsen von rechts nach links als Potenzen von 2. Das äußerste rechte Bit entspricht 1 (2^0), das Bit daneben 2 (2^1) usw.

Liegt ein Wert außerhalb des gültigen Wertebereichs, liefert die Funktion den Wert `#ZAHL!`. Ist eines der Argumente nicht numerisch, wird der Fehlerwert `#WERT!` zurückgegeben.

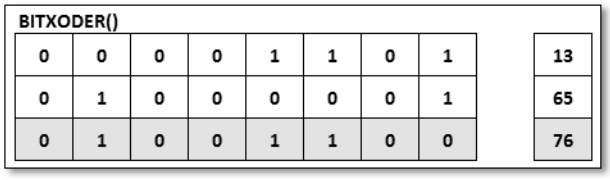


Abbildung 6.16 Ausschließendes ODER zweier Binärwerte

DELTA()

DELTA()

Syntax: `DELTA(`**`Zahl1`**`; Zahl2)`
Beispiel: `=DELTA(10/6;20/12)`
Ergebnis: `1`

Die Funktion `DELTA()` prüft, ob die mit den Argumenten `Zahl1` und `Zahl2` angegebenen Werte gleich sind. Sie ergibt 1, wenn `Zahl1` und `Zahl2` in ihrem numerischen Wert übereinstimmen, andernfalls ist das Ergebnis 0. Wird `Zahl2` nicht angegeben, wird das Argument als `0` angenommen. `=DELTA(B10)` prüft also, ob die Zelle den Wert `0` liefert.

Als Argumente sind nur Einträge erlaubt, die numerische Werte ergeben, andernfalls liefert die Funktion den Fehler `#WERT!`. Zugelassen werden aber auch Argumente, die numerische Werte als Zeichenfolge bereitstellen, wie einige der Umwandlungsfunktionen oder die Funktion `TEXT()`.

In dem Beispiel in Abbildung 6.17 wird die Funktion benutzt, um schnell die Frage zu beantworten, bei wie vielen Kapiteln der Planwert und der Istwert übereinstimmen. Dazu wird einfach die Summe der `DELTA()`-Werte gebildet.

	A	B	C	D
1				
2	**Werte auf Gleichheit prüfen**			
3				
4		Planseiten	Istseiten	Gleich
5		Zahl1	Zahl2	DELTA()
6	Kapitel 1	12	15	0
7	Kapitel 2	23	23	1
8	Kapitel 3	34	34	1
9	Kapitel 4	23	25	0
10	Kapitel 5	43	40	0
11			Passende Kapitel:	2

Abbildung 6.17 Prüfen der exakten Übereinstimmung

DEZINBIN()

DEC2BIN()

Syntax: `DEZINBIN(Zahl; Stellen)`
Beispiel: `=DEZINBIN(100)`
Ergebnis: `1100100`

Die Funktion `DEZINBIN()` liefert den Binärwert einer Dezimalzahl als Zeichenfolge. Das Argument `Zahl` kann dabei nur Werte zwischen `-512` und `+511` annehmen. Die Zahl der ausgegebenen Stellen kann mit `Stellen` festgelegt werden. Ohne Angabe der Stellenzahl werden nur die notwendigen Stellen angezeigt. Ist `Stellen` größer als deren Zahl, wird die Binärzahl mit führenden Nullen ausgegeben.

Ist `Zahl` negativ, wird die angegebene Stellenzahl ignoriert und der binäre Wert immer mit 10 Zeichen ausgegeben, wobei das erste Zeichen 1 das Vorzeichenbit ist. Sowohl beim Argument `Zahl` als auch bei `Stellen` werden Nachkommastellen ignoriert.

Ist `Zahl` < `-512` oder > `511`, liefert die Funktion den Fehler `#ZAHL!`. Das gilt auch, wenn die Funktion zur Ausgabe des Werts mehr Stellen benötigt, als angegeben ist, oder wenn der Wert für `Stellen` ≤ `0` ist. Sind `Zahl` oder `Stellen` keine numerischen Ausdrücke, erscheint der Fehler `#WERT!`.

Die Funktion `DEZINBIN()` liefert ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion `DEZIMAL()` ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. `=DEZIMAL(E10;2)` wandelt die binäre Zeichenfolge 10101 in die dezimale Zahl 21 um. Wenn Sie irrtümlich auf eine binäre Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: `10101 + 50 = 10151`.

	A	B	C	D	E	F	G	H
1								
2	**Umwandlungen von Dezimalzahlen**							
3								
4	Zahl	Stellen	DEZINBIN(Zahl)	DEZINBIN(Zahl;Stellen)	DEZINHEX(Zahl)	DEZINHEX(Zahl;Stellen)	DEZINOKT(Zahl)	DEZINOKT(Zahl;Stellen)
5	0	4	0	0000	0	0000	0	0000
6	1	4	1	0001	1	0001	1	0001
7	2	4	10	0010	2	0002	2	0002
8	3	4	11	0011	3	0003	3	0003
9	4	4	100	0100	4	0004	4	0004
10	5	4	101	0101	5	0005	5	0005
11	6	4	110	0110	6	0006	6	0006
12	7	4	111	0111	7	0007	7	0007
13	8	4	1000	1000	8	0008	10	0010
14	9	4	1001	1001	9	0009	11	0011
15	10	4	1010	1010	A	000A	12	0012

Abbildung 6.18 Beispiele für die Umwandlung dezimaler Zahlen in andere Zahlensysteme

DEZINHEX()

DEC2HEX()

Syntax: `DEZINHEX(`**`Zahl`**`; Stellen)`

Beispiel: `=DEZINHEX(15)`

Ergebnis: `F`

Die Funktion `DEZINHEX()` liefert den Hexadezimalwert einer Dezimalzahl als Zeichenfolge. Das Argument `Zahl` kann dabei eine ganze Zahl mit Werten zwischen `-549.755.813.888` und `+549.755.813.887` sein. Die Zahl der ausgegebenen Stellen kann mit `Stellen` festgelegt werden. Ohne Angabe der Stellenzahl werden nur die notwendigen Stellen ausgegeben; ist `Stellen` größer als deren Zahl, wird die hexadezimale Zahl mit führenden Nullen ausgegeben.

Ist Zahl negativ, wird die angegebene Stellenzahl ignoriert und der hexadezimale Wert immer mit 10 Zeichen (40 Bits) ausgegeben, wobei das erste Bit das Vorzeichenbit ist. Sowohl beim Argument Zahl als auch bei Stellen werden Nachkommastellen ignoriert.

Wenn Zahl keine zulässige dezimale Zahl ist, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!. Ist Zahl oder Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!.

Die Funktion DEZINHEX() liefert ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;16) wandelt die hexadezimale Zeichenfolge 000F in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine hexadezimale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0006 + 10 = 16. Kommen in der hexadezimalen Zeichenfolge Buchstaben vor, liefert die Berechnung den Fehler #WERT!.

DEZINOKT()

DEC2OCT()

Syntax: DEZINOKT(**Zahl**; Stellen)
Beispiel: =DEZINOKT(8)
Ergebnis: 10

Die Funktion DEZINOKT() liefert den Oktalwert einer Dezimalzahl als Zeichenfolge. Das Argument Zahl kann dabei Werte zwischen -536.870.912 und +536.870.911 annehmen. Die Zahl der ausgegebenen Stellen kann mit Stellen festgelegt werden. Ohne Angabe der Stellenzahl werden nur die notwendigen Stellen ausgegeben; ist Stellen größer als deren Zahl, wird die oktale Zahl mit führenden Nullen ausgegeben.

Ist Zahl negativ, wird die angegebene Stellenzahl ignoriert und der oktale Wert immer mit 10 Zeichen (30 Bits) ausgegeben, wobei das erste Bit das Vorzeichenbit ist. Sowohl beim Argument Zahl als auch bei Stellen werden Nachkommastellen ignoriert.

Wenn Zahl keine zulässige dezimale Zahl ist, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!. Ist Zahl oder Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!.

Die Funktion DEZINOKT() liefert ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;8) wandelt 0017 in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine oktale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0073 + 10 = 83.

GAUSSF.GENAU()

ERF.PRECISE()

Syntax: GAUSSF.GENAU(**x**)
Beispiel: =GAUSSF.GENAU(1)
Ergebnis: 0,8427

Die Funktion GAUSSF.GENAU() liefert Werte des gaußschen Fehlerintegrals. Die Funktion gehört eigentlich zu den Verteilungsfunktionen der Statistik. Das Fehlerintegral gibt die Wahrscheinlichkeit an, mit der ein Wert bei der mit x angegebenen Untergrenze, ab der die Funktion integriert werden soll, zu finden ist, wobei die Normalverteilung zugrunde gelegt wird.

Ist x nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Die seit Excel 2010 eingeführte Funktion GAUSSFKOMPL.GENAU() und die ältere Funktion GAUSSFEHLER() unterscheiden sich durch den Namen, der an das in der Version 2010 eingeführte Benennungssystem mit den durch Punkte gegliederten Namen angepasst wurde. Bei der neueren Funktion wurde zudem das Argument Obere_Grenze weggelassen. Beide Funktionen verwenden allerdings einen gegenüber älteren Versionen verbesserten Algorithmus, der genauere Werte liefert.

Siehe auch Abbildung 6.19 zu der Funktion GAUSSFEHLER().

GAUSSFEHLER()

ERF()

Syntax: GAUSSFEHLER(**Untere_Grenze**; Obere_Grenze)
Beispiel: =GAUSSFEHLER(0;1)
Ergebnis: 0,8427

Die Funktion GAUSSFEHLER() liefert Werte des Gaußschen Fehlerintegrals. Die Funktion gehört eigentlich zu den Verteilungsfunktionen der Statistik. Das Fehlerintegral gibt an, mit welcher Wahrscheinlichkeit ein Merkmalswert innerhalb eines bestimmten Intervalls, das durch die Argumente der Funktion angegeben wird, zu finden ist. Dabei wird die Normalverteilung (Gaußverteilung) zugrunde gelegt.

Mit Untere_Grenze (0 oder größer) wird der Wert angegeben, ab dem die Funktion integriert werden soll, mit Obere_Grenze der Wert, bis zu dem integriert wird. Wird das Argument Obere_Grenze nicht angegeben, wird die Integration von 0 bis Untere_Grenze durchgeführt. Das Ergebnis ist dann gleich dem Ergebnis der Funktion GAUSSF.GENAU().

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Gaußsche Fehlerfunktion und ihr Kompliment

x	GAUSSFEHLER()	GAUSSF.GENAU()	GAUSSFKOMPL()	GAUSSFKOMPL.GENAU()	GAUSSFEHLER() + GAUSSFKOMPL()
0	0	0	1	1	1
0,25	0,27633	0,27633	0,72367	0,72367	1
0,5	0,52050	0,52050	0,47950	0,47950	1
0,75	0,71116	0,71116	0,28884	0,28884	1
1	0,84270	0,84270	0,15730	0,15730	1
1,25	0,92290	0,92290	0,07710	0,07710	1
1,5	0,96611	0,96611	0,03389	0,03389	1
1,75	0,98667	0,98667	0,01333	0,01333	1
2	0,99532	0,99532	0,00468	0,00468	1
2,25	0,99854	0,99854	0,00146	0,00146	1
2,5	0,99959	0,99959	0,00041	0,00041	1
2,75	0,99990	0,99990	0,00010	0,00010	1
3	0,99998	0,99998	0,00002	0,00002	1

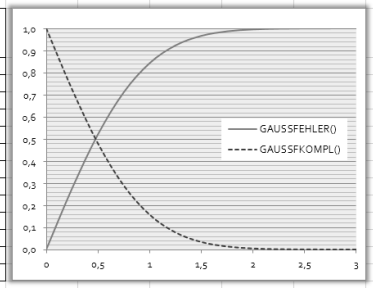


Abbildung 6.19 Die Funktionen GAUSSFEHLER() und GAUSSKOMPL()

GAUSSFKOMPL()

ERFC()

Syntax: GAUSSFKOMPL(**Untere_Grenze**)
Beispiel: =GAUSSFKOMPL(0,8)
Ergebnis: 0,2579

Die Funktion GAUSSKOMPL() liefert komplementäre Werte des Gaußschen Fehlerintegrals, vergleiche GAUSSFEHLER().

Es gilt die Beziehung:

```
1 - GAUSSFEHLER(x) = GAUSSFKOMPL(x)
```

Die Summe von GAUSSFEHLER(x) und GAUSSFKOMPL(x) ist also immer 1, wie auch in Abbildung 6.19 zu der Funktion GAUSSFEHLER() zu sehen ist.

Ist Untere_Grenze nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

GAUSSFKOMPL.GENAU()

ERFC.PRECISE()

Syntax: GAUSSFKOMPL.GENAU(**x**)
Beispiel: =GAUSSFKOMPL.GENAU (0,8)
Ergebnis: 0,2579

Die Funktion GAUSSFKOMPL.GENAU() liefert komplementäre Werte des Gaußschen Fehlerintegrals, vergleiche GAUSSF.GENAU().

Es gilt die Beziehung:

```
1 - GAUSSF.GENAU(x) = GAUSSFKOMPL.GENAU(x)
```

Ist x nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Die seit Excel 2010 eingeführte Funktion GAUSSFKOMPL.GENAU() und die ältere Funktion GAUSSFKOMPL() unterscheiden sich nur durch den Namen, der an das in der Version 2010 eingeführte Benennungssystem mit den durch Punkte gegliederten Namen angepasst wurde. Beide Funktionen verwenden allerdings einen gegenüber älteren Versionen verbesserten Algorithmus, der genauere Werte liefert.

Siehe auch Abbildung 6.19 zu der Funktion GAUSSFEHLER().

GGANZZAHL()

GESTEP()

Syntax: GGANZZAHL(**Zahl**; Schritt)
Beispiel: =GGANZZAHL(1;2)
Ergebnis: 1

Die Funktion GGANZZAHL() prüft, ob der mit Zahl angegebene Wert den mit Schritt angegebenen Schwellenwert erreicht oder überschreitet. Sie ergibt 1, wenn Zahl größer oder gleich Schritt ist, sonst 0. Wird Schritt nicht angegeben, wird er als 0 angenom-

men. Ungeachtet des missverständlichen deutschen Funktionsnamens können auch gebrochene Zahlen für die beiden Argumente angegeben werden.

Ist eines der Argumente nicht numerisch, erscheint der Fehler `#WERT!`.

In Kombination mit `SUMME()` lässt sich mit Hilfe von `GGANZZAHL()` leicht prüfen, wie viele Werte in einer Spalte einen bestimmen Schwellenwert erreicht haben. Abbildung 6.20 zeigt ein Beispiel, in dem der Bestand mit dem Mindestbestand verglichen wird. Die Formel `=GGANZZAHL(C5;D5)` verrät, ob der Bestand über dem Mindestbestand liegt oder nicht.

	A	B	C	D	E
1					
2	**Prüfen auf Schwellenwerte**				
3					
4	Artnr	Bezeichnung	Mindestbestand	Bestand	GGANZZAHL()
5	7777	Jalousie Ccxs	500	600	0
6	7778	Jalousie Ccxx	500	500	1
7	8443	Jalousie VVx	500	700	0
8	7774	Jalousie Vvxx	500	450	1
9	5667	Markise Luxor	500	890	0
10	8666	Rollo Dark	300	340	0
11	5554	Rollo PCx	300	230	1
12	6666	Rollo Top	300	700	0
13	5222	Rollo XXs	300	400	0
14	7999	Sunset	300	500	0
15		Anzahl Artikel mit zu geringem Bestand:			**3**

Abbildung 6.20 Erreicht oder unterschreitet der Bestand den vorgegebenen Mindestbestand?

HEXINBIN()

HEX2BIN()

Syntax: `HEXINBIN(`**`Zahl`**`; Stellen)`
Beispiel: `=HEXINBIN(A)`
Ergebnis: `1010`

Die Funktion `HEXINBIN()` liefert den Binärwert einer hexadezimalen Zahl als Zeichenfolge. Das Argument `Zahl` kann dabei maximal 10 Stellen lang sein und Werte zwischen `FFFFFFFE00` und `1FF` annehmen, das entspricht den dezimalen Werten –512 und +511. Die Zahl der ausgegebenen Stellen kann mit `Stellen` festgelegt werden. Ohne Angabe

der Stellenzahl werden nur die notwendigen Stellen ausgegeben; ist `Stellen` größer als deren Zahl, wird die binäre Zahl mit führenden Nullen aufgefüllt.

Ist `Zahl` negativ, wird die angegebene Stellenzahl ignoriert und der binäre Wert immer mit 10 Zeichen (10 Bits) ausgegeben, wobei das erste Bit das Vorzeichenbit ist. Sowohl beim Argument `Zahl` als auch bei `Stellen` werden Nachkommastellen ignoriert.

Wenn `Zahl` keine zulässige hexadezimale Zahl ist, gibt die Funktion den Fehler `#ZAHL!` aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist `Stellen` negativ, erscheint ebenfalls der Fehler `#ZAHL!`. Ist `Zahl` oder `Stellen` nicht numerisch, liefert die Funktion den Fehler `#WERT!`.

Die Funktion `HEXINBIN()` liefert ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion `DEZIMAL()` ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. `=DEZIMAL(E10;2)` wandelt die binäre Zeichenfolge 10101 in die dezimale Zahl 21 um. Wenn Sie irrtümlich auf eine binäre Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: `10101 + 50 = 10151`.

	A	B	C	D	E	F	G
1							
2	**Umwandlungen von Hexadezimalzahlen**						
3							
4	Zahl	Stellen	HEXINBIN(Zahl)	HEXINBIN(Zahl;Stellen)	HEXINDEZ()	HEXINOKT(Zahl)	HEXINOKT(Zahl;Stellen)
5	0	4	0	0000	0	0	0000
6	1	4	1	0001	1	1	0001
7	2	4	10	0010	2	2	0002
8	3	4	11	0011	3	3	0003
9	4	4	100	0100	4	4	0004
10	5	4	101	0101	5	5	0005
11	6	4	110	0110	6	6	0006
12	7	4	111	0111	7	7	0007
13	8	4	1000	1000	8	10	0010
14	9	4	1001	1001	9	11	0011
15	A	4	1010	1010	10	12	0012
16	B	4	1011	1011	11	13	0013
17	C	4	1100	1100	12	14	0014
18	D	4	1101	1101	13	15	0015
19	E	4	1110	1110	14	16	0016
20	F	4	1111	1111	15	17	0017
21	10	8	10000	00010000	16	20	00000020

Abbildung 6.21 Funktionen für die Umwandlung hexadezimaler Zahlen

HEXINDEZ()

HEX2DEC()

Syntax: HEXINDEZ(**Zahl**)
Beispiel: =HEXINDEZ(F)
Ergebnis: 15

Die Funktion HEXINDEZ() liefert den Dezimalwert einer hexadezimalen Zahl. Das Argument Zahl kann dabei maximal 10 Stellen (40 Bits) lang sein und Werte zwischen 8000000000 und 7FFFFFFFFF annehmen, das entspricht dezimalen Werten zwischen –549.755.813.888 und 543.755.813.887. Das erste Bit ist das Vorzeichenbit.

Wenn Zahl keine zulässige hexadezimale Zahl ist, gibt die Funktion den Fehler #ZAHL! aus. Ist der Wert nicht numerisch, erscheint der Fehler #WERT!.

Siehe auch Abbildung 6.21 zu der Funktion HEXINBIN().

HEXINOKT()

HEX2OCT()

Syntax: HEXINOKT(**Zahl**; Stellen)
Beispiel: =HEXINOKT(F)
Ergebnis: 17

Die Funktion HEXINOKT() liefert den Oktalwert einer hexadezimalen Zahl. Das Argument Zahl kann dabei maximal 10 Stellen (40 Bits) lang sein und Werte zwischen FFE0000000 und 1FFFFFFF annehmen. Das erste Bit ist das Vorzeichenbit. Stellen gibt an, wie viele Stellen angezeigt werden. Ohne Angabe werden nur die notwendigen Stellen angezeigt; ist Stellen größer als deren Anzahl, werden führende Nullen ausgegeben. Ist Zahl negativ, ignoriert die Funktion das Argument Stellen und gibt eine oktale Zahl mit 10 Zeichen aus.

Wenn Zahl keine zulässige hexadezimale Zahl ist, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!. Ist Zahl oder Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!.

Im Unterschied zu der Funktion HEXINDEZ() liefert HEXINOKT() ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;8) wandelt die oktale Zeichenfolge 0017 in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine oktale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0073 + 10 = 83.

Siehe auch Abbildung 6.21 zu der Funktion HEXINBIN().

IMABS()

IMABS()

Syntax:	IMABS(**Komplexe_Zahl**)
Beispiel:	=IMABS("3+4i")
Ergebnis:	5

Die Funktion IMABS() liefert den Absolutbetrag einer komplexen Zahl. Dieser Betrag entspricht der Länge des Vektors in der geometrischen Darstellung der komplexen Zahl, also der Entfernung des Punkts (x,x) vom Ursprung (0,0), siehe Abbildung 6.6.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!. Hilfsweise kann das Argument mit Hilfe der Funktion KOMPLEXE() aus einem Realteil beispielsweise einem imaginären Teil zusammengestellt werden, : =IMABS(KOMPLEXE(1;5)).

Das Ergebnis der Funktion ist immer eine positive reelle Zahl. Die Funktion arbeitet mit der Formel:

```
=WURZEL(x^2+y^2)
```

Also im Beispiel:

```
=WURZEL(3^2+4^2)
```

IMAGINÄRTEIL()

IMAGINARY()

Syntax: IMAGINÄRTEIL(**Komplexe_Zahl**)
Beispiel: =IMAGINÄRTEIL("5-12i")
Ergebnis: -12

Die Funktion IMAGINÄRTEIL() liefert den imaginären Teil einer komplexen Zahl. Als Ergebnis wird eine reelle Zahl angezeigt. Es gibt also an, mit welcher reellen Zahl die imaginäre Einheit i innerhalb einer komplexen Zahl multipliziert wird.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Die Funktion IMAGINÄRTEIL() kann auch als Argument in der Funktion KOMPLEXE() verwendet werden, um eine komplexe Zahl zu erzeugen:

```
=KOMPLEXE(IMREALTEIL(A17);IMAGINÄRTEIL(A16))
```

IMAPOTENZ()

IMPOWER()

Syntax: IMAPOTENZ(**Komplexe_Zahl**; **Potenz**)
Beispiel: =IMAPOTENZ("3+2i";2)
Ergebnis: 5+12i

Die Funktion IMAPOTENZ() liefert die Potenz der mit Komplexe_Zahl angegebenen komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!. Für das Argument Potenz können ganze positive oder negative Zahlen, aber auch Brüche angegeben werden. Wird dagegen als Potenz eine komplexe Zahl oder ein nicht numerischer Ausdruck angegeben, liefert die Funktion den Fehler #WERT!.

Das Ergebnis ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMARGUMENT()

IMARGUMENT()

Syntax: `IMARGUMENT(`**`Komplexe_Zahl`**`)`
Beispiel: `=IMARGUMENT("3+3i")`
Ergebnis: `0,785`

Die Funktion `IMARGUMENT()` liefert das Argument einer komplexen Zahl. Dies ist der Winkel im Bogenmaß, der für die Darstellung der komplexen Zahl in der goniometrischen Form verwendet wird.

Das Argument `Komplexe_Zahl` muss als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

IMCOS()

IMCOS()

Syntax: `IMCOS(`**`Komplexe_Zahl`**`)`
Beispiel: `=IMCOS("3+4i")`
Ergebnis: `-27,03-3,85i` (Nachkommastellen gekürzt)

Die Funktion `IMCOS()` liefert den Kosinus einer komplexen Zahl.

Das Argument `Komplexe_Zahl` muss als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Für den Kosinus gilt die Formel:

```
cos(x + yi) = cos(x) cosh(y) – sin(x) sinh(y)i
```

die beispielsweise für `3+4i` mit

```
=KOMPLEXE((COS(3)*COSHYP(4));-(SIN(3)*SINHYP(4)))
```

nachgebildet werden kann.

Das Ergebnis ist eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMCOSEC()

IMCSC()

Syntax:	IMCOSEC(**Zahl**)
Beispiel:	=IMCOSEC("4+3i")
Ergebnis:	-0,075...+0,06...i

Die in Excel 2013 neu eingeführte Funktion IMCOSEC() gibt den Kosekans einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den Kosekans einer komplexen Zahl gilt:

```
cosec(x + yi) = 1 / sin(x + yi)
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

Hilfsweise kann das Argument Zahl mit Hilfe der Funktion KOMPLEXE() aus einem Realteil und einem imaginären Teil zusammengestellt werden, z. B. : =IMABS(KOMPLEXE(1;5) ergibt 1+5i.

IMCOSECHYP()

IMCSCH()

Syntax:	IMCOSECHYP(**Zahl**)
Beispiel:	=IMCOSECHYP("4+3i")
Ergebnis:	-0,036...+0,05...i

Die in Excel 2013 neu eingeführte Funktion IMCOSECHYP() gibt den hyperbolischen Kosekans einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den hyperbolischen Kosekans einer komplexen Zahl gilt:

cosech(x + yi) = 1 / sinh(x +yi) = 2 / $e^{x} - e^{-x}$

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMCOSHYP()

IMCOSH()

Syntax:	IMCOSHYP(**Zahl**)
Beispiel:	=IMCOSHYP("4+3i")
Ergebnis:	-0,036...+0,05...i

Die in Excel 2013 neu eingeführte Funktion IMCOSHYP() gibt den hyperbolischen Kosinus einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den hyperbolischen Kosinus einer komplexen Zahl gilt:

cosh(x + yi) = cosh(x)*cos(y) + i*sinh(x)*sin(y)

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMCOT()

IMCOT()

Syntax:	IMCOT(**Zahl**)
Beispiel:	=IMCOT("4+3i")
Ergebnis:	0,004...-0,999...i

Die in Excel 2013 neu eingeführte Funktion IMCOT() gibt den Kotangens einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den Kotangens einer komplexen Zahl gilt:

```
cot(x + yi) = cos(x + yi)/sin(x + yi)
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMDIV()

IMDIV()

Syntax:	`IMDIV(`**`Komplexe_Zahl1`**`; `**`Komplexe_Zahl2`**`)`
Beispiel:	`=IMDIV("25-10i";"1+2i")`
Ergebnis:	`1-12i`

Mit Hilfe der Funktion `IMDIV()` kann eine komplexe Zahl, die mit `Komplexe_Zahl1` angegeben wird, durch eine andere mit `Komplexe_Zahl2` angegebene komplexe Zahl dividiert werden.

Beide Argumente müssen als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Das Ergebnis ist wiederum eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

Die Funktion arbeitet nach der Formel:

```
(a+bi)/(c+di) = (a+bi)(c-di)/(c+di)(c-di)
      =(ac+bd)/(c^2+d^2)+(bc-ad)/(c^2+d^2) * i
```

IMEXP()

IMEXP()

Syntax:	`IMEXP(`**`Komplexe_Zahl`**`)`
Beispiel:	`=IMEXP("0+4i")`
Ergebnis:	`-0,65-0,76i` (Nachkommastellen gekürzt)

Die Funktion `IMEXP()` liefert das Resultat der Potenzierung von e (Eulersche Zahl) mit einer komplexen Zahl. Das Argument `Komplexe_Zahl` gibt also den Exponenten an.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Das Ergebnis ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

Für eine in exponentieller Schreibweise vorliegende komplexe Zahl gilt wegen der Eulerschen Formel:

$e^{(x + yi)} = e^x * e^{yi} = e^x * (\cos y + i \sin y)$

IMEXP() ist die Umkehrfunktion von IMLN(). Wenn also IMEXP(1 + 2i) die komplexe Zahl k liefert, ergibt IMLN(k) die komplexe Zahl 1 +2i.

IMKONJUGIERTE()

IMCONJUGATE()

Syntax: IMKONJUGIERTE(**Komplexe_Zahl**)
Beispiel: =IMKONJUGIERTE("1+2i")
Ergebnis: 1-2i

Die Funktion IMKONJUGIERTE() liefert zu einer komplexen Zahl das konjugiert komplexe Komplement.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Im Ergebnis der Funktion wird das Vorzeichen des imaginären Teils umgedreht. Geometrisch handelt es sich um eine Spiegelung an der reellen Achse. Es ergeben sich beispielsweise folgende Zusammenhänge: Das Produkt einer komplexen Zahl mit dem Komplement ergibt das Quadrat ihres Absolutbetrags:

```
=IMPRODUKT("1+2i";"1-2i") = IMABS("1+2i")^2 = 5
```

Die Summe einer komplexen Zahl mit dem Komplement verdoppelt den Realteil:

```
=IMSUMME("1+2i";"1-2i") = 2
```

IMLN()

IMLN()

Syntax: IMLN(**Komplexe_Zahl**)
Beispiel: =IMLN("2+2i")
Ergebnis: 1,04+0,78i (Nachkommastellen gekürzt)

Die Funktion IMLN() liefert den natürlichen Logarithmus einer komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den natürlichen Logarithmus einer komplexen Zahl gilt:

$$\ln(x+yi) = \ln\sqrt{x^2 + y^2} + i*\arctan\frac{y}{x}$$

Das Ergebnis ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

IMLN() ist die Umkehrfunktion von IMEXP(). Wenn also IMLN(1 + 2i) die komplexe Zahl k liefert, ergibt IMEXP(k) die komplexe Zahl 1 +2i.

IMLOG10()

IMLOG10()

Syntax: IMLOG10(**Komplexe_Zahl**)
Beispiel: =IMLOG10("100-i")
Ergebnis: 2,00002-0,00434i

Die Funktion IMLOG10() liefert den dekadischen Logarithmus einer komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Der gewöhnliche Logarithmus, wie er auch genannt wird, kann auf folgende Weise aus dem natürlichen Logarithmus berechnet werden:

$$\log_{10}(x + yi) = \log_{10}e * \ln(x + yi)$$

Das Ergebnis ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

IMLOG2()

IMLOG2()

Syntax:	IMLOG2(**Komplexe_Zahl**)
Beispiel:	=IMLOG2("8+4i")
Ergebnis:	3,1609-0,6689i

Die Funktion IMLOG2() liefert den binären Logarithmus einer komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Der binäre Logarithmus kann auf folgende Weise aus dem natürlichen Logarithmus berechnet werden:

$\log_2(x + yi) = \log_2 e * \ln(x + yi)$

Das Ergebnis ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

IMPRODUKT()

IMPRODUCT()

Syntax:	IMPRODUKT(**Komplexe_Zahl1**; Komplexe_Zahl2,...)
Beispiel:	=IMPRODUKT("3+4i";"3-4i")
Ergebnis:	25

Die Funktion IMPRODUKT() berechnet seit Excel 2007 das Produkt von bis zu 255, in den älteren Versionen von bis zu 29 komplexen Zahlen.

Die Argumente müssen als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Das Ergebnis der Funktion ist entweder eine komplexe Zahl oder eine reelle Zahl wie in dem angegebenen Beispiel. Dabei wird nach folgender Formel verfahren:

(a+bi)*(c+di) = (ac – bd) + (ab+bc) *i

IMREALTEIL()

IMREAL()

Syntax: IMREALTEIL(**Komplexe_Zahl**)
Beispiel: =IMREALTEIL("12-i")
Ergebnis: 12

Die Funktion IMREALTEIL() liefert den reellen Anteil einer komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Das Ergebnis ist eine reelle Zahl. Die Funktion IMREALTEIL() kann auch als Argument in der Funktion KOMPLEXE() verwendet werden, um eine komplexe Zahl zu erzeugen, beispielsweise:

```
=KOMPLEXE(IMREALTEIL(A17);IMAGINÄRTEIL(A16))
```

IMSEC()

IMSEC()

Syntax: IMSEC(**Zahl**)
Beispiel: =IMSEC("4+3i")
Ergebnis: -0,065...-0,075...i

Die in Excel 2013 neu eingeführte Funktion IMSEC() gibt den Sekans einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den Sekans einer komplexen Zahl gilt:

```
sec(x + yi) = 1 / cos(x + yi)
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMSECHYP()

IMSECH()

Syntax: `IMSECHYP(`**`Zahl`**`)`
Beispiel: `=IMSECHYP("4+3i")`
Ergebnis: `-0,036...-0,005...i`

Die in Excel 2013 neu eingeführte Funktion `IMSECHYP()` gibt den hyperbolischen Sekans einer komplexen Zahl im Textformat `x+yi` oder `x+yj` zurück.

Das Argument `Zahl` muss als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Für den hyperbolischen Sekans einer komplexen Zahl gilt:

```
sech(x + yi) = 1 / cosh(x + yi)
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMSIN()

IMSIN()

Syntax: `IMSIN(`**`Komplexe_Zahl`**`)`
Beispiel: `=IMSIN("3,14+i")`
Ergebnis: `0,002-1,175i` (Nachkommastellen gekürzt)

Die Funktion `IMSIN()` liefert den Sinus einer komplexen Zahl.

Das Argument `Komplexe_Zahl` muss als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Für den Sinus einer komplexen Zahl gilt:

```
sin(x + yi) = sin(x) * cosh(y) + cos(x) * sinh(y)*i
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMSINHYP()

IMSINH()

Syntax: `IMSINHYP(`**`Zahl`**`)`
Beispiel: `=IMSINHYP("4+3i")`
Ergebnis: `-27,016...+3,853...i`

Die in Excel 2013 neu eingeführte Funktion `IMSINHYP()` gibt den hyperbolischen Sinus einer komplexen Zahl im Textformat `x+yi` oder `x+yj` zurück.

Das Argument `Zahl` muss als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

IMSUB()

IMSUB()

Syntax: `IMSUB(`**`Komplexe_Zahl1`**`; `**`Komplexe_Zahl2`**`)`
Beispiel: `=IMSUB("12+2i";"3-2i")`
Ergebnis: `9+4i`

Die Funktion `IMSUB()` liefert das Resultat der Subtraktion zweier komplexer Zahlen.

Die Argumente müssen als Zeichenfolge in der Form `x+yi` oder `x+yj` angegeben werden, wobei `x` und `y` zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler `#ZAHL!`.

Das Ergebnis ist entweder eine komplexe Zahl oder eine reelle Zahl, ausgegeben als Zeichenfolge. Dabei wird nach folgender Formel verfahren:

```
(a+bi)-(c+di) = (a-c) + (b-d) *i
```

IMSUMME()

IMSUM()

Syntax: `IMSUMME(`**`Komplexe_Zahl1`**`; Komplexe_Zahl2;...)`
Beispiel: `=IMSUMME("1+i";"1-i")`
Ergebnis: `2`

Die Funktion IMSUMME() berechnet seit Excel 2007 die Summe von bis zu 255, in den älteren Versionen von bis zu 29 komplexen Zahlen.

Die Argumente müssen als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Das Ergebnis ist entweder wieder eine komplexe Zahl oder eine reelle Zahl, ausgegeben als Zeichenfolge. Dabei wird nach folgender Formel verfahren:

```
(a+bi)+(c+di) = (a+c) + (b+d) *i
```

IMTAN()

IMTAN()

Syntax:	IMTAN(**Zahl**)
Beispiel:	=IMTAN("4+3i")
Ergebnis:	0,004...+1,000...i

Die in Excel 2013 neu eingeführte Funktion IMTAN() gibt den Tangens einer komplexen Zahl im Textformat x+yi oder x+yj zurück.

Das Argument Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Für den Tangens einer komplexen Zahl gilt:

```
tan(x + yi) = sin(x + yi) / cos(x + yi)
```

Das Ergebnis der Funktion ist wieder eine komplexe Zahl, die als Zeichenfolge ausgegeben wird.

IMWURZEL()

IMSQRT()

Syntax:	IMWURZEL(**Komplexe_Zahl**)
Beispiel:	=IMWURZEL("3+2i")
Ergebnis:	1,82-0,55i (Nachkommastellen gekürzt)

Die Funktion IMWURZEL() liefert die Quadratwurzel einer komplexen Zahl.

Das Argument Komplexe_Zahl muss als Zeichenfolge in der Form x+yi oder x+yj angegeben werden, wobei x und y zwei beliebige Zahlen sein können. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. Unerlaubte Zeichen führen zu dem Fehler #ZAHL!.

Die Quadratwurzel wird nach folgender Formel berechnet:

$$\sqrt{x+yi} = \sqrt{r}\left(\cos\frac{\varphi}{2} + i*\sin\frac{\varphi}{2}\right)$$

Dabei ist r der Absolutwert, der der Länge des Vektors in der Polardarstellung entspricht und φ der Polarwinkel, siehe Abbildung 6.5.

Das Ergebnis ist wieder eine komplexe Zahl, ausgegeben als Zeichenfolge.

KOMPLEXE()

COMPLEX()

Syntax:	KOMPLEXE(**Realteil**; **Imaginärteil**; Suffix)
Beispiel:	=KOMPLEXE(3;4;i)
Ergebnis:	3+4i

Die Funktion KOMPLEXE() wird dazu verwendet, eine komplexe Zahl aus zwei reellen Zahlen zu bilden, wobei das Argument Realteil den reellen Anteil der Zahl liefert, das Argument Imaginärteil den imaginären Anteil. Mit Suffix lässt sich festlegen, ob der Buchstabe i oder j zur Kennzeichnung des Imaginärteils verwendet wird. Vorgabe ist i.

Sind Realteil oder Imaginärteil nicht numerisch, gibt die Funktion den Fehler #WERT! aus. Das gilt auch, wenn für Suffix etwas anderes angegeben wird als die Kleinbuchstaben i oder j.

Soll eine komplexe Zahl in einer der Tabellenfunktionen für komplexe Zahlen als Argument verwendet werden, muss sie immer als Zeichenfolge in der algebraischen Form a+bi angegeben werden. Das heißt, die Funktion KOMPLEXE() kann verwendet werden, um Argumente für diese Funktionen entsprechend aufzubereiten.

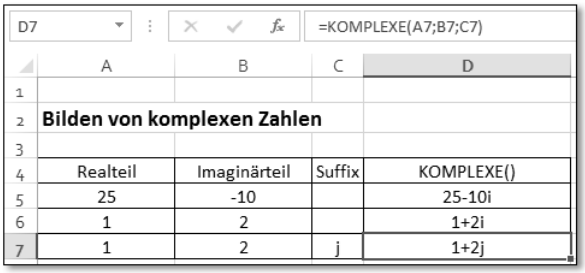


Abbildung 6.22 Verknüpfen zweier reeller Zahlen zu einer komplexen Zahl

OKTINBIN()

OCT2BIN()

Syntax: `OKTINBIN(Zahl; Stellen)`
Beispiel: `=OKTINBIN(100;10)`
Ergebnis: `0001000000`

Die Funktion `OKTINBIN()` liefert den Binärwert einer oktalen Zahl. Das Argument `Zahl` kann dabei maximal 10 Stellen lang sein und Werte zwischen `7777777000` und `777` annehmen, das entspricht den dezimalen Werten –512 und +511. Die Zahl der ausgegebenen Stellen kann mit `Stellen` festgelegt werden. Ohne Angabe der Stellenzahl werden nur die notwendigen Stellen ausgegeben; ist `Stellen` größer als deren Zahl, wird die binäre Zahl mit führenden Nullen ausgegeben.

Ist `Zahl` negativ, wird die angegebene Stellenzahl ignoriert und der binäre Wert immer mit 10 Zeichen (10 Bits) ausgegeben, wobei das erste Bit das Vorzeichenbit ist. Sowohl beim Argument `Zahl` als auch bei `Stellen` werden Nachkommastellen ignoriert.

Wenn `Zahl` keine zulässige oktale Zahl ist, gibt die Funktion den Fehler `#ZAHL!` aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist `Stellen` negativ, erscheint ebenfalls der Fehler `#ZAHL!`. Ist `Zahl` oder `Stellen` nicht numerisch, liefert die Funktion den Fehler `#WERT!`.

Die Funktion `OKTINBIN()` liefert ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion `DEZIMAL()` ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. `=DEZIMAL(E10;2)` wandelt die binäre Zeichenfolge 10101 in die dezimale Zahl 21 um. Wenn Sie irrtümlich auf eine binäre Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: `10101 + 50 = 10151`.

	A	B	C	D	E	F	G
1							
2	**Umwandlungen von oktalen Zahlen**						
3							
4	Zahl	Stellen	OKTINBIN(Zahl)	OKTINBIN(Zahl;Stellen)	OKTINDEZ()	OKTINHEX(Zahl)	OKTINHEX(Zahl;Stellen)
5	0	8	0	00000000	0	0	00000000
6	1	8	1	00000001	1	1	00000001
7	2	8	10	00000010	2	2	00000002
8	3	8	11	00000011	3	3	00000003
9	4	8	100	00000100	4	4	00000004
10	5	8	101	00000101	5	5	00000005
11	6	8	110	00000110	6	6	00000006
12	7	8	111	00000111	7	7	00000007
13	10	8	1000	00001000	8	8	00000008
14	11	8	1001	00001001	9	9	00000009
15	12	8	1010	00001010	10	A	0000000A

Abbildung 6.23 Funktionen für die Umwandlung von oktalen Zahlen

OKTINDEZ()

OCT2DEC()

Syntax: `OKTINDEZ(`**`Zahl`**`)`

Beispiel: `=OKTINDEZ(10)`

Ergebnis: `8`

Die Funktion `OKTINDEZ()` liefert den Dezimalwert einer oktalen Zahl. Das Argument `Zahl` kann dabei maximal 10 Stellen (30 Bits) lang sein. Das erste Bit ist das Vorzeichenbit.

Wenn `Zahl` keine zulässige oktale Zahl ist, gibt die Funktion den Fehler `#ZAHL!` aus. Ist der Wert nicht numerisch, erscheint der Fehler `#WERT!`.

Siehe auch Abbildung 6.23 zu der Funktion `OKTINBIN()`.

OKTINHEX()

OCT2HEX()

Syntax: `OKTINHEX(`**`Zahl`**`; Stellen)`

Beispiel: `=OKTINDEZ(100;6)`

Ergebnis: `000040`

Die Funktion OKTINHEX() liefert den hexadezimalen Wert einer oktalen Zahl. Das Argument Zahl kann dabei maximal 10 Stellen (30 Bits) lang sein. Das erste Bit ist das Vorzeichenbit.

Stellen gibt an, wie viele Stellen angezeigt werden. Ohne Angabe werden nur die notwendigen Stellen angezeigt; ist Stellen größer als deren Anzahl, werden führende Nullen ausgegeben. Ist Zahl negativ, ignoriert die Funktion das Argument Stellen und gibt eine hexadezimale Zahl mit 10 Zeichen aus.

Wenn Zahl keine zulässige oktale Zahl ist, gibt die Funktion den Fehler #ZAHL! aus. Reicht die angegebene Stellenzahl nicht für die Darstellung aus oder ist Stellen negativ, erscheint ebenfalls der Fehler #ZAHL!. Ist Zahl oder Stellen nicht numerisch, liefert die Funktion den Fehler #WERT!.

Im Unterschied zu der Funktion OKTINDEZ() liefert OKTINHEX() ihr Ergebnis in Form einer Zeichenfolge. Sie können dieses Ergebnis also nicht ohne Weiteres in eine Rechenoperation einbeziehen. Mit der in Excel 2013 neu eingeführten Funktion DEZIMAL() ist aber eine Umwandlung in eine dezimale Zahl leicht möglich. =DEZIMAL(E10;16) wandelt die hexadezimale Zeichenfolge 000F in die dezimale Zahl 15 um. Wenn Sie irrtümlich auf eine hexadezimale Zeichenfolge eine Zahl addieren, behandelt Excel die Zeichenfolge wie eine Zahl und liefert dann unsinnige Ergebnisse, etwa: 0006 + 10 = 16. Kommen in der hexadezimalen Zeichenfolge Buchstaben vor, liefert die Berechnung den Fehler #WERT!.

Siehe auch Abbildung 6.23 zu der Funktion OKTINBIN().

UMWANDELN()

CONVERT()

Syntax:	UMWANDELN(**Zahl**; **Von_Maßeinheit**; **In_Maßeinheit**)
Beispiel:	=UMWANDELN(4;"J";"cal")
Ergebnis:	0,955

Die Funktion UMWANDELN() liefert Umrechnungen zwischen verschiedenen Maßeinheiten. Mit Zahl wird angegeben, wie viele Einheiten der Von_Maßeinheit umgerechnet werden sollen. Von_Maßeinheit ist die Einheit, aus der umgerechnet wird; In_Maßeinheit ist die Einheit, in die umgerechnet werden soll.

Beide Argumente müssen als Zeichenfolge angegeben werden, also entweder direkt in Anführungszeichen oder durch einen Zellbezug oder eine Formel, die die Zeichenfolge liefert. Auf Groß- und Kleinschreibung muss geachtet werden.

Hat ein Argument einen falschen Datentyp, liefert die Funktion den Fehler #WERT!. Naturgemäß können nur Paare aus Maßeinheiten angegeben werden, die zur selben Gruppe gehören. Andernfalls oder wenn eine unbekannte Maßeinheit angegeben wird, liefert die Funktion den Fehlerwert #NV.

Die Zahl der Einheiten, die für eine Umrechnung verfügbar sind, ist in Excel 2013 beträchtlich erweitert worden, Tabelle 6.1 zeigt die Maßeinheiten aufgeteilt nach Gruppen.

Gewicht und Masse	**Maßeinheit**
Gramm	"g"
Hyl (Slug)	"sg"
Pfund (Handelsgewicht)	"lbm"
U (Atomare Masseneinheit)	"u"
Unze (Handelsgewicht)	"ozm"
Gran	"grain"
US-Hundredweight (Short)	"cwt" oder "shweight"
UK-Hundredweight (Long)	"uk_cwt" oder "lcwt" ("hweight")
Stone	"stone"
Tonne	"ton"
UK-Tonne	"uk_ton" oder "LTON" ("brton")
Entfernung	**Maßeinheit**
Meter	"m"
Meile (Statute Mile, USA)	"mi"
Nautische Meile	"Nmi"
Zoll	"in"
Fuß	"ft"
Yard	"yd"
Ångstrom	"ang"
Elle	"ell"
Lichtjahr	"ly"
Parsec	"parsec" oder "pc"

Tabelle 6.1 Liste der Maßeinheiten

Entfernung	**Maßeinheit**
Pica (1/72 Zoll)	"Picapt" oder "Pica"
Pica (1/6 Zoll)	"pica"
Meile (Survey Mile, USA)	"survey_mi"
Zeit	**Maßeinheit**
Jahr	"yr"
Tag	"day" oder "d"
Stunde	"hr"
Minute	"mn" oder "min"
Sekunde	"sec" oder "s"
Druck	**Maßeinheit**
Pascal	"Pa" (oder "p")
Atmosphäre	"atm" (oder "at")
mm Quecksilbersäule	"mmHg"
PSI	"psi"
Torr	"Torr"
Kraft	**Maßeinheit**
Newton	"N"
Dyne	"dyn" (oder "dy")
Pound-Force	"lbf"
Pond	"pond"
Energie	**Maßeinheit**
Joule	"J"
Erg	"e"
Thermodynamische Kalorie	"c"
I.T.-Kalorie	"cal"
Elektronenvolt	"eV" (oder "ev")
PS-Stunde	"HPh" (oder "hh")

Tabelle 6.1 Liste der Maßeinheiten (Forts.)

Energie	**Maßeinheit**
Wattstunde	"Wh" (oder "wh")
Foot-Pound	"flb"
BTU	"BTU" (oder "btu")
Leistung	**Maßeinheit**
Pferdestärken (PS)	"HP" (oder "h")
Pferdestärke	"PS"
Watt	"W" (oder "w")
Magnetismus	**Maßeinheit**
Tesla	"T"
Gauss	"ga"
Temperatur	**Maßeinheit**
Grad Celsius	"C" (oder "cel")
Grad Fahrenheit	"F" (oder "fah")
Kelvin	"K" (oder "kel")
Grad Rankine	"Rank"
Grad Réaumur	"Reau"
Volumen (oder Flüssigkeitsmaß)	**Maßeinheit**
Teelöffel (Tl)	"tsp"
Metrische TL (Teelöffel)	"tspm"
Esslöffel (El)	"tbs"
Flüssigunze	"oz"
Tasse	"cup"
US-Pint	"pt" (oder "us_pt")
UK-Pint	"uk_pt"
Quart	"qt"
UK-Quart	"uk_qt"

Tabelle 6.1 Liste der Maßeinheiten (Forts.)

Volumen (oder Flüssigkeitsmaß)	**Maßeinheit**
Gallone	"gal"
UK-Gallone	"uk_gal"
Liter	"l" oder "L" ("lt")
Kubikångström	"ang3" oder "ang^3"
US-Ölfass	"barrel"
US-Scheffel	"bushel"
Kubikfuß	"ft3" oder "ft^3"
Kubikzoll	"in3" oder "in^3"
Kubiklichtjahr	"ly3" oder "ly^3"
Kubikmeter	"m3" oder "m^3"
Kubikmeile	"mi3" oder "mi^3"
Kubikyard	"yd3" oder "yd^3"
Nautische Kubikmeile	"Nmi3" oder "Nmi^3"
Kubikpica	"Picapt3", "Picapt^3", "Pica3" oder "Pica^3"
Bruttoregistertonne	"GRT" ("regton")
Frachttonne	"MTON"
Fläche	**Maßeinheit**
UK-Morgen	"uk_acre"
US-Morgen	"us_acre"
Quadratångström	"ang2" oder "ang^2"
Ar	"ar"
Quadratfuß	"ft2" oder "ft^2"
Hektar	"ha"
Quadratzoll	"in2" oder "in^2"
Quadratlichtjahr	"ly2" oder "ly^2"
Quadratmeter	"m2" oder "m^2"
Morgen	"Morgen"

Tabelle 6.1 Liste der Maßeinheiten (Forts.)

Fläche	Maßeinheit
Quadratmeile	"mi2" oder "mi^2"
Nautische Quadratmeile	"Nmi2" oder "Nmi^2"
Quadratpica	"Picapt2", "Pica2", "Pica^2" oder "Picapt^2"
Quadratyard	"yd2" oder "yd^2"
Information	**Maßeinheit**
Bit	"bit"
Byte	"byte"
Geschwindigkeit	**Maßeinheit**
Admiralty-Knoten	"admkn"
Knoten	"kn"
Meter pro Stunde	"m/h" oder "m/hr"
Meter pro Sekunde	"m/s" oder "m/sec"
Meilen pro Stunde	"mph"

Tabelle 6.1 Liste der Maßeinheiten (Forts.)

Vor die beiden Einheitennamen können noch Abkürzungen gesetzt werden, die für häufig verwendete Präfixe bei Mengeneinheiten stehen, beispielsweise für Kilogramm:

```
=UMWANDELN(1;"kg";"lbm")
```

In Excel 2013 gibt es zwei Gruppen solcher Präfixe. Die erste Gruppe kommt fast immer mit einem Buchstaben aus, die zweite nutzt zwei Buchstaben. Der Multiplikator ist in diesem Fall immer eine Potenz von 2, weshalb diese Präfixe Binärpräfixe heißen.

Präfix	Multiplikator	Abkürzung
Yotta	1E+24	"Y"
Zetta	1E+21	"Z"
Exa	1E+18	"E"
Peta	1E+15	"P"
Tera	1E+12	"T"

Tabelle 6.2 Liste der Präfixe

Präfix	Multiplikator	Abkürzung
Giga	1E+09	"G"
Mega	1E+06	"M"
Kilo	1E+03	"k"
Hekto	1E+02	"h"
Deka	1E+01	"da" oder "e"
Dezi	1E-01	"d"
Zenti	1E-02	"c"
Milli	1E-03	"m"
Mikro	1E-06	"u"
Nano	1E-09	"n"
Piko	1E-12	"p"
Femto	1E-15	"f"
Atto	1E-18	"a"
Zepto	1E-21	"z"
Yocto	1E-24	"y"

Tabelle 6.2 Liste der Präfixe (Forts.)

Binär-präfix	Präfixwert	Abkürzung	Abgeleitet von
Yobi	2^80 = 1 208 925 819 614 629 174 706 176	"Yi"	Yotta
Zebi	2^70 = 1 180 591 620 717 411 303 424	"Zi"	Zetta
Exbi	2^60 = 1 152 921 504 606 846 976	"Ei"	Exa
Pebi	2^50 = 1 125 899 906 842 624	"Pi"	Peta
Tebi	2^40 = 1 099 511 627 776	"Ti"	Tera
Gibi	2^30 = 1 073 741 824	"Gi"	Giga
Mebi	2^20 = 1 048 576	"Mi"	Mega
Kibi	2^10 = 1024	"ki"	Kilo

Tabelle 6.3 Liste der Binärpräfixe

7 Statistische Funktionen

Funktion	Seite
ACHSENABSCHNITT()	465
ANZAHL()	466
ANZAHL2()	468
ANZAHLLEEREZELLEN()	471
BESTIMMTHEITSMASS()	473
BETA.INV()	474
BETA.VERT()	475
BINOM.INV()	477
BINOM.VERT()	478
BINOM.VERT.BEREICH()	480
CHIQU.INV()	481
CHIQU.INV.RE()	482
CHIQU.TEST()	482
CHIQU.VERT()	484
CHIQU.VERT.RE()	487
EXPON.VERT()	488
F.INV()	490
F.INV.RE()	491
F.TEST()	492
F.VERT()	493
F.VERT.RE()	495
FISHER()	496
FISHERINV()	497
G.TEST()	498
GAMMA()	499
GAMMA.INV()	501
GAMMA.VERT()	502
GAMMALN()	504
GAMMALN.GENAU()	504
GAUSS()	505

Funktion	Seite
GEOMITTEL()	506
GESTUTZTMITTEL()	508
HÄUFIGKEIT()	509
HARMITTEL()	511
HYPGEOM.VERT()	512
KGRÖSSTE()	514
KKLEINSTE()	515
KONFIDENZ.NORM()	517
KONFIDENZ.T()	519
KORREL()	521
KOVARIANZ.P()	523
KOVARIANZ.S()	524
KURT()	524
LOGNORM.INV()	526
LOGNORM.VERT()	527
MAX()	528
MAXA()	529
MEDIAN()	529
MIN()	531
MINA()	532
MITTELABW()	532
MITTELWERT()	534
MITTELWERTA()	536
MITTELWERTWENN()	538
MITTELWERTWENNS()	540
MODUS.EINF()	542
MODUS.VIELF()	543
NEGBINOM.VERT()	545
NORM.INV()	546
NORM.S.INV()	547

Funktion	Seite
NORM.S.VERT()	548
NORM.VERT()	549
PEARSON()	553
PHI()	554
POISSON.VERT()	555
QUANTIL.EXKL()	557
QUANTIL.INKL()	558
QUANTILSRANG.EXKL()	559
QUANTILSRANG.INKL()	561
QUARTILE.EXKL()	562
QUARTILE.INKL()	563
RANG.GLEICH()	564
RANG.MITTELW()	565
RGP()	566
RKP()	569
SCHÄTZER()	572
SCHIEFE()	574
SCHIEFE.P()	575
STABW.N()	577
STABW.S()	578
STABWA()	579
STABWNA()	581

Funktion	Seite
STANDARDISIERUNG()	581
STEIGUNG()	583
STFEHLERYX()	584
SUMQUADABW()	585
T.INV()	587
T.INV.2S()	587
T.TEST()	590
T.VERT()	592
T.VERT.2S()	594
T.VERT.RE()	595
TREND()	595
VAR.P()	598
VAR.S()	600
VARIANZA()	601
VARIANZENA()	602
VARIATION()	603
VARIATIONEN()	605
VARIATIONEN2()	606
WAHRSCHBEREICH()	607
WEIBULL.VERT()	609
ZÄHLENWENN()	610
ZÄHLENWENNS()	612

7.1 Einsatzbereiche für statistische Funktionen

Für Berechnungen aus dem Bereich der Statistik braucht Excel 2013 von der Funktionsvielfalt her gesehen den Vergleich mit professionellen Statistikprogrammen kaum zu scheuen. Um die Orientierung in diesem Bereich etwas zu erweitern, hier wenigstens ein paar kurze Vorbemerkungen.

7.1.1 Deskriptive und induktive statistische Methoden

Eine erste Aufteilung der statistischen Funktionen ergibt sich aus der Unterscheidung zwischen deskriptiven statistischen Methoden und induktiven statistischen Methoden.

Die ersteren, die beschreibenden Methoden, beziehen sich auf den gegebenen Ausgangsdatenbestand. Typische Funktionen sind hier etwa die Berechnung der verschiedenen Formen von Mittelwerten oder die Berechnung der Streuung.

Von induktiven statistischen Methoden wird gesprochen, wenn es darum geht, von Daten aus einer Stichprobe auf die Verhältnisse rückzuschließen, die in der Grundgesamtheit herrschen, aus der die Stichprobe genommen wurde.

Die Statistikfunktionen decken hier mehrere Bereiche ab: die Analyse einzelner Stichproben, bei denen eine oder mehrere Merkmale erfasst wurden, Analyse und Vergleich mehrerer Stichproben, Vergleich von Stichproben mit einer Grundgesamtheit und Aspekte wie Wahrscheinlichkeitsrechnung und Wahrscheinlichkeitsverteilungen von Zufallsvariablen.

7.1.2 Unterschiedliche Skalen

Um die Frage beantworten zu können, welche statistischen Methoden sich auf ein bestimmtes Datenmaterial anwenden lassen, ist es notwendig, die Art der Skalierung zu berücksichtigen. Grundsätzlich kann unterschieden werden zwischen nominalskalierten, ordinalskalierten und metrischskalierten Werten.

Wird beispielsweise bei Personen der Familienstand erfasst, stellen die möglichen Merkmale Werte einer Nominalskala dar. Es handelt sich dabei um qualitative Merkmale, bei denen nur festgestellt werden kann, dass sie sich unterscheiden. Sie lassen sich aber nicht in eine Ordnung bringen, die etwa einem Merkmal einen höheren Rang zuweisen würde als einem anderen. Es lässt sich auch nichts über die Abstände zwischen den Merkmalen aussagen.

Bei ordinalskalierten Werten kann von den Daten darüber hinaus auch eine bestimmte Rangordnung abgelesen werden, wie es etwa bei Schulnoten der Fall ist. Die Note 1 ist besser als die Note 2, 4 besser als 5, aber dennoch lässt sich nicht behaupten, dass der Abstand zwischen 1 und 2 genau dem Abstand zwischen 4 und 5 entspricht.

Liefern zusätzlich auch die genauen Abstände zwischen Werten verwertbare Informationen, wie etwa bei Daten über Gewicht und Größe, wird für die Daten eine metrische Skala verwendet.

Es ist evident, dass bestimmte statistische Auswertungen für eine bestimmte Skalierung wenig Sinn ergeben, wie etwa ein Mittelwert zwischen »verheiratet« und »ledig«.

Eine weitere grundlegende Unterscheidung in Bezug auf die Art des Datenmaterials ist die zwischen diskreten Werten und stetigen Werten. Während der Familienstand einer Person nur wenige Werte annehmen kann, lässt sich die Körpergröße in einer beliebigen Genauigkeit, also auch mit beliebigen Zwischenwerten, angeben.

7.1.3 Urliste und Merkmalsverteilung

Die erste Aufzeichnung der Merkmalswerte für die Einheiten, die statistisch untersucht werden, wird auch als *Urliste* bezeichnet. Sie enthält die Rohdaten für die weiteren Untersuchungen. Die Tabellenstruktur, die Excel vorgibt, ist gut geeignet, eine solche Urliste anzulegen.

	A	B	C	D	E	F	G	H	I	J
1										
2	**Beispiel für eine Urliste**									
3										
4	**Körpergröße von Jugendlichen**									
5										
6	184	195	183	184	170	179	166	183	155	186
7	168	173	164	195	185	187	174	187	182	163
8	152	158	181	178	175	167	170	184	157	164
9	153	162	166	168	191	179	163	186	165	196
10	162	158	181	160	183	168	184	185	194	183
11	180	183	191	190	173	184	176	167	192	167
12	182	160	167	187	162	169	180	159	173	173
13	176	163	163	157	161	184	177	197	195	172
14	151	177	160	177	173	196	166	174	198	163
15	186	158	174	177	196	185	183	161	154	166

Abbildung 7.1 Beispiel für eine Urliste zum Merkmal »Körpergröße«

In der Urliste werden alle vorkommenden Merkmale genauso oft notiert, wie sie tatsächlich vorkommen. Ein erster Schritt, um die Datenmenge zu reduzieren, ist deshalb regelmäßig, die Häufigkeit, mit der die verschiedenen Merkmale auftreten, zu berechnen und in der Folge mit der Verteilung der Häufigkeiten zu arbeiten. Bei Daten zur Körpergröße bietet es sich an, die erfassten Werte in Intervalle zu gliedern. Dazu kann in Excel die Funktion `HÄUFIGKEIT()` verwendet werden.

Dabei wird unterschieden zwischen der absoluten Häufigkeit, die angibt, wie oft ein Merkmal auftritt, und der relativen Häufigkeit, bei der die absolute Häufigkeit in Bezug gesetzt wird zur Gesamtzahl der beobachteten Werte. Die relative Häufigkeit kann nur Werte zwischen `0` und `1` annehmen und wird oft als Prozentsatz dargestellt.

Abbildung 7.2 zeigt eine erste Auswertung der abgebildeten Urliste, wobei die Daten in 10 Intervalle oder Klassen eingeteilt werden.

B19 {=HÄUFIGKEIT(A6:J15;A19:A28)}

	A	B	C	D	E
18	**Klassen**	**abs. Häufigkeit**	**rel. Häufigkeit**	**in %**	
19	155	5	0,05	5%	
20	160	8	0,08	8%	
21	165	14	0,14	14%	
22	170	12	0,12	12%	
23	175	11	0,11	11%	
24	180	11	0,11	11%	
25	185	19	0,19	19%	
26	190	8	0,08	8%	
27	195	6	0,06	6%	
28	200	6	0,06	6%	
29		100	1	100%	

Abbildung 7.2 Erste Auswertung der Häufigkeitsverteilung

7.2 Stichproben und Grundgesamtheiten

Fast alle statistischen Verfahren haben entweder mit Stichproben oder mit Grundgesamtheiten zu tun, häufig auch direkt oder indirekt mit beiden. Eine Stichprobe ist eine Untergruppe von Elementen aus einer Grundgesamtheit, die zufällig aus dieser Grundgesamtheit ausgewählt wurden. Zufällig heißt hier, dass bei der Auswahl darauf geachtet wird, dass nicht bestimmte Elemente der Grundgesamtheit bevorzugt werden. Die Grundgesamtheit ist die Menge der Elemente, aus denen die Stichprobe genommen wird.

Geht es beispielsweise um Wahlprognosen, besteht die Grundgesamtheit aus der Summe aller Wahlberechtigten in einem bestimmten Gebiet zu einer bestimmten Zeit. Soll nun etwa für eine Meinungsumfrage eine Stichprobe der Wahlberechtigten genommen werden, genügt es nicht, ein oder alle Telefonbücher zufällig aufzuschlagen (Telefonbesitzer werden bevorzugt) oder auf der Straße Leute anzusprechen (zu Hause Bleibende werden benachteiligt). Entsprechend haben die Institute, die Meinungsumfragen durchführen, wohlgehütete Geheimnisse, wie sie zu ihren repräsentativen Stichproben kommen.

7.3 Zufallsvariable und Wahrscheinlichkeit

Auf der Grundlage statistischer Erhebungen oder zuweilen auch theoretischer Überlegungen lässt sich häufig für das Auftauchen bestimmter Ereignisse eine bestimmte

Wahrscheinlichkeit angeben. So ist etwa die Wahrscheinlichkeit, beim Münzwurf eine Zahl zu werfen, 1/2 (0,5 oder 50%).

Ob Zahl oder Wappen nach dem Wurf oben liegt, hängt von zahlreichen physikalischen Ursachen ab, die sich normalerweise nicht eindeutig ermitteln lassen, und insofern kann davon gesprochen werden, dass das Ergebnis des Münzwurfs zufällig ist. Ähnlich ist es beim Würfeln. Die Wahrscheinlichkeit, eine 6 zu schaffen, wird mit 1/6 (0,166666) angegeben, vorausgesetzt, das Ergebnis ist wieder zufällig und nicht durch Manipulationen am Würfel beeinflusst.

7.3.1 Theoretische Wahrscheinlichkeit

In solchen Fällen, wo die Zahl der möglichen Ereignisse feststeht, wird von einer theoretischen Wahrscheinlichkeit gesprochen. Sie wird als Bruch

```
p = g / m
```

dargestellt, wobei `g` die Anzahl der günstigen Ereignisse und `m` die Anzahl der möglichen Fälle darstellt. Die Wahrscheinlichkeit, eine bestimmte Karte aus einem Skatspiel zu ziehen, ist also 1/32.

In anderen Fällen wird die Wahrscheinlichkeit durch Abzählen der Grundgesamtheit ermittelt. Ist z. B. bekannt, dass in einem Land 51% der Bevölkerung weiblich sind, dann ist die Wahrscheinlichkeit, dass ein zufällig ausgewählter Mensch weiblich ist, 0,51.

Eine Größe dieser Art heißt eine Zufallsvariable. Ist diese Variable so wie in den genannten Beispielen diskret, dann lässt sich direkt eine Wahrscheinlichkeit dafür angeben, dass sie einen bestimmten Wert oder einen von mehreren Werten annimmt.

7.3.2 Empirische Wahrscheinlichkeit

Häufig besteht aber gar nicht die Möglichkeit, die Zahl der möglichen Ereignisse genau zu bestimmen. Auch die Frage der Gleichberechtigung von Ereignissen ist vielfach nicht so einfach zu beantworten. Dies gilt jedenfalls, wenn es um nicht diskrete Variable geht. Sind etwa die Körpergrößen einer Grundgesamtheit von Menschen erfasst, dann stellt sich sowohl bei der Erfassung als auch bei der Angabe der Wahrscheinlichkeiten die Frage nach der Messgenauigkeit und nach der Einordnung.

Die Frage, wie wahrscheinlich die Größe von z. B. 1,73 m ist, kann nicht ohne Weiteres beantwortet werden: Ist etwa 1,7299999 mitgemeint oder nicht? Die Werte müssen in solchen Fällen in der Regel in Klassen eingeteilt werden (z. B. 172,5 bis 173,49999...).

In all diesen Fällen wird statt der theoretischen Wahrscheinlichkeit die empirische Wahrscheinlichkeit in Anspruch genommen. Als ihr Maß gilt die Formel:

```
w = h / n
```

Dabei ist `h` die Häufigkeit, mit der ein Ereignis auftritt, und `n` die Anzahl der Versuche, die zur Ermittlung der Wahrscheinlichkeit unternommen wurden. Dieses Maß kann natürlich nicht so exakt sein wie das Maß der theoretischen Wahrscheinlichkeit. Hier hängt alles von der Anzahl der Versuche ab; je höher diese Anzahl sein kann, umso genauer kann der errechnete empirische Wert der Wahrscheinlichkeit an den theoretischen Wert heranrücken.

Für alle derartigen Größen arbeitet die Statistik mit sogenannten Wahrscheinlichkeitsverteilungen für kontinuierliche Variablen, von denen Excel mehrere zur Verfügung stellt.

7.3.3 Untersuchung von Stichproben

Bei der Untersuchung von Stichproben stellen sich meist zwei Fragen: Was hat die Stichprobe ergeben, und welche Schlüsse erlaubt sie auf die Grundgesamtheit? Für die erste Frage gibt es zunächst zwei Größen: den Mittelwert und die Streuung. Als Maße sind hier eigentlich nur zwei gebräuchlich: das arithmetische Mittel und die Standardabweichung (oder das Quadrat der Standardabweichung, die Varianz). Aus diesen beiden Größen lassen sich dann auch die entsprechenden Parameter der Grundgesamtheit schätzen, wobei die Schätzung umso verlässlicher wird, je größer die Stichprobe ist, vergleiche hierzu `MITTELWERT()`, `VAR.S()` und `STABW.S()`.

7.3.4 Berechnung der Standardabweichung bei Testergebnissen

Bei der Auswertung von Testergebnissen stellt sich regelmäßig die Frage, welche durchschnittlichen Werte zustande kommen und wie groß die Streuung ist. Hier ein einfaches Beispiel: ein Test, an dem zahlreiche Personen beteiligt sind und der im Ergebnis verschiedene Punktwerte geliefert hat. Die Punktwerte sind in der Spalte B aufgelistet.

Liegen die gesamten Werte vor, kann leicht der arithmetische Mittelwert berechnet werden. Sie benutzen die Funktion `MITTELWERT()` und geben als Argumente die Adressen des Bereichs mit den Punktwerten an. Ein anderer mittlerer Wert ist der `MEDIAN()`, der auf dieselbe Weise errechnet werden kann. Das Ergebnis ist hier 1.208 und ist so zu verstehen, dass in diesem Beispiel genau die Hälfte der Werte über 1.208 und die andere Hälfte unter 1.208 liegt.

Der Mittelwert selbst sagt aber noch nichts über die Streuung der Ergebnisse aus. Bei gleichem Mittelwert können ja die Werte eng um den Mittelpunkt herumliegen oder auch ziemlich weit davon entfernt.

In der Spalte C sind hier zur Veranschaulichung die Differenzen zum Mittelwert berechnet worden, einfach durch Subtraktion des Einzelwerts vom Mittelwert. Wird von diesen Differenzen der Mittelwert gebildet, zeigt sich, dass sich die negativen und positiven Abweichungen aufheben. Das Ergebnis ist also nicht aussagekräftig.

Verbessert werden kann die Situation, wenn wie in Spalte D mit Hilfe der Funktion `ABS()` die absolute Differenz zum Mittelwert gebildet wird. Das Ergebnis in Zelle D17 ist die durchschnittliche Abweichung.

F5 =STABW.N(B5:B16)

	A	B	C	D	E	F
1						
2	**Berechnung der Standardabweichung und der Varianz bei Testergebnissen**					
3						
4	Test 1	Punkte	Abweichung vom Mittelwert	Abweichung absolut	Quadrate der Abweichung	Standardabweichung
5	Testperson 1	970	153	153	23285	248
6	Testperson 2	1389	-267	267	71096	Varianzen
7	Testperson 3	878	244	244	59647	61727
8	Testperson 4	673	450	450	202073	
9	Testperson 5	700	423	423	178587	
10	Testperson 6	1258	-135	135	18226	
11	Testperson 7	1108	15	15	227	
12	Testperson 8	1302	-179	179	32084	
13	Testperson 9	1350	-227	227	51740	
14	Testperson 10	1158	-35	35	1235	
15	Testperson 11	1395	-272	272	74248	
16	Testperson 12	1291	-168	168	28273	
17	Mittelwert	1123	0	214	61727	
18	Median	1208				

Abbildung 7.3 Berechnung der Standardabweichung bei Testergebnissen

Diese einfache Berechnung der Abweichung hat aber Nachteile, wenn wenige extrem große oder extrem kleine Werte vorkommen. Deshalb wird bei den Funktionen für die

Berechnung der Varianz und der Standardabweichung mit den Quadraten der Abweichung gearbeitet. Dies sorgt dafür, dass die großen Abweichungen stärker ins Gewicht fallen als die kleinen. Die Varianz ist gleich dem Mittelwert der Quadrate der Abweichung. Die Formel in Excel 2013 heißt:

```
=VAR.P(B5:B16)
```

Die Varianz ist in der deskriptiven Statistik ein Maß für die Streuungsbreite.

Zur Ermittlung der Standardabweichung wird dann wieder die Wurzel aus der Varianz gezogen.

```
=STABW.N(B5:B16)
```


Varianz und Standardabweichung lassen sich wie die Mittelwertberechnung nur bei metrischen Werten anwenden. Wie Sie an den Ergebnissen sehen können, ist die Standardabweichung in diesem Beispiel doch deutlich höher als die mittlere Abweichung.

7.4 Korrelation

Eine andere Fragestellung bei einer Stichprobe ist, ob die ermittelten Werte einer bestimmten Gesetzmäßigkeit gehorchen. Soll etwa untersucht werden, ob es einen Zusammenhang zwischen dem persönlichen Einkommen und der Größe des genutzten Wohnraums gibt, dann ist anzunehmen, dass eine Beziehung besteht: je mehr Einkommen, umso mehr Quadratmeter. Maße für diese Beziehung sind der Korrelationskoeffizient (`KORREL()`) und die Kovarianz (`KOVARIANZ.S()`), die Angaben darüber liefern, ob und wie stark die Daten zusammenhängen.

Der Nachweis der Korrelation darf allerdings nicht verwechselt werden mit einem Nachweis eines kausalen Zusammenhangs. Zwei Wertereihen können eine sehr hohe Korrelation haben, die aber durch einen ganz anderen Faktor verursacht werden mag, der sich auf beide auswirkt.

7.5 Regressionsanalyse

Kann vermutet werden, dass der Zusammenhang zwischen verschiedenen Faktoren linear oder exponential ist, dann lässt sich dieser Zusammenhang häufig durch eine Regressionsanalyse klären. Dabei geht es darum, ein mathematisches Modell zu finden, das den Zusammenhang zwischen verschiedenen Variablen näherungsweise beschreibt.

Die durch ein solches Modell gelieferten Daten lassen sich anschließend mit den tatsächlich zugänglichen Daten vergleichen, beispielsweise mit den im letzten Abschnitt angesprochenen Korrelationsfunktionen, um abzuschätzen, wie gut das Modell mit der Realität übereinstimmt. Auf der Basis solcher Vergleiche lassen sich dann mit guten Gründen auch Voraussagen über zukünftige Entwicklungen treffen.

Für die Regressionsanalyse stehen mehrere Verfahren zur Verfügung sowie eine ganze Reihe von Funktionen: TREND(), VARIANZ(), ACHSENABSCHNITT(), STEIGUNG(), BESTIMMTHEISMASS(), SCHÄTZER(), sowie die Funktionen RGP() und RKP(), die zahlreiche Kenngrößen für eine statistische Auswertung liefern.

7.5.1 Ein Beispiel für lineare Regression

Die einfachste Methode ist die lineare Regression. Sie kann immer angewendet werden, wenn zwischen einer unabhängigen Variablen x und einer abhängigen Variablen y eine Beziehung besteht, die sich näherungsweise durch eine lineare Gleichung in der folgenden Form beschreiben lässt:

```
y = mx + b
```

Dabei steht b für den Ordinatenabschnitt, also den Startpunkt auf der y-Achse, wenn die Daten in einem X/Y-Diagramm geplottet werden, während m die Steigung der Gerade bestimmt, die die Beziehung der beiden Variablen darstellt. Dabei wird meist davon ausgegangen, dass das Auftreten der mit den beiden Variablen bezeichneten Merkmale im Zeitablauf beobachtet wird. Wenn x bekannt ist, kann daraus y abgeleitet werden. Für die Bestimmung von b kann auch die Funktion ACHSENABSCHNITT() verwendet werden, für die Bestimmung von m die Funktion STEIGUNG(). Die Funktion RGP() liefert beide Werte gleichzeitig.

Die Regressionsanalyse geht davon aus, dass zwischen den Variablen eine gerichtete Beziehung besteht, der Wert von x also den Wert von y auf irgendeine Weise bestimmt. Daraus kann aber nicht gleich auch ein kausaler Zusammenhang abgeleitet werden, denn die Veränderung zweier Merkmale im Zeitablauf kann ganz andere Ursachen haben oder auch ganz unbekannt sein.

In vielen Fällen wird die Veränderung eines Merkmals y im Zeitablauf auch nicht nur mit der Veränderung eines Merkmals x zusammenhängen, sondern von mehreren unabhängigen Merkmalen abhängig sein. Eine solche multiple Regressionsanalyse wird von den Funktionen, die Excel hier anbietet, ebenfalls unterstützt.

Ein einfaches Beispiel soll zunächst die Arbeitsweise der Regressionsanalyse veranschaulichen. Die in Abbildung 7.4 gezeigte Tabelle listet für 1½ Jahre die monatlichen Ausgaben für Werbung und die monatlichen Umsätze auf. Es ist zu vermuten, dass die Höhe der Werbeausgaben die Höhe der Umsätze positiv beeinflusst. Der Werbeaufwand wird dabei als die unabhängige Variable, der Umsatz als die abhängige Variable betrachtet.

Einen ersten Aufschluss, ob ein Zusammenhang besteht, kann ein X/Y-Diagramm der Tabellenwerte liefern. Dazu wird der Bereich B4 bis C22 markiert und über **Einfügen • Diagramme** ein Punktdiagramm eingefügt.

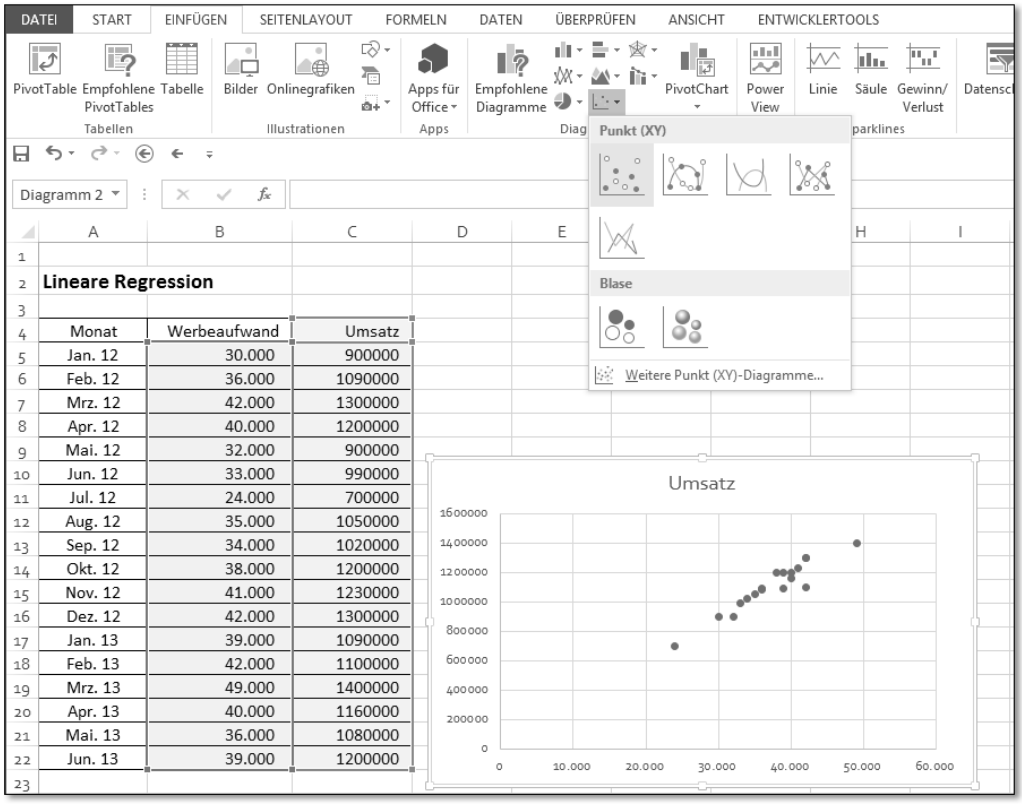


Abbildung 7.4 X/Y-Diagramm zum Zusammenhang zwischen Werbeaufwand und Umsatz

Es ist hier sinnvoll, anschließend die Minimalwerte für beide Achsen so zu ändern, dass sie nahe der untersten Werte für den Werbeaufwand und den Umsatz liegen. Das Diagramm zeigt mit der so erzeugten Punktwolke eine Verteilung, die einen linearen Zusammenhang nahelegt. Das wird noch deutlicher, wenn Sie über das Plussymbol des

Diagramms eine lineare Trendlinie einfügen und anschließend über den Aufgabenbereich **Trendlinie formatieren** die entsprechenden Formeln anzeigen lassen. Die Optionen dafür sind **Formel im Diagramm anzeigen** und **Bestimmtheitsmaß im Diagramm darstellen**.

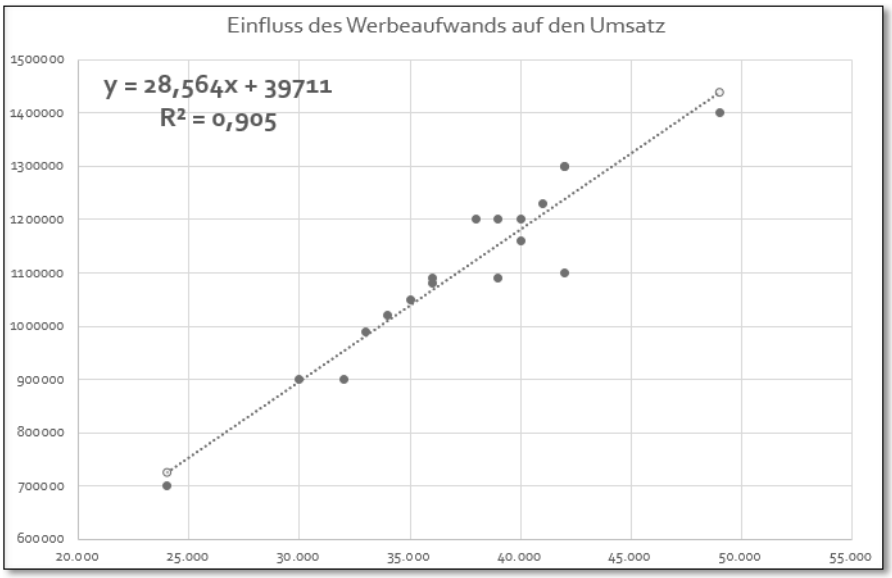


Abbildung 7.5 Punktwolke und lineare Trendlinie im überarbeiteten Diagramm

Die in das Diagramm eingefügte Formel liefert die Werte, die Excel aus den vorhandenen Werten für `m` und `b` berechnet und für die Zeichnung der linearen Trendlinie in die oben beschriebene Gleichung eingesetzt hat:

```
b = 39711
m = 28,564
```

Der erste Wert ist der Startwert für den Trend, der zweite Wert gibt an, um wie viel der Werbeaufwand je Monat steigt. Diese Werte entsprechen dem Ergebnis der Funktionen `ACHSENABSCHNITT()` und `STEIGUNG()`.

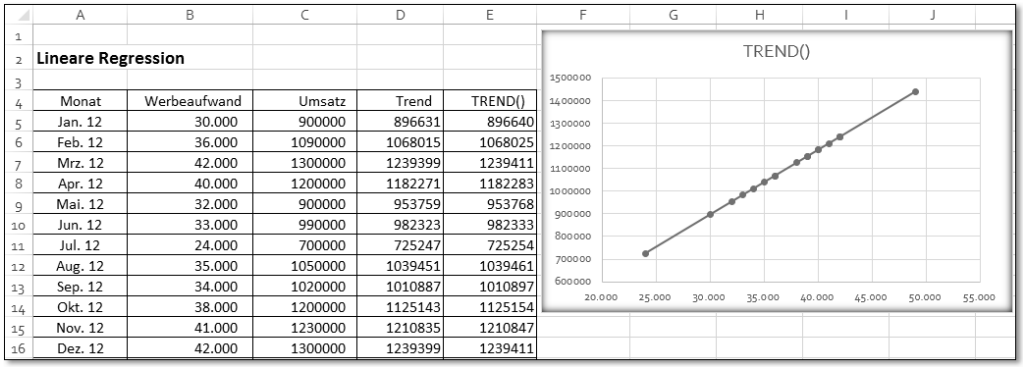
Der Wert, der für `R2` in das Diagramm eingefügt wird, ist der Wert für das Bestimmtheitsmaß. Er kann separat auch für die Funktion `BESTIMMTHEITSMASS()` ermittelt werden oder als einer der Werte, die die Funktion `RGP()` liefert. Dieser Wert gibt an, wie stark der Wert der Variablen `y`, hier die Umsatzwerte, durch den Wert der Variablen `x`, hier die Werte für den Werbeaufwand, beeinflusst wird. Der hier ausgegebene Wert von 0,905 liegt nahe bei der 1, was auf eine starke Bestimmtheit schließen lässt.

7.5.2 Vergleich der realen und der generierten Werte

In der Beispieltabelle kann neben der Spalte mit den tatsächlichen Umsätzen eine Spalte eingefügt werden, die mit der im Diagramm angezeigten Regressionsformel arbeitet. Die erste Formel in D5 ist dann:

```
=28,564*B5+39711
```

Wenn Sie die Formel nach unten kopieren, erhalten Sie die Trendwerte für den gesamten Zeitraum. In einem X/Y-Diagramm zu diesen Werten liegen dann alle Punkte aus dieser Spalte genau auf der Linie, die in Abbildung 7.5 per Aufgabenbereich als Trendlinie in das Diagramm eingefügt worden ist.



Lineare Regression

Monat	Werbeaufwand	Umsatz	Trend	TREND()
Jan. 12	30.000	900000	896631	896640
Feb. 12	36.000	1090000	1068015	1068025
Mrz. 12	42.000	1300000	1239399	1239411
Apr. 12	40.000	1200000	1182271	1182283
Mai. 12	32.000	900000	953759	953768
Jun. 12	33.000	990000	982323	982333
Jul. 12	24.000	700000	725247	725254
Aug. 12	35.000	1050000	1039451	1039461
Sep. 12	34.000	1020000	1010887	1010897
Okt. 12	38.000	1200000	1125143	1125154
Nov. 12	41.000	1230000	1210835	1210847
Dez. 12	42.000	1300000	1239399	1239411

Abbildung 7.6 Trendberechnung per Formel oder mit der Funktion `TREND()`

Die mit dieser Formel berechneten Werte entsprechen ziemlich gut den Werten, die Sie auch mit der Funktion `TREND()` erzeugen können, die Excel für die lineare Regression anbietet. In diesem Fall gilt die Formel:

```
=TREND($C$5:$C$22;$B$5:$B$22;;WAHR)
```

Diese Formel wird als Matrixformel für den gesamten Bereich angelegt, muss also mit Strg + ⇧ + ↵ beendet werden. Die leichten Abweichungen gegenüber den Werten in der Spalte D kommen durch die Rundung der im Diagramm angezeigten Werte für `b` und `m` zustande.

7.5.3 Andere Methoden der Regression

Wenn sich herausstellt, dass die realen Daten schlecht zu einer linearen Trendlinie passen, die Punkte im Diagramm also zu sehr streuen, können Sie versuchen, ob eine an-

dere Regressionsmethode zu besseren Ergebnissen führt. Als ersten Schritt können Sie wieder versuchen, zu der Punktwolke in einem entsprechenden X/Y-Diagramm eine andere Trendlinie hinzuzufügen, um zu sehen, ob sie besser zu den vorhandenen Daten passt. Excel bietet im Aufgabenbereich **Trendlinie formatieren** sechs verschiedene Linien an.

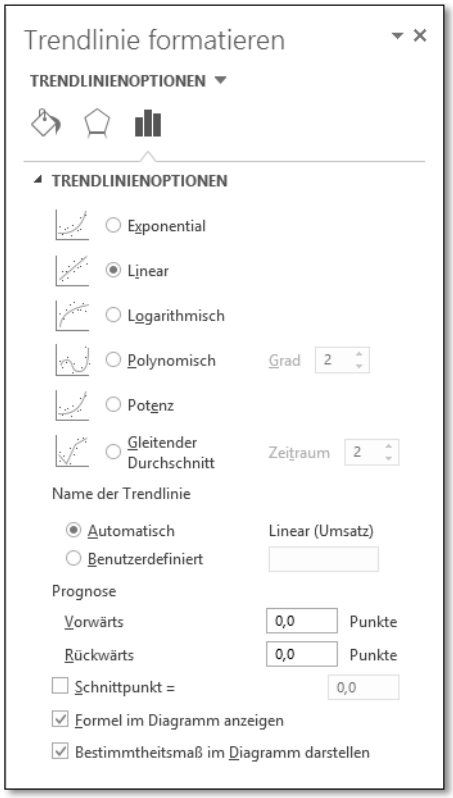


Abbildung 7.7 Der Aufgabenbereich für Trendlinien in einem Diagramm

7.6 Statistische Tests

Die Aufgabe statistischer Tests ist, ganz allgemein gesprochen, festzustellen, mit welcher Sicherheit oder Unsicherheit von Werten einer Stichprobe auf Werte der Grundgesamtheit geschlossen werden kann. Hierfür gibt es seit Excel 2007 zwei Funktionsgruppen. Die eine bietet die Möglichkeit, direkt Tests anhand von Stichproben durchzuführen, die andere liefert Werte aus Wahrscheinlichkeitsverteilungen, anhand deren aus den Stichproben gewonnene Parameter überprüft werden können.

Beim t-Test (T.TEST()) wird beispielsweise die Frage geprüft, ob zwei Stichproben sich in ihrem Mittelwert nur zufällig unterscheiden (dann wären beide Stichproben derselben Grundgesamtheit zufällig entnommen) oder ob sie sich nicht zufällig unterscheiden (dann stammen sie entweder aus verschiedenen Grundgesamtheiten oder sind nicht zufällig entnommen). Hier liefert die Funktion T.TEST() direkt einen Wahrscheinlichkeitswert.

Mit dem F-Test (F.TEST()) wird geprüft, ob zwei Stichproben sich in ihrer Varianz zufällig unterscheiden oder nicht. Auch hier ist wieder der Umweg über die F-Verteilung (F.VERT()) gangbar und bei manchen Fragestellungen notwendig.

Der Chi-Test (CHIQU.TEST()) schließlich dient der Überprüfung der Frage, ob eine Stichprobe, mit der mehrere Werte erfasst sind, mit einer Grundgesamtheit übereinstimmt, aus der für diese Werte Erwartungswahrscheinlichkeiten bekannt sind. Auch hier steht zusätzlich die Verteilungsfunktion zur Verfügung.

7.7 Verteilungsfunktionen

Von Zufallsgrößen war oben schon die Rede. Es wurde die Frage gestellt, mit welcher Wahrscheinlichkeit ein bestimmtes Ereignis auftritt. Für die Beantwortung dieser Frage steht eine Anzahl von Funktionen zur Verfügung, die das sonst notwendige Nachschlagen in umfangreichen statistischen Tabellenwerken ersparen kann.

Gemäß der Unterscheidung in diskrete und stetige Zufallsgrößen lassen sich auch die zugehörigen Verteilungen in diskrete und stetige unterscheiden. Hier kurz ein Überblick mit einigen Hinweisen zur Anwendung.

7.7.1 Diskrete Verteilungen

Binomialverteilung: Gehört zu den diskreten Verteilungen. Grundlage ist ein Ereignis, das jeweils mit einer bestimmten Wahrscheinlichkeit eintreten kann oder aber nicht. Die einzelnen Versuche müssen dabei voneinander unabhängig sein. Beispiele hierfür sind Münzwürfe, Würfeln oder Kartenziehen, wobei aber nach jedem Versuch die Karte anschließend zurückgesteckt werden muss; dadurch wird der Ausgangszustand immer wieder hergestellt. Siehe BINOM.VERT().

Negative Binomialverteilung: Wahrscheinlichkeitsverteilung für die Anzahl der erforderlichen Versuche, um eine vorgegebene Anzahl von Erfolgen zu erzielen. Siehe dazu NEGBINOM.VERT().

Hypergeometrische Verteilung: Wird bei einem Beispiel wie Kartenziehen die Karte nicht zurückgesteckt, dann ändert sich beim nächsten Versuch die theoretische Wahrscheinlichkeit. In solchen Fällen wird die hypergeometrische Verteilung benutzt. Siehe HYPGEOM.VERT().

Poisson-Verteilung: Diese diskrete Verteilung wird normalerweise als Näherung für die Binomialverteilung bei sehr großen Zahlen und sehr kleinen Wahrscheinlichkeiten genommen. Da Excel 2013 aber genauso gut mit der Binomialverteilung rechnen kann, ist dieser Ausweg nicht unbedingt erforderlich. Siehe POISSON.VERT().

7.7.2 Stetige Verteilungen

Betaverteilung: Dies ist eine stetige Verteilung über dem Intervall [0,1]. Sie kann bei Berechnung der Verteilung von Größen aus beliebigen, gleichmäßig stetig verteilten Grundgesamtheiten verwendet werden, beispielsweise um die Streuung von Stichproben zu vergleichen. Mit den beiden Formparametern lässt sich die Verteilung an den Rändern modellieren. Siehe BETA.VERT().

Chi-Quadrat-Verteilung: Diese Verteilung ist definiert als die Verteilung der Summe der Quadrate von n unabhängigen und standardnormalverteilten Zufallsvariablen. Sie wird insbesondere für Streuungstests verwendet. Siehe CHIQU.VERT()und CHIQU.VERT.RE().

Exponentialverteilung: Die Exponentialverteilung ist ein Spezialfall der Gammaverteilung für p = 1. Siehe EXPON.VERT().

F-Verteilung: Die F-Verteilung – das F erinnert an den britischen Statistiker Fisher – steht in Zusammenhang mit der Chi-Quadrat-Verteilung. Wird der Quotient aus zwei durch die Zahl der Freiheitsgrade dividierten Chi-Quadrat-Variablen gebildet, entsteht eine Variable, die F-verteilt ist. Diese Werte werden für Testverfahren verwendet, bei denen die Varianz zweier unabhängiger normalverteilter Merkmale verglichen werden soll. Siehe F.VERT()und F.VERT.RE().

Gammaverteilung: Eine kontinuierliche Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen, die beispielsweise in der Warteschlangentheorie verwendet wird. Siehe GAMMA.VERT().

Logarithmische Normalverteilung: Eine kontinuierliche Verteilung über der Menge der reellen Zahlen für Zufallsvariablen, wenn ln(x) normalverteilt ist. Siehe dazu LOGNORM.VERT().

Normalverteilung: Diese kontinuierliche Verteilung ist definiert für Werte zwischen – und +. In zahlreichen Fällen, wo einer Zufallsvariablen eine Grundgesamtheit zugrunde liegt, die sehr groß ist (ab 1.000), und wo eine stetige Größe gemessen wird, können Sie davon ausgehen, dass sie wenigstens annäherungsweise normalverteilt ist.

Daraus folgt, dass die Normalverteilung im Prinzip die wichtigste der stetigen Verteilungen ist. Sie wird auch die Gaußsche Verteilung oder Gaußsche Fehlerkurve genannt. Eine besondere Stellung nimmt hier die Standardnormalverteilung ein, bei der der Mittelwert 0 und die Standardabweichung 1 ist. Siehe `NORM.VERT()` und `NORM.S.VERT()`.

Student-t-Verteilung: Diese Verteilung ist eine stetige Verteilungsfunktion, die 1908 von dem englischen Statistiker Gosset unter dem Pseudonym »Student« publiziert worden ist. Sie ähnelt der Normalverteilung. Je umfangreicher der Stichprobenumfang ist, umso mehr nähert sich diese Verteilung der Standardnormalverteilung an. Die t-Verteilung wird häufig verwendet, wenn Mittelwerte verglichen werden und die Varianz dabei geschätzt werden muss. Siehe `T.VERT()`, `T.VERT.2S()` und `T.VERT.RE()`.

Weibull-Verteilung: Diese Verteilung wird gerne zur Beschreibung der Ausfallraten von Produkten in der Qualitätssicherung verwendet. Siehe `WEIBULL.VERT()`.

7.7.3 Dichtefunktion und Verteilungsfunktion

Bei allen diskreten Verteilungen mit einem bestimmten Wertebereich kann jedem möglichen Elementarereignis `i` eine ganz bestimmte Wahrscheinlichkeit zugeordnet werden. Der entsprechende Wert wird durch die Dichtefunktion geliefert. Die Wahrscheinlichkeit, dass bei einem Versuch ein Ausgang `x < i` beobachtet wird, wird durch die Verteilungsfunktion geliefert.

Bei stetigen Verteilungen ist es nicht möglich, einfach einem bestimmten Wert `x` eine Wahrscheinlichkeit zuzuordnen. Es ist ja die Besonderheit der stetigen Verteilung, dass selbst in dem kleinsten Intervall immer noch unendlich viele `x`-Werte zu finden sind, so dass die Wahrscheinlichkeit für einen einzelnen `x`-Wert null ist. Eine Wahrscheinlichkeit, die ungleich null ist, kann deshalb nur für bestimmte Intervalle existieren.

Den Wahrscheinlichkeitsverteilungen liegt deshalb mathematisch immer eine Dichtefunktion zugrunde, bei der Normalverteilung etwa die berühmte Glockenkurve. Der jeweilige `y`-Wert sagt hier noch nichts über die Wahrscheinlichkeit des dazugehörenden `x`-Werts aus. Erst die Fläche zwischen zwei `x`-Werten (mathematisch das bestimmte Integral) ist ein Maß für die Wahrscheinlichkeit. In Abbildung 7.8 liefert die Fläche unter der Glocke vom linken Ende bis zu der senkrechten Linie bei `x1` die Wahrscheinlichkeit

eines Werts bis zu x1, die Fläche zwischen x1 und x2 die Wahrscheinlichkeit eines Werts in diesem Bereich.

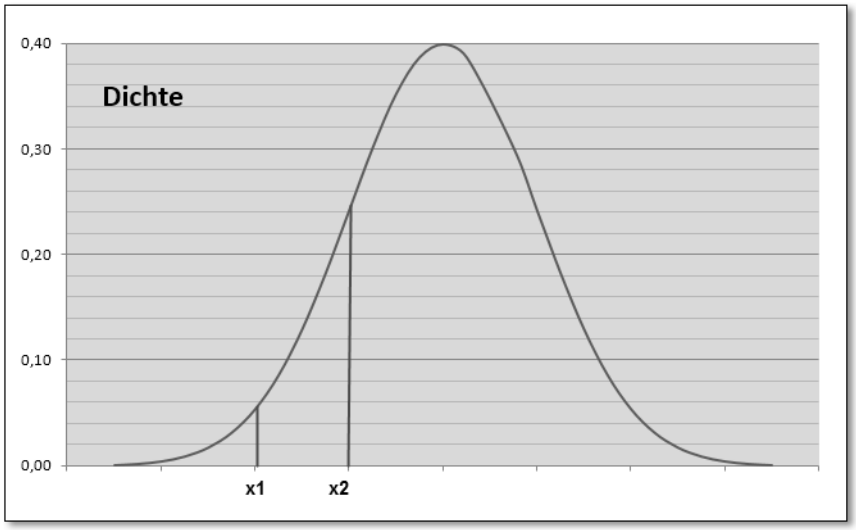


Abbildung 7.8 Wahrscheinlichkeit als Flächenanteil einer Verteilung

In Anlehnung an die diskreten Verteilungen verwendet Excel 2013 bei den stetigen Verteilungsfunktionen einen Wahrheitswert Kumuliert, der die Funktionen veranlasst, entweder die Dichte (Kumuliert = FALSCH) oder die kumulierte Wahrscheinlichkeit (Kumuliert = WAHR) zu berechnen; das Erste liefert also den Wert der Dichtefunktion, das Zweite den Wert der Verteilungsfunktion.

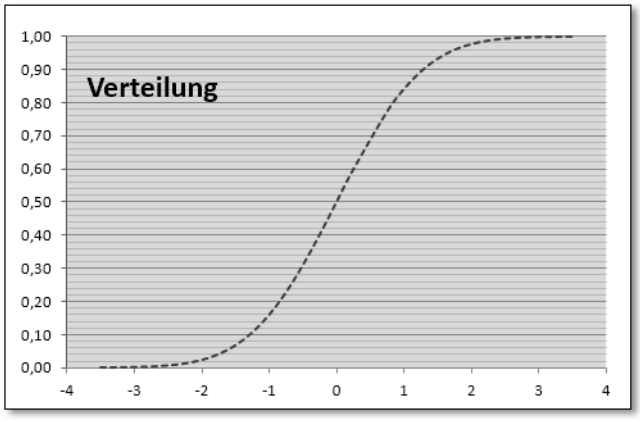


Abbildung 7.9 Die Verteilungsfunktion liefert die kumulierten Wahrscheinlichkeitswerte.

7.7.4 Umkehrfunktionen

Für die meisten Verteilungsfunktionen, die alle mit dem Suffix `VERT` gekennzeichnet sind – `NORM.VERT()`, `LOGNORM.VERT()` etc. –, stehen obendrein inverse Funktionen zur Verfügung, die alle das Suffix `INV` verwenden, etwa `NORM.INV()`, `LOG.INV()` etc. Das Verhältnis der entsprechenden Funktionen zueinander ist folgendes:

- Die Verteilungsfunktion liefert die Wahrscheinlichkeit dafür, dass eine Zufallsvariable einen Wert gleich oder kleiner als einen vorgegebenen Wert annimmt, der mit dem Argument `x` angegeben wird.
- Die inverse Funktion liefert zu einer angegebenen Wahrscheinlichkeit den Wert, der gleich oder kleiner dem der Zufallsvariable mit der angegebenen Wahrscheinlichkeit ist. Da dieser Wert Quantil genannt wird, lässt sich der Zusammenhang auch so angeben:

  `...VERT(q) = p`
  `...INV(p) = q`

  mit `p` = Wahrscheinlichkeit und `q` = Quantil.

7.8 Neue und geänderte Funktionen

Wie im Vorwort schon angesprochen, wurden mit der Version 2010 zahlreiche Änderungen an der Funktionsbibliothek vorgenommen. Die Mehrzahl der Änderungen und Neuerungen betrifft dabei die Gruppe der statistischen Funktionen.

7.8.1 Benennungsschema

Zunächst wurde ein neues Benennungsschema eingeführt, um die verschiedenen Funktionstypen deutlicher zu kennzeichnen. Dazu wurden zusammengesetzte Namen eingeführt, bei denen die einzelnen Bestandteile durch Punkte getrennt sind. Die Funktion `FVERT()` beispielsweise heißt jetzt `F.VERT()`. Das Suffix `VERT` kennzeichnet die Funktion als Verteilungsfunktion. Ähnlich wird das Suffix `INV` für inverse Funktionen verwendet. `F.INV()` ersetzt `FINV()`. Soll die Funktion nicht linksseitig, sondern rechtsseitig verwendet werden, wird noch ein zweites Suffix `RE` angehängt: `F.VERT.RE()` und `F.INV.RE()`. Die bisherige Funktion `FVERT()` wird also durch zwei Funktionen ersetzt, `F.VERT()` und `F.VERT.RE()`, wobei das Ergebnis von `F.VERT.RE()` dem Ergebnis der alten Funktion entspricht.

Die neuen Funktionen mit dem Suffix VERT haben jetzt alle ein Argument Kumuliert, dessen Wert den Typ der Funktion bestimmt: WAHR liefert einen Wert der Verteilungsfunktion, FALSCH einen Wert der Dichtefunktion.

Bei zweiseitigen Funktionen wird 2S angehängt. Das Suffix P – für Population –, beispielsweise in KOVARIANZ.P(), gibt an, dass sich die Funktion auf die Grundgesamtheit bezieht, Suffix S – für sample – wird für Funktionen verwendet, die sich auf eine Stichprobe beziehen. Einige der umbenannten Funktionen haben zugleich zusätzliche Parameter erhalten.

Wird eine Funktion mit einem solchen Namen in einem VBA-Makro als Methode des WorksheetFunction-Objekts verwendet, wird der Punkt übrigens durch einen Unterstrich ersetzt, also etwa Beta_Dist() für den englischen Funktionsnamen BETA.DIST().

In Tabelle 7.1 sind die umbenannten statistischen Funktionen den alten Namen gegenübergestellt, die jetzt in der Kategorie **Kompatibilität** geparkt sind.

Umbenannte Funktion	Kompatible Funktion
BETA.INV()	BETAINV()
BETA.VERT()	BETAVERT()
BINOM.INV()	KRITBINOM()
BINOM.VERT()	BINOMVERT()
CHIQU.INV()	CHIINV()
CHIQU.INV.RE()	CHIINV()
CHIQU.TEST()	CHITEST()
CHIQU.VERT()	CHIVERT()
CHIQU.VERT.RE()	CHIVERT()
EXPON.VERT()	EXPONVERT()
F.INV()	FINV()
F.INV.RE()	FINV()
F.TEST()	FTEST()
F.VERT()	FVERT()

Tabelle 7.1 Gegenüberstellung der neueren und der kompatiblen Funktionen

Umbenannte Funktion	Kompatible Funktion
F.VERT.RE()	FVERT()
G.TEST()	GTEST()
GAMMA.INV()	GAMMAINV()
GAMMA.VERT()	GAMMAVERT()
HYPGEOM.VERT()	HYPGEOMVERT()
KONFIDENZ.NORM()	KONFIDENZ()
KONFIDENZ.T()	KONFIDENZ()
KOVARIANZ.P()	KOVAR()
KOVARIANZ.S()	KOVAR()
LOGNORM.INV()	LOGINV()
LOGNORM.VERT()	LOGNORMVERT()
MODUS.EINF()	MODALWERT()
MODUS.VIELF()	MODALWERT()
NEGBINOM.VERT()	NEGBINOMVERT()
NORM.INV()	NORMINV()
NORM.S.INV()	STANDNORMINV()
NORM.S.VERT()	STANDNORMVERT()
NORM.VERT()	NORMVERT()
POISSON.VERT()	POISSON()
QUANTIL.EXKL()	QUANTIL()
QUANTIL.INKL()	QUANTIL()
QUANTILSRANG.EXKL()	QUANTILSRANG()
QUANTILSRANG.INKL()	QUANTILSRANG()
QUARTILE.EXKL()	QUARTILE()
QUARTILE.INKL()	QUARTILE()

Tabelle 7.1 Gegenüberstellung der neueren und der kompatiblen Funktionen (Forts.)

Umbenannte Funktion	Kompatible Funktion
RANG.GLEICH()	RANG()
STABW.N()	STABWN()
STABW.S()	STABW()
T.INV()	TINV()
T.INV.2S()	TINV()
T.TEST()	TTEST()
T.VERT()	TVERT()
T.VERT.2S()	TVERT()
T.VERT.RE()	TVERT()
VAR.P()	VARIANZEN()
VAR.S()	VARIANZ()
WEIBULL.VERT()	WEIBULL()

Tabelle 7.1 Gegenüberstellung der neueren und der kompatiblen Funktionen (Forts.)

7.8.2 Funktionen für Kompatibilität

Damit es keine Probleme mit älteren Arbeitsmappen gibt, bleiben die umbenannten Funktionen weiterhin auch unter ihrem alten Namen verfügbar. Im Dialog **Funktion einfügen** sind diese Funktionen in der Kategorie **Kompatibilität** zu finden. Die Beschreibung dieser Funktionen finden Sie im nächsten Kapitel. Wenn Sie Funktionen direkt einfügen und die Option **AutoVervollständigen-Formel** über **Datei • Optionen • Formeln** eingeschaltet lassen, werden die kompatiblen Funktionen jeweils mit einem Symbol angezeigt, das ein kleines Warnzeichen enthält.

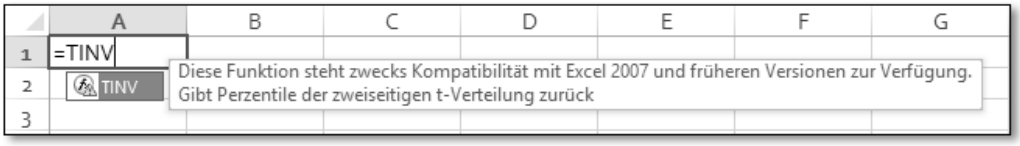


Abbildung 7.10 Hinweis bei kompatiblen Funktionen

7.8.3 Neue statistische Funktionen in Excel 2013

Excel 2013 erweitert die Reihe der statistischen Funktionen noch um einige neue Funktionen:

- `BINOM.VERT.BEREICH()` gibt die Wahrscheinlichkeit dafür an, dass die Anzahl erfolgreicher Versuche in einem bestimmten Bereich liegt.
- `GAMMA()` liefert direkt den Wert der Gammafunktion für die mit `x` angegebene Variable. Die Funktion erspart Ihnen also das Ablesen der Werte aus Tabellen zur Gammafunktion.
- `GAUSS()` berechnet die Wahrscheinlichkeit, dass ein Element zwischen zwei Standardabweichungen vom Mittelwert liegt.
- `PHI()` liefert direkt den Wert der Dichtefunktion für eine Standardnormalverteilung.
- `SCHIEFE.P()` liefert ein Maß für die Asymmetrie der Häufigkeitsverteilung einer Grundgesamtheit.
- `VARIATIONEN2()` berechnet die Zahl der möglichen Anordnungen von Elementen, wenn Wiederholungen zugelassen sind.

7.9 Referenz der statistischen Funktionen

ACHSENABSCHNITT()

INTERCEPT()

Syntax:	`ACHSENABSCHNITT(`**`Y_Werte`**`; `**`X_Werte`**`)`
Beispiel:	`=ACHSENABSCHNITT(B7:B16;A7:A16)`
Ergebnis:	`2,8` (im abgebildeten Beispiel)

Die Funktion `ACHSENABSCHNITT()` liefert den Ordinatenabschnitt für den Schnittpunkt der aus den Argumenten `Y_Werte` und `X_Werte` errechneten Regressionsgeraden oder Trendlinie einer linearen Regressionsfunktion mit der y-Achse, vergleiche hierzu die Funktion `RGP()`.

Für `Y_Werte` und `X_Werte` kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden. Dabei stellen die `X_Werte` die unabhängigen Daten oder Messwerte dar, die `Y_Werte` dagegen die abhängigen. Die Funktion ermittelt also den Wert, den die abhängige Variable annimmt, wenn die unabhängige Variable den Wert `0` hat.

Die Argumente `Y_Werte` und `X_Werte` müssen jeweils die gleiche Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Nicht numerische x-Werte ergeben den Fehler `#WERT!`.

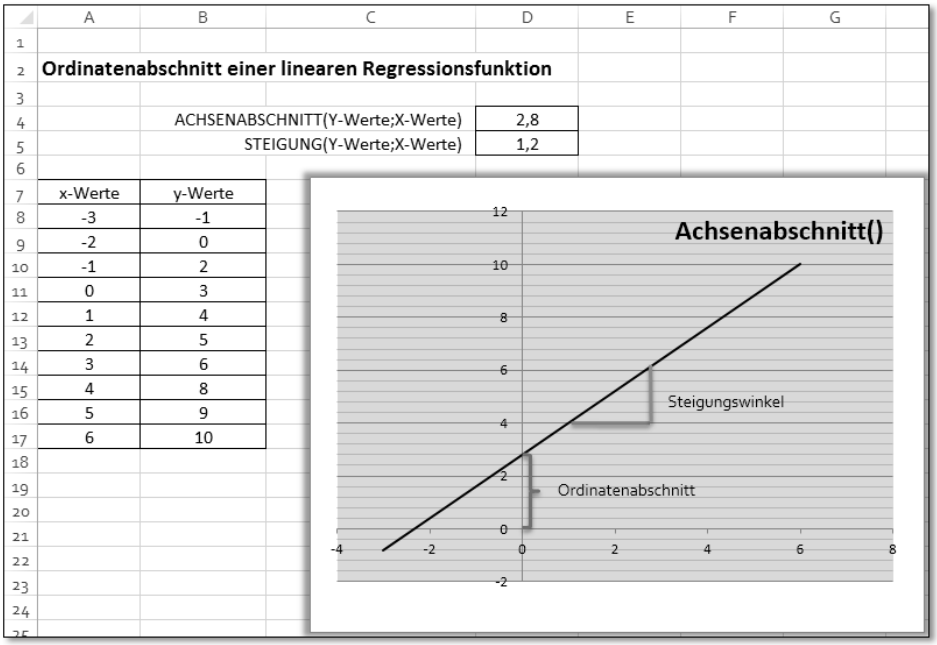


Abbildung 7.11 Ordinatenabschnitt und Steigung einer Regressionsgeraden

Die Funktionsgleichung für die Regressionsgerade lautet:

```
y = b + m * x
```

wobei `b` der Schnittpunkt mit der y-Achse ist.

Siehe auch die Erläuterungen zur Regressionsanalyse in Abschnitt 7.5, »Regressionsanalyse«.

ANZAHL()

COUNT()

Syntax: `ANZAHL(`**`Wert1`**`; Wert2; ...)`

Beispiel: `=ANZAHL(1;4;7;"fünf")`

Ergebnis: `3`

Die Funktion ANZAHL() ergibt die Anzahl der numerischen Werte, die in der Argumentliste bzw. in Bereichen, auf die sie sich beziehen, enthalten sind. Als numerische Werte gelten Zahlen einschließlich der Null sowie Datums- und Zeitwerte.

	A	B	C
1			
2	**Die Anzahl vorhandener Werte berechnen**		
3			
4		Besucher vormittags	Besucher nachmittags
5	06.03.2013	200	200
6	07.03.2013	300	300
7	08.03.2013	320	unbekannt
8	09.03.2013		
9	10.03.2013		200
10	11.03.2013	440	440
11	12.03.2013	240	240
12	13.03.2013	800	800
13	14.03.2013	240	240
14	15.03.2013	300	300
15	16.03.2013		
16	17.03.2013		
17	18.03.2013	260	260
18	19.03.2013	480	480
19	20.03.2013	260	260
20	21.03.2013	#NV	#NV
21	22.03.2013		
22	ANZAHL()	11	11
23	ANZAHL2()	12	13
24	ANZAHLLEEREZELLEN()	5	4

Abbildung 7.12 Während die Funktion ANZAHL() nur die numerischen Werte zählt, berücksichtigt ANZAHL2 alle nicht leeren Zellen.

Seit Excel 2007 sind insgesamt 255 Argumente möglich, in den älteren Versionen nur 30. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen nicht mitgezählt. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie mitgezählt:

```
=ANZAHL(1;4;7;"5";WAHR)
```

ergibt also 5.

Die Funktion kann auch auf Matrixkonstanten angewendet werden. Im Beispiel in Abbildung 7.12 wird mit Hilfe von ANZAHL() festgestellt, für wie viele Tage die Besucherzahlen eingetragen sind.

Die Funktion ANZAHL() übernimmt bei statistischen Auswertungen gewissermaßen die Basisaufgabe: das Zählen der Größe einer Grundgesamtheit oder einer Stichprobe.

Wenn Sie die Anzahl der numerischen Werte in einem markierten Zellbereich sofort sehen wollen, können Sie diesen Wert auch in der Statuszeile anzeigen lassen. Das Kontextmenü der Statusleiste bietet dazu die Option **Numerische Zahl** an.

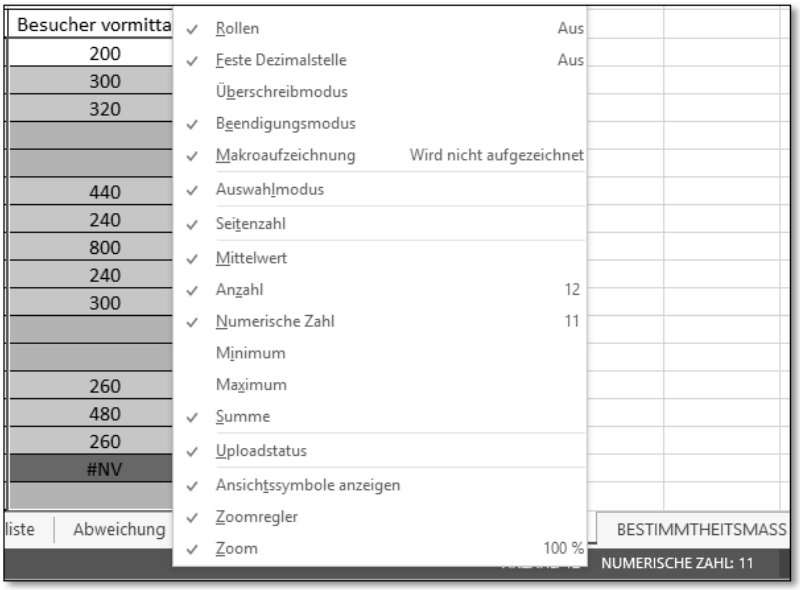


Abbildung 7.13 Anzeige der ausgewählten numerischen Werte in der Statusleiste

ANZAHL2()

COUNTA()

Syntax: ANZAHL2(**Wert1**; Wert2; ...)
Beispiel: =ANZAHL2(1;4;7;"fünf")
Ergebnis: 4

Anders als die Funktion ANZAHL(), die nur die numerischen Werte berücksichtigt, ermittelt ANZAHL2() die Werte insgesamt, die in der Argumentliste bzw. in angegebenen Bereichen oder Matrixkonstanten enthalten sind. Texteinträge – auch leere wie "" –, Wahrheitswerte und Fehlerwerte werden mitgezählt, nur wirklich leere Zellen (oder Argumente) werden abgezogen. Seit Excel 2007 sind insgesamt 255 Argumente möglich,

in den älteren Versionen nur 30. Das erste Argument ist erforderlich. Die Funktion kann auch auf Matrixkonstanten angewendet werden.

Die Funktion ist besonders praktisch, wenn bei Beschriftungsspalten oder -zeilen festgestellt werden soll, wie viele Einträge aktuell vorhanden sind. Mit Bezug auf eine Namensliste ermittelt die Funktion beispielsweise sofort, wie viele Personen, Kunden etc. vorkommen, siehe Abbildung 7.14.

	A	B
1		
2	**Zählen der Elemente einer Liste**	
3		
4	Name	Lieblingssport
5	Hans	Fußball
6	Werner	Tennis
7	Lotte	Tennis
8	Lisa	Joggen
9	Franz	Joggen
10	Rita	Turnen
11	Henning	Fußball
12	Wanda	Tennis
13	Lori	Tennis
14	Bella	Tennis
15	Arne	Joggen
16	Lukas	Fußball
17	ANZAHL2()	12
18		

Abbildung 7.14 Zählen der Elemente einer Liste

Soll ermittelt werden, wie viele Einträge in einem Bereich keine numerischen Werte enthalten, kann einfach die Differenz der beiden Funktionen gebildet werden:

```
=ANZAHL2(B5:B21)-ANZAHL(B5:B21)
```

Ähnlich kann verfahren werden, wenn geprüft werden soll, ob für alle Werte in Spalte A bereits Werte in Spalte B eingetragen sind:

```
=ANZAHL2(A5:A21)-ANZAHL(B5:B21)
```

In dem Beispiel in Abbildung 7.12 zählt `ANZAHL2()` auch die Tage mit, bei denen etwas anderes als eine Zahl eingetragen ist.

Die Funktionen `ANZAHL()` und `ANZAHL2()` sind nicht nur statistisch verwendbar, sie lassen sich auch als Zähler für VBA-Makros einsetzen, wenn es beispielsweise darum geht, alle Einträge einer Spalte oder Zeile in einer Schleife abzuarbeiten.

7.9.1 Beispiel Rückstandsberechnung

Angenommen, Sie planen, eine bestimmte Arbeit einigermaßen gleichmäßig auf einen bestimmten Zeitraum zu verteilen. Die Arbeit ist in Stückzahlen messbar. Dann können Sie dafür eine Tabelle verwenden, die mit den Funktionen SUMME(), ANZAHL() und ANZAHL2() arbeitet. Etwas knifflig an der Lösung ist nur der Umgang mit den Bezügen, die mal relativ und mal absolut sein müssen.

Die Zelle B4 in der in Abbildung 7.15 gezeigten Tabelle wird verwendet, um die geplante Gesamtmenge einzugeben, die innerhalb des vorgegebenen Zeitraums produziert werden soll. In der ersten Spalte sind die Arbeitstage durchnummeriert. In der zweiten Spalte ist das Datum des ersten Tages eingegeben. Wenn Samstag und Sonntag nicht gearbeitet werden soll, können Sie das Ausfüllkästchen mit gedrückter rechter Maustaste nach unten ziehen und im Menü den Befehl **Wochentage ausfüllen** verwenden. In die Spalte C sollen dann Tag für Tag die an dem jeweiligen Tag erledigten Mengen eingetragen werden. In Spalte D wird geprüft, ob das geplante Soll erfüllt, überschritten oder unterschritten ist. Ist das Soll überschritten, haben Sie einen Vorsprung, der als positiver Wert ausgegeben wird, liegen Sie dagegen zurück, wird ein negativer Wert angezeigt. Bei Nullwerten liegen Sie genau im Plan.

Die Formel, die in allen Zellen verwendet wird, rechnet immer die insgesamt erledigte Menge zusammen und vergleicht sie mit der Menge, die bis zu dem gerade erreichten Tag erledigt sein müsste. Die Formel soll aber nur dann einen Wert ausrechnen, wenn in der Spalte C in der entsprechenden Zeile etwas eingetragen ist. Deshalb ist die gesamte Berechnung noch mal in eine WENN()-Funktion eingepackt, die prüft, ob an dem Tag schon eine Menge eingegeben worden ist. Solange das nicht der Fall ist, bleibt die Zelle in Spalte D leer.

Die Berechnung des Rückstands oder Vorsprungs geschieht mit folgender Formel:

```
=SUMME($C$5:C5)-(ANZAHL($C$5:C5)*$B$4/ANZAHL2($A$5:$A$19))
```

Wie Sie sehen, ist in den beiden ersten Funktionen das erste Argument jeweils absolut, das zweite relativ. Der Summenbereich wächst also jeden Tag um eine Position, wenn die Formel nach unten kopiert wird.

Von dem Ergebnis der Summenfunktion, die immer die aufgelaufene Gesamtmenge liefert, werden die aufgelaufenen Planmengen abgezogen, die durch den Rest der Formel ermittelt werden.

Mit der ANZAHL2()-Funktion wird festgestellt, wie viele Werte die Spalte mit den Tagesnummern insgesamt enthält. Sie könnten hier natürlich auch direkt die Zahl der Tage eingeben.

Eingepackt in die WENN()-Funktion, sieht die Formel folgendermaßen aus:

```
=WENN(C5>0; SUMME($C$5:C5)-(ANZAHL($C$5:C5)*$B$4/ANZAHL2($A$5:$A$19));"")
```

Abbildung 7.15 zeigt die Tabelle mit einigen Werten für die ersten Tage.

	A	B	C	D
1				
2	**Berechnen von Rückstand und Vorsprung**			
3				
4	Geplante Gesamtmenge:	300	Erledigte Menge	Rückstand/-Vorsprung
5	Tag 1	10. Okt	20	0
6	Tag 2	11. Okt	32	12
7	Tag 3	12. Okt	32	24
8	Tag 4	13. Okt	5	9
9	Tag 5	16. Okt	3	-8
10	Tag 6	17. Okt	1	-27
11	Tag 7	18. Okt		
12	Tag 8	19. Okt		
13	Tag 9	20. Okt		
14	Tag 10	23. Okt		
15	Tag 11	24. Okt		
16	Tag 12	25. Okt		
17	Tag 13	26. Okt		
18	Tag 14	27. Okt		
19	Tag 15	30. Okt		

Abbildung 7.15 Berechnen von Rückstand oder Vorsprung

Wenn Sie sich im Verlauf der Arbeit dazu entschließen, die geplante Gesamtmenge zu erhöhen, werden die Werte sofort an die neue Anforderung angepasst.

Das Beispiel kann leicht an beliebige Zeiträume angepasst werden. Sie müssen dann nur die letzte Adresse in der Formel entsprechend ändern. Anstelle eines Tagesintervalls kann das Ganze natürlich auch etwa für Stundenintervalle verwendet werden.

ANZAHLLEEREZELLEN()

COUNTBLANK()

Syntax: ANZAHLLEEREZELLEN(**Bereich**)

Beispiel: =ANZAHLLEEREZELLEN(A1:A100)

Ergebnis: 5, wenn der Bereich 5 leere Zellen enthält

Die Funktion ANZAHLLEEREZELLEN() ermittelt die Anzahl der leeren Zellen im angegebenen Bereich. Zellen, die Leerzeichen enthalten, gelten dabei nicht als leer.

Dagegen wird eine leere Zeichenfolge als leer gezählt, anders also als es bei der Funktion ANZAHL2() der Fall ist. Abbildung 7.16 zeigt eine Tabelle, in der in Spalte B die Werktage mit 1 gekennzeichnet sind, der Samstag und der Sonntag dagegen mit einer leeren Zeichenfolge. Dazu wird folgende Formel verwendet:

```
=WENN(ODER(WOCHENTAG(A5)=7;WOCHENTAG(A5)=1);"";1)
```

Folglich stehen in den Zellen B8 und B9 leere Zeichenfolgen. Die beiden Formeln in B10 und B11 zeigen, dass

```
=ANZAHL2(B5:B9)
```

die Zellen mit den leeren Zeichenfolgen mitzählt, während

```
=ANZAHLLEEREZELLEN(B5:B9)
```

angibt, dass in dem Bereich zwei Zellen leer sind.

	A	B
1		
2	**Die Anzahl leerer Zellen berechnen**	
3		
4		**=WENN(ODER(WOCHENTAG(A5)=7; WOCHENTAG(A5)=1);"";1)**
5	Mittwoch, 6. März 2013	1
6	Donnerstag, 7. März 2013	1
7	Freitag, 8. März 2013	1
8	Samstag, 9. März 2013	
9	Sonntag, 10. März 2013	
10	ANZAHL2()	5
11	ANZAHLLEEREZELLEN()	2

Abbildung 7.16 Unterschiedliche Bewertung von Zellen mit leeren Zeichenfolgen

Die Funktion kann beispielsweise dazu verwendet werden, um festzustellen, wie viele Werte in eine Tabellenspalte noch nicht eingetragen sind. Das ist bei großen Tabellen unter Umständen eine wichtige Information, um den Aufwand abzuschätzen, der für die Dateneingabe noch ansteht.

BESTIMMTHEITSMASS()

RSQ()

Syntax: BESTIMMTHEITSMASS(**Y_Werte**; **X_Werte**)
Beispiel: =BESTIMMTHEITSMASS({3;5;8;7};{5;6;5;4})
Ergebnis: 0,13559

Mit der Funktion BESTIMMHEITSMASS() wird der Wert für r2, das Quadrat des Pearsonschen Korrelationskoeffizienten, ermittelt, vergleiche PEARSON(). Das Bestimmtheitsmaß ist auch als Determinationskoeffizient bekannt und ist ein Maß für die Güte der Anpassung, die eine Regression erzielt, d.h., es ist ein Maß dafür, wie dicht die Datenpunkte an der Regressionsgeraden liegen.

Für Y_Werte und X_Werte kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden. Texte, leere Zellen oder Wahrheitswerte werden ignoriert. Dabei stellen die X_Werte die unabhängigen Daten oder Messwerte dar, die Y_Werte dagegen die abhängigen. Der ermittelte Wert r2 kann als der Anteil der Varianz von Y interpretiert werden, der durch die Varianz von X erklärt wird.

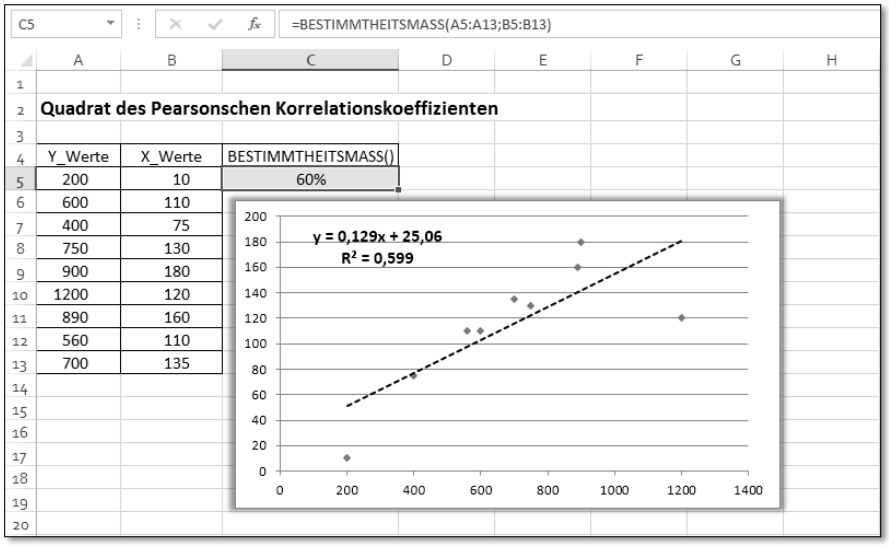


Abbildung 7.17 Das Bestimmtheitsmaß lässt sich auch im Diagramm ausgeben.

Enthalten Y_Werte und X_Werte unterschiedlich viele Datenelemente, liefert die Funktion den Fehler #NV. Enthalten beide nur ein oder gar kein Datenelement, erscheint der Fehler #DIV/0!.

Die Funktion liefert Werte von 0 bis +1. 1 bedeutet, dass sich die Werte der abhängigen Variablen allein durch die Werte der unabhängigen Variablen erklären lassen. In diesem Fall liegen alle Datenpunkte direkt auf der Regressionsgeraden. Andere Faktoren spielen also keine Rolle. Ein Wert von 0,13 bedeutet, dass sich der Wert der abhängigen Variablen nur zu 13% durch die unabhängige Variable erklären lässt, der Rest deutet auf den Einfluss anderer Faktoren.

Siehe dazu auch den Abschnitt 7.5, »Regressionsanalyse«.

BETA.INV()

BETA.INV()

Syntax:	BETA.INV(**Wahrsch**; **Alpha**; **Beta**; A; B)
Beispiel:	=BETA.INV(0,1; 3; 4)
Ergebnis:	0,2009

Die Funktion BETA.INV() liefert Quantile der Betaverteilung für eine Zufallsvariable und ist die Umkehrung von BETA.VERT(). Die Funktion ersetzt seit Excel 2010 die bisherige Funktion BETAINV().

Als notwendige Argumente sind mit Wahrsch die Wahrscheinlichkeit und mit Alpha und Beta Parameter der Verteilung einzutragen. A und B sind optionale Argumente, die die Intervallgrenzen für x bezeichnen. Werden sie nicht angegeben, dann wird A = 0 gesetzt und B = 1. In diesem Fall ist die Betaverteilung eine stetige Verteilung über dem Intervall [0,1]. Vergleiche BETA.VERT().

Die Funktion BETA.INV() ist die Umkehrung der Funktion BETA.VERT(). Wenn:

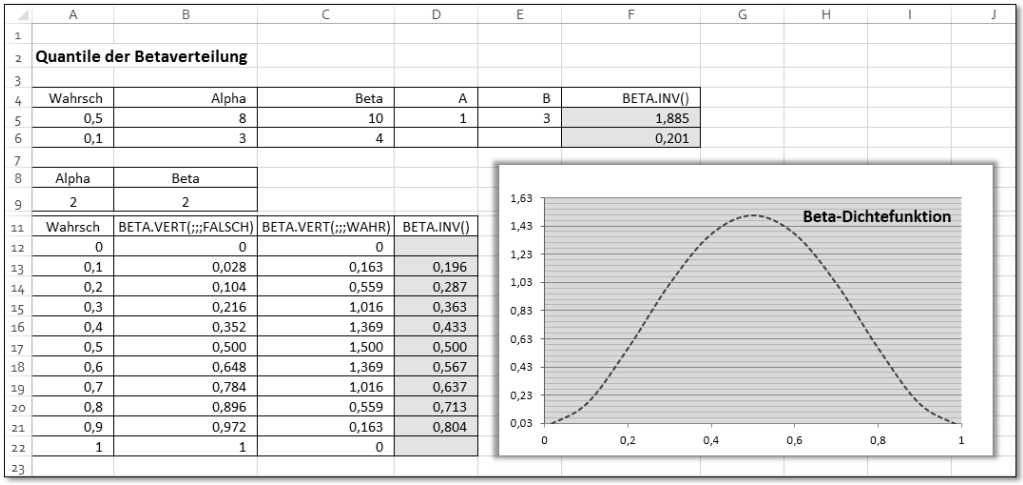
Wahrsch = BETA.VERT(x; ... WAHR)

dann ist:

x = BETA.INV(Wahrsch; ...)

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Alpha ≤ 0 oder Beta ≤ 0, erscheint der Fehler #ZAHL!. Das gilt auch, wenn Wahrsch ≤ 0 oder > 1.

Abbildung 7.18 zeigt einige Beispiele für von der Funktion berechnete Werte und den Graphen der Dichtefunktion der Betaverteilung.



Quantile der Betaverteilung

Wahrsch	Alpha	Beta	A	B	BETA.INV()
0,5	8	10	1	3	1,885
0,1	3	4			0,201

Alpha	Beta
2	2

Wahrsch	BETA.VERT(;;;FALSCH)	BETA.VERT(;;;WAHR)	BETA.INV()
0	0	0	
0,1	0,028	0,163	0,196
0,2	0,104	0,559	0,287
0,3	0,216	1,016	0,363
0,4	0,352	1,369	0,433
0,5	0,500	1,500	0,500
0,6	0,648	1,369	0,567
0,7	0,784	1,016	0,637
0,8	0,896	0,559	0,713
0,9	0,972	0,163	0,804
1	1	0	

Abbildung 7.18 Die Betaverteilung und der Graph der Dichtefunktion

Siehe auch `BETA.VERT()`.

BETA.VERT()

BETA.DIST()

Syntax: `BETA.VERT(`**`x`**`;` **`Alpha`**`;` **`Beta`**`;` **`Kumuliert`**`; A; B)`

Beispiel: `=BETA.VERT(0,5; 3; 4; Wahr)`

Ergebnis: `0,65625`

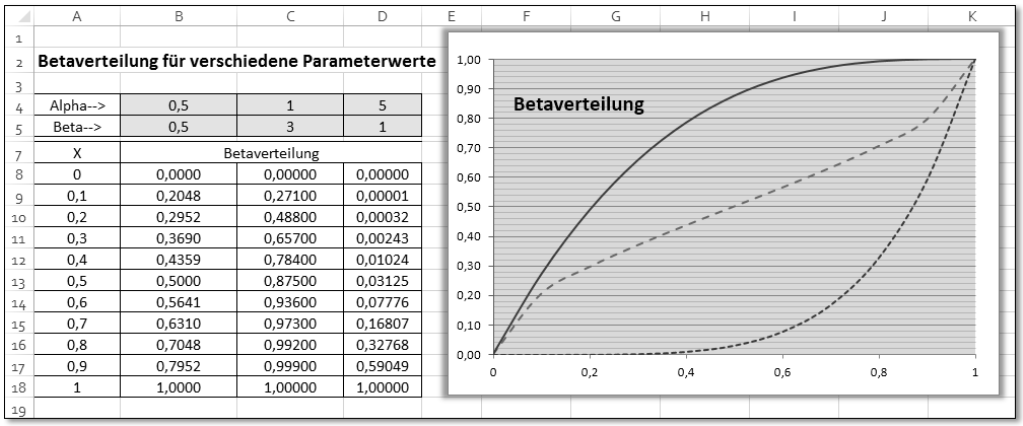
Die Funktion `BETA.VERT()` liefert die Wahrscheinlichkeitsverteilung für eine betaverteilte Zufallsvariable. Die Betaverteilung ist eine stetige Verteilung über dem Intervall `[0,1]` oder in der allgemeineren Version über dem Intervall `[A,B]`. Sie wird eingesetzt, um Ereignisse zu modellieren, die innerhalb eines Intervalls stattfinden, das durch einen minimalen und einen maximalen Wert bestimmt ist. Dadurch eignet sich die Funktion beispielsweise für die Projektplanung oder für Kontrollsysteme zu zeitkritischen Abläufen.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `BETAVERT()`, wobei `Kumuliert` als zusätzliches Argument eingebaut ist.

Es wird berechnet, mit welcher Wahrscheinlichkeit die Zufallsvariable einen Wert zwischen `A` und `x` annimmt. Das Argument `x` ist die Größe der Zufallsvariablen im Intervall

A bis B, Alpha und Beta – beide müssen größer als 0 sein – sind Parameter der Verteilung. Die Verteilung ist für Alpha = Beta symmetrisch um x = ½. (In der Literatur werden normalerweise die Bezeichnungen p und q verwendet.)

Das neue Argument Kumuliert ist ein Wahrheitswert, der den Typ der Funktion bestimmt. Mit Kumuliert = WAHR liefert die Funktion den Wert der kumulierten Verteilungsfunktion, andernfalls den Wert der Dichtefunktion.



Betaverteilung für verschiedene Parameterwerte

Alpha-->	0,5	1	5
Beta-->	0,5	3	1

X	Betaverteilung		
0	0,0000	0,00000	0,00000
0,1	0,2048	0,27100	0,00001
0,2	0,2952	0,48800	0,00032
0,3	0,3690	0,65700	0,00243
0,4	0,4359	0,78400	0,01024
0,5	0,5000	0,87500	0,03125
0,6	0,5641	0,93600	0,07776
0,7	0,6310	0,97300	0,16807
0,8	0,7048	0,99200	0,32768
0,9	0,7952	0,99900	0,59049
1	1,0000	1,00000	1,00000

Abbildung 7.19 Einfluss der Parameter Alpha und Beta auf den Verlauf der Betaverteilung

A und B sind optionale Argumente und bezeichnen die untere und obere Grenze des Intervalls. Werden für A und B keine Werte angegeben, dann gilt die standardmäßige Betaverteilung über dem Intervall [0,1], also A = 0 und B = 1.

Ist eines der Argumente x, Alpha, Beta, A oder B nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Alpha ≤ 0 oder Beta ≤ 0, erscheint der Fehler #ZAHL!. Das gilt auch, wenn x < A, oder x > B oder A = B.

Abbildung 7.19 zeigt den Einfluss verschiedener Parameter auf den Verlauf der Funktion.

Die Betafunktion, auch Eulersches Integral erster Gattung genannt, ist eine Funktion mit zwei Parametern, die durch folgende Formel definiert ist:

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$$

Siehe auch BETA.INV().

BINOM.INV()

BINOM.INV()

Syntax: `BINOM.INV(Versuche; Erfolgswahrsch; Alpha)`
Beispiel: `=BINOM.INV(200; 0,9; 0,01)`
Ergebnis: `170`

Die Funktion `BINOM.INV()` liefert die kleinste Anzahl erfolgreicher Versuche, für die die kumulierte Wahrscheinlichkeit größer oder gleich der mit `Alpha` angegebenen Irrtums- oder Grenzwahrscheinlichkeit ist. Voraussetzung ist, dass die Zufallsgröße binomialverteilt ist. Das bedeutet, die Versuche sind voneinander unabhängig, und es gibt immer nur zwei mögliche Ergebnisse.

Mit `Versuche` wird die Zahl der Versuche angegeben, mit `Erfolgswahrsch` die Wahrscheinlichkeit für den erfolgreichen Ausgang eines Versuchs.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Versuche` < `0` oder `Erfolgswahrsch` < `0` oder > `1`, erscheint der Fehler `#ZAHL!`; das gilt auch für `Alpha` < `0` oder > `1`.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `KRITBINOM()`.

Das Ergebnis der Funktion kann als Akzeptanzkriterium verwendet werden, um z. B. zu entscheiden, ob die Fehlerrate in einem Fertigungslos noch geduldet werden kann oder nicht. Bei dieser Fragestellung gibt es nur zwei mögliche Ergebnisse: Ein Prüfling ist in Ordnung oder nicht in Ordnung. Die Betrachtung der einzelnen Prüflinge geschieht unabhängig voneinander, also sind die Bedingungen für eine Binomialverteilung gegeben.

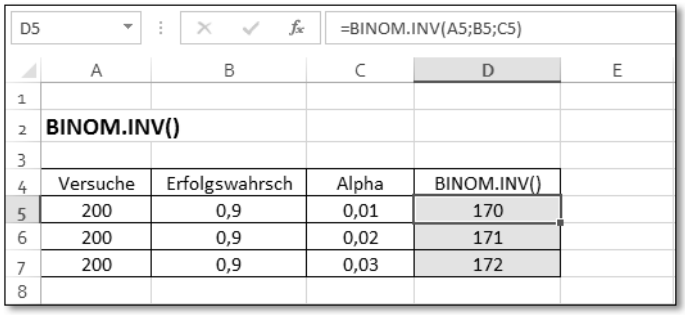


Abbildung 7.20 Berechnen der kritischen Grenze bei binomialen Verteilungen

Im Beispiel in Abbildung 7.20 wird nun angenommen, dass bei einer gegebenen Maschineneinstellung von 200 Prüflingen im Durchschnitt 180 (= 90%) korrekt sind, die

Wahrscheinlichkeit für einen korrekten Prüfling also 0,9 ist. Die Fragestellung lautet: Mit wie vielen korrekten Prüflingen können Sie mit einer Irrtumswahrscheinlichkeit von 0,01 mindestens rechnen? Das Ergebnis lautet 170, d.h., in 99% aller 200-Stück-Lieferungen sind 170 korrekte Produkte zu erwarten.

Die Funktion BINOM.INV() ist die Umkehrung der Funktion BINOM.VERT(). Siehe auch BINOM.VERT() und die in Excel 2013 neue Funktion BINOM.VERT.BEREICH().

BINOM.VERT()

BINOM.DIST()

Syntax:	BINOM.VERT(**Zahl_Erfolge**; **Versuche**; **Erfolgswahrsch**; **Kumuliert**)
Beispiel:	=BINOM.VERT(3; 10; 1/6; FALSCH)
Ergebnis:	15,5%

Die Funktion BINOM.VERT() gibt die Wahrscheinlichkeit dafür an, dass bei alternativen diskreten Versuchsergebnissen bei einer mit Versuche angegebenen Anzahl von Versuchen ein bestimmtes Ergebnis mit einer durch Zahl_Erfolge angegebenen Häufigkeit auftritt. Die (vorweg ermittelte) Wahrscheinlichkeit für das Einzelergebnis wird mit Erfolgswahrsch (zwischen 0 und 1) angegeben.

Kumuliert verlangt einen Wahrheitswert und beschreibt den Typ der Funktion. WAHR liefert Werte der Verteilungsfunktion, FALSCH Werte der Dichtefunktion.

Zahl_Erfolge muss ≥ 0 und ≤ Versuche sein, ansonsten gibt die Funktion den Fehlerwert #ZAHL! zurück. Das gilt auch für Erfolgswahrsch < 0 oder > 1. Ist eines der Argumente Zahl_Erfolge, Versuche, Erfolgswahrsch nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Die Werte der Dichtefunktion der Binomialverteilung lassen sich mit folgender Formel nachrechnen:

```
= KOMBINATIONEN(Versuche; Zahl_Erfolge) * Erfolgswahrsch^Zahl_Erfolge *
(1 - Erfolgswahrsch) ^ (Versuche - Erfolge)
```


Beispiele für die Anwendung der Funktion sind Münzwürfe (Erfolgswahrscheinlichkeit 1/2), Würfel (1/6) oder Kartenziehen (1/32), wobei aber nach jedem Versuch die Karte anschließend zurückgesteckt werden muss, also jedes Mal der Ausgangszustand wieder hergestellt wird.

	A	B	C	D	E	F
1						
2	**Wahrscheinlichkeit einer binomialverteilten Zufallsvariablen**					
3						
4	Zahl_Erfolge	Versuche	Erfolgswahrsch	Kumuliert	BINOM.VERT()	
5	40	100	1/2	WAHR	2,8%	
6	40	100	1/2	FALSCH	1,1%	
7	20	100	1/4	WAHR	14,9%	
8	3	10	1/6	FALSCH	15,5%	
9						

Abbildung 7.21 Berechnungen mit der Funktion `BINOM.VERT()`

Das oben angeführte Beispiel liefert die Wahrscheinlichkeit dafür, dass bei zehn Würfen mit einem Würfel genau dreimal die Sechs gewürfelt wird. Wird `Kumuliert` dagegen mit `WAHR` belegt, wird die Wahrscheinlichkeit berechnet, dass die Sechs bis zu dreimal gewürfelt wird.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `BINOMVERT()`.

7.9.2 Anteile von Merkmalen ermitteln

Das folgende Beispiel geht von der Fragestellung aus, wie viele Menschen aus einer Zufallsgruppe von 10 Personen weiblich sind, wobei unterstellt wird, dass die Wahrscheinlichkeit, dass eine einzelne Person eine Frau ist, genau 50% beträgt. In Zelle B17 wird nun beispielsweise berechnet, wie hoch die Wahrscheinlichkeit ist, dass in der Zehnergruppe 3 Frauen sind. Sie liegt bei 11,7%.

	A	B	C
9			
10	**Wahrscheinlichkeit von x Frauen**		
11	**in einer Zufallsgruppe von 10 Personen**		
12			
13	x	f(x)	F(x)
14	0	0,1%	0,1%
15	1	1,0%	1,1%
16	2	4,4%	5,5%
17	3	11,7%	17,2%
18	4	20,5%	37,7%
19	5	24,6%	62,3%
20	6	20,5%	82,8%
21	7	11,7%	94,5%
22	8	4,4%	98,9%
23	9	1,0%	99,9%
24	10	0,1%	100,0%
25			

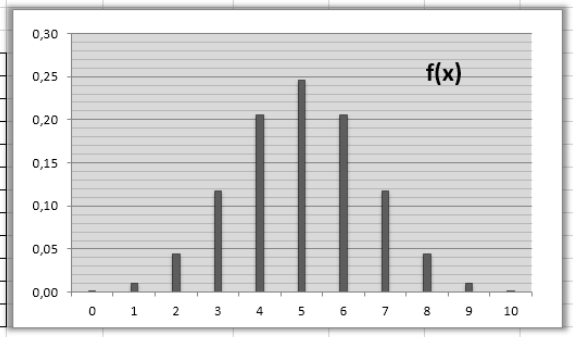


Abbildung 7.22 Die Binomialverteilung und der Graph der Dichtefunktion

Der Graph bildet die Werte von f(x) für x von 0 bis 10 ab. Die Summe der Werte muss immer 1 betragen. Diese Funktion wird als Dichtefunktion bezeichnet. In der Spalte C werden die kumulierten Wahrscheinlichkeiten angezeigt. Die Wahrscheinlichkeit, dass 0 bis 10 Personen Frauen sind, muss natürlich ebenfalls 1 sein. Die Funktion zu Spalte C wird als Verteilungsfunktion bezeichnet.

Siehe auch BINOM.INV() und die in Excel 2013 neue Funktion BINOM.VERT.BEREICH().

BINOM.VERT.BEREICH()

BINOM.DIST.RANGE()

Syntax: BINOM.VERT.BEREICH(**Versuche**; **Erfolgswahrsch**; **Zahl_Erfolge**; Zahl2_Erfolge)

Beispiel: =BINOM.VERT.BEREICH(60; 0,65; 40; 45)

Ergebnis: 0,415

Die in Excel 2013 neu eingeführte Funktion BINOM.VERT.BEREICH() gibt die Wahrscheinlichkeit dafür an, dass bei alternativen diskreten Versuchsergebnissen bei einer mit Versuche angegebenen Anzahl von Versuchen ein bestimmtes Ergebnis mit einer durch Zahl_Erfolge angegebenen Häufigkeit auftritt. Wird das optionale Argument Zahl2_Erfolge angegeben, errechnet die Funktion die Wahrscheinlichkeit, dass die Anzahl der erfolgreichen Versuche zwischen Zahl_Erfolge und Zahl2_Erfolge liegt.

Zahl_Erfolge muss ≥ 0 und ≤ Versuche sein, Zahl2_Erfolge muss ≥ Zahl_Erfolge und ≤ Versuche sein, ansonsten gibt die Funktion den Fehlerwert #ZAHL! zurück. Ist eines der Argumente nicht numerisch, zeigt die Funktion den Fehlerwert #WERT!.

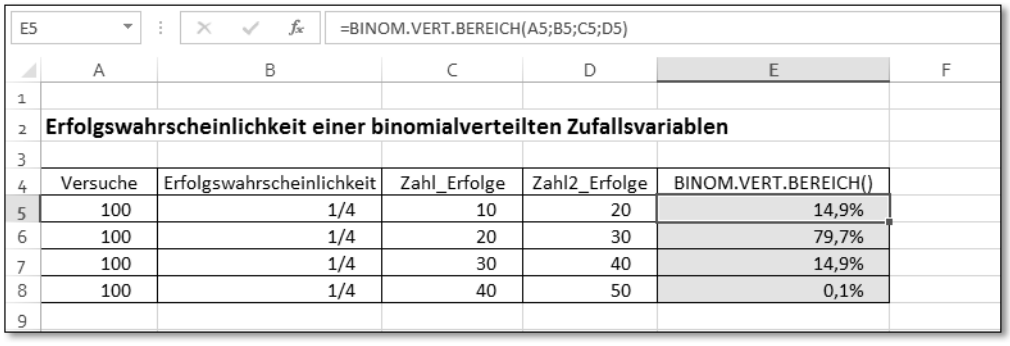


Abbildung 7.23 Berechnungen mit der Funktion BINOM.VERT.BEREICH()

Die (vorweg ermittelte) Wahrscheinlichkeit für das Einzelergebnis wird mit Erfolgswahrsch (zwischen 0 und 1) angegeben. Beispiele sind Münzwürfe (Erfolgswahrscheinlichkeit 1/2), Würfel (1/6) oder Kartenziehen (1/32), wobei aber nach jedem Versuch die Karte anschließend zurückgesteckt werden muss, also jedes Mal der Ausgangszustand wiederhergestellt wird.

Siehe auch BINOM.INV() und BINOM.VERT().

CHIQU.INV()

CHISQ.INV()

Syntax: CHIQU.INV(**Wahrsch**; **Freiheitsgrade**)
Beispiel: =CHIQU.INV(0,05; 3)
Ergebnis: 0,3518

Die Funktion CHIQU.INV() liefert die (z.B. in statistischen Tabellenwerken tabellierten) Quantile der linksseitigen Chi-Quadrat-Verteilung χ^2. Sie wird verwendet, um einen Vergleichswert zu berechnen, mit dem Hypothesen über die Übereinstimmung von beobachteten und erwarteten Ergebnissen bewertet werden können.

Für das Argument Wahrsch erwartet die Funktion Wahrscheinlichkeitswerte aus einer Chi-Quadrat-Verteilung und dazu die Anzahl der Freiheitsgrade.

Ist Wahrsch < 0 oder > 1 oder Freiheitsgrade < 1 oder > 10^10, liefert die Funktion den Fehlerwert #ZAHL!, ist eines der Argumente nicht numerisch, den Fehler #WERT!.

Die Funktion CHIQU.INV() ist zugleich die Umkehrfunktion von CHIQU.VERT(). Es gilt also: Ist:

```
w = CHIQU.INV(x;...)
```

dann ist:

```
x = CHIQU.VERT(w;...;WAHR)
```

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von CHIINV() angeboten werden. Die andere ist CHIQU.INV.RE().

Siehe auch CHIQU.INV.RE(), CHIQU.TEST(), CHIQU.VERT() und CHIQU. VERT.RE() und Abbildung 7.24.

CHIQU.INV.RE()

CHISQ.INV.RT()

Syntax:	CHIQU.INV.RE(**Wahrsch**; **Freiheitsgrade**)
Beispiel:	=CHIQU.INV.RE(0,05; 3)
Ergebnis:	7,8147

Die Funktion CHIQU.INV.RE() liefert die (z. B. in statistischen Tabellenwerken tabellierten) Quantile der rechtsseitigen Chi-Quadrat-Verteilung χ^2. Für das Argument Wahrsch erwartet die Funktion Wahrscheinlichkeitswerte aus einer Chi-Quadrat-Verteilung und dazu die Anzahl der Freiheitsgrade.

Ist Wahrsch < 0 oder > 1 oder Freiheitsgrade < 1 oder > 10^10, liefert die Funktion den Fehlerwert #ZAHL!, ist eines der Argumente nicht numerisch, den Fehler #WERT!.

Die Funktion CHIQU.INV.RE() ist zugleich die Umkehrfunktion von CHIQU. VERT.RE(). Es gilt also: Ist:

```
w = CHIQU.INV.RE(x;...)
```

dann ist:

```
CHIQU.VERT.RE(w;...) = x
```

Die Funktion sucht den Wert für x in einem iterativen Verfahren, das nach 64 Schritten abgeschlossen sein muss, andernfalls ist das Ergebnis der Fehlerwert #NV.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion CHIINV(), die ebenfalls Werte der rechtsseitigen Chi-Quadrat-Verteilung liefert.

Siehe auch CHIQU.INV(), CHIQU.TEST(),CHIQU.VERT() und CHIQU.VERT.RE()und Abbildung 7.24.

CHIQU.TEST()

CHISQ.TEST()

Syntax:	CHIQU.TEST(**Beob_Messwerte**; **Erwart_Werte**)
Beispiel:	=CHIQU.TEST({9; 11; 9; 12; 10; 9}; {10; 10; 10; 10; 10; 10})
Ergebnis:	0,977

Der Chi-Test prüft die Frage, ob eine Stichprobe, mit der mehrere Werte erfasst sind, mit einer Grundgesamtheit übereinstimmt, aus der für diese Werte Erwartungswahrscheinlichkeiten bekannt sind. Die Funktion `CHIQU.TEST()` liefert also direkt den Wahrscheinlichkeitswert für den Chi-Quadrat-Test beim Vergleich zwischen beobachteten und erwarteten Größen.

Als Argumente werden je ein Bereichsbezug oder eine Matrix für die beobachteten Werte `Beob_Messwerte` und die theoretisch erwarteten Werte `Erwart_Werte` eingetragen.

Beide Argumente müssen die gleiche Anzahl von Datenelementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Sind die Argumente nicht numerisch, erscheint der Fehler `#WERT!`.

Abbildung 7.24 zeigt ein einfaches Beispiel für ein solches Testverfahren. Es soll geprüft werden, wie sehr sich bei 60-maligem Würfeln die beobachteten Ergebnisse, die in Spalte B abgelegt sind, an die Ergebnisse anpassen, die aufgrund der theoretischen Wahrscheinlichkeit zu erwarten sind. Deshalb wird auch von Anpassungstests gesprochen. Die theoretische Wahrscheinlichkeit ergibt sich aus der Formel `60 * 1/6`, sie setzt also eine sogenannte Gleichverteilung voraus. Deshalb ist in Spalte C für alle Wurfergebnisse der Wert `10` abgelegt.

Die Nullhypothese, die durch den Chi-Quadrat-Test geprüft werden soll, lässt sich so formulieren: Die Differenzen zwischen den beobachteten und den theoretischen Häufigkeiten sind rein zufällig und nicht signifikant, die empirische Häufigkeitsverteilung passt sich in einem ausreichenden Maße der theoretischen Häufigkeitsverteilung an. Es gibt also kein Indiz dafür, dass der Würfel beispielsweise gezinkt oder defekt ist.

Die Funktion `CHIQU.TEST()` rechnet nach folgendem Verfahren: Zunächst wird für alle Variablen die Differenz zwischen dem beobachteten und dem erwarteten Ergebnis gebildet und diese dann quadriert, so dass die negativen Vorzeichen keine Rolle mehr spielen. Das Ergebnis wird jedes Mal durch den erwarteten Wert geteilt, um die Abweichungen zu relativieren:

```
(Beobachtungswert-Erwartungswert)^2/Erwartungswert
```

Aus diesen Einzelergebnissen wird die Summe ermittelt, um den Wert der chi-quadrierten Verteilung zu erhalten, der auch als *Chi-Quadrat* oder mit `u` bezeichnet wird. Dieser Wert wird als Prüfgröße verwendet. Dieser Wert würde bei einer perfekten Übereinstimmung zwischen dem erwarteten und dem beobachteten Ergebnis `0` sein. Je größer der Wert ist, umso fragwürdiger ist die Übereinstimmung.

Im letzten Schritt ermittelt die Funktion nun die Wahrscheinlichkeit dafür, dass u den errechneten Wert annimmt. In diesem Fall ergibt sich der Wert 0,98 oder 98%. Dieser Wert liegt deutlich über dem vorgegebenen Signifikanzniveau. Mit der Funktion CHIQU.INV.RE() kann zum Vergleich mit dem Prüfwert ein kritischer Wert aus der Chi-Quadrat-Verteilung berechnet werden.

```
=CHIQU.INV.RE(0,05;5)
```

ergibt mit einer Irrtumswahrscheinlichkeit von 5% und mit 5 Freiheitsgraden den Wert 11,07, der wesentlich höher als der Prüfwert ist. Der Prüfwert liegt also nicht im kritischen Bereich.

	A	B	C	D	E	F
1						
2	**Chi-Test Beispiel: Würfeleichung: 60-mal gewürfelt**					
3						
4	Wurf	Beob	Erw	Beob-Erw	(beob-erw)^2/erw	
5	1	9	10	-1	0,1	
6	2	11	10	1	0,1	
7	3	12	10	2	0,4	
8	4	9	10	-1	0,1	
9	5	10	10	0	0	
10	6	9	10	-1	0,1	
11	Summe	60	60	0	0,8	**<-- u = Summe der relativierten**
12						**quadrierten Abweichungen**
13			=CHIQU.TEST(B5:B10;C5:C10)		0,98	
14			=CHIQU.INV.RE(0,05;5)		11,07	**<-- kritischer Wert**
15			=CHIQU.VERT.RE(E11;5)		0,98	
16						

Abbildung 7.24 Beispiel für die Funktion CHIQU.TEST()

Die Nullhypothese muss nicht verworfen werden, die Abweichungen von den theoretisch erwarteten Ergebnissen können als rein zufällig eingestuft werden.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion CHITEST(). Siehe auch CHIQU.INV(), CHIQU.INV.RE(), CHIQU.VERT() und CHIQU.VERT.RE().

CHIQU.VERT()

CHISQ.DIST()

Syntax: CHIQU.VERT(**x; Freiheitsgrade; Kumuliert**)

Beispiel: =CHIQU.VERT(10; 3; WAHR)

Ergebnis: 0,98

Die Funktion CHIQU.VERT() liefert Werte der linksseitigen Verteilungsfunktion für eine mit x angegebene Chi-Quadrat-verteilte Zufallsvariable und den angegebenen Wert für Freiheitsgrade, also die Wahrscheinlichkeit für die Übereinstimmung von beobachteten und erwarteten Werten, vergleiche CHIQU.TEST().

Kumuliert bestimmt den Typ der Funktion. WAHR liefert die kumulative Verteilungsfunktion, FALSCH den Wert der Dichtefunktion.

Ist x < 0, gibt die Funktion den Fehler #ZAHL! zurück. Das gilt auch, wenn Freiheitsgrade < 1 oder > 10^10 ist. Ist eines der beiden Argumente nicht numerisch, erscheint der Fehler #WERT!.

Die Chi-Quadrat-Verteilung ist eine Wahrscheinlichkeitsverteilung, die sich über die Summe von n unabhängigen, quadrierten, standardnormalverteilten Variablen und einer Anzahl von Freiheitsgraden definiert.

Die Funktion wird für den Chi-Quadrat-Test benötigt, der beim Vergleich von empirischen zu theoretisch erwarteten Häufigkeiten zum Einsatz kommt.

Je nach Anzahl der Freiheitsgrade ändert sich der Charakter der Verteilung, mit steigender Anzahl wird die Kurve der Dichtefunktion flacher und verschiebt sich nach rechts, wie Abbildung 7.25 zeigt, die die Kurven für die Dichtefunktion mit den Freiheitsgraden 4, 10 und 20 anzeigt.

Chi-Quadrat-Verteilung für verschiedene Freiheitsgrade

	Kumuliert(Verteilung)			Nicht kumuliert(Dichte)		
FG	4	10	20	4	10	20
x						
0	0	0	0	0	0	0
1	0,09020401	0,000172116	1,70967E-10	0,151632665	0,000789753	1,63226E-09
2	0,264241118	0,003659847	1,11425E-07	0,183939721	0,007664155	5,06889E-07
3	0,4421746	0,018575936				
4	0,59399415	0,052653017				
5	0,712702505	0,108821981				
6	0,800851727	0,184736755				
7	0,864111775	0,274555047				
8	0,908421806	0,371163065				
9	0,938900519	0,467896424				
10	0,959572318	0,559506715				
11	0,973435986	0,642481998				
12	0,982648735	0,7149435				
13	0,988724206	0,776328183				
14	0,992704944	0,827008392				
15	0,995298783	0,867938144				
16	0,996980836	0,9003676				
17	0,99806705	0,92563602				
18	0,998765902	0,945036359				
19	0,999214056	0,959737318				
20	0,999500601	0,970747312				
21	0,999683331	0,978906434	0,602867401	0,000144566	0,006973068	0,058859776

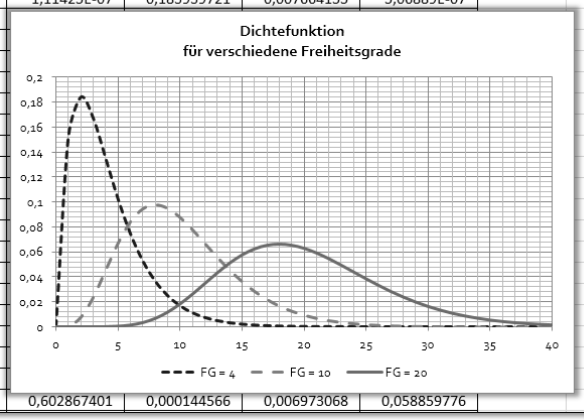


Abbildung 7.25 Die Kurven der Chi-Quadrat-Dichtefunktion für unterschiedliche Freiheitsgrade

Abbildung 7.26 zeigt den Graph der Verteilungsfunktion für die gleichen Werte.

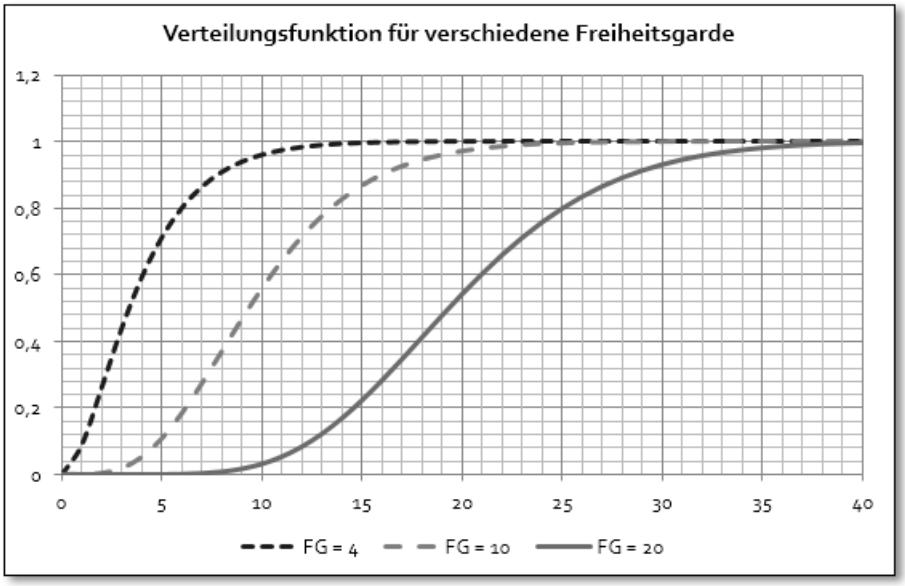


Abbildung 7.26 Die Kurven der Chi-Quadrat-Verteilungsfunktion für unterschiedliche Freiheitsgrade

Die Freiheitsgrade lassen sich bei einer Datenspalte oder Zeile am einfachsten ermitteln mit:

```
Anzahl Möglichkeiten (bei kontinuierlichen Größen die Klassen) - 1
```

Für zweidimensionale Wertetabellen gilt:

```
(Anzahl der Zeilen - 1) * (Anzahl der Spalten - 1)
```

Dass immer ein Freiheitsgrad »verloren« geht, lässt sich an dem Beispiel mit den 60 Würfeltests aus Abbildung 7.24 zu `CHIQU.TEST()` leicht verstehen. Wenn nämlich für 5 mögliche Augenergebnisse die zufälligen Häufigkeiten feststehen, ist die Häufigkeit für das sechste mögliche Ergebnis nicht mehr zufällig, sondern vorgegeben als die Differenz der Summe der 5 Häufigkeiten zur Zahl der Würfe insgesamt.

Die Funktion ist eine der beiden Funktionen in Excel 2013, die anstelle der bisherigen Funktion `CHIVERT()` angeboten werden. `CHIQU.VERT.RE()` liefert die rechtsseitigen Werte.

Siehe auch `CHIQU.INV()`, `CHIQU.INV.RE()`, `CHIQU.TEST()` und `CHIQU.VERT.RE()`.

CHIQU.VERT.RE()

CHISQ.DIST.RT()

Syntax: `CHIQU.VERT.RE(x; Freiheitsgrade)`
Beispiel: `=CHIQU.VERT.RE(10; 3)`
Ergebnis: `0,018`

Die Funktion `CHIQU.VERT.RE()` liefert Werte der rechtsseitigen Verteilungsfunktion (`1-Alpha`) für eine mit `x` angegebene Chi-Quadrat-verteilte Zufallsvariable und den angegebenen Wert für `Freiheitsgrade`.

Ist `x < 0`, gibt die Funktion den Fehler `#ZAHL!` zurück. Das gilt auch, wenn `Freiheitsgrade < 1` oder `> 10^10`. Ist eines der beiden Argumente nicht numerisch, erscheint der Fehler `#WERT!`.

Die Chi-Quadrat-Verteilung ist eine Wahrscheinlichkeitsverteilung, die sich über die Summe von n unabhängigen, quadrierten, standardnormalverteilten Variablen und einer Anzahl von Freiheitsgraden definiert. Die Funktion wird für den Chi-Quadrat-Test benötigt, der beim Vergleich von empirischen zu theoretisch erwarteten Häufigkeiten zum Einsatz kommt.

Rechtsseitige Chi-Quadrat-Verteilung für verschiedene Freiheitsgrade

	Freiheitsgrade		
FG	4	10	20
x			
0	1	1	1
1	0,90979599	0,999827884	1
2	0,735758882	0,996340153	0,999999889
3	0,5578254	0,981424064	0,999995902
4	0,40600585	0,947346983	0,999953502
5	0,287297495	0,891178019	0,999722648
6	0,199148273	0,815263245	0,998897512
7	0,135888225	0,725444953	0,996685056
8	0,091578194	0,628836935	0,991867757
9	0,061099481	0,532103576	0,982907267
10	0,040427682	0,440493285	0,968171943
11	0,026564014	0,357518002	0,94622253
12	0,017351265	0,2850565	0,916075983
13	0,011275794	0,223671817	0,877384049
14	0,007295056	0,172991608	0,830495937
15	0,004701217	0,132061856	0,776407613
16	0,003019164	0,0996324	0,716624259

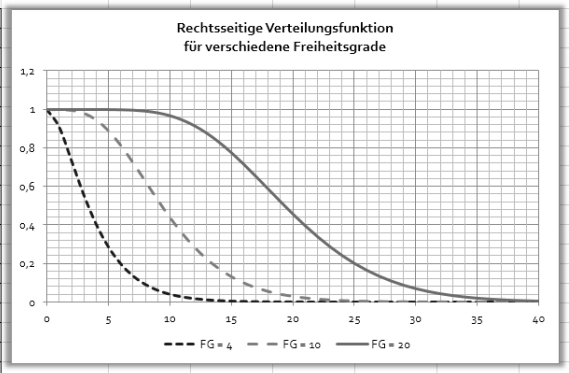


Abbildung 7.27 Die Kurven der rechtsseitigen Chi-Quadrat-Verteilungsfunktion für unterschiedliche Freiheitsgrade

Je nach Anzahl der Freiheitsgrade ändert sich der Charakter der Verteilung, mit steigender Anzahl wird die Funktion flacher und verschiebt sich nach rechts.

Die Freiheitsgrade lassen sich bei einer Datenspalte oder Zeile am einfachsten ermitteln mit:

```
Anzahl Möglichkeiten (bei kontinuierlichen Größen die Klassen) - 1
```

Für zweidimensionale Wertetabellen gilt:

```
(Anzahl der Zeilen - 1) * (Anzahl der Spalten - 1)
```

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 die bisherige Funktion CHIVERT() ersetzen. CHIQU.VERT() liefert die Werte der linksseitigen Verteilungsfunktion.

Siehe auch CHIQU.INV(), CHIQU.INV.RE(), CHIQU.TEST(), und CHIQU.VERT().

EXPON.VERT()

EXPON.DIST()

Syntax: EXPON.VERT(**x; Lambda; Kumuliert**)
Beispiel: =EXPON.VERT(0,5; 3; WAHR)
Ergebnis: 0,777

Die Funktion EXPON.VERT() liefert Wahrscheinlichkeiten für eine exponentialverteilte Zufallsvariable. Diese Verteilung beschreibt die zufällige zeitliche Dauer, bis das nächste Ausnahmeereignis eintritt, wobei aber die noch ausstehende Wartezeit von der bereits verstrichenen Wartezeit unabhängig sein muss. Diese Anforderung ist nicht gegeben, wenn Objekte altern, denn dann hat das bereits erreichte Alter Einfluss auf die zukünftige Lebensdauer. Für derartige Prozesse verwenden Sie besser die Funktionen WEIBULL(), GAMMA() oder LOGNORM.VERT().

Mit x wird das Quantil angegeben, für das die Wahrscheinlichkeit ermittelt werden soll. Lambda ist ein Parameter, der bei der Dichtefunktion den Anfangswert bei $x = 0$ sowie den Grad des Abfalls bestimmt. Kumuliert ist ein Wahrheitswert, mit dem der Typ der Funktion bestimmt wird. Ist Kumuliert mit WAHR belegt, wird der Wert der Verteilungsfunktion geliefert (die Fläche bis zum Quantil), mit FALSCH belegt ergibt sich der Wert für die Dichtefunktion (der Wert auf der y-Achse).

Die Funktion bestimmte die Dichte nach der folgenden Formel:

$$= \lambda e^{-\lambda x}$$

Dabei ist λ der Lambdawert und e die Eulersche Zahl.

Ist `x` oder `Lambda` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `x` < `0` oder `Lambda` ≤ `0`, erscheint der Fehler `#ZAHL!`.

Exponentialverteilte Zufallsvariable

	Kumuliert(Verteilung)			Nicht kumuliert(Dichte)		
Lambda->	0,1	1	3	0,1	1	3
x						
0	0	0	0	0	0	0
0,1	0,0099502	0,0951626	0,2591818	0,0990050	0,9048374	2,2224547
0,2	0,0198013	0,1812692	0,4511884	0,0980199		
0,3	0,0295545	0,2591818	0,5934303	0,0970446		
0,4	0,0392106	0,3296800	0,6988058	0,0960789		
0,5	0,0487706	0,3934693	0,7768698	0,0951229		
0,6	0,0582355	0,4511884	0,8347011	0,0941765		
0,7	0,0676062	0,5034147	0,8775436	0,0932394		
0,8	0,0768837	0,5506710	0,9092820	0,0923116		
0,9	0,0860688	0,5934303	0,9327945	0,0913931		
1	0,0951626	0,6321206	0,9502129	0,0904837		

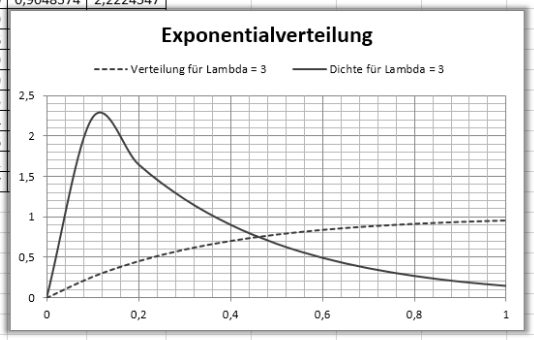


Abbildung 7.28 Beispiele für die Verteilungsfunktion und die Dichtefunktion der Exponentialverteilung

Die Exponentialverteilung wird insbesondere für die Berechnung der Dauer von zufälligen Zeitintervallen benutzt. So können beispielsweise die Haltbarkeit von Bauteilen, die Halbwertzeiten radioaktiver Elemente etc. gut mit einer Exponentialverteilung dargestellt werden.

In dem in Abbildung 7.29 gezeigten Beispiel wird davon ausgegangen, dass ein Gerät eine Ausfallrate (`Lambda`) von 0,1 Promille pro Tag hat. Es interessiert die Frage, wie viele Geräte nach einem Jahr wahrscheinlich defekt sein werden. Die entsprechende Funktion `=EXPON.VERT(365; 0,0001; WAHR)` liefert als Ergebnis den Wert `0,0358` für die Wahrscheinlichkeit, dass ein Gerät höchstens 1 Jahr funktioniert. Oder anders gesprochen: 3,58% der Geräte sind im Durchschnitt nach einem Jahr defekt.

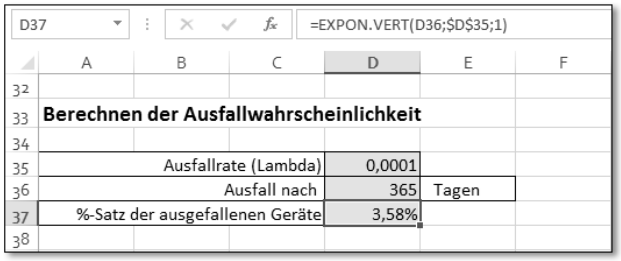


Abbildung 7.29 Lambda als Ausfallrate verstanden

Die Exponentialverteilung setzt in diesem Fall eine gleichmäßige Ausfallrate voraus. Nimmt die Ausfallrate dagegen mit der Zeit zu (oder auch ab), arbeiten Sie besser mit Funktionen wie WEIBULL.VERT(), GAMMA.VERT() oder LOGNORM.VERT().

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion EXPONVERT().

F.INV()

F.INV()

Syntax: F.INV(**Wahrsch**; **Freiheitsgrade1**; **Freiheitsgrade2**)
Beispiel: =F.INV(0,05; 7; 7)
Ergebnis: 0,2640

Die Funktion F.INV() liefert Quantile der linksseitigen F-Verteilung (d.h. die Werte, die in statistischen Tabellenwerken tabelliert sind). Die F-Verteilung kann für F-Tests verwendet werden, bei denen die Varianzen zweier unabhängig χ^2-verteilten Zufallsvariablen untersucht werden.

Mit Wahrsch wird die zur F-Verteilung gehörige Wahrscheinlichkeit angegeben. Als Werte für die Argumente Freiheitsgrade1 und Freiheitsgrade2 werden jeweils die Größen der beiden miteinander zu vergleichenden Stichproben minus 1 angegeben. Sie lassen sich mit der Funktion ANZAHL() ermitteln.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch < 0 oder Wahrsch > 1 oder Freiheitsgrade1 oder Freiheitsgrade2 < 1, erscheint der Fehler #ZAHL!.

Die Funktion ist die Umkehrung von F.VERT(). Bei einem gegebenen Wert für Wahrsch sucht die Funktion durch ein Iterationsverfahren einen Wert x, so dass die Gleichung gilt:

```
Wahrsch = F.VERT(x;..;..)
```

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von FINV() angeboten werden. Die andere ist F.INV.RE().

Siehe auch F.INV.RE(), F.TEST(), F.VERT(), F.VERT.RE(), Abbildung 7.30 zu F.TEST() und die Abschnitte 17.2 bis 17.4 und 17.9.

F.INV.RE()

F.INV.RT()

Syntax: `F.INV.RE(`**`Wahrsch; Freiheitsgrade1; Freiheitsgrade2`**`)`
Beispiel: `=F.INV.RE(0,95; 7; 7)`
Ergebnis: `0,2640`

Die Funktion `F.INV.RE()` liefert Quantile der rechtsseitigen F-Verteilung (d.h. die Werte, die in statistischen Tabellenwerken tabelliert sind). Die F-Verteilung kann für F-Tests verwendet werden, bei denen die Varianzen zweier unabhängig χ^2-verteilten Zufallsvariablen untersucht werden.

Mit `Wahrsch` wird die zur F-Verteilung gehörige Wahrscheinlichkeit angegeben. Als Werte für die Argumente `Freiheitsgrade1` und `Freiheitsgrade2` werden die Größen der beiden miteinander verglichenen Stichproben minus 1 angegeben.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` < `0` oder `Wahrsch` > `1` oder `Freiheitsgrade1` oder `Freiheitsgrade2` < `1` oder > `10^10`, erscheint der Fehler `#ZAHL!`.

`F.INV.RE()` ist die Umkehrung von `F.VERT.RE()`. Bei einem gegebenen Wert für `Wahrsch` sucht die Funktion durch ein Iterationsverfahren einen Wert `x`, so dass die Gleichung gilt:

```
Wahrsch = F.VERT.RE(x;..;..)
```

Die Funktion kann insbesondere auch dazu verwendet werden, um den kritischen Wert für einen F-Test zu berechnen. Dazu wird als Wert für das Argument `Wahrsch` das für den Test maßgebliche Signifikanzniveau angegeben.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von `FINV()` angeboten werden. Die andere ist `F.INV()`.

Die Funktion `F.INV.RE()` liefert das gleiche Ergebnis wie `F.INV()`, die Berechnung ist aber in Bezug auf die Genauigkeit verbessert worden.

Siehe auch `F.INV()`, `F.TEST()`, `F.VERT()`, `F.VERT.RE()`, Abbildung 7.30 zu `F.TEST()` und die Abschnitte 17.2 bis 17.4 und 17.9.

F.TEST()

F.TEST()

Syntax: `F.TEST(`**Matrix1**`;` **Matrix2**`)`
Beispiel: `=FTEST({12;19;13;14;17}; {15;17;16;15;17})`
Ergebnis: `0,0618`

Mit dem F-Test wird geprüft, ob sich zwei Stichproben mit unabhängig χ^2-verteilten Zufallsvariablen in ihrer Varianz zufällig unterscheiden – das ist die Nullhypothese – oder ob sie sich nicht zufällig, sondern »echt« unterscheiden – das ist die Alternativhypothese. Im ersten Fall kann angenommen werden, dass die beiden Stichproben aus derselben normalverteilten Grundgesamtheit stammen können, im zweiten Fall stammen die Stichproben entweder aus verschiedenen Grundgesamtheiten oder sie sind nicht zufällig entnommen.

Die Funktion `F.TEST()` liefert also die Wahrscheinlichkeit der Übereinstimmung zweier Stichproben hinsichtlich ihrer Varianz. `Matrix1` und `Matrix2` sind die Einzelwerte zweier Stichproben. Die Argumente müssen nicht denselben Umfang haben. `F.TEST()` liefert dabei ein Ergebnis zwischen 0 und 1.

Enthalten die mit `Matrix1` und `Matrix2` angegebenen Datenreihen nicht numerische Elemente, werden sie ignoriert. Sind aber jeweils weniger als zwei numerische Elemente vorhanden, gibt die Funktion den Fehler `#DIV/0!` zurück.

Im Beispiel in Abbildung 7.30 wird getestet, ob die beiden Stichproben aus derselben Grundgesamtheit stammen können. In A14 und B14 sind die Varianzen der einzelnen Stichproben mit der Funktion `VAR.S()` berechnet.

Wenn Sie von diesen beiden Varianzwerten den Quotienten bilden, wobei der größere Wert in den Zähler gesetzt wird, ergibt sich ein F-Wert, der die Ausprägung einer Zufallsvariablen ist, die zu einer F-Verteilung gehört. Der in diesem Fall berechnete F-Wert `1,31` ist niedriger als der in E4 mit der Funktion `F.INV.RE()` berechnete kritische F-Wert mit einem Signifikanzniveau von 5% und den Freiheitsgraden der beiden Stichproben – jeweils Anzahl Werte `-1`. Die Nullhypothese muss also nicht verworfen werden.

Sie können nun für den F-Wert `1,31` die Überschreitungswahrscheinlichkeit berechnen, indem Sie für diesen Wert die Funktion `F.VERT.RE()` mit den beiden Werten für Freiheitsgrade berechnen und das Ergebnis mit 2 multiplizieren. Das ergibt hier den Wert `0,73`. Dieser Wert ist größer als das Signifikanzniveau von `0,05`, was noch einmal die Nullhypothese bestätigt. Denselben Wert erhalten Sie mit der Funktion `F.TEST()` auch

direkt, wenn Sie als Argumente die beiden Wertebereiche angeben, wie in Zelle E5 zu sehen ist.

	A	B	C	D	E	F
1						
2	**F-Test**					
3						
4	x-Werte	y-Werte		=F.INV.RE(0,05;7;7)	3,79	
5	4	3		=F.TEST(A5:A12;B5:B12)	0,73	
6	2	2		=F.VERT.RE(E4;7;7)	0,05	
7	3	5		=F.VERT.RE(E14;7;7)*2	0,73	
8	5	6				
9	3	5				
10	6	5				
11	3	2				
12	5	3				
13	Varianz1	Varianz2			F-Wert	
14	1,839285714	2,410714286		Varianz2/Varianz1	1,31	
15						

Abbildung 7.30 Beispiel für einen F-Test

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `FTEST()`. Siehe auch `F.INV()`, `F.INV.RE()`, `F.TEST()`, `F.VERT.RE()`.

F.VERT()

F.DIST()

Syntax: `F.VERT(`**`x; Freiheitsgrade1; Freiheitsgrade2; Kumuliert`**`)`

Beispiel: `=F.VERT(3,787;7;7;WAHR)`

Ergebnis: `0,95`

Die Funktion `F.VERT()` liefert Wahrscheinlichkeitswerte der F-Verteilung. Die wichtigste Anwendung der F-Verteilung liegt in Signifikanztests für zwei unabhängige Stichproben mit χ^2-verteilten Zufallsvariablen.

Mit `x` wird das Quantil der Verteilung angegeben. Je nach der Anzahl `Freiheitsgrade1` (Größe der ersten Stichprobe –1) und der Anzahl `Freiheitsgrade2` (Größe der zweiten Stichprobe –1) unterscheiden sich die F-Verteilungen und nehmen verschiedene Gestalt an, wie Abbildung 7.31 zeigt.

Die Formel für die Dichtefunktion der F-Verteilung ist die folgende:

$$= \frac{\left(\frac{m}{n}\right)^{\frac{m}{2}} * x^{\frac{m}{2}-1}}{\frac{\Gamma\left(\frac{m}{2}\right) * \Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{m}{2}+\frac{n}{2}\right)}} * \left(1+\frac{m}{n} * x\right)^{-(m+m)/2}$$

Dabei steht `m` für `Freiheitsgrade1` und `n` für `Freiheitsgrade2` und Γ für die Funktion `GAMMA()`.

Ist eines der Argumente `x`, `Freiheitsgrade1` oder `Freiheitsgrade2` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `x` < 0 oder `Freiheitsgrade1` oder `Freiheitsgrade2` < 1, erscheint der Fehler `#ZAHL!`.

	A	B	C	D
2	**F-Verteilung**			
4	Freiheitsgrade1	4	10	10
5	Freiheitsgrade2	10	10	100
7	x-Werte	F.VERT()		
8	0	0	0	0
9	0,1	0,36476055	0,02428923	0,009148
10	0,2	0,560150779	0,16279763	0,08513584
11	0,3	0,65138287	0,37016218	0,25202265
12	0,4	0,679352697	0,55757098	0,468175
13	0,5	0,669795953	0,68282274	0,67525915
14	0,6	0,63891148	0,74258423	0,83135081
15	0,7	0,596855898	0,75031477	0,91893819
16	0,8	0,54993797	0,72272894	0,93984299
17	0,9	0,502008867	0,674178	0,90680637
18	1	0,45534963	0,61523438	0,83638113
19	1,1	0,411241068	0,55299136	0,74440421
20	1,2	0,370331348	0,49185685	0,64377729
21	1,3	0,332873693	0,43434577	0,54382721
22	1,4	0,298880158	0,38171592	0,4505544
23	1,5	0,268220901	0,33443021	0,36727023
24	1,6	0,240687914	0,29247382	0,29531971
25	1,7	0,216035532	0,25556336	0,23473357
26	1,8	0,194005755	0,22328022	0,18474876
27	1,9	0,174343662	0,19515288	0,14418943
28	2	0,156806369	0,17070569	0,11172577
29	2,1	0,14116784	0,14948609	0,0860367
30	2,2	0,127221049	0,13107851	0,06590251

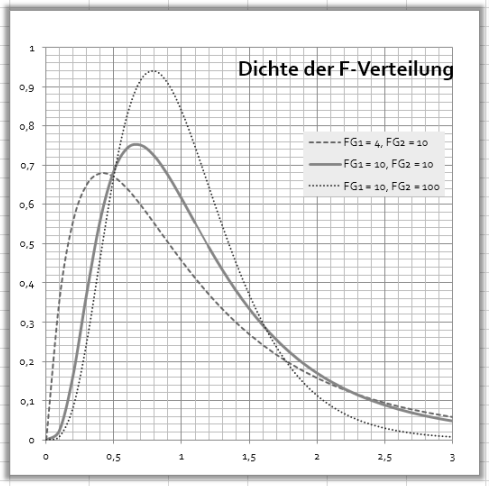


Abbildung 7.31 Dichtefunktion der F-Verteilung

Für den Zusammenhang zwischen `F.VERT()` und `F.INV()` gilt: Wenn:

```
Wert = F.INV(a; ...)
```

dann ist:

```
a = F.VERT(Wert; ...; WAHR)
```

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von FVERT() angeboten werden. Die andere ist F.VERT.RE().

Siehe auch F.INV(), F.INV.RE(), F.TEST(), F.VERT.RE() und GAMMA().

F.VERT.RE()

F.DIST.RT()

Syntax: F.VERT.RE(**x**; **Freiheitsgrade1**; **Freiheitsgrade2**)
Beispiel: =F.VERT.RE(3,787;7;7)
Ergebnis: 0,05

Die Funktion F.VERT.RE() liefert Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitigen F-verteilten Zufallsvariablen zurück. Die wichtigste Anwendung der F-Verteilung liegt in Signifikanztests für zwei unabhängige Stichproben mit χ^2-verteilten Zufallsvariablen.

Mit x wird das Quantil der Verteilung angegeben. Je nach der Anzahl der Freiheitsgrade1 (Größe der ersten Stichprobe –1) und der Anzahl der Freiheitsgrade2 (Größe der zweiten Stichprobe –1) unterscheiden sich die F-Verteilungen und nehmen verschiedene Gestalt an.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist $x < 0$ oder Freiheitsgrade1 oder Freiheitsgrade2 < 1, erscheint der Fehler #ZAHL!.

Für den Zusammenhang zwischen F.VERT.RE() und F.INV.RE() gilt: Wenn:

```
Wert = F.INV.RE(a; ...)
```

dann ist:

```
a = F.VERT.RE(Wert; ...)
```

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von FVERT() angeboten werden. Die andere ist F.VERT(). Die Funktion F.VERT.RE() liefert das gleiche Ergebnis wie FVERT(), die Berechnung ist aber in Bezug auf die Genauigkeit verbessert worden.

Siehe auch F.INV(), F.INV.RE(), F.TEST() und F.VERT().

FISHER()

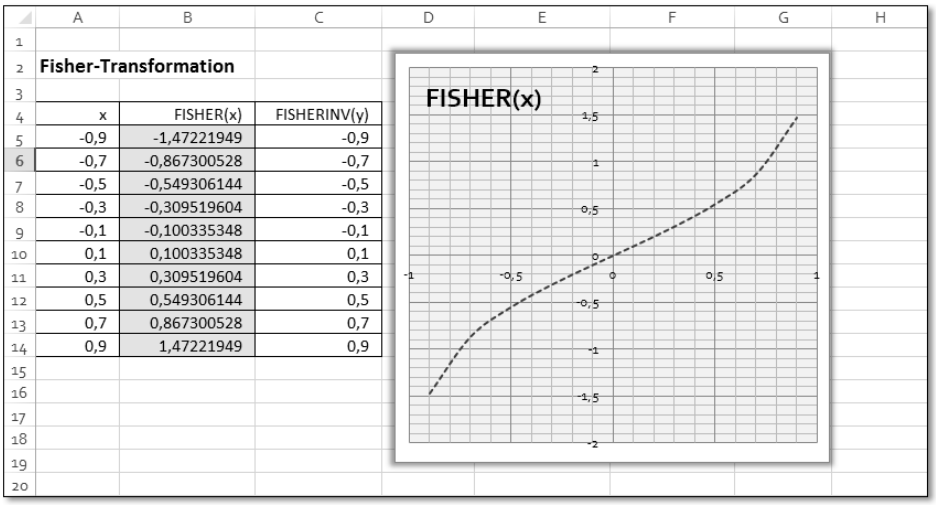
FISHER()

Syntax: `FISHER(x)`
Beispiel: `=FISHER(0,3)`
Ergebnis: `0,30952`

Durch die sogenannte Fisher-Transformation mit Hilfe der Funktion `FISHER()` lässt sich ein mit dem Argument `x` angegebener Korrelationskoeffizient `r` zwischen zwei Variablen in eine annähernd normalverteilte Größe überführen und so anhand der Normalverteilung untersuchen. Für `r` gilt dabei `-1 < r < 1`.

Dazu wird folgende Formel verwendet:

```
z = LN((1+r)/(1-r))/2
```



Fisher-Transformation

x	FISHER(x)	FISHERINV(y)
-0,9	-1,47221949	-0,9
-0,7	-0,867300528	-0,7
-0,5	-0,549306144	-0,5
-0,3	-0,309519604	-0,3
-0,1	-0,100335348	-0,1
0,1	0,100335348	0,1
0,3	0,309519604	0,3
0,5	0,549306144	0,5
0,7	0,867300528	0,7
0,9	1,47221949	0,9

Abbildung 7.32 Fisher-Transformation und ihr Graph

Ist `x` kein numerischer Wert, gibt die Funktion den Fehler `#WERT!` zurück. Ist `x` ≤ `-1` oder ≥ `1`, erscheint der Fehler `#ZAHL!`.

7.9.3 Prüfen der Korrelation zweier Faktoren

In dem Beispiel in Abbildung 7.33 geht es darum, die Korrelationskoeffizienten von zwei Jahren zu vergleichen, die die Enge des Zusammenhangs zwischen Werbeaufwand und Umsatz angeben.

	A	B	C	D	E	F	G
23							
24	**Ermitteln des mittleren Korrelationskoeffizienten**						
25							
26	Monat	Werbeaufwand	Umsatz	Monat	Werbeaufwand	Umsatz	
27	Jan 12	10000	900000	Jan 13	11000	1090000	
28	Feb 12	12000	1090000	Feb 13	12000	1100000	
29	Mrz 12	14000	1300000	Mrz 13	15000	1400000	
30	Apr 12	12000	1200000	Apr 13	12000	1160000	
31	Mai 12	11000	900000	Mai 13	11000	1080000	
32	Jun 12	10000	990000	Jun 13	10000	950000	
33	Jul 12	10000	980000	Jul 13	10000	1000000	
34	Aug 12	11000	1050000	Aug 13	11000	1090000	
35	Sep 12	12000	1020000	Sep 13	13000	1240000	
36	Okt 12	13000	1200000	Okt 13	14000	1360000	
37	Nov 12	13500	1230000	Nov 13	13500	1290000	
38	Dez 12	14000	1300000	Dez 13	16000	1700000	
39	**Korrelationskoeffizient 1**		0,911836646	**Korrelationskoeffizient 2**		0,967833513	
40	**Mit Fisher() transformierter Wert 1**		1,538314133	**transformierter Wert 2**		2,056881666	
41	**mittlerer transformierter Wert**		1,797597899	**=MITTELWERT(C40;F40)**			
42	**mittlerer Korrelationskoeffizient**		0,946556689	**=FISHERINV(C41)**			
43							

Abbildung 7.33 Anwendung der Fisher-Transformation beim Vergleich von Korrelationen

Der naheliegende Gedanke, einfach den Mittelwert zwischen den beiden Koeffizienten zu bilden, scheitert daran, dass der Korrelationskoeffizient nicht intervallskaliert ist. Die Werte sind nicht normalverteilt. Hier hilft der Umweg über die Fisher-Transformation. In Zelle C33 und F33 wird zunächst jeweils der errechnete Korrelationskoeffizient 1 + 2 transformiert – mit `=FISHER(C32)` bzw. `=FISHER(F32)`.

Da diese Werte nun angehend normalverteilt sind, lässt sich in Zelle C34 mit `=MITTELWERT(C33;F33)` der Mittelwert bilden. Dieser muss nun wieder rückübersetzt werden in einen Korrelationskoeffizienten. Dazu wird einfach die Umkehrfunktion verwendet: `=FISHERINV(C34)`. Dieser Wert kann nun bewertet werden, er belegt in diesem Fall mit `0,946` eine im Durchschnitt der beiden Jahre sehr enge Korrelation. Siehe auch `FISHERINV()`.

FISHERINV()

FISHERINV()

Syntax: `FISHERINV(`**y**`)`

Beispiel: `=FISHERINV(0,5)`

Ergebnis: `0,4621`

Die Funktion FISHERINV() ist die Umkehrfunktion von FISHER(). Sie rechnet aus dem transformierten Wert wieder auf den entsprechenden Korrelationskoeffizienten zurück. Dabei ist y der transformierte Wert. Wenn:

```
y = FISHER(x)
```

dann ist:

```
FISHERINV(y) = x
```

Ist y kein numerischer Wert, gibt die Funktion den Fehler #WERT! zurück.

Seltsamerweise wird bei dieser Funktion das Suffix INV weiterhin ohne Punkt angehängt. Siehe auch FISHER(), Abbildung 7.32 und Abbildung 7.33.

G.TEST()

G.TEST()

Syntax:	G.TEST(**Matrix**; **x**; Sigma)
Beispiel:	=G.TEST({11.19.18.21.13.17.9.14};12;4)
Ergebnis:	0,01078

Die Funktion G.TEST() liefert den einseitigen Wahrscheinlichkeitswert für einen Gauß-Test bei normalverteilten Daten. Für den mit x angegebenen Erwartungswert einer Zufallsvariablen gibt G.TEST() die Wahrscheinlichkeit zurück, mit der der Stichprobenmittelwert größer ist als der Durchschnitt der in Matrix angegebenen Werte. Mit diesem Test kann die Wahrscheinlichkeit dafür geschätzt werden, dass ein bestimmter Wert aus derselben (normalverteilten) Grundgesamtheit stammt wie eine gegebene Stichprobe.

Mit Matrix wird der Datenbereich der Stichprobe angegeben, mit der der Wert x verglichen werden soll. Nicht numerische Elemente in Matrix werden ignoriert.

Das optionale Argument Sigma gibt die Standardabweichung der Grundgesamtheit an, wenn sie bekannt ist. Wird Sigma nicht angegeben, dann verwendet die Funktion die Standardabweichung der Stichprobe als Schätzwert für Sigma.

Ist Matrix leer, gibt die Funktion den Fehler #NV zurück. Nicht numerische Werte für x oder Sigma führen zu dem Fehler #WERT!, negative Werte für Sigma zu dem Fehler #ZAHL!.

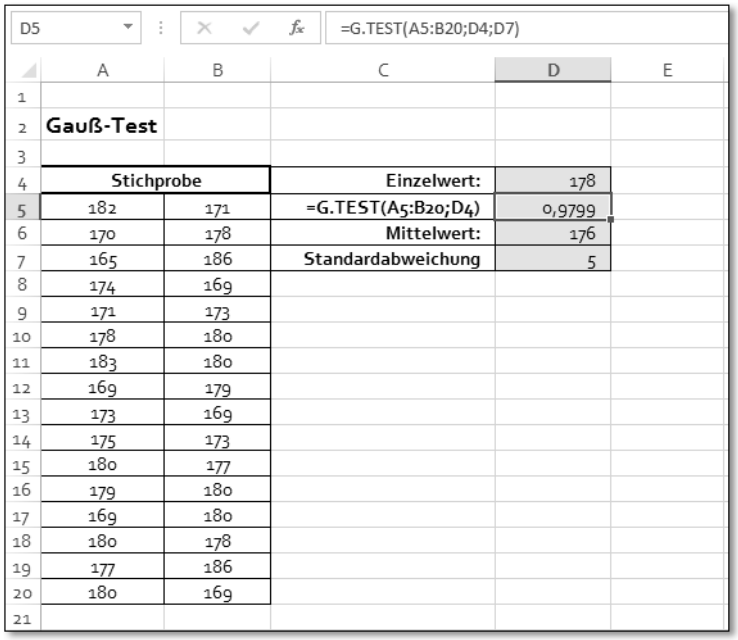


Abbildung 7.34 Beispiel für einen Gauß-Test

Der Test liefert als Wahrscheinlichkeit `0,97` für die Hypothese, dass der Einzelwert zur gleichen Grundgesamtheit gehört. Der Test wird genauer, wenn die Stichprobe größer ist und die Standardabweichung der Grundgesamtheit vorliegt. Damit der Test zu brauchbaren Ergebnissen führt, sollte die Matrix wenigstens 30 Werte enthalten. Siehe auch Abschnitt 17.18, »Gaußtest«.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `GTEST()`.

GAMMA()

GAMMA()

Syntax: `GAMMA(`**x**`)`
Beispiel: `=GAMMA(2,5)`
Ergebnis: `1,329`

Die in Excel 2013 neu eingeführte Funktion `GAMMA()` liefert direkt den Wert der Gammafunktion für die mit `x` angegebene Variable. Die Funktion erspart Ihnen also das Ablesen der Werte aus Tabellen zur Gammafunktion.

	A	B
1		
2	**Werte der GAMMA()-Funktion**	
3		
4	x	GAMMA()
5	-4	
6	-3,8	0,299632
7	-3,6	0,246857
8	-3,4	0,325891
9	-3,2	0,689056
10	-3	
11	-2,8	-1,138602
12	-2,6	-0,888686
13	-2,4	-1,108030
14	-2,2	-2,204981
15	-2	
16	-1,8	3,188086
17	-1,6	2,310583
18	-1,4	2,659272
19	-1,2	4,850957
20	-1	
21	-0,8	-5,738555
22	-0,6	-3,696933
23	-0,4	-3,722981
24	-0,2	-5,821149
25	0	
26	0,2	4,590844
27	0,4	2,218160
28	0,6	1,489192
29	0,8	1,164230
30	1	1,000000
31	1,2	0,918169
32	1,4	0,887264
33	1,6	0,893515
34	1,8	0,931384

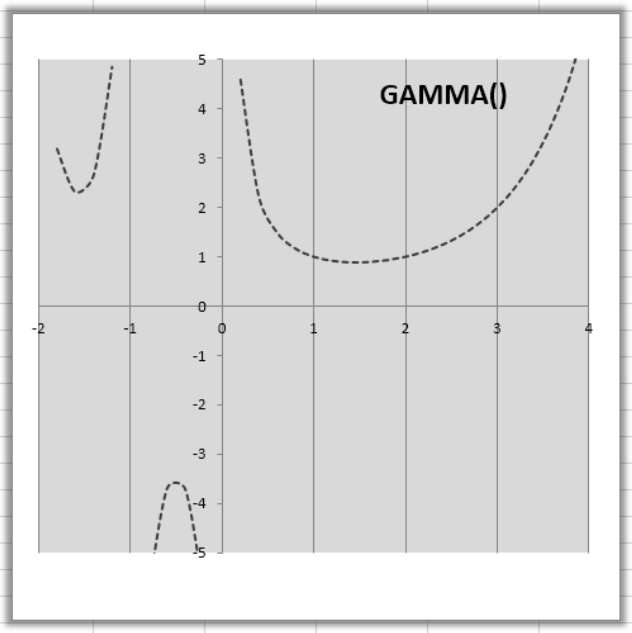


Abbildung 7.35 Werte der Funktion `GAMMA()` und ihr Graph

Die Funktion wird auch als Eulersches Integral zweiter Gattung bezeichnet. Der Graph der Funktion hat für 0 und alle negativen Ganzzahlen Polstellen.

Der Wert für `x` ist eingegrenzt auf Werte > `0` oder negative Werte, die keine Ganzzahlen sind. Für negative Ganzzahlen oder `0` gibt die Funktion den Fehler `#ZAHL!` aus. Ist `x` nicht numerisch, erscheint der Fehler `#WERT!`.

Die Funktion arbeitet für `x` > `0` mit folgender Gleichung:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

Für alle positiven ganzen Zahlen gilt dabei:

Γ(n) = (n-1)!

Die GAMMA()-Funktion liefert in diesem Fall also für ganze Zahlen eine Parallelverschiebung der Funktion FAKULTÄT().

	I	J	K	L
1				
2	x	GAMMA()	FAKULTÄT()	
3	1	1	1	
4	2	1	2	
5	3	2	6	
6	4	6	24	
7	5	24	120	
8	6	120	720	
9	7	720	5040	
10	8	5040	40320	
11	9	40320	362880	
12	10	362880	3628800	
13				

Abbildung 7.36 Vergleich GAMMA() und FAKULTÄT() für ganze Zahlen

Von Gauß wurde die Definition der Gammafunktion auf alle nicht in der Menge {0, –1, –2, ...} enthaltenen Zahlen erweitert:

$$= \lim_{n\to\infty} \frac{n!n^{x-1}}{x(x+1)(x+2)...(x+n-1)}$$

Siehe auch GAMMA.INV() und GAMMA.VERT().

GAMMA.INV()

GAMMA.INV()

Syntax:	GAMMA.INV(**Wahrsch**; **Alpha**; **Beta**)
Beispiel:	=GAMMA.INV(0,05; 3; 1)
Ergebnis:	0,8176

Die Funktion GAMMA.INV() berechnet Quantile der Gammaverteilung.

Für das Argument Wahrsch erwartet die Funktion Wahrscheinlichkeitswerte aus einer Gammaverteilung. Alpha und Beta sind Funktionsparameter (in der Literatur werden

als Parameter meist b und p angegeben). Beta = 1 liefert die standardisierte Gammaverteilung.

Ist Wahrsch, Alpha oder Beta nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch < 0 oder > 1 oder Alpha oder Beta ≤ 0, erscheint der Fehler #ZAHL!.

Die Funktion ist die Umkehrfunktion von GAMMA.VERT(). Zwischen GAMMA.INV() und GAMMA.VERT() besteht folgender Zusammenhang:

Wenn:

```
Wert = GAMMA.INV(x; ...)
```

dann ist:

```
x = GAMMA.VERT(Wert; ...WAHR)
```

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion GAMMAINV(). Siehe auch GAMMA.VERT(), GAMMA() und Abbildung 7.37.

GAMMA.VERT()

GAMMA.DIST()

Syntax:	GAMMA.VERT(**x**; **Alpha**; **Beta**; **Kumuliert**)
Beispiel:	=GAMMA.VERT(1,5; 2; 1; WAHR)
Ergebnis:	0,44217

Die Funktion GAMMA.VERT() liefert Wahrscheinlichkeiten für eine gammaverteilte Zufallsvariable. Die Gammaverteilung, auch Eulersches Integral zweiter Gattung genannt, ist eine stetige Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen. Sie gilt als sehr anpassungsfähig, da sie auch die Untersuchung von schiefen Verteilungen erlaubt. Sie findet vor allem in der Warteschlangen- (oder Bedienungs-) und Zuverlässigkeitstheorie Anwendung. Von den Argumenten bezeichnet x das Quantil, für das die Wahrscheinlichkeit berechnet werden soll, Alpha und Beta sind Parameter der Verteilung (vergleiche GAMMA.INV()).

Kumuliert bestimmt den Typ der Verteilung: Mit WAHR wird der Wert der Verteilungsfunktion berechnet, also die Wahrscheinlichkeit, dass eine Zufallsgröße einen Wert zwischen 0 und x annimmt; mit FALSCH wird der Wert der Dichtefunktion berechnet, also die Wahrscheinlichkeit für den Wert x. Wird Beta = 1 gesetzt, ergibt dies die Werte für die standardisierte Gammaverteilung.

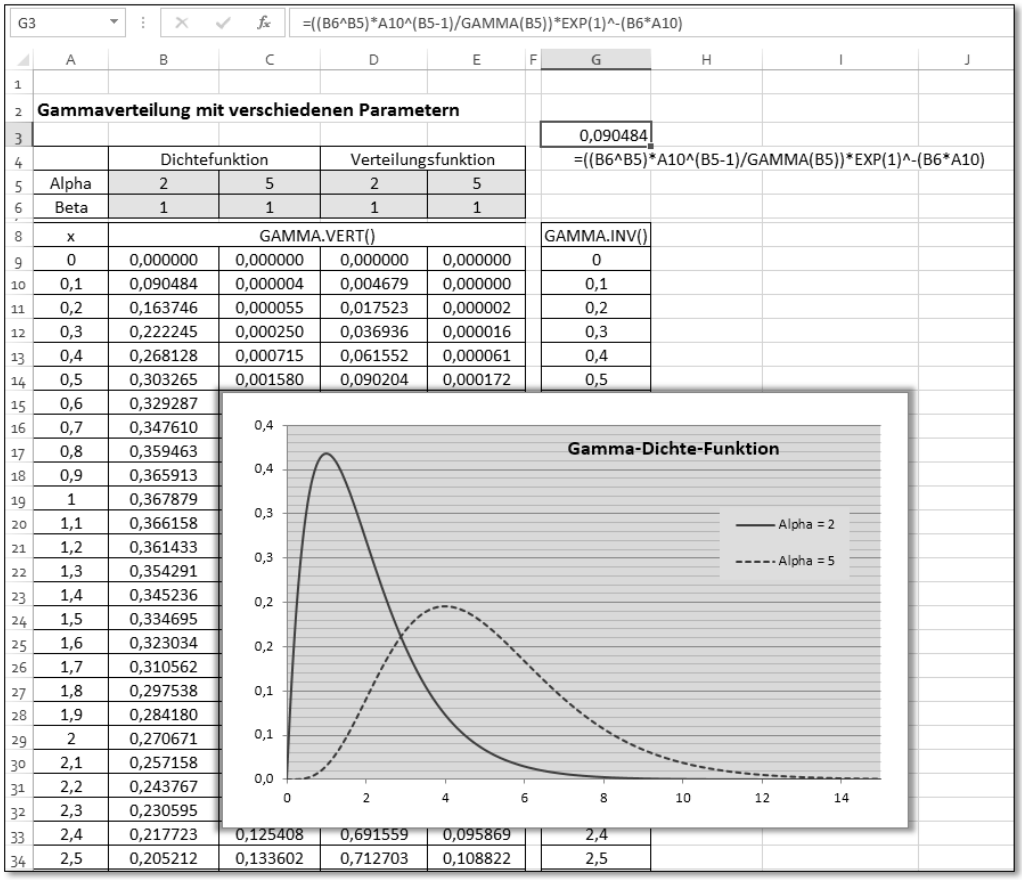


Abbildung 7.37 Gammaverteilung und Dichtefunktion

Ist `x`, `Alpha` oder `Beta` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist $x < 0$ oder `Alpha` oder `Beta` ≤ 0, erscheint der Fehler `#ZAHL!`.

Durch geeignete Wahl der Parameter lässt sich die Gammaverteilung in andere Verteilungen überführen (Chi-Quadrat-Verteilung, Weibullverteilung etc.). Wird `Alpha = 1` gesetzt, ergibt sich eine Exponentialverteilung mit `Lambda = 1/Beta`.

Die Formel für die Dichtefunktion der Gammaverteilung ist folgende:

$$= \frac{\beta^{\alpha} x^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta x}$$

für $x \geq 0$ und α (`Alpha`) und β (`Beta`) > 0. Den Wert für $\Gamma(\alpha)$ liefert dabei die Funktion `GAMMA()`.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion GAMMAVERT().

Siehe auch GAMMA.INV() und GAMMA().

GAMMALN()

GAMMALN()

Syntax:	GAMMALN(**x**)
Beispiel:	=GAMMALN(6)
Ergebnis:	4,7875

Die Funktion GAMMALN() liefert den natürlichen Logarithmus zu Werten der Gammafunktion.

Ist x nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist x ≤ 0, liefert die Funktion den Fehler #ZAHL!.

Die Funktion rechnet nach folgender Formel:

```
=LN(GAMMA(x))
```

Seit Excel 2010 wird für GAMMALN() die umbenannte Funktion GAMMALN.GENAU() angeboten. (Seltsamerweise wird GAMMALN() aber nicht in der Kategorie **Kompatibilität** angeboten.)

Siehe auch GAMMA() und GAMMALN.GENAU().

GAMMALN.GENAU()

GAMMALN.PRECISE()

Syntax:	GAMMALN.GENAU(**x**)
Beispiel:	=GAMMALN.GENAU(6)
Ergebnis:	4,7875

Die Funktion GAMMALN.GENAU() liefert den natürlichen Logarithmus zu Werten der Gammafunktion. Sie ersetzt seit Excel 2010 die bisherige Funktion GAMMALN().

Ist x nicht numerisch, liefert die Funktion den Fehler #WERT!. Ist x ≤ 0, liefert die Funktion den Fehler #ZAHL!.

	A	B	C	D	E
1					
2	**Logarithmus zu Werten der Gamma-Funktion**				
3					
4	x	GAMMA()	GAMMALN.GENAU()	GAMMALN()	
5	0,2	4,590844	1,52406382243078	1,52406382243078	
6	0,4	2,218160	0,79667781770178	0,79667781770178	
7	0,6	1,489192	0,39823385806924	0,39823385806924	
8	0,8	1,164230	0,15205967839984	0,15205967839984	
9	1	1,000000	0,00000000000000	0,00000000000000	
10	1,2	0,918169	-0,08537409000332	-0,08537409000332	
11	1,4	0,887264	-0,11961291417237	-0,11961291417237	
12	1,6	0,893515	-0,11259176569676	-0,11259176569676	
13	1,8	0,931384	-0,07108387291437	-0,07108387291437	
14	2	1,000000	0,00000000000000	0,00000000000000	
15	2,2	1,101802	0,09694746679064	0,09694746679064	
16	2,4	1,242169	0,21685932244885	0,21685932244885	
17	2,6	1,429625	0,35741186354899	0,35741186354899	
18	2,8	1,676491	0,51670279198776	0,51670279198776	
19	3	2,000000	0,69314718055996	0,69314718055996	
20					

Abbildung 7.38 Werte der Funktionen GAMMALN.GENAU() und GAMMALN()

Bei ganzen Zahlen gilt folgende Beziehung für diese Funktion:

EXP(GAMMALN.GENAU(x))

ergibt:

FAKULTÄT(x-1)

Siehe auch GAMMALN().

GAUSS()

GAUSS()

Syntax: GAUSS(**x**)
Beispiel: =GAUSS(2)
Ergebnis: 0,47725

Werte der GAUSS()-Funktion

x	GAUSS()	NORM.S.VERT()
-3,5	-0,499767	0,000232629
-3,25	-0,499423	0,000577025
-3	-0,498650	0,001349898
-2,75	-0,497020	0,002979763
-2,5	-0,493790	0,006209665
-2,25	-0,487776	0,012224473
-2	-0,477250	0,022750132
-1,75	-0,459941	0,040059157
-1,5	-0,433193	0,066807201
-1,25	-0,394350	0,105649774
-1	-0,341345	0,158655254
-0,75	-0,273373	0,226627352
-0,5	-0,191462	0,308537539
-0,25	-0,098706	0,401293674
0	0,000000	0,5
0,25	0,098706	0,598706326
0,75	0,273373	0,773372648
1	0,341345	0,841344746
1,25	0,394350	0,894350226
1,5	0,433193	0,933192799
1,75	0,459941	0,959940843
2	0,477250	0,977249868
2,25	0,487776	0,987775527
2,5	0,493790	0,993790335
2,75	0,497020	0,997020237
3	0,498650	0,998650102
3,25	0,499423	0,999422975
3,5	0,499767	0,999767371

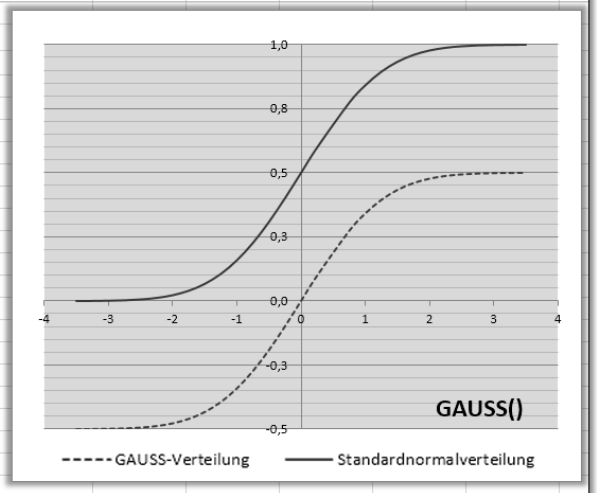


Abbildung 7.39 GAUSS() und Standardnormalverteilung im Vergleich

Die in Excel 2013 neu eingeführte Funktion GAUSS() berechnet die Wahrscheinlichkeit, dass ein Element einer Standardgrundgesamtheit zwischen dem Mittelwert und der mit x angegebenen Zahl von Standardabweichungen liegt. Der im Beispiel angezeigte Wert gibt also die Wahrscheinlichkeit an, dass ein Element zwischen zwei Standardabweichungen vom Mittelwert liegt.

Die Funktion gibt für das Argument x einen Wert aus, der genau 0,5 unter dem Wert liegt, den die Standardnormalverteilung liefert.

GEOMITTEL()

GEOMEAN()

Syntax: GEOMITTEL(**Zahl1**; Zahl2; ...)

Beispiel: =GEOMITTEL(2;5;6;8;9)

Ergebnis: 5,33

Die Funktion GEOMITTEL() berechnet das geometrische Mittel für eine Reihe von Daten, die positive Zahlen und größer 0 sein müssen. Berechnet wird das geometrische Mittel, indem alle Beobachtungen miteinander multipliziert werden und dann die n-te Wurzel aus dem Ergebnis gezogen wird, wobei n die Anzahl der Beobachtungen ist. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30.

Das Ergebnis von GEOMITTEL() ist immer kleiner als das Ergebnis für MITTELWERT(). Ein wichtiges Anwendungsgebiet ist die Errechnung von Mittelwerten für Zuwachsraten.

Ist einer der für die Berechnung aufgenommenen Werte 0 oder kleiner, so gibt Excel den Fehler #DIV/0! aus. Wahrheitswerte und in Textform angegebene Zahlen, die direkt als Argumente angegeben sind, werden berücksichtigt. Nicht numerische Werte führen zu dem Fehler #WERT!.

Das Beispiel in Abbildung 7.40 wendet die Funktion auf Zuwachsraten an, die über einen Index in Spalte D angegeben werden, und zwar als Verhältnis des aktuellen Jahresbetrags zum Vorjahresbetrag. Von diesen Indexwerten wird in Zelle C15 das geometrische Mittel berechnet:

```
=GEOMITTEL(C6:C14)
```

Zum Vergleich wird in C16 noch der arithmetische Mittelwert der Wachstumsindizes berechnet. Während dieser Wert noch im Mittel ein Wachstum angibt, zeigt der Wert für GEOMITTEL() einen leichten Rückgang an.

	A	B	C
1			
2	**Geometrisches Mittel**		
3			
4	Jahr	Umsatz	Index
5	1. Jahr	10000	
6	2. Jahr	9800	0,9800
7	3. Jahr	10298	1,0508
8	4. Jahr	10360	1,0060
9	5. Jahr	8000	0,7722
10	6. Jahr	10900	1,3625
11	7. Jahr	10690	0,9807
12	8. Jahr	10800	1,0103
13	9. Jahr	10020	0,9278
14	10. Jahr	9900	0,9880
15		GEOMITTEL()	0,9989
16		MITTELWERT()	1,0087
17			

Abbildung 7.40 Das geometrische Mittel im Vergleich zum Mittelwert

Um zu belegen, dass die Funktion GEOMITTEL() in diesem Fall die geeignetere Funktion ist, wird in Zelle C17 vom Umsatz des ersten Jahres aus auf den Umsatz des 10. Jahres mit der Formel

```
=B5*C15^9
```

hochgerechnet und in Zelle C18 eine entsprechende Hochrechnung mit dem arithmerischen Mittel vorgenommen. Die erste Formel stimmt exakt mit dem Umsatz im 10. Jahr überein, die Berechnung mit dem arithmetischen Mittel liefert dagegen einen deutlich zu hohen Wert.

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen MITTELWERT(), MITTELWERTA(), GESTUTZTMITTEL(), HARMITTEL(), MEDIAN(), MODUS.EINF() sowie MODUS.VIELF().

GESTUTZTMITTEL()

TRIMMEAN()

Syntax: GESTUTZTMITTEL(**Matrix**; **Prozent**)

Beispiel: =GESTUTZTMITTEL({2;9;7;5;8;7;9};0,3)

Ergebnis: 7,2

Die Funktion GESTUTZTMITTEL() gibt das arithmetische Mittel eines Datensatzes zurück, bei dem aber die niedrigsten und die höchsten Werte nicht berücksichtigt werden. Das Verfahren ist dazu gedacht, »Ausreißer« aus dem Mittelwert herauszuhalten, die den Gesamteindruck zu sehr verzerren würden.

Die zu mittelnden Werte werden mit dem Argument Matrix angegeben. Das kann ein Zellbereich oder der Name eines solchen Bereichs oder eine Matrixkonstante sein. Sollen nicht zusammenhängende Zellbereiche ausgewertet werden, müssen Sie die einzelnen Bereiche durch Semikola trennen und alle Bereiche insgesamt einklammern, beispielsweise:

```
=GESTUTZTMITTEL((A10:A100;C10:C100);2%)
```

Wie viele der mit Matrix angegebenen Daten jeweils oben und unten abgeschnitten werden, wird durch das Argument Prozent bestimmt. Wird für Prozent 0,1 (10%) angegeben, werden 5% der niedrigsten und 5% der höchsten Werte für die Berechnung nicht berücksichtigt. Dabei wird bei Bedarf auf das kleinste Vielfache von 2 gerundet, so dass am Anfang und am Ende der Skala immer dieselbe Anzahl von Werten aus der Berechnung herausgenommen wird.

	A	B	C
1			
2	**Gestutzter Mittelwert**		
3			
4	Monat	Umsatz	GESTUTZTMITTEL()
5	Jan 12	900000	1091667
6	Feb 12	1090000	MITTELWERT()
7	Mrz 12	1300000	1104000
8	Apr 12	1200000	
9	Mai 12	900000	
10	Jun 12	990000	
11	Jul 12	30000	
12	Aug 12	1050000	
13	Sep 12	1020000	
14	Okt 12	1200000	
15	Nov 12	1230000	
16	Dez 12	1300000	
17	Jan 13	1090000	
18	Feb 13	1100000	
19	Mrz 13	2400000	
20	Apr 13	1160000	
21	Mai 13	1080000	
22	Jun 13	950000	
23	Jul 13	1000000	
24	Aug 13	1090000	
25			

Abbildung 7.41 Die markierten Ausreißer werden von `GESTUTZTMITTEL()` ignoriert.

Der Wert für `Prozent` darf nicht < 0 und auch nicht ≥ 1 sein, sonst gibt die Funktion den Fehler `#ZAHL!` aus. Nicht numerische Werte oder Wahrheitswerte in `Matrix` werden ignoriert.

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen `MITTELWERT()`, `MITTELWERTA()`, `GEOMITTEL()`, `HARMITTEL()`, `MEDIAN()`, `MODUS.EINF()` sowie `MODUS.VIELF()`.

HÄUFIGKEIT()

FREQUENCY()

Syntax: `HÄUFIGKEIT(`**`Daten; Klassen`**`)`

Beispiel: siehe Abbildung 7.42

Die Funktion HÄUFIGKEIT() wertet die mit Daten angegebene Menge numerischer Werte nach der Häufigkeit des Vorkommens innerhalb der durch Klassen definierten Intervalle aus. Das Argument Daten kann dabei als Bezug auf einen ein- oder mehrspaltigen Bereich oder als Matrixkonstante angegeben werden.

Das Argument Klassen gibt die Intervallgrenzen an, in der Regel in Form eines einspaltigen oder einzeiligen Bereichs oder als Matrixkonstante.

Die Funktion ignoriert bei dem Argument Daten sowohl leere Zellen als auch Zellen, die Texte oder als Text eingegebene Zahlen enthalten. Enthält Daten keine numerischen Werte, ist das Ergebnis eine Matrix mit lauter Nullen.

Da die Funktion als Ergebnis eine einspaltige Matrix liefern soll, muss sie auch als Matrixformel eingegeben werden. Der Bereich, in dem die Ergebnisse ausgegeben werden sollen, muss also vor der Eingabe der Funktion markiert werden. Die Formel selbst wird mit Strg + ⇧ + ↵ abgeschlossen. Der für die Funktionsergebnisse gewählte Bereich muss immer eine Zelle mehr umfassen als die Zahl der Elemente in Klassen.

Bei der Auswertung ordnet die Funktion dem ersten Element von Klassen alle Daten zu, die kleiner oder gleich der ersten angegebenen Klassengrenze sind; in der überzähligen Zelle des Ausgabebereichs werden alle Datenwerte gezählt, die größer als die letzte angegebene Klassengrenze sind.

7.9.4 Daten klassifizieren

Die Klassifizierung von Daten ist eine der bewährten Methoden, um Übersicht über größere Datenmengen zu gewinnen. Die Kunst liegt dabei in einer sinnvollen Bestimmung der Klassengrenzen. In der Regel sollten die Klassen gleich breit sein, wobei allerdings die Anfangs- und Endklassen häufig offen bleiben, um seltenere Extremwerte aufzunehmen. Außerdem sollten die Klassen weder zu groß noch zu klein gewählt werden, wobei es von dem jeweiligen Gegenstandsbereich und den vorgesehenen Fragestellungen abhängt, was als sinnvolle Klasseneinteilung verstanden werden kann.

Abbildung 7.42 zeigt ein einfaches Beispiel, in dem die Häufigkeit bestimmter Körpergrößen ermittelt wird. Das Diagramm wertet die Bereiche D5:E9 aus, wobei die Klassenbezeichnungen die Daten für die x-Achse liefern. Da das Beispiel mit einheitlichen Intervallbreiten arbeitet, kann einfach ein Säulendiagramm verwendet werden, bei dem die Abstände zwischen den Säulen auf 0 gesetzt sind, um ein Histogramm zu erzeugen, das mit den Flächen der Säulen die Häufigkeit sichtbar macht.

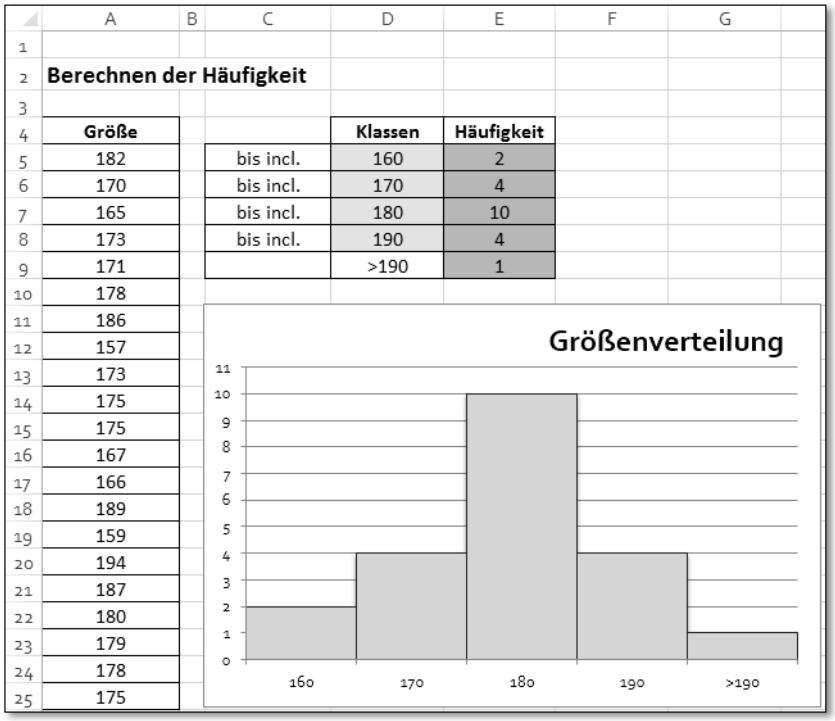


Abbildung 7.42 Auswertung von Daten nach Klassen

Excel bietet unter den in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, vorgestellten Analyse-Funktionen noch eine ähnliche Darstellung der Häufigkeit an, die gleichzeitig auch die Ausgabe der Häufigkeitsverteilung als Histogramm einschließt.

HARMITTEL()

HARMEAN()

Syntax: `HARMITTEL(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=HARMITTEL(2;5;6;8;9)`

Ergebnis: `4,53`

Die Funktion `HARMITTEL()` gibt das harmonische Mittel der angegebenen Werte zurück. Das harmonische Mittel ist der Kehrwert eines aus Kehrwerten errechneten arithmetischen Mittels. Bezogen auf das Beispiel also:

```
=1/MITTELWERT(1/2;1/5;1/6;1/8;1/9)
```

Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30. Das harmonische Mittel wird zuweilen im Rahmen varianzanalytischer Problemstellungen benötigt.

Eine typische Anwendung ist die Ermittlung von Durchschnittsgeschwindigkeiten. In Abbildung 7.43 ist dargestellt, dass nacheinander drei Strecken von je 100 km mit unterschiedlichen Durchschnittsgeschwindigkeiten zurückgelegt wurden. Die Formel in C5 ermittelt nun die Durchschnittsgeschwindigkeit, die für die gesamte Strecke erreicht werden müsste, um zum selben Zeitpunkt anzukommen. Das Ergebnis fällt deutlich kleiner aus als bei `GEOMITTEL()` und `MITTELWERT()`.

	A	B	C	D	E	F
1						
2	**Harmonisches Mittel einer Datenmenge**					
3						
4	km	Geschwindigkeit	HARMITTEL()	GEOMITTEL()	MITTELWERT()	
5	100	100	42,86	50,00	58,33	
6	100	50				
7	100	25				
8	300		km nach 7 Std			
9	Gesamtdauer in Std	7	300	350	408	
10						

Abbildung 7.43 Berechnung der Durchschnittsgeschwindigkeit für mehrere Strecken

Bei Bezügen auf Zellen, die nicht numerische Werte, Wahrheitswerte oder gar nichts enthalten, werden diese ignoriert, Wahrheitswerte oder Zahlen in Textform, die direkt als Argumente eingegeben sind, werden dagegen berücksichtigt.

Beträgt der Wert eines Elements der Daten `0` oder weniger, dann gibt das Programm den Fehler `#ZAHL!` aus.

HYPGEOM.VERT()

HYPGEOM.DIST()

Syntax: `HYPGEOM.VERT(`**`Erfolge_S`**`; `**`Umfang_S`**`; `**`Erfolge_G`**`; `**`Umfang_G`**`; `**`Kumuliert`**`)`
Beispiel: `=HYPGEOM.VERT(1; 2; 4; 32)`
Ergebnis: `0,2258`

Die Funktion `HYPGEOM.VERT()` berechnet die Wahrscheinlichkeiten für hypergeometrisch verteilte Zufallsvariablen. Die Funktion wird in Fällen angewendet, in denen es

durch Entnahme aus der Grundgesamtheit – ohne anschließendes Zurücklegen – jedes Mal zu einer Änderung ihrer Zusammensetzung und damit der Erfolgswahrscheinlichkeit bei den nachfolgenden Entnahmen kommt, so dass hier die Binomialverteilung nicht eingesetzt werden kann. Vergleiche BINOM.VERT().

Mit Umfang_S und Umfang_G werden die Größe der entnommenen Stichprobe und die Größe der Grundgesamtheit angegeben. Erfolge_G gibt an, wie oft das zu testende Ereignis in der Grundgesamtheit enthalten ist, Erfolge_S, wie oft es in der Stichprobe enthalten sein soll. Das neue Argument Kumuliert ist ein Wahrheitswert, der den Typ der Funktion bestimmt. Mit Kumuliert = WAHR liefert die Funktion den Wert der kumulierten Verteilungsfunktion, andernfalls den Wert der Dichtefunktion.

Ist Erfolge_S < 0 oder Erfolge_S größer als der kleinere der Werte von Umfang_S oder Erfolge_G, gibt die Funktion den Fehlerwert #ZAHL! zurück. Das gilt auch für folgende Fälle:

- Erfolge_S ist kleiner als der größere Wert von 0 bzw. (Umfang_S – Umfang_G + Erfolge_G).
- Umfang_S ist ≤ 0 oder Umfang_S > Umfang_G.
- Erfolge_G ist ≤ 0 oder Erfolge_G > Umfang_G.
- Umfang_G ist ≤ 0.

Ist eines der Argumente Erfolge_S, Umfang_S, Erfolge_G oder Umfang_G nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Die Dichtefunktion der hypergeometrischen Verteilung ist durch folgende Formel definiert:

$$= \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$$

Dabei ist N = Umfang_G, M = Umfang_S, n = Erfolge_G und k = Erfolge_S.

Im Beispiel in Abbildung 7.44 wird die Wahrscheinlichkeit dafür ermittelt, dass beim 6-aus-49-Lotto 0 bis 6 Richtige erreicht werden. Da bei diesem Spiel jede Zahl höchstens einmal gezogen werden kann, ist die Fragestellung ein Fall für eine hypergeometrische Verteilung.

	A	B	C	D	E	F	G
1							
2	**Gewinnchance im Lotto 6 aus 49:**						
3							
4	Erfolge_S	Umfang_S	Erfolge_G	Umfang_G		HYPGEOM.VERT()	
5	6	6	6	49	6 Richtige	0,0000072%	
6	5	6	6	49	5 Richtige	0,0018450%	
7	4	6	6	49	4 Richtige	0,0968620%	
8	3	6	6	49	3 Richtige	1,7650404%	
9	2	6	6	49	2 Richtige	13,2378029%	
10	1	6	6	49	1 Richtige	41,3019450%	
11	0	6	6	49	0 Richtige	43,5964976%	
12							

Abbildung 7.44 Gewinnchancen im Lotto

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `HYPGEOMVERT()`.

KGRÖSSTE()

LARGE()

Syntax: `KGRÖSSTE(`**`Matrix; k`**`)`
Beispiel: `=KGRÖSSTE({3;7;5;4;8;2};2)`
Ergebnis: 7

Die Funktion `KGRÖSSTE()` gibt den `k`-größten Wert zurück. Das Argument `k` bestimmt, der wievielt größte Wert aus der angegebenen `Matrix` gesucht wird. `Matrix` kann ein Bereichsbezug oder eine Matrixkonstante sein. Der Bereichsbezug kann auch aus mehreren Bereichen bestehen, wenn die Bereichsadressen noch einmal eingeklammert werden:

```
=KGRÖSSTE((D3:D30;F3:F30);3)
```

Enthält der angegebene Bereich Zeichenfolgen oder leere Zellen, liefert die Funktion den Fehlerwert `#Zahl!`, wenn die entsprechende Zelle durch den Wert für `k` ausgegeben werden müsste. Dieser Fehler erscheint auch, wenn `Matrix` keine numerischen Werte enthält oder leer ist oder wenn $k \leq 0$ oder > als die Zahl der Elemente in `Matrix`. Enthält der Bereich einen Fehlerwert, liefert die Funktion auch diesen Fehlerwert.

Ist `k = 5`, so gibt die Funktion den fünftgrößten Wert als Ergebnis aus; mit `k = 1` wird der größte Wert ausgegeben. In Abbildung 7.45 werden beispielsweise die Umsätze der drei besten Monate ermittelt:

	A	B	C	D	E
1					
2	**Berechnen des k-größten Werts**				
3					
4	Monat	Umsatz	k	KGRÖSSTE()	
5	Jan 13	900000	1	1400000	
6	Feb 13	1090000	2	1300000	
7	Mrz 13	1300000	3	1230000	
8	Apr 13	1200000			
9	Mai 13	900000			
10	Jun 13	990000			
11	Jul 13	30000			
12	Aug 13	1050000			
13	Sep 13	1020000			
14	Okt 13	1200000			
15	Nov 13	1230000			
16	Dez 13	1400000			
17					

Abbildung 7.45 Der größte, der zweitgrößte und der drittgrößte Wert

Statt der einzelnen Formeln in Zelle D5:D7 kann auch eine Matrixformel verwendet werden, wenn dieser Bereich vorher markiert wird:

```
{=KGRÖSSTE(B5:B16;C5:C7)}
```

Für `k` kann in diesem Fall ein Zellbereich oder auch eine Matrixkonstante angegeben werden.

Die Funktion lässt sich auch gut für bedingte Formate verwenden. In Abbildung 7.45 werden beispielsweise für die Umsatzspalte drei Regeln für die Hintergrundfarbe verwendet, und zwar mit der Syntax:

```
=B5=KGRÖSSTE($B$5:$B$16;1)
```

Siehe auch `KKLEINSTE()`.

KKLEINSTE()

SMALL()

Syntax: `KKLEINSTE(`**`Matrix`**`; `**`k`**`)`

Beispiel: `=KKLEINSTE({3;7;5;4;8;2};2)`

Ergebnis: `3`

Die Funktion KKLEINSTE() gibt den k-kleinsten Wert zurück. Das Argument k bestimmt, der wievielt kleinste Wert aus der Matrix gesucht wird. Matrix kann ein Bereichsbezug oder eine Matrixkonstante sein. Der Bereichsbezug kann auch aus mehreren Bereichen bestehen, wenn die Bereichsadressen noch einmal eingeklammert werden:

```
=KKLEINSTE((D3:D30;F3:F30);3)
```

Enthält der angegebene Bereich Zeichenfolgen oder leere Zellen, liefert die Funktion den Fehlerwert #Zahl!, wenn die entsprechende Zelle durch den Wert für k ausgegeben werden müsste. Dieser Fehler erscheint auch, wenn Matrix keine numerischen Werte enthält oder leer ist oder wenn $k \leq 0$ oder > als die Zahl der Elemente in Matrix. Enthält der Bereich einen Fehlerwert, liefert die Funktion auch diesen Fehlerwert.

Ist k = 3, so gibt die Funktion den drittkleinsten Wert als Ergebnis aus; mit k = 1 wird der kleinste Wert ausgegeben. In Abbildung 7.46 werden beispielsweise die Umsätze der drei schlechtesten Monate ermittelt.

	A	B	C	D	E
1					
2	**Berechnen des k-kleinsten Werts**				
3					
4	Monat	Umsatz	k	KKLEINSTE()	
5	Jan 13	910000	1	30000	
6	Feb 13	1090000	2	900000	
7	Mrz 13	1300000	3	910000	
8	Apr 13	1200000			
9	Mai 13	900000			
10	Jun 13	990000			
11	Jul 13	30000			
12	Aug 13	1050000			
13	Sep 13	1020000			
14	Okt 13	1200000			
15	Nov 13	1230000			
16	Dez 13	1300000			
17					

Abbildung 7.46 Der kleinste, der zweitkleinste und der drittkleinste Wert

Statt der einzelnen Formeln in Zelle D5:D7 kann auch eine Matrixformel verwendet werden, wenn dieser Bereich vorher markiert wird:

```
{=KKLEINSTE(B5:B16;C5:C7)}
```

Für k kann in diesem Fall ein Zellbereich oder auch eine Matrixkonstante angegeben werden.

Die Funktion lässt sich auch gut für bedingte Formate verwenden. In Abbildung 7.46 werden beispielsweise für die Umsatzspalte drei Regeln für die Hintergrundfarbe verwendet, und zwar mit der Syntax:

```
=B5=KKLEINSTE($B$5:$B$16;1)
```

Siehe auch `KGRÖSSTE()`.

KONFIDENZ.NORM()

CONFIDENCE.NORM()

Syntax:	`KONFIDENZ.NORM(`**Alpha**`;` **Standardabwn**`;` **Umfang**`)`
Beispiel:	`=KONFIDENZ.NORM(0,05; 2,6; 200)`
Ergebnis:	`0,36033`

Die Funktion `KONFIDENZ.NORM()` dient der Schätzung eines Konfidenzintervalls (auch Vertrauensbereich oder Mutungsintervall genannt) für den Erwartungswert von Zufallsvariablen aus einer normalverteilten Grundgesamtheit anhand einer Stichprobe aus dieser Grundgesamtheit.

Bei ein- wie zweiseitigen Fragestellungen wird ein bestimmter Prozentsatz (`Alpha`) extremer Fälle der Stichprobenverteilung als unwahrscheinlich ausgeschlossen. Diese Extremwerte liegen an dem einen Ende oder an den beiden Enden der Verteilung, wie Abbildung 7.47 und Abbildung 7.48 zeigen. Der übrig bleibende Bereich beidseitig vom Mittelwert ist das Konfidenzintervall. Die entsprechende Fläche in der Dichtefunktion entspricht jeweils `1 – Alpha`, was auch als Konfidenz- oder Vertrauensniveau bezeichnet wird.

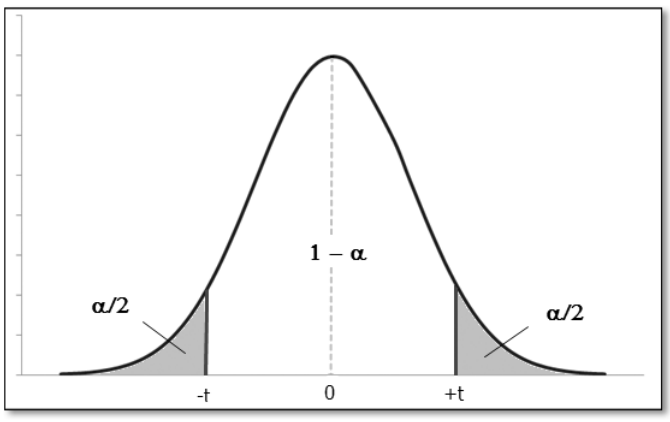


Abbildung 7.47 Konfidenzintervall bei zweiseitigen Fragestellungen

Das Konfidenzintervall ist ein Wertebereich, in dessen Mittelpunkt der Stichprobenmittelwert liegt. Der Bereich bei einer zweiseitigen Fragestellung ist `Mittelwert ± KONFIDENZ.NORM()`.

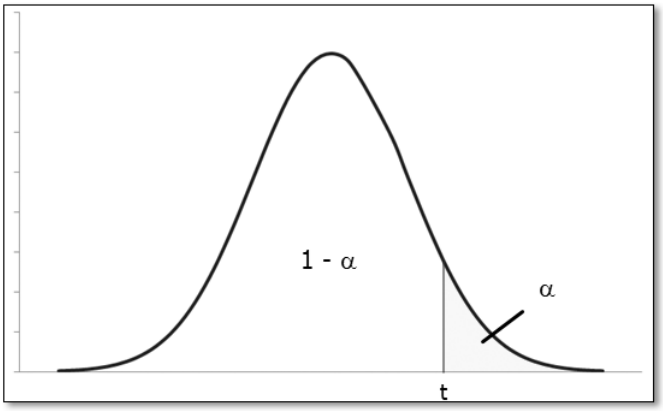


Abbildung 7.48 Konfidenzintervall bei einseitiger Fragestellung

`Alpha` ist die Irrtumswahrscheinlichkeit (gewählt wird zumeist 0,05, 0,01 oder 0,001). `Standardabwn` ist die als bekannt angenommene Standardabweichung der Grundgesamtheit (sie muss > `0` sein). Ist sie nicht bekannt, wird ersatzweise die Standardabweichung der Stichprobe als Schätzwert verwendet. `Umfang` ist die Größe der Stichprobe. `Alpha` muss zwischen `0` und `1` ausschließlich liegen.

Für den Mittelwert der Grundgesamtheit gilt:

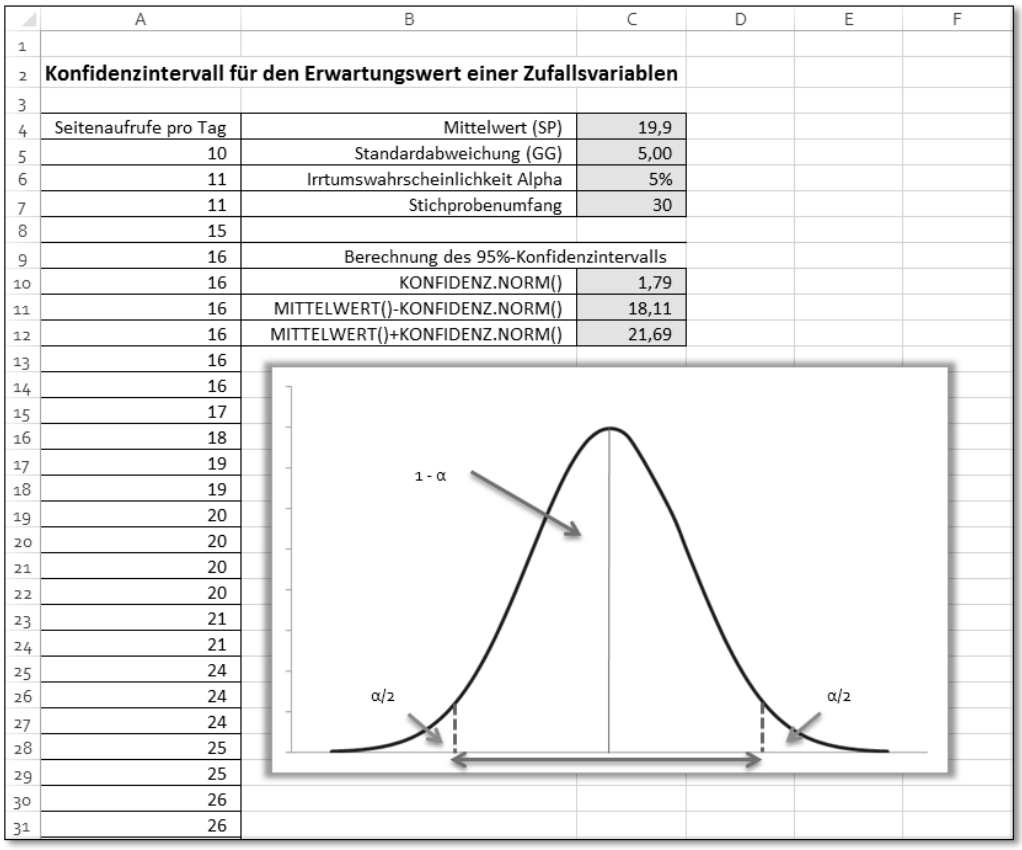
```
Mgg = Mst +- k*(s/WURZEL(n))
```

Dabei sind `Mgg` und `Mst` die Mittelwerte von Grundgesamtheit und Stichprobe, `k` ist der von der Funktion `KONFIDENZ.NORM()` ermittelte Wert, `s` die Standardabweichung der Stichprobe und `n` die Größe der Stichprobe.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Alpha` ≤ `0` oder ≥ `1`, erscheint der Fehler `#ZAHL!`. Das gilt auch, wenn `Standardabwn` gleich `0` oder `Umfang` < `1` ist.

In dem fiktiven Beispiel in Abbildung 7.49 wird das Konfidenzintervall für den vermuteten Mittelwert von Daten zur Häufigkeit von Seitenaufrufen im Web (pro Person und Tag) auf der Basis einer Stichprobe mit 30 Teilnehmern geschätzt. Mit den in Spalte C errechneten Werten kann anschließend die untere und die obere Grenze des Vertrauensbereichs durch Subtrahieren vom und Addieren zum Mittelwert der Stichprobe angegeben werden. Mit einem Vertrauensniveau von `95%` kann für den unbekannten Mit-

telwert der Grundgesamtheit angenommen werden, dass er zwischen 18,11 und 21,69 liegt. Wird der Stichprobenumfang erhöht, wird das Konfidenzintervall verringert.



	A	B	C	D	E	F
1						
2	**Konfidenzintervall für den Erwartungswert einer Zufallsvariablen**					
3						
4	Seitenaufrufe pro Tag	Mittelwert (SP)	19,9			
5	10	Standardabweichung (GG)	5,00			
6	11	Irrtumswahrscheinlichkeit Alpha	5%			
7	11	Stichprobenumfang	30			
8	15					
9	16	Berechnung des 95%-Konfidenzintervalls				
10	16	KONFIDENZ.NORM()	1,79			
11	16	MITTELWERT()-KONFIDENZ.NORM()	18,11			
12	16	MITTELWERT()+KONFIDENZ.NORM()	21,69			
13	16					
14	16					
15	17					
16	18					
17	19					
18	19					
19	20					
20	20					
21	20					
22	20					
23	21					
24	21					
25	24					
26	24					
27	24					
28	25					
29	25					
30	26					
31	26					

Abbildung 7.49 Schätzen des Konfidenzintervalls

Die Funktion KONFIDENZ.NORM() ist eine der beiden Funktionen, die seit Excel 2010 anstelle von KONFIDENZ() angeboten werden. Die andere ist KONFIDENZ.T().

KONFIDENZ.T()

CONFIDENCE.T()

Syntax: KONFIDENZ.T(**Alpha**; **Standardabwn**; **Umfang**)

Beispiel: =KONFIDENZ.T(0,05; 2,6; 200)

Ergebnis: 0,36254

Die Funktion `KONFIDENZ.T()` dient der Schätzung des Konfidenzintervalls (auch Vertrauensbereich oder Mutungsintervall genannt) für den Erwartungswert einer Zufallsvariablen aus einer t-verteilten Grundgesamtheit anhand einer Stichprobe aus dieser Grundgesamtheit.

Insbesondere wenn die Standardabweichung der Grundgesamtheit nicht bekannt ist, wird die Schätzung des Mittelwerts unsicherer, und es ist dann sinnvoll, das Konfidenzintervall etwas breiter anzulegen. Deshalb wird statt von einer Normalverteilung von einer t-Verteilung ausgegangen, die stärker vom Umfang der Stichprobe beeinflusst wird.

Bei ein- wie zweiseitigen Fragestellungen wird ein bestimmter Prozentsatz (`Alpha`) extremer Fälle der Stichprobenverteilung als unwahrscheinlich ausgeschlossen. Diese Extremwerte liegen an den beiden Enden der Verteilung, der Bereich zwischen den beiden Extremwerten beidseitig vom Mittelwert ist das Konfidenzintervall.

	A	B	C	D	E
1					
2	**Konfidenzintervall für den Erwartungswert einer t-verteilten Zufallsvariablen**				
3					
4	Seitenaufrufe pro Tag	Mittelwert (SP)	19,9		
5	10	Standardabweichung (GG)	5,00		
6	11	Irrtumswahrscheinlichkeit Alpha	5%		
7	11	Stichprobenumfang	30		
8	15				
9	16	Berechnung des 95%-Konfidenzintervalls			
10	16	KONFIDENZ.T()	1,87		
11	16	MITTELWERT()-KONFIDENZ.T()	18,03		
12	16	MITTELWERT()+KONFIDENZ.T()	21,77		
13	16				
14	16				
15	17				
16	18				
17	19				
18	19				
19	20				
20	20				
21	20				
22	20				
23	21				
24	21				
25	24				
26	24				
27	24				
28	25				
29	25				
30	26				

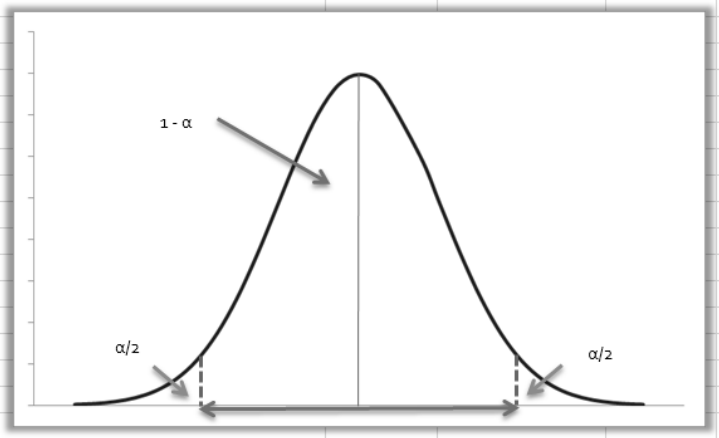


Abbildung 7.50 Schätzen des Konfidenzintervalls bei t-verteilten Variablen

Alpha ist die Irrtumswahrscheinlichkeit (gewählt wird zumeist 0,05, 0,01 oder 0,001), Standardabwn ist die als bekannt angenommene Standardabweichung der Grundgesamtheit (sie muss > 0 sein). Ersatzweise wird die Standardabweichung der Stichprobe verwendet. Umfang ist die Größe der Stichprobe. Alpha muss zwischen 0 und 1 ausschließlich liegen.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Alpha = 0 oder = 1, erscheint der Fehler #ZAHL!. Das gilt auch, wenn Standardabwn gleich 0 ist. Ist Umfang = 1, erscheint der Fehler #DIV/0!.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von KONFIDENZ() angeboten werden. Die andere ist KONFIDENZ.NORM(). Siehe dazu auch T.VERT().

KORREL()

CORREL()

Syntax:	KORREL(**Matrix1**; **Matrix2**)
Beispiel:	=KORREL({1;2;3;4};{2;4;6;8})
Ergebnis:	1

Die Funktion KORREL() liefert den Korrelationskoeffizienten zweier Datenreihen, also von paarweise ermittelten Daten, die mit Hilfe der Argumente Matrix1 und Matrix2 angegeben werden.

Beide Argumente müssen dieselbe Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler #NV. Dabei werden in den angegebenen Zellbereichen oder Matrizen Texteinträge, Wahrheitswerte und leere Elemente ignoriert. Enthalten beide Argumente nur ein oder gar kein Datenelement, erscheint der Fehler #DIV/0!.

Mit der Funktion kann festgestellt werden, ob es einen berechenbaren Zusammenhang, also eine Korrelation, zwischen zwei Variablen gibt.

Anders als bei der Funktion PEARSON(), die ebenfalls einen Korrelationskoeffizienten errechnet, wird beim Einsatz von KORREL() nicht schon vorausgesetzt, dass ein linearer Zusammenhang zwischen den Werten in Matrix1 und Matrix2 besteht. Es wird also auch nicht unterstellt, dass Matrix1 die unabhängigen und Matrix2 die abhängigen Werte enthält. Deshalb wäre in dem Beispiel in Abbildung 7.51 auch das Ergebnis der Funktion das gleiche, wenn die Werte für den Umsatz als Matrix1 und die Werte für den Werbeaufwand als Matrix2 genommen würden.

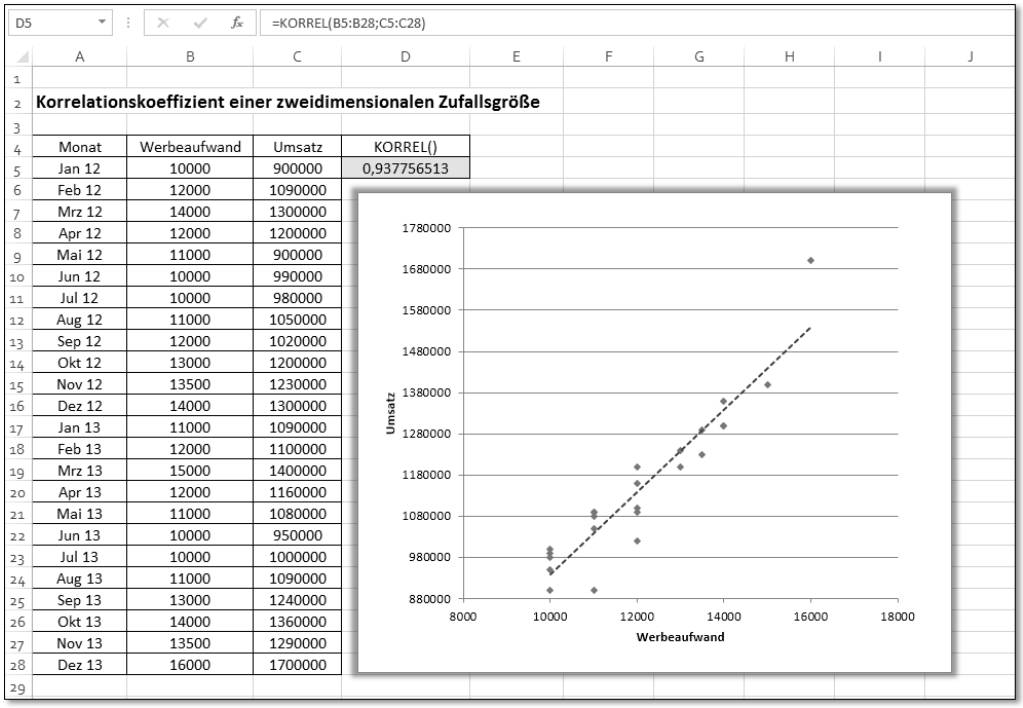


Abbildung 7.51 Korrelation von Datenreihen

Die Funktion ergibt den Wert `1` bei direktem linearem Zusammenhang (die beiden Regressionsgeraden der Daten sind dann direkt proportional). Sie liefert `-1` bei einem indirekten gegenläufigen Zusammenhang, die beiden Regressionsgeraden sind dann umgekehrt proportional; wenn also der Wert aus `Matrix1` größer wird, wird der entsprechende Wert aus `Matrix2` kleiner. Ein Wert von `0` ergibt sich, wenn kein Zusammenhang besteht.

Werte von `0,3` bis `0,5` zeigen einen geringen bis mäßigen Zusammenhang, Werte von `0,7` bis `0,9` zeigen einen engen Zusammenhang, alles, was über `0,9` liegt, deutet auf einen sehr engen Zusammenhang. In Abbildung 7.51 zeigt sich beispielsweise ein sehr enger Zusammenhang zwischen dem Werbeaufwand und dem im gleichen Zeitraum erzielten Umsatz. Steigender Werbeaufwand führt zu steigenden Umsätzen. Die Datenpunkte im Diagramm liegen deshalb auch meist nahe der Regressionsgeraden.

Zu Korrelationen vergleiche auch den Abschnitt 7.4, »Korrelation«, und den Abschnitt 7.7, »Verteilungsfunktionen«.

KOVARIANZ.P()

COVARIANCE.P()

Syntax: `KOVARIANZ.P(`**`Array1`**`; `**`Array2`**`)`
Beispiel: `=KOVARIANZ.P({2; 4; 6; 8; 10; 12}; {12; 2; 10; 4; 8; 6})`
Ergebnis: `-3`

Die Funktion `KOVARIANZ.P()` liefert ähnlich wie die Funktion `KORREL()` ein Maß für den Zusammenhang zwischen den Daten zweier Datenreihen, die mit Hilfe der Argumente `Array1` und `Array2` angegeben werden. Sie liefert die Kovarianz einer Grundgesamtheit, also den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen. So wird ermittelt, in welchem Maß die Daten der beiden Datenreihen gemeinsam von ihrem jeweiligen Mittelwert abweichen. Entscheidend am Ergebnis der Funktion ist nur die Richtung. Positive Werte deuten auf einen linearen Zusammenhang hin (wenn x größer wird, wird auch y größer), negative auf einen gegensinnigen Zusammenhang (wenn x größer wird, wird y kleiner). Null heißt: kein Zusammenhang.

Beide Argumente müssen dieselbe Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Dabei werden in den angegebenen Zellbereichen oder Matrizen Texteinträge, Wahrheitswerte und leere Elemente ignoriert. Enthält eines oder beide Argumente kein numerisches Datenelement, erscheint der Fehler `#DIV/0!`.

	A	B	C	D	E
1					
2	**Kovarianz**				
3					
4	Matrix1	Matrix2	Matrix3	Matrix4	
5	2	12	12	20	
6	4	10	2	40	
7	6	8	10	60	
8	8	6	4	80	
9	10	4	8	100	
10	12	2	6	120	
11					
12	KOVARIANZ.P() 1-2		-11,67		
13	KOVARIANZ.P() 2-3		3		
14	KOVARIANZ.P() 1-4		116,67		
15	KOVARIANZ.P() 1-3		-3		
16					
17	KOVARIANZ.S() 1-2		-14		
18	KOVARIANZ.S() 2-3		3,6		
19	KOVARIANZ.S() 1-4		140		
20	KOVARIANZ.S() 1-3		-3,6		
21					

Abbildung 7.52 Kovarianz zwischen Datenreihen

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von KOVAR() angeboten werden. Die andere ist KOVARIANZ.S().

KOVARIANZ.S()

COVARIANCE.S()

Syntax: KOVARIANZ.S(**Array1**; **Array2**)
Beispiel: =KOVARIANZ.S({2;4;6;8;10;12}; {12;2;10;4;8;6})
Ergebnis: -3,6

Die Funktion KOVARIANZ.S() liefert die Kovarianz von Stichproben, also den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen. Dabei wird die Summe der Produkte aber nicht durch n für die Anzahl der Paare geteilt, sondern durch n – 1.

Die Funktion ermittelt, in welchem Maß die Daten der beiden Datenreihen gemeinsam von ihrem jeweiligen Mittelwert abweichen.

Entscheidend am Ergebnis der Funktion ist nur die Richtung. Positive Werte deuten auf einen linearen Zusammenhang hin (wenn x größer wird, wird auch y größer), negative auf einen gegensinnigen Zusammenhang (wenn x größer wird, wird y kleiner). Null heißt: kein Zusammenhang.

Beide Argumente müssen dieselbe Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler #NV. Dabei werden in den angegebenen Zellbereichen oder Matrizen Texteinträge, Wahrheitswerte und leere Elemente ignoriert. Enthält eines oder beide Argumente kein oder nur ein numerisches Datenelement, erscheint der Fehler #DIV/0!.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von KOVAR() angeboten werden. Die andere ist KOVARIANZ.P(). Siehe dazu auch Abbildung 7.52 zu KOVARIANZ.P().

KURT()

KURT()

Syntax: KURT(**Zahl1**; Zahl2; ...)
Beispiel: =KURT(2;3;2;7;9;6;4;2)
Ergebnis: -0,77

Die Funktion KURT() liefert die Kurtosis, also die Wölbung einer Häufigkeitsverteilung. Während die Funktionen zur Berechnung der Schiefe prüfen, ob der Gipfel der Verteilung links oder rechts vom »normalen« Gipfel angesiedelt ist, prüft KURT(), wie flach oder wie gipflig die Verteilungskurve ist. Verglichen wird mit einer Normalverteilung mit gleichem Mittelwert und gleicher Streuung.

Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen nur 30. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie berücksichtigt.

Sind weniger als vier Datenelemente angegeben oder ist die Standardabweichung der Stichprobe gleich 0, gibt die Funktion den Fehler #DIV/0! zurück. Fehlerwerte in den Datenelementen führen dazu, dass die Funktion diesen Fehler als Ergebnis ausgibt.

Ein negatives Ergebnis weist dabei auf einen stumpferen, flacheren Verlauf hin. Die Werte streuen stark vom Mittelwert weg. Ein positives Ergebnis deutet auf einen steileren, spitzeren Verlauf hin. Die Werte liegen also näher am Mittelwert, als es normalerweise zu erwarten ist. Null entspricht der Normalverteilung.

Wölbung einer Verteilung

Urliste mit Körpergrößen von 70 Schülern

177	174	183	184	185	172	166	176	163	163
168	173	172	177	185	177	174	151	172	173
179	158	181	178	185	167	180	186	158	177
153	162	166	168	177	179	173	157	179	173
172	158	181	179	173	168	184	177	173	196
176	183	177	177	178	184	176	172	177	185
182	160	163	187	162	169	180	172	166	183

80,000	=VAR.S(A5:J11)
8,944	=STABW.S(A5:J11)
-0,461	=SCHIEFE(A5:J11)
0,002	=KURT(A5:J11)
173,843	=MITTELWERT(A5:J11)
175,835	=MEDIAN(A5:J11)
177,000	=MODUS.EINF(A5:J11)

	abs. Häufigkeit	rel. Häufigkeit	normalverteilt
155	2	3%	5%
160	5	7%	9%
165	5	7%	13%
170	8	11%	17%
175	14	20%	18%
180	20	29%	16%
185	13	19%	11%
190	2	3%	7%
195	0	0%	3%
200	1	1%	1%
	70	100%	100%

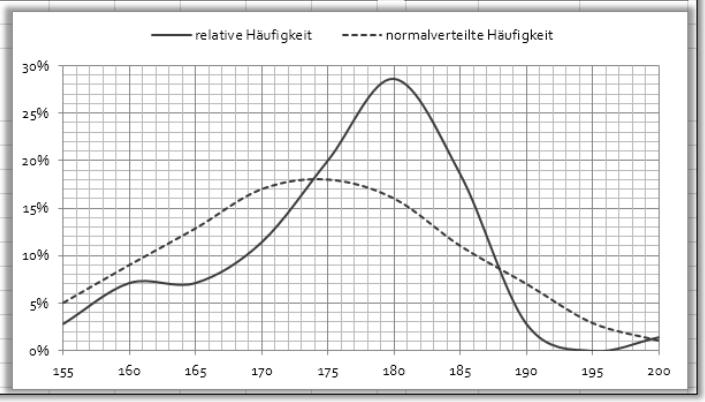


Abbildung 7.53 Wölbung einer realen Verteilung im Vergleich zur Normalverteilung

Abbildung 7.53 zeigt eine Auswertung der Verteilung von Körpergrößen bei Jugendlichen. Im Diagramm sind die relativen Häufigkeiten und die bei einer Normalverteilung zu erwartenden Häufigkeiten gegenübergestellt. Die Kurve der relativen Häufigkeiten ist deutlich steiler.

LOGNORM.INV()

LOGNORM.INV()

Syntax: `LOGNORM.INV(`**`Wahrsch`**`; `**`Mittelwert`**`; `**`Standabwn`**`)`

Beispiel: `=LOGNORM.INV(0,01; 0; 1)`

Ergebnis: `0,098`

Die Funktion `LOGNORM.INV()` liefert Quantile einer logarithmischen Normalverteilung. Die Funktion hilft bei der Analyse von Daten, deren Logarithmus normalverteilt ist. Anwendungsbereiche sind Statistiken über Einkommensverteilungen oder Schadensfälle in der Versicherungsbranche.

Das Argument `Wahrsch` ist die zur Lognormalverteilung gehörende Wahrscheinlichkeit, als Zweites wird der `Mittelwert` von `ln(x)` und als Drittes mit `Standabwn` die Standardabweichung von `ln(x)` angegeben.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` ≤ 0 oder ≥ 1, erscheint der Fehler `#ZAHL!`; das gilt auch für `Standabwn` ≤ 0.

Die Funktion ist die Umkehrung von `LOGNORM.VERT()`. Es gilt also:

Wenn: `LOGNORM.INV(p;...)= x`

dann ist: `LOGNORM.VERT(x;...WAHR)= p`

	A	B	C	D	E
1					
2	**Quantile einer logarithmischen Normalverteilung**				
3					
4	Wahrsch	Mittelwert	Standabwn	LOGNORM.INV()	
5	5%	0	1	0,193040817	
6	25%	0	1	0,509416284	
7	50%	0	1	1	
8	75%	0	1	1,963031084	
9	95%	0	1	5,180251602	
10					

Abbildung 7.54 `LOGNORM.INV()` für verschiedene Wahrscheinlichkeiten

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `LOGINV()`. Siehe dazu auch `LOGNORM.VERT()`.

LOGNORM.VERT()

LOGNORM.DIST()

Syntax: `LOGNORM.VERT(x; Mittelwert; Standabwn; Kumuliert)`

Beispiel: `=LOGNORM.VERT(1; 0; 1; WAHR)`

Ergebnis: `0,5`

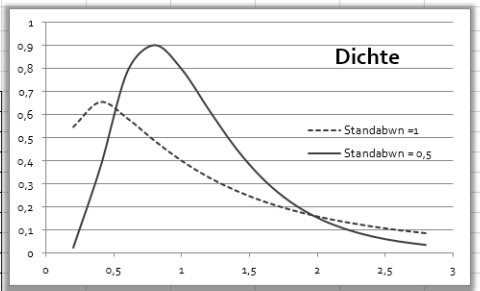
Die Funktion `LOGNORM.VERT()` liefert Werte der Wahrscheinlichkeitsverteilung für eine logarithmische Normalverteilung. Wenn eine Zufallsvariable `y = ln(x)` normalverteilt ist, dann wird die Verteilung von `x` als Lognormalverteilung bezeichnet.

Bei einigen Experimenten, z.B. über Reaktionszeiten, ergibt sich als Häufigkeitsverteilung ein asymmetrischer linkssteiler Kurvenzug. Durch Logarithmieren lassen sich daraus häufig normalverteilte Messwerte erstellen. Auch bei Daten über die Einkommensverteilung ist die Verteilung meist linkssteil, wenn wenige Personen mit sehr hohem Einkommen auf der einen Seite des Mittelwerts stehen und sehr viele Personen mit niedrigem Einkommen auf der anderen Seite.

Wahrscheinlichkeit bei logarithmischer Normalverteilung

Standabwn	1	0,5
Mittelwert	0	0

	LOGNORM.VERT()			
x	Dichte		Verteilung	
0,2	0,546267871	0,022439461	0,05376031	0,00064347
0,4	0,655444168	0,372068229	0,179757214	0,03343242
0,6	0,583573823	0,789108569	0,304736583	0,153473
0,8	0,486415781	0,902818366	0,411711892	0,32769494
1	0,39894228	0,797884561	0,5	0,5
1,2	0,326972024	0,622136841	0,572334809	0,64231109
1,4	0,269276229	0,454439394	0,631742608	0,74950869
1,6	0,223265447	0,320586931	0,680823788	0,82639308
1,8	0,186472449	0,222114038	0,721662252	0,88011729
2	0,156874019	0,152613826	0,755891404	0,91717148
2,2	0,132890686	0,104603451	0,784785385	0,94259243
2,4	0,113309754	0,071779877	0,809340543	0,96002166
2,6	0,097203259	0,049424598	0,83034044	0,97199918
2,8	0,083859205	0,034196121	0,848405652	0,98026432



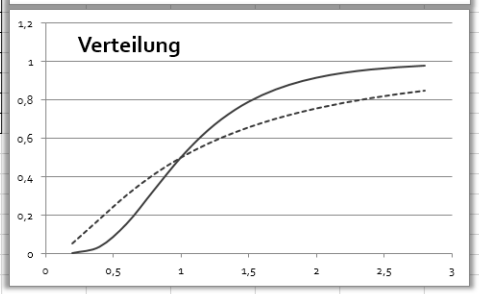


Abbildung 7.55 Verteilung logarithmischer Werte

Das Argument `x` bezeichnet den Wert des Quantils, `Mittelwert` das arithmetische Mittel und `Standabwn` die Standardabweichung der Werte. `Kumuliert` bestimmt den Typ der

Funktion. WAHR liefert die kumulative Verteilungsfunktion, FALSCH den Wert der Dichtefunktion.

Sind x, Mittelwert oder Standabwn nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist x ≤ 0 oder Standabwn ≤ 0, erscheint der Fehler #ZAHL!.

Die Dichtefunktion der Lognormalverteilung wird durch folgende Formel definiert:

$$= \frac{1}{\sigma\sqrt{2\pi}} \frac{1}{x} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$$

Dabei ist μ das arithmetische Mittel und σ die Standardabweichung.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion LOGNORMVERT(). Siehe auch LOGNORM.INV().

MAX()

MAX()

Syntax:	MAX(**Zahl1**; Zahl2; ...)
Beispiel:	=MAX(4;6;2;3)
Ergebnis:	6

Mit der Funktion MAX() wird der größte Wert aus der Liste der Argumente ermittelt. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30.

Jedes Argument kann eine Zahl, ein Bezug oder ein benannter Bereich oder eine Matrixkonstante sein. Ist ein Argument ein Bezug auf einen Bereich oder eine Matrixkonstante, werden darin nur numerische Werte ausgewertet, Texteinträge, Wahrheitswerte und leere Zellen werden dagegen ignoriert. Enthält der Bereich keine Zahlen, ist das Ergebnis 0. Als numerische Werte gelten Zahlen sowie Datums- und Zeitwerte.

Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie als Zahl ausgewertet, also WAHR als 1 und FALSCH als 0.

```
=MAX("-6";WAHR;0)
```

ergibt also 1.

Siehe auch Abbildung 7.58 zu MITTELWERT().

MAXA()

MAXA()

Syntax: `MAXA(`**`Wert1`**`; Wert2; ...)`
Beispiel: `=MAXA(-4;-6;WAHR;-3)`
Ergebnis: `1`

Mit der Funktion `MAXA()` wird der größte Wert aus der Liste der Argumente ermittelt. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30.

Jedes Argument kann eine Zahl, ein Bezug oder ein benannter Bereich oder eine Matrixkonstante sein. Ist ein Argument ein Bezug auf einen Bereich oder eine Matrixkonstante, werden darin nur numerische Werte ausgewertet, als Text eingegebene Werte und leere Zellen werden dagegen ignoriert. Logische Werte werden mit dem Wert `1` für `WAHR` und `0` für `FALSCH` berechnet. Enthält der Bereich keine Werte, ist das Ergebnis `0`. Als numerische Werte gelten Zahlen sowie Datums- und Zeitwerte. Fehlerwerte führen dazu, dass die Funktion den zuerst vorkommenden Fehlerwert ausgibt.

Wird eine Zahl in Textform dagegen direkt als Argument angegeben, wird sie als Zahl ausgewertet.

```
=MAXA(WAHR;"2")
```

ergibt also `2`.

Siehe auch Abbildung 7.60 zu `MITTELWERTA()`.

MEDIAN()

MEDIAN()

Syntax: `MEDIAN(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=MEDIAN(2;3;2;4;1000)`
Ergebnis: `3`

Die Funktion `MEDIAN()` ermittelt den Median einer Reihe von numerischen Werten. Der Median ist der Zentralwert, der genau auf der Mitte einer Skala liegt, deren untere und obere Grenze durch den tiefsten und den höchsten Wert der Zahlenreihe gebildet wird. Über dem Median liegen also in einer nach der Größe sortierten Reihe genauso viele Werte wie unter ihm. Bei einer ungeraden Anzahl von Werten ist der Median einer der

vorhandenen Werte. Bei einer geraden Anzahl von Werten ermittelt Excel den Mittelwert der beiden mittleren Werte, um ihn als Median auszugeben. Der Wert wird in diesem Fall interpoliert, weil er in der Menge der Werte nicht vorkommt. Streng genommen ist dieses Verfahren allerdings nur für metrische Werte sinnvoll.

Der Median wird deshalb auch als lagetypischer Mittelwert bezeichnet. Im Vergleich zu `MITTELWERT()` ist das Ergebnis von `MEDIAN()` unempfindlicher gegenüber einzelnen Extremwerten, wie schon das oben angegebene Beispiel zeigt; `=MITTELWERT(2;3;2;4;1000)` würde `202,2` ergeben. Die Funktion lässt sich nicht nur bei metrischen, sondern auch bei ordinalskalierten Werten – etwa bei der Auswertung von Schulnoten – sinnvoll anwenden.

Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen nur 30. Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Bezüge auf leere Zellen oder Zellen mit Texten oder Wahrheitswerten werden ignoriert. Wird allerdings eine Zahl in Textform oder ein Wahrheitswert direkt als Argument angegeben, werden sie als Zahl ausgewertet.

Der Median im Vergleich zum Mittelwert

Person	Geschlecht	Alter
1	m	23
2	m	25
3	w	29
4	m	33
5	w	35
6	w	35
7	m	43
8	w	43
9	m	43
10	w	43
11	w	45
12	m	46
13	m	47
14	m	47
15	w	51
16	w	53
17	w	53
18	m	54
19	w	55
20	m	63
MIN()		23
MAX()		63
MITTELWERT()		43,3
MEDIAN()		44

Abbildung 7.56 Der Median im Vergleich zum Mittelwert

Abbildung 7.56 zeigt die Berechnung des Medians für das Merkmal »Alter«. Die Werte in Spalte C sind sortiert und die unterschiedlichen Farben zeigen, wo die Mitte der Werte liegt. Da die Zahl der Werte gerade ist, ist der Median der Mittelwert der beiden Werte in den Zellen C14 und C15.

Der Median kann auch mit Hilfe der Funktion `QUARTILE.INCL()` berechnet werden, da er dem Wert des 2. Quartils entspricht.

```
=QUARTILE.INCL(C5:C25;2)
```

ergibt ebenfalls den Wert 44.

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen `MITTELWERT()`, `MITTELWERTA()`, `GEOMITTEL()`, `GESTUTZTMITTEL()`, `HARMITTEL()`, `MODUS.EINF()` sowie `MODUS.VIELF()`.

MIN()

MIN()

Syntax:	`MIN(`**`Zahl1`**`; Zahl2; ...)`
Beispiel:	`=MIN(5;2;3;6;1)`
Ergebnis:	`1`

Mit der Funktion `MIN()` wird der kleinste Wert aus der Liste der Argumente ermittelt. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30.

Jedes Argument kann eine Zahl, ein Bezug oder ein benannter Bereich oder eine Matrixkonstante sein. Ist ein Argument ein Bezug auf einen Bereich oder eine Matrixkonstante, werden darin nur numerische Werte ausgewertet, Texteinträge, Wahrheitswerte und leere Zellen werden dagegen ignoriert. Enthält der Bereich keine Zahlen, ist das Ergebnis `0`. Als numerische Werte gelten Zahlen sowie Datums- und Zeitwerte.

Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie als Zahl ausgewertet, also `WAHR` als `1` und `FALSCH` als `0`.

```
=MIN("6";WAHR;5)
```

ergibt also `1`.

Siehe auch Abbildung 7.58 zu `MITTELWERT()`.

MINA()

MINA()

Syntax: `MINA(`**`Wert1`**`; Wert2; ...)`
Beispiel: `=MINA(5;2;WAHR;6;1)`
Ergebnis: `1`

Mit der Funktion `MINA()` wird der kleinste Wert aus der Liste der Argumente ermittelt. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen 30.

Jedes Argument kann eine Zahl, ein Bezug oder ein benannter Bereich oder eine Matrixkonstante sein. Ist ein Argument ein Bezug auf einen Bereich oder eine Matrixkonstante, werden darin nur numerische Werte ausgewertet, als Text eingegebene Werte und leere Zellen werden dagegen ignoriert. Logische Werte werden mit dem Wert `1` für `WAHR` und `0` für `FALSCH` berechnet. Enthält der Bereich keine Werte, ist das Ergebnis `0`. Als numerische Werte gelten Zahlen sowie Datums- und Zeitwerte. Fehlerwerte führen dazu, dass die Funktion den zuerst vorkommenden Fehlerwert ausgibt.

Wird eine Zahl in Textform dagegen direkt als Argument angegeben, wird sie als Zahl ausgewertet.

```
=MINA(WAHR;"2")
```

ergibt also `1`.

Siehe auch Abbildung 7.60 zu `MITTELWERTA()`.

MITTELABW()

AVEDEV()

Syntax: `MITTELABW(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=MITTELABW(4;6;5;7;3;5)`
Ergebnis: `1`

Die Funktion `MITTELABW()` liefert die mittlere lineare Abweichung der mit `Zahl1` bis `Zahl(n)` angegebenen Argumente. Sie gehört damit zu den Streuungsmaßen in der Statistik. Da sich die negativen und positiven Abweichungen bei der Mittelwertberechnung immer aufheben, bestimmt die Funktion stattdessen den Durchschnitt der Absolutwerte der gemessenen Abweichungen aller Messwerte vom arithmetischen Mittel.

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen nur 30. Wahrheitswerte und Zahlen in Textform werden nur berücksichtigt, wenn sie direkt als Argument angegeben werden, bei Bezügen auf Zellbereiche werden Wahrheitswerte, nicht numerische Werte und leere Zellen ignoriert.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler `#ZAHL!`.

In Abbildung 7.57 sind in den Spalten E und F als Beleg für die Arbeitsweise der Funktion die Absolutwerte der Abweichungen von den in der Zeile 25 berechneten Mittelwerten eingetragen.

```
=ABS($C$25-C5)
```

ist die erste Formel. Die Mittelwerte dieser Werte in den Zellen E25 und F25 entsprechen den Ergebnissen der Funktion `MITTELABW()` in der Zeile 26.

	A	B	C	D	E	F	G
1							
2	**Berechnung der mittleren linearen Abweichung**						
3							
4	Person	Geschlecht	Alter	Gewicht	Abweichung 1	Abweichung 2	
5	1	m	23	69	20,3	2,9	
6	2	m	25	66	18,3	0,1	
7	3	w	29	68	14,3	1,9	
8	4	m	33	72	10,3	5,9	
9	5	w	35	74	8,3	7,9	
10	6	w	35	63	8,3	3,1	
11	7	m	43	68	0,3	1,9	
12	8	w	43	79	0,3	12,9	
13	9	m	43	58	0,3	8,1	
14	10	w	43	79	0,3	12,9	
15	11	w	45	68	1,7	1,9	
16	12	m	46	72	2,7	5,9	
17	13	m	47	77	3,7	10,9	
18	14	m	47	56	3,7	10,1	
19	15	w	51	60	7,7	6,1	
20	16	w	53	62	9,7	4,1	
21	17	w	53	58	9,7	8,1	
22	18	m	54	61	10,7	5,1	
23	19	w	55	55	11,7	11,1	
24	20	m	63	57	19,7	9,1	
25	MITTELWERT()		43,3	66,1	8,1	6,5	
26	MITTELABW()		8,1	6,5			
27	STABW.S()		10,6	7,7			
28							

Abbildung 7.57 Mittelwert und mittlere lineare Abweichung

Anstelle der durchschnittlichen Abweichung wird in der statistischen Praxis heute meist mit der Standardabweichung gearbeitet, also mit der mittleren quadratischen Abweichung, die ein zuverlässigeres Streuungsmaß liefert.

MITTELWERT()

AVERAGE()

Syntax: `MITTELWERT(Zahl1; Zahl2; ...)`
Beispiel: `=MITTELWERT(33;22;28;17;23;26)`
Ergebnis: `24,83`

Die Funktion `MITTELWERT()` liefert das arithmetische Mittel der mit `Zahl1` bis `Zahl(n)` angegebenen Argumente. Diese wohl bekannteste Lagemaßzahl wird auch als Mittelwert bezeichnet. Sie liefert die Mitte von metrischen Daten. Dabei werden alle Werte aufsummiert und durch die Zahl der Werte geteilt.

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Seit Excel 2007 sind insgesamt 255 Argumente möglich, in den älteren Versionen nur 30.

Innerhalb eines Bezugs oder einer Matrix werden Text- oder Wahrheitswerte oder leere Zellen ignoriert. Fehlerwerte führen dazu, dass die Funktion diesen Fehlerwert ausgibt. Wahrheitswerte und Zahlen in Textform werden nur berücksichtigt, wenn sie direkt als Argument angegeben werden.

`=MITTELWERT(WAHR; FALSCH)`

ergibt `0,5`, aber:

`=MITTELWERT(K2:K3)`

ergibt `#DIV/0!`, wenn in K2 und K3 die Fehlerwerte `WAHR` und `FALSCH` stehen, weil diese Zellen ignoriert werden und die Funktion dann eine Division durch 0 versuchen muss.

Im Gegensatz zu leeren Zellen werden Zellen mit dem Wert `0` in jedem Fall berücksichtigt, also auch dann, wenn die Anzeige von Nullen über **Excel-Optionen • Erweitert • Optionen für dieses Arbeitsblatt anzeigen • In Zellen mit Nullwert eine Null anzeigen** abgeschaltet ist.

Abbildung 7.58 zeigt eine Tabelle mit den Mittelwerten zu den Merkmalen »Alter« und »Gewicht« bei einer Gruppe von Personen. Zum Vergleich sind noch einige andere Lagemaße angegeben.

	A	B	C	D	E
1					
2	**Minimum, Maximum und Mittelwerte berechnen**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	29	68	
8	4	m	33	72	
9	5	w	35	74	
10	6	w	35	63	
11	7	m	43	68	
12	8	w	43	79	
13	9	m	43	58	
14	10	w	43	79	
15	11	w	45	68	
16	12	m	46	72	
17	13	m	47	77	
18	14	m	47	56	
19	15	w	51	60	
20	16	w	53	62	
21	17	w	53	58	
22	18	m	54	61	
23	19	w	55	55	
24	20	m	63	57	
25		MIN()	23	55	
26		MAX()	63	79	
27		MITTELWERT()	43,3	66,1	
28		MEDIAN()	44,0	67,0	
29		MODUS.EINF()	43,0	68,0	
30					

Abbildung 7.58 Mittelwerte für zwei Merkmale

Fehlerwerte führen dazu, dass die Funktion den ersten Fehlerwert als Ergebnis ausgibt. (Um Bereiche mit Fehlerwerten dennoch statistisch auswerten zu können, sollten Sie die Funktion `AGGREGAT()` verwenden, die in der Lage ist, Fehlerwerte zu ignorieren.)

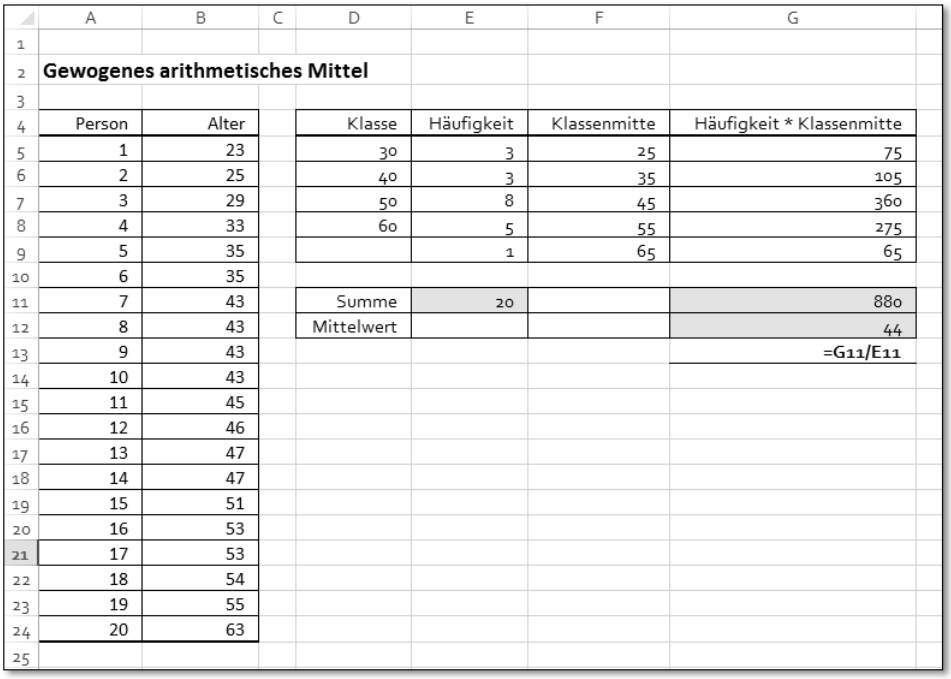
Auch wenn die vorhandenen Daten eine Berechnung des arithmetischen Mittels erlauben, ist das Ergebnis der Funktion nicht immer sinnvoll. Beispiel dafür ist etwa die mittlere Kinderzahl der Frauen in Deutschland, die zwischen 1 und 2 liegt.

7.9.5 Mittelwert bei klassifizierten Daten

Wenn die Daten, deren Mittel berechnet werden soll, bereits als klassifizierte Daten vorliegen, wird mit dem gewogenen Mittel gearbeitet. Abbildung 7.59 zeigt neben der Urliste mit Werten zum Alter eine Tabelle, die zunächst in Spalte E die Häufigkeiten in den verschiedenen Altersklassen anzeigt. Diese lassen sich mit der Funktion `HÄUFIGKEIT()` berechnen. In die Spalte daneben werden die Klassenmitten eingetragen, also jeweils

der Mittelwert zwischen dem höchsten und niedrigsten Wert in der Klasse. Diese Mitten werden in der nächsten Spalte mit den Häufigkeiten multipliziert, um sie zu gewichten.

In Zelle E11 wird dann zunächst die Summe der Häufigkeiten und in Zelle G12 die Summen der gewichteten Mittel berechnet. Um das Mittel der klassifizierten Daten zu finden, wird schließlich die Summe der gewichteten Mittel durch die Summen der Häufigkeiten geteilt.



Gewogenes arithmetisches Mittel

Person	Alter
1	23
2	25
3	29
4	33
5	35
6	35
7	43
8	43
9	43
10	43
11	45
12	46
13	47
14	47
15	51
16	53
17	53
18	54
19	55
20	63

Klasse	Häufigkeit	Klassenmitte	Häufigkeit * Klassenmitte
30	3	25	75
40	3	35	105
50	8	45	360
60	5	55	275
	1	65	65
Summe	20		880
Mittelwert			44
			=G11/E11

Abbildung 7.59 Mittelwert für klassifizierte Daten

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen `MITTELWERTA()`, `GEOMITTEL()`, `GESTUTZTMITTEL()`, `HARMITTEL()`, `MEDIAN()`, `MODUS.EINF()` sowie `MODUS.VIELF()`.

MITTELWERTA()

AVERAGEA()

Syntax: `MITTELWERTA(`**`Wert1`**`; Wert2; ...)`

Beispiel: `=MITTELWERTA(10;20;WAHR)`

Ergebnis: `10,33`

Die Funktion MITTELWERTA() liefert das arithmetische Mittel der mit Zahl1 bis Zahl(n) angegebenen Argumente. Diese wohl bekannteste Lagemaßzahl wird auch als Mittelwert bezeichnet. Sie liefert die Mitte von metrischen Daten. Dabei werden alle Werte aufsummiert und durch die Zahl der Werte geteilt.

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Seit Excel 2007 sind insgesamt 255 Argumente möglich, in den älteren Versionen nur 30.

Anders als bei der verwandten Funktion MITTELWERT() werden Argumente, die Zahlen in Textform enthalten, ebenso berücksichtigt wie Wahrheitswerte. Texte, die nicht als Zahl ausgewertet werden können, werden mit dem Wert 0 berechnet, Wahrheitswerte mit 1 für WAHR, 0 für FALSCH. Bezüge auf Zellen, die Zahlen in Textform enthalten, werden ebenfalls mit dem Wert 0 berechnet. Zellen mit dem Wert 0 werden in jedem Fall berücksichtigt, also auch dann, wenn die Anzeige von Nullen über **Excel-Optionen • Erweitert • Optionen für dieses Arbeitsblatt anzeigen • In Zellen mit Nullwert eine Null anzeigen** abgeschaltet ist.

Fehlerwerte führen dazu, dass die Funktion den ersten Fehlerwert als Ergebnis ausgibt.

	A	B	C
1			
2	**Mittelwert aller Werte**		
3			
4	Spender	Spende	
5	Hansen	120,00 €	
6	Lauber	100,00 €	
7	Korn	60,00 €	
8	Bernstein	90,00 €	
9	Otto	angefragt	
10	Klare	angefragt	
11	Bonin	200,00 €	
12	Kerr	170,00 €	
13	Horst	120,00 €	
14	Emen	abgelehnt	
15	SUMME()	860,00 €	
16	ANZAHL2()	10	
17	MITTELWERTA()	86,00 €	
18	ANZAHL()	7	
19	MITTELWERT()	122,86 €	
20			

Abbildung 7.60 Diese Funktionsvariante berücksichtigt nicht nur numerische Werte in der Spalte B.

Fragt sich, wann die Anwendung dieser Funktion sinnvoll ist. Das Beispiel in Abbildung 7.60 deutet an, in welchen Fällen sie von Nutzen sein kann. Der errechnete Mittelwert

in Zelle B17 legt den Gesamtbetrag der Spenden auf alle tatsächlichen und potenziellen Spender um, also nicht nur auf die, die schon etwas gespendet haben. Die Funktion ermittelt also die Anzahl, durch die der Gesamtbetrag dividiert wird, auf die gleiche Weise wie die Funktion ANZAHL2(). Das Ergebnis ist deshalb kleiner als das der Funktion MITTELWERT(), die die Gesamtsumme durch den Wert teilt, den die Funktion ANZAHL() ermitteln würde.

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen MITTELWERT(), GEOMITTEL(), GESTUTZTMITTEL(), HARMITTEL(), MEDIAN(), MODUS.EINF() sowie MODUS.VIELF().

MITTELWERTWENN()

AVERAGEIF()

Syntax: MITTELWERTWENN(**Bereich**; **Kriterien**; Mittelwert_Bereich)
Beispiel: =MITTELWERTWENN(C5:C24;">40";D5:D24)
Ergebnis: 65 (für die in Abbildung 7.61 gezeigten Werte)

Die seit Excel 2007 angebotene Funktion MITTELWERTWENN() erlaubt es, das arithmetische Mittel von Werten zu berechnen, die vorher mit einem Kriterium ausgefiltert werden. Das Argument Bereich gibt den Bereich an, auf den das mit Kriterien angegebene Kriterium angewendet werden soll. Bereich kann ein Bezug auf einen Bereich, ein Bereichsname oder eine Matrixkonstante sein. Wahrheitswerte in Bereich werden ignoriert.

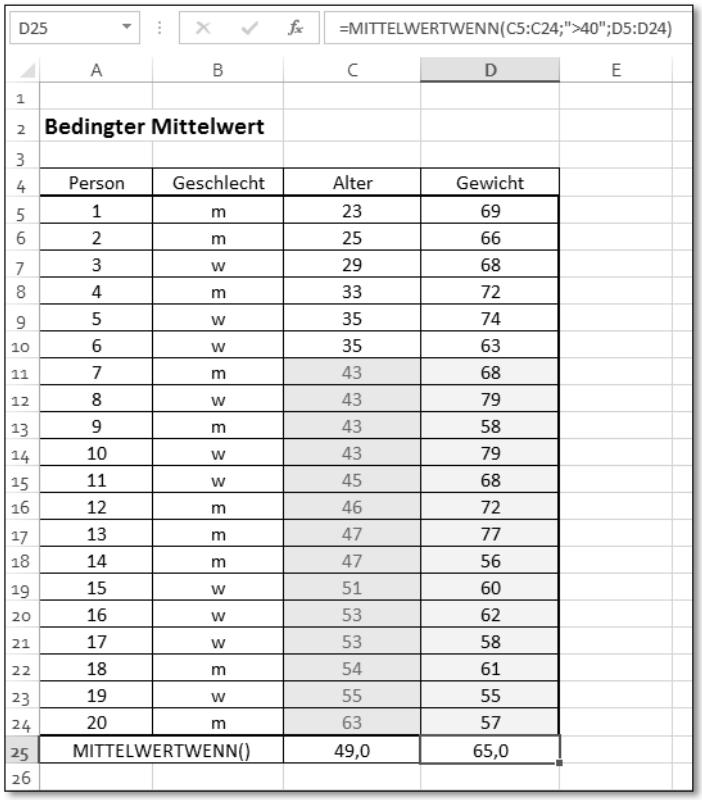
Enthält Bereich keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!. Dieser Fehler erscheint auch, wenn keines der Elemente in Bereich den Kriterien entspricht.

Kriterien können Konstanten oder einfache Vergleiche wie ">40" oder auch Zelladressen sein, die auf entsprechende Inhalte verweisen. Auch Joker wie ? für ein beliebiges Zeichen oder * für eine beliebige Anzahl von Zeichen sind erlaubt.

Wird das Argument Mittelwert_Bereich angegeben, wird der Mittelwert für die Werte in diesem Zellbereich gebildet. Ansonsten wird der Mittelwert für die in Bereich vorhandenen Werte berechnet. Die angegebene Größe von Mittelwert_Bereich kann von der von Bereich abweichen, der Funktion genügt es, wenn die linke obere Eckzelle von Mittelwert_Bereich bekannt ist. Sie dehnt den Bereich dann automatisch von dort auf die Größe von Bereich aus.

In dem Beispiel in Abbildung 7.61 wird in C25 das mittlere Alter der Personen berechnet, die älter als 40 sind. Die Formel ist:

```
=MITTELWERTWENN(C5:C24;">40")
```



	A	B	C	D	E
1					
2	**Bedingter Mittelwert**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	29	68	
8	4	m	33	72	
9	5	w	35	74	
10	6	w	35	63	
11	7	m	43	68	
12	8	w	43	79	
13	9	m	43	58	
14	10	w	43	79	
15	11	w	45	68	
16	12	m	46	72	
17	13	m	47	77	
18	14	m	47	56	
19	15	w	51	60	
20	16	w	53	62	
21	17	w	53	58	
22	18	m	54	61	
23	19	w	55	55	
24	20	m	63	57	
25	MITTELWERTWENN()		49,0	65,0	
26					

Abbildung 7.61 Bedingter Mittelwert mit einem Kriterium

Der Bereich, in dem mit dem Kriterium gefiltert wird, ist also zugleich der, für den das Mittel errechnet wird.

In D25 dagegen wird der Mittelwert der Gewichtswerte für die Personen errechnet, deren Alter größer als 40 ist. Deshalb wird in der Formel das Argument `Mittelwert_Bereich` eingesetzt:

```
=MITTELWERT(C5:C24;">40";D5:D24)
```

Die Suche findet in diesem Fall also in einem anderen Bereich statt als die Berechnung des Mittelwerts.

Die Funktion MITTELWERTWENN() ist eine häufig verwendete Alternative zu der Funktion DBMITTELWERT() aus der Kategorie der Datenbankfunktionen. Der Vorteil ist, dass Sie nicht extra einen Kriterienbereich anlegen müssen.

Zur Bedeutung des arithmetischen Mittels siehe auch die Beschreibung der Funktion MITTELWERT(). Um mit mehr als einem Kriterium zu arbeiten, verwenden Sie die Funktion MITTELWERTWENNS().

Bedingte Zusammenfassungen von Werten stellt Excel auch über andere Funktionen zur Verfügung: SUMMEWENN(), SUMMEWENNS(), ZÄHLENWENN(), ZÄHLENWENNS(), MITTELWERTWENNS().

MITTELWERTWENNS()

AVERAGEIFS()

Syntax:	MITTELWERTWENNS(**Mittelwert_Bereich**; **Kriterien_Bereich1**; **Kriterien1**; Kriterien_Bereich2; Kriterien2 ...)
Beispiel:	=MITTELWERTWENNS(D5:D24;C5:C24;">27";C5:C24;"<55")
Ergebnis:	67,2 (für die in Abbildung 7.62 gezeigten Werte)

Die seit Excel 2007 angebotene Funktion MITTELWERTWENNS() erlaubt es, das arithmetische Mittel aus Werten in dem mit Mittelwert_Bereich angegebenen Zellbereich auszurechnen, die mit einem oder mehreren Kriterien gefiltert werden.

Mittelwert_Bereich kann ein Bezug auf einen Bereich, ein Bereichsname oder eine Matrixkonstante sein. Wahrheitswerte werden mit 1 für WAHR und 0 für FALSCH gewertet.

Enthält Mittelwert_Bereich keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!. Dieser Fehler erscheint auch, wenn keines der Elemente in Mittelwert_Bereich den Kriterien entspricht.

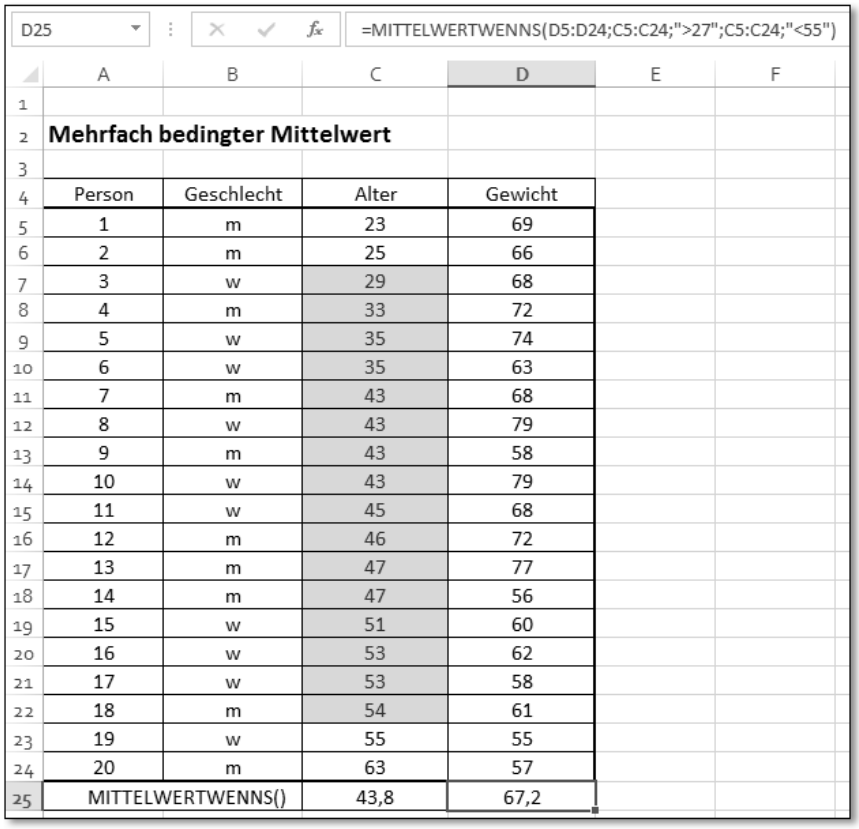
Für jeden Filter wird jeweils mit Kriterien_Bereich(n) der Bereich angegeben, auf den das mit Kriterien(n) angegebene Kriterium angewendet werden soll. Anders als bei der Funktion MITTELWERTWENN() müssen die Bereiche die gleiche Größe haben wie Mittelwert_Bereich, sonst liefert die Funktion den Fehler #WERT!.

Kriterien können Konstanten oder einfache Vergleiche wie ">27" oder auch Zelladressen sein, die auf entsprechende Inhalte verweisen. Auch Joker wie ? für ein beliebiges Zeichen oder * für eine beliebige Anzahl von Zeichen sind erlaubt. Bis zu 127 Paare von Kriterienbereichen und Kriterien sind möglich.

In dem Beispiel in Abbildung 7.62 wird in D25 der Mittelwert der Gewichtswerte für die Personen errechnet, deren Alter größer als 27 und kleiner als 55 ist. Die Formel

```
=MITTELWERTWENNS(D5:D25;C5:C25;">27";C5:C25";"<55")
```

verwendet für die beiden Kriterien in diesem Fall also zweimal den gleichen Kriterienbereich.



D25 | =MITTELWERTWENNS(D5:D24;C5:C24;">27";C5:C24;"<55")

	A	B	C	D	E	F
1						
2	**Mehrfach bedingter Mittelwert**					
3						
4	Person	Geschlecht	Alter	Gewicht		
5	1	m	23	69		
6	2	m	25	66		
7	3	w	29	68		
8	4	m	33	72		
9	5	w	35	74		
10	6	w	35	63		
11	7	m	43	68		
12	8	w	43	79		
13	9	m	43	58		
14	10	w	43	79		
15	11	w	45	68		
16	12	m	46	72		
17	13	m	47	77		
18	14	m	47	56		
19	15	w	51	60		
20	16	w	53	62		
21	17	w	53	58		
22	18	m	54	61		
23	19	w	55	55		
24	20	m	63	57		
25	MITTELWERTWENNS()		43,8	67,2		

Abbildung 7.62 Bedingter Mittelwert mit zwei Kriterien

Zur Bedeutung des arithmetischen Mittels siehe auch die Beschreibung der Funktion `MITTELWERT()`.

Bedingte Zusammenfassungen von Werten stellt Excel auch über andere Funktionen zur Verfügung: `SUMMEWENN()`, `SUMMEWENNS()`, `ZÄHLENWENN()`, `ZÄHLENWENNS()`, `MITTELWERTWENN()`.

MODUS.EINF()

MODE.SNGL()

Syntax: `MODUS.EINF(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=MODUS.EINF(2; 6; 3; 6; 1; 5; 6)`
Ergebnis: `6`

Die Funktion `MODUS.EINF()` ermittelt aus den mit den Argumenten `Zahl1` bis `Zahl(n)` angegebenen Daten den am häufigsten vorkommenden Wert. Damit gehört die Funktion zu den grundlegenden statistischen Kennwerten für Maße der zentralen Tendenz. Mit dem Modalwert oder Modus lassen sich schnell Informationen über den Schwerpunkt einer Verteilung gewinnen. Betrachten Sie eine Verteilung, so ist das Maximum der Verteilung gleich dem Modalwert.

	A	B	C	D	E
1					
2	**Modalwerte berechnen**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	35	68	
8	4	m	35	72	
9	5	w	34	74	
10	6	w	35	63	
11	7	m	43	68	
12	8	w	22	79	
13	9	m	43	58	
14	10	w	43	79	
15	11	w	45	68	
16	12	m	47	72	
17	13	m	47	77	
18	14	m	47	56	
19	15	w	47	60	
20	16	w	33	62	
21	17	w	53	58	
22	18	m	53	61	
23	19	w	53	55	
24	20	m	63	57	
25		MODUS.EINF()	47	68	
26		MODALWERT()	47	68	
27					

Abbildung 7.63 Einfacher Modalwert

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Seit Excel 2007 sind insgesamt 255 Argumente möglich, in den älteren Versionen nur 30. Wahrheitswerte und Zahlen in Textform werden nur berücksichtigt, wenn sie direkt als Argument angegeben werden. Bei gleich häufigem Vorkommen verschiedener Werte wird der in der Liste zuerst vorkommende ausgegeben.

Enthält eines der Argumente einen Fehlerwert, gibt auch die Funktion diesen Fehlerwert aus. Kann die Funktion keinen Modalwert angeben, weil keiner der Werte zumindest zweimal vorkommt, wird der Fehlerwert `#NV` ausgegeben.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von `MODALWERT()` angeboten werden. Die andere ist `MODUS.VIELF()`.

Alternativen zur Berechnung von Mittelwerten bieten die Funktionen `MITTELWERT()`, `MITTELWERTA()`, `GEOMITTEL()`, `GESTUTZTMITTEL()`, `HARMITTEL()`, `MEDIAN()` sowie `MODUS.VIELF()`.

MODUS.VIELF()

MODE.MULT()

Syntax: `MODUS.VIELF(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=MODUS.VIELF(2; 6; 3; 6; 3; 5; 6)`

Ergebnis: `6 und 3`

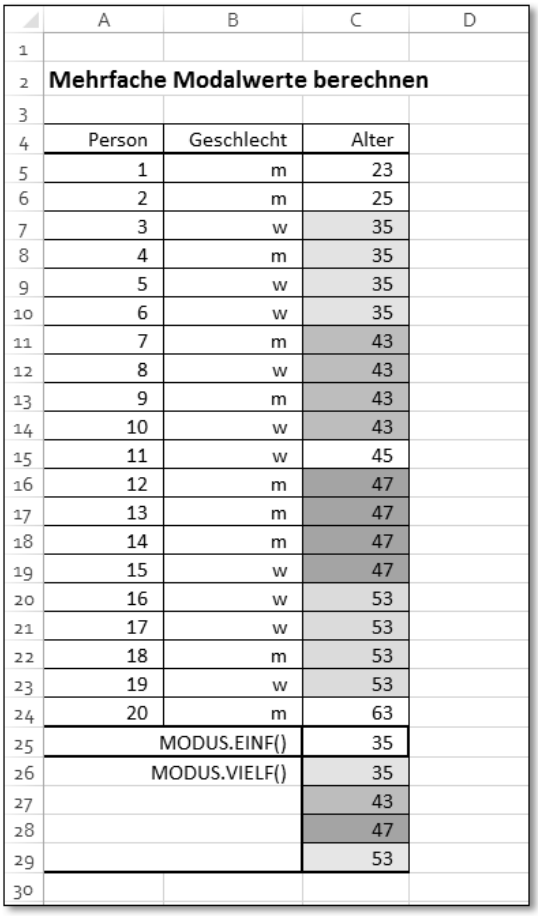
Die Funktion `MODUS.VIELF()` ermittelt aus den mit den Argumenten `Zahl1` bis `Zahl(n)` angegebenen Daten die am häufigsten vorkommenden Werte. Die Anwendung der Funktion ist hauptsächlich bei mehrgipfligen Verteilungen sinnvoll.

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Bis zu 255 Argumente sind möglich. Wahrheitswerte und Zahlen in Textform werden nur berücksichtigt, wenn sie direkt als Argument angegeben werden.

Enthält eines der Argumente einen Fehlerwert, gibt auch die Funktion diesen Fehlerwert aus. Kann die Funktion keine Modalwerte angeben, weil keiner der Werte zumindest zweimal vorkommt, wird der Fehler `#NV` ausgegeben.

Die Funktion muss innerhalb eines vertikalen Arrays verwendet werden, um mehrere Werte zu liefern. Sie müssen also vor der Formeleingabe einen genügend großen Be-

reich markieren und die Formeleingabe mit Strg + ⇧ + ↵ abschließen. Ist der Bereich zu groß gewählt, erscheint in den Zellen, für die keine Modalwerte angezeigt werden können, der Fehler #NV. Ist der Bereich zu klein, um alle Modalwerte anzuzeigen, werden nur die in der Reihenfolge ersten Werte angezeigt.



	A	B	C	D
1				
2	**Mehrfache Modalwerte berechnen**			
3				
4	Person	Geschlecht	Alter	
5	1	m	23	
6	2	m	25	
7	3	w	35	
8	4	m	35	
9	5	w	35	
10	6	w	35	
11	7	m	43	
12	8	w	43	
13	9	m	43	
14	10	w	43	
15	11	w	45	
16	12	m	47	
17	13	m	47	
18	14	m	47	
19	15	w	47	
20	16	w	53	
21	17	w	53	
22	18	m	53	
23	19	w	53	
24	20	m	63	
25		MODUS.EINF()	35	
26		MODUS.VIELF()	35	
27			43	
28			47	
29			53	
30				

Abbildung 7.64 Mehrere Modalwerte

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von MODALWERT() angeboten werden. Die andere ist MODUS.EINF().

Weitere Funktionen, die mit der Berechnung von Mittelwerten zu tun haben, sind MITTELWERT(), MITTELWERTA(), GEOMITTEL(), GESTUTZTMITTEL(), HARMITTEL() sowie MEDIAN().

NEGBINOM.VERT()

NEGBINOM.DIST()

Syntax:	NEGBINOM.VERT(**Zahl_Misserfolge**; **Zahl_Erfolge**; **Erfolgswahrsch**; **Kumuliert**)
Beispiel:	=NEGBINOM.VERT(5; 1; 1/6; FALSCH)
Ergebnis:	0,067

Die Funktion NEGBINOM.VERT() benutzt als Grundlage ihrer Berechnungen ebenso wie BINOM.VERT() die Binomialverteilung und wird auch als negative Binomialverteilung bezeichnet. Sie berechnet, mit welcher Wahrscheinlichkeit ein zusammengesetztes Ereignis auftritt: erst eine bestimmte Zahl von Misserfolgen, dann eine bestimmte Zahl von Erfolgen.

Als Argumente werden Zahl_Misserfolge und Zahl_Erfolge angegeben. Erfolgswahrsch gibt die Wahrscheinlichkeit für den erfolgreichen Ausgang an. Kumuliert bestimmt den Typ der Funktion. WAHR liefert die kumulative Verteilungsfunktion, FALSCH den Wert der Dichtefunktion.

Ist Zahl_Erfolge < 0 oder Zahl_Erfolge < 1, gibt die Funktion den Fehlerwert #ZAHL! zurück. Das gilt ebenfalls für Erfolgswahrsch < 0 oder > 1. Ist eines der Argumente Zahl_Misserfolge, Zahl_Erfolge, Erfolgswahrsch nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Im Beispiel oben wird die Wahrscheinlichkeit ermittelt, hintereinander genau fünfmal nicht die Sechs und dann die Sechs zu werfen. Abbildung 7.65 zeigt weitere Berechnungen für den Münzwurf.

	A	B	C	D	E
1					
2	**Wahrscheinlichkeit von negativbinomialverteilten Zufallsvariablen**				
3					
4	Zahl_Misserfolge	Zahl_Erfolge	Erfolgswahrsch	NEGBINOM.VERT()	
5	0	5	0,5	3,13%	
6	3	5	0,5	13,67%	
7	6	5	0,5	10,25%	
8	10	5	0,5	3,05%	
9	20	5	0,5	0,03%	
10					

Abbildung 7.65 Berechnen der Wahrscheinlichkeit bei zusammengesetzten Ereignissen

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion NEGBINOMVERT(). Siehe auch BINOM.VERT().

NORM.INV()

NORM.INV()

Syntax:	NORM.INV(**Wahrsch**; **Mittelwert**; **Standabwn**)
Beispiel:	=NORM.INV(0,5; 20; 30)
Ergebnis:	20

Die Funktion NORM.INV() liefert Quantile der Normalverteilung und ist die Umkehrung von NORM.VERT() (siehe dort).

Als Argumente werden Wahrsch (die Wahrscheinlichkeit, mit der das Quantil gesucht wird) sowie Mittelwert und Standabwn (Standardabweichung) der Verteilung angegeben. Wie bei der Normalverteilung gilt auch hier, dass bei Mittelwert = 0 und Standardabweichung = 1 eine Standardnormalverteilung vorliegt. In diesem Fall kann auch NORM.S.INV() aufgerufen werden.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch < 0 oder > 1, erscheint der Fehler #ZAHL!; das gilt auch für Standabwn ≤ 0.

Die Funktion sucht in einem iterativen Verfahren zu einem für Wahrsch angegebenen Wert einen Wert x, so dass gilt:

```
NORM.VERT(x; Mittelwert; Standabwn; WAHR) = Wahrsch
```

Ist nach 100 Schritten keine ausreichende Annäherung erreicht, liefert die Funktion den Fehler #NV.

Die Tabelle in Abbildung 7.66 zeigt in der Spalte D die Werte an, die die Funktion für die 25%-, 50%- und 75%-Quantile ausrechnet. 25 % der Werte sind nicht größer als 896,48, die Hälfte der Werte ist nicht größer als der Mittelwert, 75 % der Werte erreichen maximal 1503,52.

	A	B	C	D	E
1					
2	**Berechnen von Quantilen der Normalverteilung**				
3					
4	Wahrsch	Mittelwert	Standabwn	NORM.INV()	
5	25%	1200	450	896,48	
6	50%	1200	450	1200,00	
7	75%	1200	450	1503,52	
8					

Abbildung 7.66 Rückrechnen von der Wahrscheinlichkeit in einer Normalverteilung auf das Quantil

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion NORMINV(). Siehe auch NORM.VERT(), NORM.S.INV(), NORM.S.VERT().

NORM.S.INV()

NORM.S.INV()

Syntax: NORM.S.INV(**Wahrsch**)
Beispiel: =NORM.S.INV(0,90)
Ergebnis: 1,28

Die Funktion NORM.S.INV() liefert Quantile der Standardnormalverteilung für die mit Wahrsch angegebene Wahrscheinlichkeit. Die Funktion ist die Umkehrung von NORM.S.VERT().

Ist Wahrsch < 0 oder > 1, erscheint der Fehler #ZAHL!, ist der Wert nicht numerisch, der Fehler #WERT!.

	A	B	C
1			
2	**Quantile der Standardnormalverteilung**		
3			
4	z	NORM.S.INV()	
5	5%	-1,64	
6	10%	-1,28	
7	15%	-1,04	
8	20%	-0,84	
9	25%	-0,67	
10	30%	-0,52	
11	35%	-0,39	
12	40%	-0,25	
13	45%	-0,13	
14	50%	0,00	
15	55%	0,13	
16	60%	0,25	
17	65%	0,39	
18	70%	0,52	
19	75%	0,67	
20	80%	0,84	
21	85%	1,04	
22	90%	1,28	
23	95%	1,64	
24			

Abbildung 7.67 Berechnen des Quantils zu einer Wahrscheinlichkeit

Die Funktion sucht in einem iterativen Verfahren zu einem für Wahrsch angegebenen Wert einen Wert z, so dass gilt:

NORM.S.VERT(z; WAHR) = Wahrsch

Ist nach 100 Schritten keine ausreichende Annäherung erreicht, liefert die Funktion den Fehler #NV.

Die Standardnormalverteilung ist eine Variante der Normalverteilung und dadurch gekennzeichnet, dass der Mittelwert gleich 0 ist und die Standardabweichung gleich 1.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion STANDNORMINV(). Siehe auch NORM.INV(), NORM.VERT(), NORM.S.VERT().

NORM.S.VERT()

NORM.S.DIST()

Syntax:	NORM.S.VERT(**z**; **Kumuliert**)
Beispiel:	=NORM.S.VERT(2; WAHR)
Ergebnis:	0,9772

Die Funktion NORM.S.VERT() gibt die Wahrscheinlichkeit dafür aus, dass eine Zufallsvariable aus einer Standardnormalverteilung den Wert z oder kleiner annimmt.

z bezeichnet den Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit berechnet werden soll. Wird die Funktion grafisch dargestellt, ist das der Wert auf der x-Achse. Dabei ergibt sich für die Darstellung der Dichte ein glockenförmiger Verlauf.

Mit Kumuliert = WAHR erhalten Sie die Verteilungsfunktion (die Wahrscheinlichkeit dafür, dass die Zufallsvariable einen Wert von z oder kleiner annimmt). Mit FALSCH erhalten Sie die Werte der Dichtefunktion.

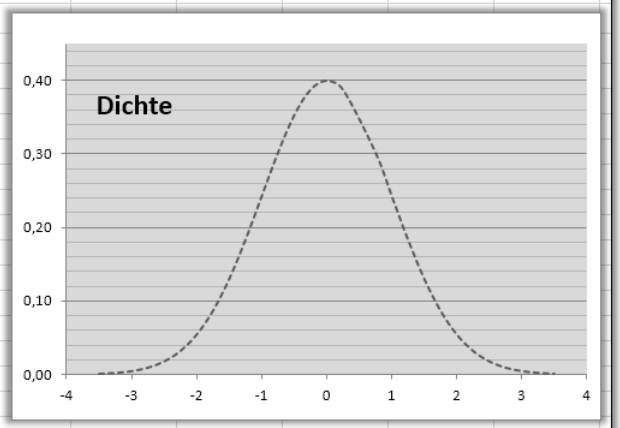
Ist z nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Die Standardnormalverteilung ist eine Variante der Normalverteilung und dadurch gekennzeichnet, dass der Mittelwert gleich 0 ist und die Standardabweichung gleich 1. Die von dieser Funktion ermittelten Werte ließen sich deshalb auch über NORM.VERT(z; 0; 1; WAHR) berechnen. Soll die Dichtefunktion berechnet werden, muss mit NORM.VERT(z; 0; 1; FALSCH) gearbeitet werden.

Mit Hilfe der Funktion STANDARDISIERUNG() kann jede Normalverteilung in eine Standardnormalverteilung transformiert werden, siehe STANDARDISIERUNG().

Standardnormalverteilung

	NORM.S.VERT()	
z	Dichte	Verteilung
-3,5	0,0009	0,0002
-3,25	0,0020	0,0006
-3	0,0044	0,0013
-2,75	0,0091	0,0030
-2,5	0,0175	0,0062
-2,25	0,0317	0,0122
-2	0,0540	0,0228
-1,75	0,0863	0,0401
-1,5	0,1295	0,0668
-1,25	0,1826	0,1056
-1	0,2420	0,1587
-0,75	0,3011	0,2266
-0,5	0,3521	0,3085
-0,25	0,3867	0,4013
0	0,3989	0,5000
0,25	0,3867	0,5987
0,75	0,3011	0,7734
1	0,2420	0,8413
1,25	0,1826	0,8944
1,5	0,1295	0,9332
1,75	0,0863	0,9599
2	0,0540	0,9772
2,25	0,0317	0,9878
2,5	0,0175	0,9938
2,75	0,0091	0,9970
3	0,0044	0,9987
3,25	0,0020	0,9994
3,5	0,0009	0,9998



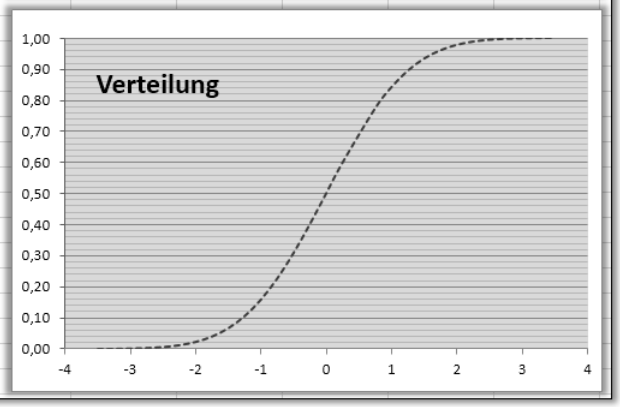


Abbildung 7.68 Die Standardnormalverteilung und der Glocken-Graph

Abbildung 7.69 zeigt den Graphen der Dichtefunktion, der die Spalte B auswertet, und den Graphen der Verteilungsfunktion, der die Spalte C auswertet.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `STANDNORMVERT()`. Siehe auch `NORM.INV()`, `NORM.VERT()`, `NORM.S.INV()`.

NORM.VERT()

NORM.DIST()

Syntax: `NORM.VERT(`**`x`**`; `**`Mittelwert`**`; `**`Standabwn`**`; `**`Kumuliert`**`)`

Beispiel: `=NORM.VERT(9; 9; 4; WAHR)`

Ergebnis: `0,5`

Die Funktion NORM.VERT() liefert Wahrscheinlichkeitswerte einer normalverteilten Zufallsvariablen. x bezeichnet den Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit berechnet werden soll. Wird die Funktion grafisch dargestellt, ist das der Wert auf der x-Achse. Dabei ergibt sich für die Darstellung der Dichte ein glockenförmiger Verlauf. Wie er im Einzelnen ausfällt, hängt von den Argumenten Mittelwert und Standabwn ab.

Mittelwert gibt zugleich den Erwartungswert an, also den wahrscheinlichsten Wert. Er gibt die Lage der Funktion auf der x-Achse an und markiert dabei den Gipfel dieser Funktion.

Standabwn (Standardabweichung) gibt die Streuung an und bestimmt damit, wie flach oder steil die Funktion verläuft. Wie die Berechnung der Standardabweichung erfolgt, hängt davon ab, ob es sich bei den Messdaten um eine Stichprobe oder um eine Grundgesamtheit handelt. Im ersten Fall verwenden Sie die Funktion STABW.S(), im zweiten Fall STABW.N().

Mit Kumuliert = WAHR erhalten Sie die Verteilungsfunktion (die Wahrscheinlichkeit dafür, dass die Zufallsvariable einen Wert von x oder kleiner annimmt). Mit FALSCH erhalten Sie die Werte der Dichtefunktion.

Mit Mittelwert = 0 und Standabwn = 1 erhalten Sie die Standardnormalverteilung, die Sie auch mit NORM.S.VERT() direkt erhalten können.

Sind x, Mittelwert oder Standabwn nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Standabwn ≤ 0, erscheint der Fehler #ZAHL!.

7.9.6 Allgemeine Merkmale der Normalverteilung

Eine Zufallsgröße heißt normalverteilt, wenn ihre Dichte durch folgende Formel definiert ist:

$$= \frac{1}{\left(\sqrt{2\pi}\right)\sigma} e^{-\frac{1}{2}\frac{(x-a)^2}{\sigma^2}}$$

Dabei sind a (Erwartungswert = Mittelwert) und σ (Standardabweichung) die Parameter der Verteilung.

Ein besonderes Merkmal der Normalverteilung ist, dass die quadrierten Abweichungen der Einzelwerte vom Mittelwert selbst wieder normalverteilt sind. Deshalb lässt sich ein

präzises mathematisches Modell für zufällige Abweichungen von einem Mittelwert angeben. Aus diesem Grund kann die Normalverteilung in vielen Situationen, in denen es um Abweichungen vom Mittelwert geht, die von einer Vielzahl unabhängiger Faktoren beeinflusst sind, wenigstens als erste Annäherung angewendet werden, auch wenn die Verteilung nicht rein zufällig ist.

Außerdem gelten folgende Eigenschaften:

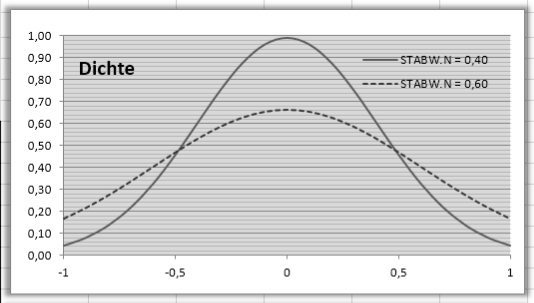
Die Verteilung ist glockenförmig und eingipflig. Sie nähert sich asymptotisch der x-Achse. Zugleich ist sie symmetrisch. Der höchste Wert ist auch der Mittelwert, wobei der arithmetische Mittelwert mit dem Median zusammenfällt. Da der Mittelwert die Glockenfläche genau in der Mitte teilt, hat der Mittelwert die Wahrscheinlichkeit von 50%. Je 50% der Fläche liegen also beidseitig vom Mittelwert. Das heißt, die Hälfte der Fälle liegt unter, die andere Hälfte über dem Mittelwert.

Die Wendepunkte liegen bei `Mittelwert + Standardabweichung` bzw. `Mittelwert – Standardabweichung`. Zwischen den beiden Wendepunkten liegen 68,3% der Fälle.

Normalverteilung

MITTELWERT()	0	0
STABW.N()	0,40	0,60

x	Verteilung		Dichte	
-1	0,00656	0,04779	0,04562	0,16580
-0,9	0,01279	0,06681	0,08186	0,21586
-0,8	0,02360	0,09121	0,13812	0,27335
-0,7	0,04124	0,12167	0,21913	0,33666
-0,6	0,06832	0,15866	0,32690	0,40328
-0,5	0,10742	0,20233	0,45858	0,46985
-0,4	0,16053	0,25249	0,60489	0,53241
-0,3	0,22838	0,30854	0,75027	0,58678
-0,2	0,30990	0,36944	0,87505	0,62897
-0,1	0,40204	0,43382	0,95967	0,65573
0	0,50000	0,50000	0,98965	0,66490
0,1	0,59796	0,56618	0,95967	0,65573
0,2	0,69010	0,63056	0,87505	0,62897
0,3	0,77162	0,69146	0,75027	0,58678
0,4	0,83947	0,74751	0,60489	0,53241
0,5	0,89258	0,79767	0,45858	0,46985
0,6	0,93168	0,84134	0,32690	0,40328
0,7	0,95876	0,87833	0,21913	0,33666
0,8	0,97640	0,90879	0,13812	0,27335
0,9	0,98721	0,93319	0,08186	0,21586
1	0,99344	0,95221	0,04562	0,16580



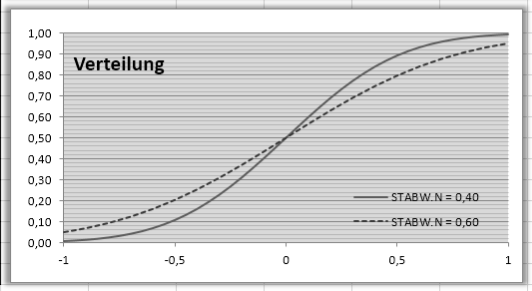


Abbildung 7.69 Normalverteilungsfunktion und Dichtefunktion

Die Fläche unter der Dichtekurve hat immer den Wert 1. Die Wahrscheinlichkeit, dass eine Zufallsvariable einen Wert zwischen `x1` und `x2` annimmt, wird ermittelt, indem die

entsprechende Fläche unter der Dichtekurve berechnet wird. Dazu kann einfach der Funktionswert von `x1` von dem von `x2` abgezogen werden:

```
=NORM.VERT(x1; Mittelwert; Standabwn; FALSCH) -
NORM.VERT(x2; Mittelwert; Standabwn; FALSCH)
```

Die Normalverteilung wird auch als Gaußsche Fehlerkurve bezeichnet, was damit zusammenhängt, dass der bekannte Astronom und Mathematiker sich bei Messungen der Planetenbahnen mit der Streuung von Messfehlern beschäftigte und dabei auf diese Form der Verteilung stieß.

Sie ist die wichtigste und auch die bekannteste unter den stetigen Verteilungen. Sie lässt sich aus der Binomialverteilung ableiten, indem der Wert für `n`, also die Anzahl der Ereignisse, gegen unendlich ausgeweitet wird. Dadurch gelingt der Übergang von einer diskreten zu einer stetigen Verteilung.

7.9.7 Wahrscheinlichkeit einer Größe

Wenn davon ausgegangen wird, dass die Körpergröße in einem bestimmten Altersjahrgang normalverteilt ist, und sowohl die Durchschnittsgröße als auch die Standardabweichung bekannt sind, kann die Wahrscheinlichkeit berechnet werden, dass eine Person eine bestimmte Größe hat oder dass seine Größe in einem bestimmten Intervall liegt.

In dem folgenden Beispiel soll herausgefunden werden, wie wahrscheinlich es ist, dass eine Person eine Größe von 2 m oder mehr hat. In E33 ist ein hypothetischer Mittelwert angegeben, in E34 die Standardabweichung. Mit der kumulierten Variante von `NORM.VERT()` kann berechnet werden, wie wahrscheinlich eine Person 2 m groß oder größer ist. Da in diesem Fall der noch verbleibende Flächenteil an der rechten Seite der Dichteglocke interessiert, muss der errechnete Wert von 1 (das ist ja die gesamte Fläche) abgezogen werden, um den Rest der Fläche zu erhalten.

	A	B	C	D	E	F	G	H
30								
31	**Größenwahrscheinlichkeit**							
32								
33	Mittlere Größe eines Jahrgangs bei Männern				175			
34	Standardabweichung				12			
35	Wahrscheinlichkeit einer Größe von 2 m und mehr?				2%	**=1-NORM.VERT(200;E33;E34;WAHR)**		
36								

Abbildung 7.70 Berechnen der kumulierten Wahrscheinlichkeit

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion NORMVERT(). Siehe auch NORM.INV(), NORM.S.VERT() und NORM.S.INV().

PEARSON()

PEARSON()

Syntax: PEARSON(**Matrix1**; **Matrix2**)
Beispiel: =PEARSON({1;2;3;4};{10;9;8,7})
Ergebnis: -1

Die Funktion PEARSON() liefert den Pearsonschen Korrelationskoeffizienten r zweier Datenreihen, der auch als empirischer Korrelationskoeffizient bezeichnet wird. Er gibt an, in welchem Umfang eine lineare Abhängigkeit zwischen zwei metrischen Merkmalen besteht.

Beide Argumente müssen dieselbe Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler #NV. Dabei werden in den angegebenen Matrizen oder Zellbereichen Texteinträge, Wahrheitswerte und leere Elemente ignoriert. Enthalten beide Argumente nur ein oder gar kein Datenelement, erscheint der Fehler #DIV/0!.

Die Funktion ergibt den Wert 1 bei direktem linearem Zusammenhang (die beiden Regressionsgeraden der Daten sind dann direkt proportional). Sie liefert -1 bei einem indirekten gegenläufigen Zusammenhang, die beiden Regressionsgeraden sind dann umgekehrt proportional; wenn also der Wert aus Matrix1 größer wird, wird der entsprechende Wert aus Matrix2 kleiner. Ein Wert von 0 ergibt sich, wenn kein Zusammenhang besteht. Werte von 0,3 bis 0,5 zeigen einen geringen bis mäßigen Zusammenhang, Werte von 0,7 bis 0,9 zeigen einen engen Zusammenhang. Alles, was über 0,9 liegt, deutet auf einen sehr engen Zusammenhang. In der Abbildung 7.71 zeigt sich beispielsweise ein sehr enger Zusammenhang zwischen dem Werbeaufwand und dem im gleichen Zeitraum erzielten Umsatz. Steigender Werbeaufwand führt zu steigenden Umsätzen. Die Datenpunkte im Diagramm liegen deshalb auch meist nahe der Regressionsgeraden.

Beachtet werden muss, dass die Funktion nicht lineare Abhängigkeiten nicht erfassen kann. Es ist deshalb sinnvoll, über die Punktverteilung in einem entsprechenden Diagramm zu prüfen, ob nicht beispielsweise ein exponentieller Zusammenhang vorliegen könnte. Zu Korrelationen vergleiche auch den Abschnitt 7.4, »Korrelation«, und den Abschnitt 7.7, »Verteilungsfunktionen«.

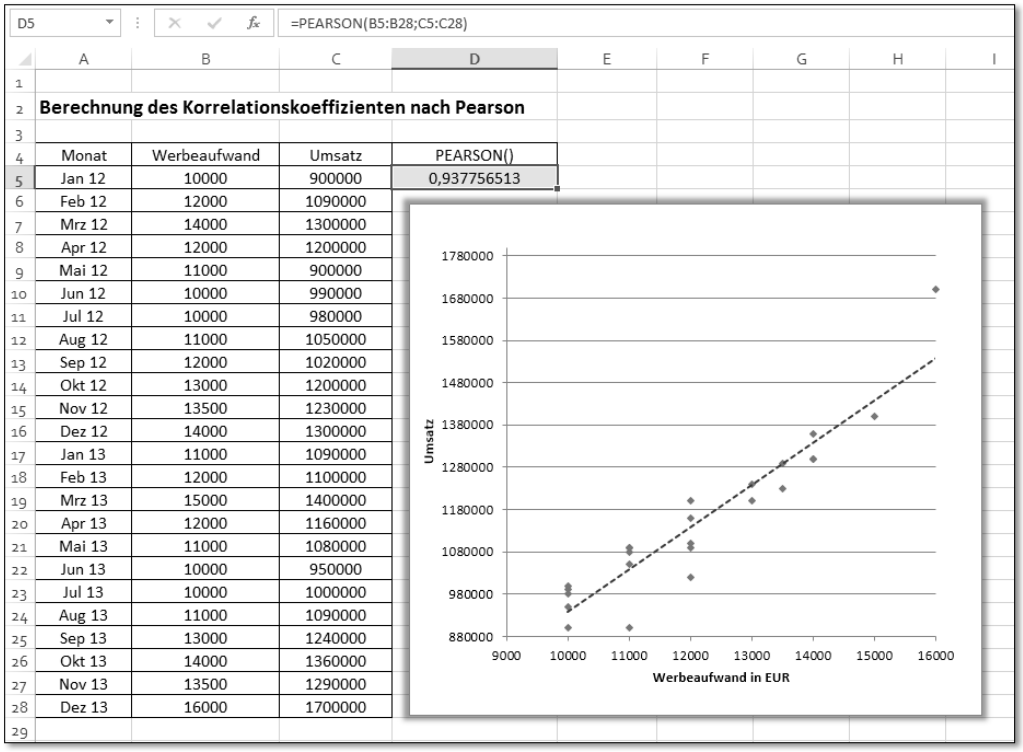


Abbildung 7.71 Der Korrelationskoeffizient nach Pearson

Im Unterschied zur Funktion `KORREL()` wird bei dieser Funktion bereits davon ausgegangen, dass zwischen den paarweise mit Hilfe der Argumente `Matrix1` und `Matrix2` angegebenen Daten eine lineare Abhängigkeit besteht, wobei `Matrix1` die unabhängigen Werte angibt und `Matrix2` die abhängigen. Es geht also nur noch darum, zu prüfen, wie stark die Korrelation ist.

Siehe auch `KORREL()` und den Abschnitt 7.4, »Korrelation«.

PHI()

PHI()

Syntax: `PHI(x)`
Beispiel: `=PHI(0,5)`
Ergebnis: `0,352`

	A	B	C
1			
2	**PHI()**		
3			
4	z	PHI()	NORM.S.VERT()
5	-3,5	0,0009	0,0009
6	-3,25	0,0020	0,0020
7	-3	0,0044	0,0044
8	-2,75	0,0091	0,0091
9	-2,5	0,0175	0,0175
10	-2,25	0,0317	0,0317
11	-2	0,0540	0,0540
12	-1,75	0,0863	0,0863
13	-1,5	0,1295	0,1295
14	-1,25	0,1826	0,1826
15	-1	0,2420	0,2420
16	-0,75	0,3011	0,3011
17	-0,5	0,3521	0,3521
18	-0,25	0,3867	0,3867
19	0	0,3989	0,3989
20	0,25	0,3867	0,3867
21	0,75	0,3011	0,3011
22	1	0,2420	0,2420

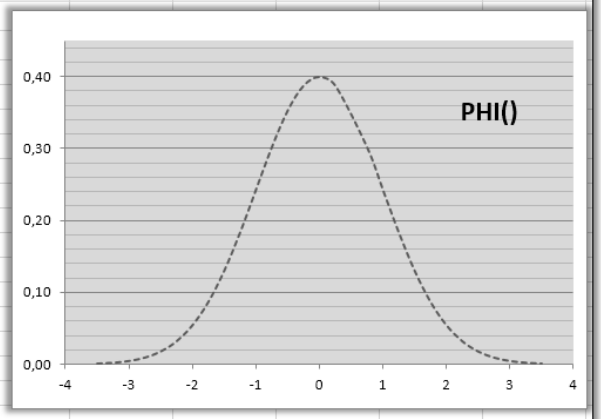


Abbildung 7.72 Werte und Graph der Dichtefunktion der Standardnormalverteilung

Die in Excel 2013 neu eingeführte Funktion `PHI()` liefert den Wert der Dichtefunktion für eine Standardnormalverteilung, das Ergebnis entspricht also dem der Funktion `NORM.S.VERT(x;FALSCH)`.

Das Argument `x` ist der Wert, für den Sie die Dichte ermitteln wollen.

Ist `x` nicht numerisch, liefert die Funktion den Fehler `#WERT!`.

Siehe auch `NORM.S.VERT()`.

POISSON.VERT()

POISSON.DIST()

Syntax:	`POISSON.VERT(`**`x; Mittelwert; Kumuliert`**`)`
Beispiel:	`=POISSON.VERT(50; 60; WAHR)`
Ergebnis:	`0,1077`

Die Funktion `POISSON.VERT()` liefert die Werte für eine Poisson-Verteilung. Die Poisson-Verteilung ist wie die Binomial- und die hypergeometrische Verteilung eine Verteilung, die nur jeweils diskrete Werte annehmen kann. Sie ist für große Zahlen eine gute Näherung an die Binomialverteilung.

An Argumenten verlangt die Funktion `x` für die Anzahl der Ereignisse und `Mittelwert` für den Erwartungswert, das ist hier die durchschnittliche Anzahl von Ereignissen in einer Bezugseinheit, also beispielsweise die durchschnittliche Anzahl von Fehlern je Monat oder je Fertigungsserie. Mit `Kumuliert = FALSCH` wird die Wahrscheinlichkeit dafür berechnet, dass die Zufallsvariable den Wert `x` annimmt, mit `Kumuliert = WAHR` die Wahrscheinlichkeit dafür, dass die Zufallsvariable einen Wert von `x` oder kleiner annimmt.

Ist `x` oder `Mittelwert` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `x < 0` oder `Mittelwert < 0`, erscheint der Fehler `#ZAHL!`.

Die Funktion kann angewendet werden, wenn für eine sehr große Zahl von Fällen die Wahrscheinlichkeit von seltenen Ausnahmeereignissen geschätzt werden soll. Dabei muss nur bekannt sein, wie häufig im Durchschnitt das Ausnahmeereignis auftritt.

Die Funktion arbeitet für die Dichtefunktion mit der Formel:

$$= \frac{\lambda^x}{x!} * e^{-\lambda}$$

Dabei steht λ für die durchschnittliche Anzahl der Ereignisse, `e` für die Eulersche Zahl.

Poisson-Verteilung

x	POISSON.VERT()	kumuliert
0	1%	1%
1	3%	4%
2	8%	12%
3	14%	27%
4	18%	44%
5	18%	62%
6	15%	76%
7	10%	87%
8	7%	93%
9	4%	97%
10	2%	99%

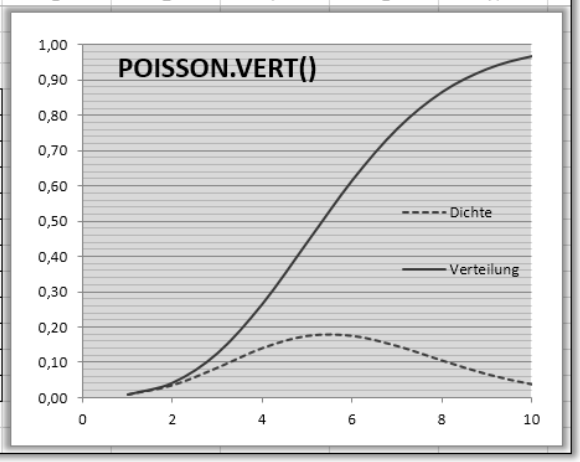


Abbildung 7.73 Wahrscheinlichkeit von Variablen, die eine Poisson-Verteilung darstellen

Die Tabelle in Abbildung 7.73 kann beispielsweise benutzt werden, um die Wahrscheinlichkeit von Bitübertragungsfehlern im Netz zu schätzen. Angenommen, bei einer Übertragung von 1 Million Bits treten durchschnittlich 5 Fehler auf, dann kann die Funktion

POISSON.VERT() die Frage beantworten, mit welcher Wahrscheinlichkeit bei der nächsten Million die Fehleranzahl x auftritt.

Da die Poisson-Verteilung normalerweise dazu verwendet wurde, die bei großen Zahlen schwer zu berechnende Binomialverteilung anzunähern, gibt es allerdings selten einen Grund, sie noch zu verwenden. Schließlich nimmt Ihnen die Funktion BINOM.VERT() die ganze Rechenarbeit ab.

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion POISSON().

QUANTIL.EXKL()

PERCENTILE.EXC()

Syntax: QUANTIL.EXKL(**Array**; **k**)
Beispiel: siehe Abbildung 7.74

Die Funktion QUANTIL.EXKL() liefert denjenigen Wert einer Datenreihe, unterhalb dessen ein mit k angegebener Bruchteil der Daten liegt. Mit dieser Funktion wird eine Verteilung nach einer Skala unterteilt, deren unterster und deren oberster Punkt den tiefsten und den höchsten Wert der Datenreihe bilden. Die mit der Funktion berechneten Lagemaße werden Quantile genannt.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von QUANTIL() angeboten werden. Die andere ist QUANTIL.INKL().

Array sind die zu unterteilenden Daten. Sie müssen nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein.

Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert. Enthält Array gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn k ≤ 0 oder ≥ 1 ist. Ist k nicht numerisch, erscheint der Fehler #WERT!.

Mit dem Argument k wird ein bestimmtes Lagemaß (Quantil) angegeben. Das Maß 0,25 (25 %) bezeichnet z. B. den Punkt, unterhalb dessen ein Viertel aller Beobachtungen liegen. Einige Quantile, die besonders oft verwendet werden, haben eigene Bezeichnungen wie Quartil für 25%-Abschnitte, Dezil für 10%-Abschnitte, Perzentile für 1%-Abschnitte. Das zweite Quartil oder ein Quantil von 0,5 entspricht dem Median.

Das Argument k kann jeden Wert zwischen 0 und 1 annehmen, bei dieser Funktion aber ausschließlich 0 und 1, worauf das Suffix des Funktionsnamens hinweist; liegt ein Quantil zwischen zwei Beobachtungen, wird durch Interpolation der entsprechende

Wert ermittelt. Das geschieht auch, wenn `k` kein Vielfaches von `1/n – 1` ist, wobei `n` die Anzahl der Werte in `Array` ist. Wenn bei einem bestimmten Wert für `k` keine Interpolation möglich ist, liefert die Funktion den Fehler `#ZAHL!`.

Die Tabelle in Abbildung 7.74 kann so gelesen werden: 50% der erfassten Personen haben ein Gewicht von unter 67 kg.

Quantile berechnen

Person	Gewicht	k(Alpha)	QUANTIL.EXKL()	QUANTIL.INKL()	QUANTIL()	
1	69	0,0		55	55	
2	66	0,1	56	57	57	
3	68	0,2	58	58	58	
4	72	0,3	60	61	61	
5	74	0,4	62	63	63	
6	63	0,5	67	67	67	=MEDIAN()
7	68	0,6	68	68	68	
8	79	0,7	71	70	70	
9	58	0,8	74	72	72	
10	79	0,9	79	77	77	
11	68	1,0		79	79	
12	72					
13	77					
14	56					
15	60					
16	62					
17	58					
18	61					
19	55					
20	57					

Abbildung 7.74 Einteilung von Daten in Quantile

Siehe auch `QUANTIL()`, `QUANTIL.INKL()`, `QUANTILSRANG.EXKL()`, `QUANTILSRANG.INKL()`, `QUARTILE()`, `QUARTILE.EXKL()` und `QUARTILE.INKL()`.

QUANTIL.INKL()

PERCENTILE.INC()

Syntax: `QUANTIL.INKL(`**`Array`**`;` **`k`**`)`

Beispiel: siehe Abbildung 7.74 zu `QUANTIL.EXKL()`

Die Funktion QUANTIL.INKL() entspricht der Funktion QUANTIL.EXKL() bis auf die möglichen Werte für k. In diesem Fall reicht der Wertebereich von 0 bis 1, diesmal also einschließlich von 0 und 1, worauf das Suffix des Funktionsnamens hinweist. Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von QUANTIL() angeboten werden. Sie liefert das gleiche Ergebnis wie die ältere Funktion. Die andere der beiden Funktionen ist QUANTIL.EXKL().

Array sind die zu unterteilenden Daten. Sie müssen nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein.

Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert. Enthält Array aber gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn k < 0 oder > 1 ist. Ist k nicht numerisch, erscheint der Fehler #WERT!.

Liegt ein Quantil zwischen zwei Beobachtungen, wird durch Interpolation der entsprechende Wert ermittelt. Das geschieht auch, wenn k kein Vielfaches von 1/n – 1 ist, wobei n die Anzahl der Werte in Array ist. Wenn bei einem bestimmten Wert für k allerdings keine Interpolation möglich ist, liefert die Funktion den Fehler #ZAHL!.

Die Tabelle in Abbildung 7.74 kann so gelesen werden: 50% der erfassten Personen haben ein Gewicht von unter 67 kg.

Siehe auch QUANTIL(), QUANTIL.EXKL(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL(), QUARTILE(), QUARTILE.EXKL() und QUARTILE.INKL().

QUANTILSRANG.EXKL()

PERCENTRANK.EXC()

Syntax: QUANTILSRANG.EXKL(**Array**; **x**; Genauigkeit)

Beispiel: siehe Abbildung 7.75

Die Funktion QUANTILSRANG.EXKL() liefert den prozentualen Rang (0..1 ausschließlich) eines Werts in einer mit dem Argument Array angegebenen Datenreihe. Array muss nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein.

Das Argument x bezeichnet den Wert, dessen relative Position in der Datenreihe ermittelt werden soll. Wenn x selbst als Wert nicht in Array auftaucht, wird der entsprechende Wert interpoliert. Mit Genauigkeit ist die Anzahl der Stellen für die Ausgabe des Ergebnisses bestimmbar. Wird Genauigkeit nicht angegeben, wird 3 angenommen.

Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert. Enthält Array aber gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Ist x größer oder kleiner als der größte oder kleinste Wert, liefert die Funktion den Fehler #NV.

Das Ergebnis der Funktion in Abbildung 7.75 lässt sich folgendermaßen lesen: Die Person, deren Gewicht 63 kg beträgt, liegt auf einer Skala von 0 bis 100 auf Rang 43, d.h., 43% der Personen haben ein Gewicht von unter oder gleich 63 kg.

Wird die Tabelle nach dem Gewicht aufsteigend sortiert, zeigen sich auch die QUANTILSRANG.EXKL()-Werte in aufsteigender Reihenfolge.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von QUANTILSRANG() angeboten werden. Die andere Funktion ist QUANTILSRANG.INKL().

	A	B	C	D	E	F
1						
2	**Quantilsrang berechnen**					
3						
4	Person	Gewicht	QUANTILSRANG.EXKL()	QUANTILSRANG.INKL()	QUANTILSRANG()	
5	19	55	0,05	0,00	0,00	
6	14	56	0,10	0,05	0,05	
7	20	57	0,14	0,11	0,11	
8	9	58	0,19	0,16	0,16	
9	17	58	0,19	0,16	0,16	
10	15	60	0,29	0,26	0,26	
11	18	61	0,33	0,32	0,32	
12	16	62	0,38	0,37	0,37	
13	6	63	0,43	0,42	0,42	
14	2	66	0,48	0,47	0,47	
15	3	68	0,52	0,53	0,53	
16	7	68	0,52	0,53	0,53	
17	11	68	0,52	0,53	0,53	
18	1	69	0,67	0,68	0,68	
19	4	72	0,71	0,74	0,74	
20	12	72	0,71	0,74	0,74	
21	5	74	0,81	0,84	0,84	
22	13	77	0,86	0,89	0,89	
23	8	79	0,90	0,95	0,95	
24	10	79	0,90	0,95	0,95	
25						

Abbildung 7.75 Berechnen des Quantilsrangs

Der Zusammenhang mit QUANTIL.EXKL() sieht so aus:

```
x = QUANTIL.EXKL(Array; 0,2)
0,2 = QUANTILSRANG.EXKL(Array; x)
```

QUANTILSRANG.EKKL() ist also die Umkehrfunktion von QUANTIL.EKKL().

Siehe auch QUANTIL(), QUANTIL.EXKLU(), QUANTIL.INKL(), QUANTILSRANG.INKL(), QUARTILE(), QUARTILE.EXKL() und QUARTILE.INKL().

QUANTILSRANG.INKL()

PERCENTRANK.INC()

Syntax: QUANTILSRANG.INKL(**Array**; **x**; Genauigkeit)
Beispiel: siehe Abbildung 7.75

Die Funktion QUANTILSRANG.INKL() liefert den prozentualen Rang (0..1 einschließlich) eines Werts in einer mit dem Argument Array angegebenen Datenreihe.

Array muss nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein. Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert.

Das Argument x bezeichnet den Wert, dessen relative Position in der Datenreihe ermittelt werden soll; Wenn x selbst als Wert nicht in Array auftaucht, wird der entsprechende Wert interpoliert. Mit Genauigkeit ist die Anzahl der Stellen für die Ausgabe des Ergebnisses bestimmbar. Wird Genauigkeit nicht angegeben, wird 3 angenommen.

Enthält Array gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Ist x größer oder kleiner als der größte oder kleinste Wert, liefert die Funktion den Fehler #NV.

Der Zusammenhang mit QUANTIL.INKL() sieht so aus:

```
x = QUANTIL.INKL(Array; 0,2)
0,2 = QUANTILSRANG.INKL(Array; x)
```

QUANTILSRANG.INKL() ist also die Umkehrfunktion von QUANTIL.INKL().

Das Ergebnis der Funktion in Abbildung 7.75 lässt sich folgendermaßen lesen: Die Person, deren Gewicht 63 kg beträgt, liegt auf einer Skala von 0 bis 100 auf Rang 42, d.h., 42% der Personen haben ein Gewicht von unter oder gleich 63 kg.

Wird die Tabelle nach dem Gewicht aufsteigend sortiert, zeigen sich auch die QUANTILSRANG.INKL()-Werte in aufsteigender Reihenfolge.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von QUANTILSRANG() angeboten werden. Sie liefert dasselbe Ergebnis. Die andere Funktion ist QUANTILSRANG.EXKL().

Siehe auch QUANTIL(), QUANTIL.EXKLU(), QUANTIL.INKL(), QUANTILSRANG.EXKL(), QUARTILE(), QUARTILE.EXKL() und QUARTILE.INKL().

QUARTILE.EXKL()

QUARTILE.EXC()

Syntax:	QUARTILE.EXKL(**Array**; **Quartile**)
Beispiel:	siehe Abbildung 7.76

Die Funktion QUARTILE.EXKL() unterteilt – basierend auf Quantilswerten zwischen 0 und 1 (also ausschließlich 0 und 1) – die Daten von Array in Bereiche mit je gleichen Anteilen von Daten und ist damit ein Spezialfall der Funktion QUANTIL.EXKL() (siehe dort).

Array sind die zu unterteilenden Daten. Sie müssen nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein. Quartile gibt die Nummer des Quartils an, das ausgegeben werden soll. Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert.

Enthält Array gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn Quartile ≤ 0 oder ≥ 4 ist. Ist Quartile nicht numerisch, erscheint der Fehler #WERT!.

Für Quartile sind bei dieser Funktion nur drei Belegungen möglich:

1 – 25%-Quantil
2 – 50%-Quantil (= Median)
3 – 75%-Quantil

Das 25%-Quantil liefert beispielsweise in dem Beispiel in Abbildung 7.76 das Gewicht, das das leichteste Viertel der Personen maximal erreicht.

Die Funktion ist eine der beiden Funktionen, die seit der Version Excel 2010 anstelle von QUARTILE() angeboten werden. Die andere Funktion ist QUARTILE.INKL().

	A	B	C	D	E	F	G
1							
2	**Quartile berechnen**						
3							
4	Person	Gewicht	QUARTILE.EXKL()	QUARTILE.INKL()	QUARTILE()		
5	1	69		55,00	55,00	Kleinster Wert	=MIN()
6	2	66	58,50	59,50	59,50	25%-Quartil	
7	3	68	67,00	67,00	67,00	50%-Quartil	=MEDIAN()
8	4	72	72,00	72,00	72,00	75%-Quartil	
9	5	74		79,00	79,00	Größter Wert	=MAX()
10	6	63					
11	7	68					
12	8	79					
13	9	58					
14	10	79					
15	11	68					
16	12	72					
17	13	77					
18	14	56					
19	15	60					
20	16	62					
21	17	58					
22	18	61					
23	19	55					
24	20	57					

Abbildung 7.76 Einteilung der Daten in Quartile

Siehe auch QUANTIL(), QUANTIL.EXKLU(), QUANTIL.INKL(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL(), QUARTILE() und QUARTILE.INKL().

QUARTILE.INKL()

QUARTILE.INC()

Syntax: QUARTILE.INKL(**Array**; **Quartile**)

Beispiel: siehe Abbildung 7.76

Die Funktion QUARTILE.INKL() unterteilt – basierend auf Quantilswerten von 0 bis 1 (also einschließlich 0 und 1) – die Daten von Array in Bereiche mit je gleichen Anteilen von Daten und ist damit ein Spezialfall von QUANTIL.INKL() (siehe dort).

Array sind die zu unterteilenden Daten. Sie müssen nicht sortiert sein. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein. Quartile gibt die Nummer des Quartils an, das ausgegeben werden soll. Nicht numerische Werte oder Wahrheitswerte in Array werden ignoriert.

Enthält Array gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn Quartile < 0 oder > 4 ist. Ist Quartile nicht numerisch, erscheint der Fehler #WERT!.

Für Quartile sind in diesem Fall fünf Belegungen möglich:

0 – niedrigster Wert
1 – 25%-Quantil
2 – 50%-Quantil (= Median)
3 – 75%-Quantil
4 – höchster Wert

Die Funktion ist eine der beiden Funktionen, die seit der Version Excel 2010 anstelle von QUARTILE() angeboten werden. Sie liefert das gleiche Ergebnis wie die ältere Funktion. Die andere Funktion ist QUARTILE.EXKL().

Siehe auch QUANTIL(), QUANTIL.EXKLU(), QUANTIL.INKL(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL(), QUARTILE() und QUARTILE.EXKL().

RANG.GLEICH()

RANK.EQ()

Syntax: RANG.GLEICH(**Zahl**; **Bezug**; Reihenfolge)
Beispiel: siehe Abbildung 7.77

Die Funktion RANG.GLEICH() liefert den Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt. Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von RANG() angeboten werden. Sie liefert das gleiche Ergebnis. Die andere Funktion ist RANG.MITTELW().

Mit Zahl wird der Wert aus der Datenreihe angegeben, dessen Rang bestimmt werden soll; Bezug liefert die Datenreihe. Mit Reihenfolge kann noch angegeben werden, ob in fallender oder steigender Ordnung gezählt wird. Vorgegeben ist die fallende Ordnung, die dann verwendet wird, wenn das Argument nicht oder mit 0 belegt ist. Bei jedem anderen Wert zählt Excel in steigender Ordnung.

Nicht numerische Werte in Bezug werden bei der Rangberechnung ignoriert. Soll aber der Rang eines solchen Werts angegeben werden, liefert die Funktion den Fehler #WERT!. Ist eine Zelle in Bezug leer, liefert die Funktion für diese Zelle den Fehler #NV.

	A	B	C	D	E	F
1						
2	**Rang von Laufergebnissen berechnen**					
3						
4	Teilnehmer	Zeit	RANG.GLEICH()	RANG.MITTELW()	RANG()	
5	5	10,60	1	1,0	1	
6	9	10,70	2	2,0	2	
7	4	10,80	3	3,0	3	
8	18	10,90	4	5,0	4	
9	3	10,90	4	5,0	4	
10	14	10,90	4	5,0	4	
11	17	11,30	7	7,5	7	
12	13	11,30	7	7,5	7	
13	2	11,50	9	9,0	9	
14	15	disqual.	#WERT!	#WERT!	#WERT!	
15						

Abbildung 7.77 Rangordnung von Laufzeiten

Sind zwei Werte in `Bezug` gleich, erhalten sie auch den gleichen Rang, dafür wird die nachfolgende Rangnummer übersprungen.

RANG.MITTELW()

RANK.AVG()

Syntax: `RANG.MITTELW(`**`Zahl`**`; `**`Bezug`**`; Reihenfolge)`

Beispiel: siehe Abbildung 7.77 zu `RANG.GLEICH()`

Die Funktion `RANG.MITTELW()` liefert den Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt. Wenn mehrere Werte die gleiche Rangzahl aufweisen, wird die durchschnittliche Rangzahl zurückgegeben. Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von `RANG()` angeboten werden. Die andere Funktion ist `RANG.GLEICH()`. Mit `Zahl` wird der Wert angegeben, dessen Rang bestimmt werden soll; `Bezug` liefert die Datenreihe. Mit `Reihenfolge` wird angegeben, ob in fallender oder steigender Ordnung gezählt wird. Vorgegeben ist die fallende Ordnung, die dann verwendet wird, wenn das Argument nicht oder mit `0` belegt ist. Bei jedem anderen Wert zählt Excel in steigender Ordnung.

Nicht numerische Werte in `Bezug` werden bei der Rangberechnung ignoriert. Soll aber der Rang eines solchen Werts angegeben werden, liefert die Funktion den Fehler `#WERT!`. Ist eine Zelle in `Bezug` leer, liefert die Funktion für diese Zelle den Fehler `#NV`.

RGP()

LINEST()

Syntax:	RGP(**Y_Werte**; X_Werte; Konstante; Stats)
Beispiel:	siehe Abbildung 7.78

Die Funktion RGP() liefert Kennziffern zur linearen Regression, mit deren Hilfe in einem X/Y-Diagramm auf der Basis der mit den Argumenten Y_Werte und X_Werte angegebenen Daten eine gerade Linie eingezeichnet werden kann, die an die vorhandenen Datenpunkte möglichst gut angepasst ist. Die Funktion verwendet dazu die Methode der kleinsten Quadrate. Hierbei wird davon ausgegangen, dass die vorhandenen Daten sich durch eine lineare Gleichung beschreiben lassen:

```
y = mx + b
```

wobei m die Steigung der Geraden und b ihren Schnittpunkt mit der y-Achse festlegt.

Werden für die x-Werte mehrere Spalten angegeben, dann wird als Gleichung für die Gerade angenommen:

```
y = m1*x1 + m2*x2 + ... + b
```

Mit Y_Werte werden die abhängigen Werte angegeben, für die eine lineare Regression durchgeführt werden soll. Alle anderen Argumente sind optional. X_Werte sind die zu den y-Werten gehörenden unabhängigen x-Werte. Wird dieses Argument nicht angegeben, benutzt Excel an ihrer Stelle eine Matrix mit fortlaufenden Nummern, {1.2.3.4....}, die genauso viele Elemente wie die Datenreihe der Y_Werte enthält. Für Y_Werte und X_Werte kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden.

Anders als bei anderen Funktionen werden nicht numerische Werte, leere Zellen oder Wahrheitswerte nicht ignoriert, sondern führen zu dem Fehler #WERT!. Enthalten Y_Werte und X_Werte unterschiedlich viele Datenelemente, liefert die Funktion den Fehler #BEZUG.

Mit Konstante lässt sich bestimmen, ob b berechnet (WAHR oder weggelassen) oder mit 0 angesetzt (FALSCH) werden soll. In diesem Fall wird m so angepasst, dass die Regressionsgleichung auf

```
y = mx
```

vereinfacht werden kann.

Letzteres ist erforderlich, wenn bei den Daten von vornherein klar ist, dass zu `x = 0` ein y-Wert `0` gehört.

`Stats` ist ein Wahrheitswert, mit dem entschieden wird, ob nur die Werte für `b` und `m` (`FALSCH`) oder auch weitere Kennziffern ermittelt werden sollen (`WAHR`).

Die Funktion gibt die Kennziffern in Form einer Matrix aus, sie muss also auch in der für Matrixfunktionen üblichen Form eingegeben werden. Wenn Sie mit `Stats = FALSCH` arbeiten, markieren Sie eine 1x2-Matrix, also zwei nebeneinanderliegende Zellen, für `Stats = WAHR` eine 5x2-Matrix. Wenn Sie mit mehreren x-Werten arbeiten, müssen Sie die Matrix um entsprechend viele Spalten erweitern, wie in Abbildung 7.78 zu sehen ist. Ist die Funktion eingetragen, schließen Sie sie mit Strg + ⇧ + ↵ ab.

	A	B	C	D	E	F	G
1							
2	**Kennziffern der Funktion RGP()**						
3							
4	**Ausgabe bei einer (oder keiner) x-Spalte**			**Ausgabe bei drei x-Spalten**			
5	m	b		m3	m2	m1	b
6	se(m)	se(b)		se(m3)	se(m2)	se(m1)	se(b)
7	r^2	se(y)		r^2	se(y)		
8	F	df		F	df		
9	ss(reg)	ss(res)		ss(reg)	ss(res)		

Abbildung 7.78 Anordnung der Kennziffern in der Ausgabematrix von `RGP()`

In Tabelle 7.2 sind die Kennziffern kurz erläutert:

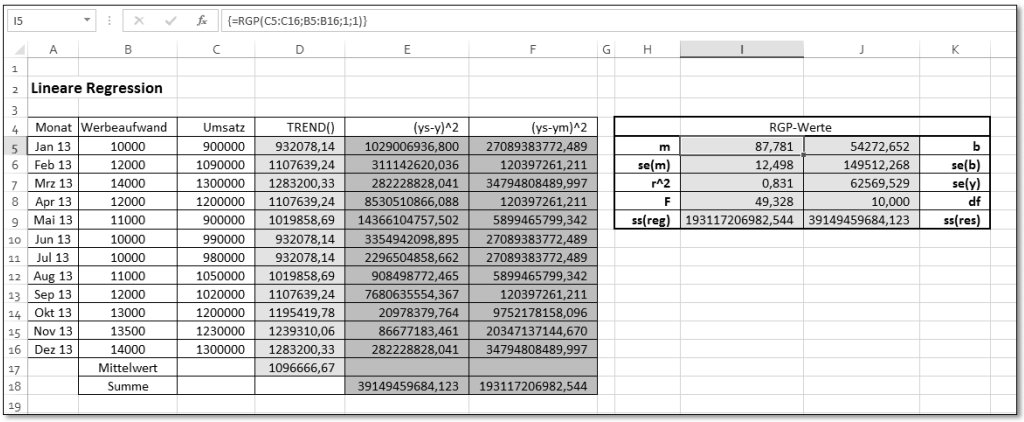
Kennziffer	Ausgabe
`m`	Die Steigung der Regressionsgeraden. Ihr Wert kann einzeln mit der Funktion `STEIGUNG()` ermittelt werden.
`b`	Der Schnittpunkt mit der y-Achse. Der Wert kann einzeln mit der Funktion `ACHSENABSCHNITT()` ermittelt werden.
`se(m)`	Der Standardschätzfehler für die Steigung. Kann für Signifikanztests verwendet werden.
`se(b)`	Der Standardschätzfehler des Achsenabschnitts. Kann für Signifikanztests verwendet werden.
`r^2`	Das Bestimmtheitsmaß. Es kann auch einzeln berechnet werden mit der Funktion `BESTIMMTHEITSMASS()`.

Tabelle 7.2 Bedeutung der von der Funktion `RGP()` ausgegebenen Kennziffern

Kennziffer	Ausgabe
se(y)	Der Standardschätzfehler der aus der Regression berechneten y-Werte kann einzeln auch mit STFEHLERYX() berechnet werden.
F	Ein F-Wert, der mit der Funktion FVERT() weiter ausgewertet werden kann. Es ist sinnvoller, gleich die Funktion FTEST() auf die Daten anzuwenden.
df	Freiheitsgrade (**d**egrees of **f**reedom) für den F-Test
ss(reg)	Die Quadratsumme der Regression ist die Summe der quadratischen Abweichungen der geschätzten y-Werte von ihrem Mittelwert.
ss(res)	Die Quadratsumme der Residuen ist die Summe der quadratischen Abweichungen der geschätzten y-Werte von ihren arithmetischen gegebenen y-Werten.

Tabelle 7.2 Bedeutung der von der Funktion RGP() ausgegebenen Kennziffern (Forts.)

Abbildung 7.79 zeigt ein Beispiel für die Anwendung der Funktion:



I5 {=RGP(C5:C16;B5:B16;1;1)}

Lineare Regression

Monat	Werbeaufwand	Umsatz	TREND()	(ys-y)^2	(ys-ym)^2
Jan 13	10000	900000	932078,14	1029006936,800	27089383772,489
Feb 13	12000	1090000	1107639,24	311142620,036	120397261,211
Mrz 13	14000	1300000	1283200,33	282228828,041	34794808489,997
Apr 13	12000	1200000	1107639,24	8530510866,088	120397261,211
Mai 13	11000	900000	1019858,69	14366104757,502	5899465799,342
Jun 13	10000	990000	932078,14	3354942098,895	27089383772,489
Jul 13	10000	980000	932078,14	2296504858,662	27089383772,489
Aug 13	11000	1050000	1019858,69	908498772,465	5899465799,342
Sep 13	12000	1020000	1107639,24	7680635554,367	120397261,211
Okt 13	13000	1200000	1195419,78	20978379,764	9752178158,096
Nov 13	13500	1230000	1239310,06	86677183,461	20347137144,670
Dez 13	14000	1300000	1283200,33	282228828,041	34794808489,997
	Mittelwert		1096666,67		
	Summe			39149459684,123	193117206982,544

RGP-Werte			
m	87,781	54272,652	**b**
se(m)	12,498	149512,268	**se(b)**
r^2	0,831	62569,529	**se(y)**
F	49,328	10,000	**df**
ss(reg)	193117206982,544	39149459684,123	**ss(res)**

Abbildung 7.79 Beispiel für die lineare Regression

In diesem (fiktiven) Beispiel sind die x-Werte Daten über den Werbeaufwand in einem Monat und die y-Werte die dadurch vermutlich beeinflussten Monatsumsätze. Die Spalte »Trend« übernimmt die in der Ausgabematrix angezeigten Werte für m und b in die lineare Regressionsgleichung:

```
=$i$5*B5+$J$5
```

Die errechneten Werte entsprechen denen, die auch die Funktion TREND() für diese Werte liefern würden. Das Diagramm zu diesen Werten liefert die gewünschte Trendlinie.

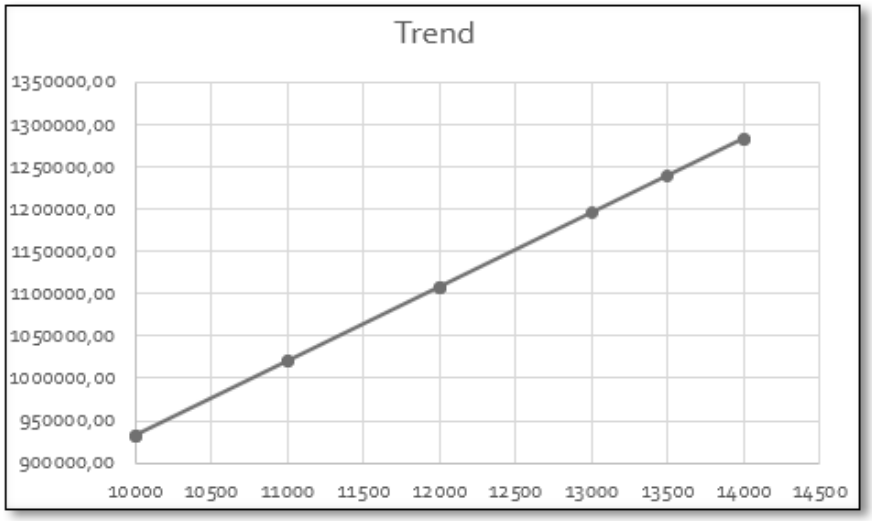


Abbildung 7.80 Die Trendlinie auf der Basis der geschätzten Werte

In der Spalte E und F werden zur Kontrolle der Werte für ss(reg) und ss(res) die Quadrate der Differenzen zwischen den geschätzten und den tatsächlichen y-Werten und die Quadrate der Differenzen zwischen dem Mittelwert der geschätzten y-Werte und den einzelnen geschätzten y-Werten ausgegeben und summiert.

Zu beachten ist, dass die Genauigkeit einer mit RGP() berechneten Geraden von der Streuung der Daten abhängt. Je geringer die Streuung, umso zuverlässiger sind die ermittelten Werte.

Siehe dazu auch den Abschnitt 7.5, »Regressionsanalyse«.

RKP()

LOGEST()

Syntax:	RKP(**Y_Werte**; X_Werte; Konstante; Stats)
Beispiel:	siehe Abbildung 7.81

Die Funktion RKP() liefert Kennziffern zur exponentiellen Regression, mit deren Hilfe in einem X/Y-Diagramm auf der Basis der mit den Argumenten Y_Werte und X_Werte angegebenen Daten eine Exponentialkurve eingezeichnet werden kann, die an die vorhandenen Datenpunkte möglichst gut angepasst ist. Hierbei wird davon ausgegangen, dass sich die vorhandenen Daten durch eine exponentielle Gleichung beschreiben lassen:

```
y = b * m^x
```

wobei b den Schnittpunkt der Regressionskurve mit der y-Achse liefert und m die Basis ist, für die jeweils x als Exponent verwendet wird. Mit m > 1 erhalten Sie eine stetig steigende, mit m < 1 eine stetig fallende Kurve. Werden mehrere Spalten mit x-Werten benutzt, dann wird eine Regression nach folgender Gleichung durchgeführt:

```
y = b * m1^x1 * m2^x2 ...
```

Mit Konstante lässt sich bestimmen, ob b berechnet (WAHR oder weggelassen) oder mit 1 angesetzt werden soll (FALSCH).

Stats ist ein Wahrheitswert, mit dem entschieden wird, ob nur die Werte für b und m (FALSCH) oder auch weitere Kennziffern ermittelt werden sollen (WAHR).

Anders als bei anderen Funktionen werden in den Wertebereichen nicht numerische Werte, leere Zellen oder Wahrheitswerte nicht ignoriert, sondern führen zu dem Fehler #WERT!. Enthalten Y_Werte und X_Werte unterschiedlich viele Datenelemente, liefert die Funktion den Fehler #BEZUG.

Die Funktion gibt die Kennziffern in Form einer Matrix aus, sie muss also auch in der für Matrixfunktionen üblichen Form eingegeben werden. Wenn Sie mit Stats = FALSCH arbeiten, markieren Sie eine 1x2-Matrix, also zwei nebeneinanderliegende Zellen, für Stats = WAHR eine 5x2-Matrix. Wenn Sie mit mehreren x-Werten arbeiten, müssen Sie die Matrix um entsprechend viele Spalten erweitern. Ist die Funktion eingetragen, schließen Sie sie mit Strg + ⇧ + ↵ ab.

Die ausgegebenen Kennziffern stimmen sinngemäß mit denen von RGP() überein (vergleiche dort). Die dort gegebenen Hinweise zur Einzelberechnung über die Funktionen BESTIMMTHEITSMASS(), ACHSENABSCHNITT() und STEIGUNG() gelten allerdings nur für die lineare Regression. Es sind zudem einige Besonderheiten zu beachten. Excel bedient sich bei den Berechnungen zu RKP() der Formel:

```
ln(y) = ln(b) + x * ln(m)
```

und berechnet mit dieser Gleichung eine lineare Regression. Hierdurch werden auch die Ausgabewerte für die Kennziffern beeinflusst: `se(m)` und `se(b)` liefern die Schätzfehler für `ln(m)` und `ln(b)`.

In dem in Abbildung 7.81 gezeigten Beispiel deuten die y-Werte auf ein exponentielles Wachstum. In der Spalte unter dem Titel »Variation« sind die in der `RKP()`-Matrix berechneten Werte für `b` und `m` in die Regressionsformel eingesetzt worden.

```
=$I$5 * $H$5^A5
```

heißt die erste Formel in Zelle D5. Die in dieser Spalte ausgegebenen y-Werte entsprechen denen, die Sie auch mit der Funktion `VARIATION()` erzeugen können.

Wie Abbildung 7.81 zeigt, liefert die Funktion in den Zellen H9 und I9 allerdings weder die Quadratsumme der Residuen noch die Quadratsumme der Regression. Welche Werte stattdessen ausgegeben werden, war bisher nicht zu ergründen.

H5 {=RKP(B5:B17;A5:A17;1;1)}

	A	B	C	D	E	F	G	H	I	J	K
1											
2	**Exponentielle Regression**										
3											
4	x-Wert	y-Wert	Variation	(ys-y)^2	(ys-ym)^2			RKP-Werte			
5	13	135	175	1603	1384031119		**m**	1,810878	0,077729	**b**	
6	14	320	317	9	1373490327		**se(m)**	0,012094	0,234200	**se(b)**	
7	15	674	574	9997	1354504788		**r^2**	0,995458	0,163157	**se(y)**	
8	16	1.100	1039	3664	1320460583		**F**	2410,771294	11,000000	**df**	
9	17	2.200	1882	100900	1259913460		**ss(reg)**	64,175453	0,292823	**ss(res)**	
10	18	3.800	3409	153108	1153886322						
11	19	5.000	6173	1375357	973743003						
12	20	12.600	11178	2021779	686414019						
13	21	19.000	20242	1543029	293622768						
14	22	41.000	36656	18869218	520537						
15	23	50.000	66380	268296934	841125321						
16	24	125.000	120206	22985694	6860485659						
17	25	230.000	217678	151837358	32508147076						
18		Mittelwert	37378								
19			Summe	467198652	50010344984						

Abbildung 7.81 Beispiel für die exponentielle Regression

In Abbildung 7.82 wird ein Verbunddiagramm gezeigt, das die tatsächlichen y-Werte aus Spalte B als Punktwolke und die errechneten Trendwerte aus Spalte C als Kurve anzeigt. Es ist gut zu sehen, dass die Kurve eine gute Annäherung an die Punktwolke darstellt. Dem entspricht auch, dass der RKP-Wert für das Bestimmtheitsmaß, also `r^2`, sehr nahe an der 1 liegt.

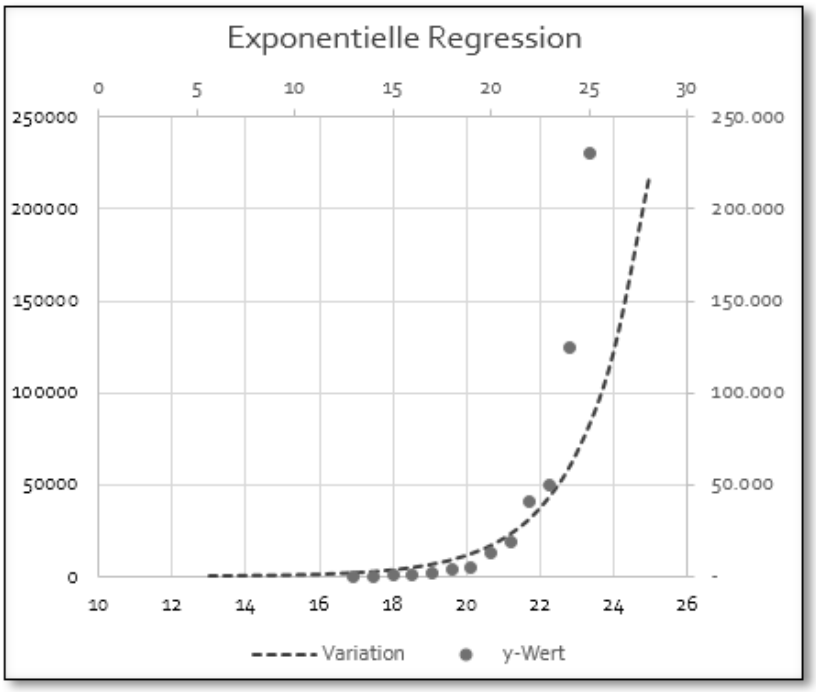


Abbildung 7.82 Punktwolke und exponentielle Kurve

Siehe dazu auch den Abschnitt 7.5, »Regressionsanalyse«.

SCHÄTZER()

FORECAST()

Syntax:	SCHÄTZER(**x**; **Y_Werte**; **X_Werte**)
Beispiel:	=SCHÄTZER(3;{4;5;6};{1;5;10})
Ergebnis:	4,48

Die Funktion SCHÄTZER() liefert für den angegebenen Wert x einen Schätzwert für den entsprechenden Wert für y anhand einer linearen Regression, die die mit Y_Werte und X_Werte angegebenen bereits bekannten Werte verwendet. Dabei stellen die X_Werte die unabhängige, die Y_Werte die abhängige Variable dar. Die Funktion dient insbesondere der Prognose zukünftiger Werte auf der Basis bereits bekannter Beziehungen zwischen zwei Merkmalen.

Für Y_Werte und X_Werte kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden. Texte, leere Zellen oder Wahrheitswerte werden ignoriert.

Enthalten Y_Werte und X_Werte unterschiedlich viele Datenelemente, liefert die Funktion den Fehler #NV. Enthalten beide nur ein oder gar kein Datenelement, erscheint der Fehler #DIV/0!. Das geschieht auch, wenn die Varianz der X_Werte gleich 0 ist. Wenn x nicht numerisch ist, liefert die Funktion den Fehler #WERT!.

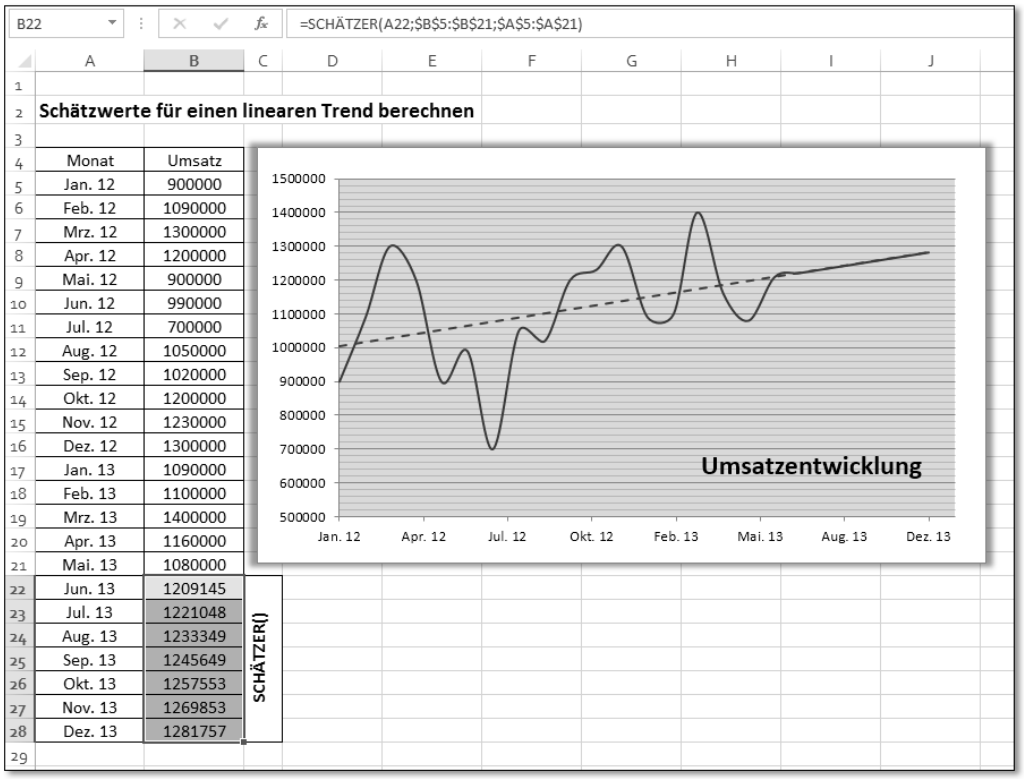


Abbildung 7.83 Errechnen von Schätzwerten bei linearer Regression

Die Funktion liefert die gleichen Schätzwerte für y wie die Funktion TREND(), wenn bei dieser Funktion mit dem Argument Neue_x_Werte gearbeitet wird. Bei SCHÄTZER() werden einzelne y-Werte ausgegeben, bei TREND() wird in der Regel eine Matrix mit mehreren Werten ausgegeben.

Siehe dazu auch den Abschnitt 7.5, »Regressionsanalyse«.

SCHIEFE()

SKEW()

Syntax: `SCHIEFE(Zahl1; Zahl2; ...)`
Beispiel: `=SCHIEFE(2;3;2;7;9;6;4;2)`
Ergebnis: `0,785`

Die Funktion `SCHIEFE()` liefert ein Maß für die Asymmetrie der Häufigkeitsverteilung einer Stichprobe. Verglichen wird dabei mit einer Normalverteilung mit gleichem Mittelwert und gleicher Streuung.

Seit Excel 2007 sind 255 Argumente möglich, in den älteren Versionen nur 30. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie berücksichtigt.

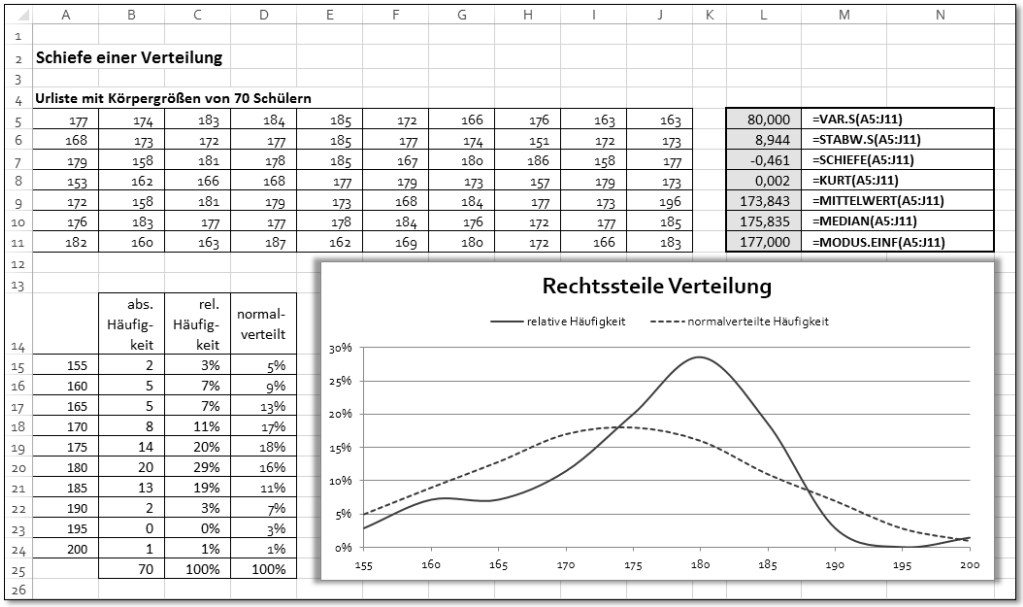
Sind weniger als drei Datenelemente angegeben oder ist die Standardabweichung der Stichprobe gleich `0`, gibt die Funktion den Fehler `#DIV/0!` zurück. Fehlerwerte in den Datenelementen führen dazu, dass die Funktion diesen Fehler als Ergebnis ausgibt.

Ist das Ergebnis größer als `0`, dann ist die linke Seite steiler, die Verteilung heißt linkssteil oder rechtsschief. Eine rechtsschiefe Verteilung ist beispielsweise oft bei Daten über die Einkommensverteilung zu finden, wenn wenige Personen mit sehr hohem Einkommen auf der einen Seite des Mittelwerts stehen und sehr viele Personen mit niedrigem Einkommen auf der anderen Seite.

Die Einkommenshöhe, die 50% der Personen erreichen, liegt dabei immer unter dem Durchschnittseinkommen. Der Mittelwert ist also größer als der Median. Der Modus ist kleiner oder höchstens gleich dem Median. Bei einem negativen Wert für die Schiefe ist die Verteilung linksschief oder rechtssteil. Der Mittelwert ist kleiner als der Median und dieser kleiner oder gleich dem Modus. Vergleiche auch `KURT()`.

In Abbildung 7.84 ist zunächst eine Urliste mit Körpergrößen für 70 Schüler angelegt. Es zeigt sich, dass die Schiefe einen negativen Wert ergibt, was auf eine rechtssteile Verteilung schließen lässt. Das wird auch bestätigt durch die Tatsache, dass der Mittelwert kleiner ist als der Median.

Im Diagramm sind die relativen Häufigkeiten und die bei einer Normalverteilung zu erwartenden Häufigkeiten gegenübergestellt. Der Gipfel der Kurve der relativen Häufigkeiten ist deutlich nach rechts verschoben.



Schiefe einer Verteilung

Urliste mit Körpergrößen von 70 Schülern

177	174	183	184	185	172	166	176	163	163
168	173	172	177	185	177	174	151	172	173
179	158	181	178	185	167	180	186	158	177
153	162	166	168	177	179	173	157	179	173
172	158	181	179	173	168	184	177	173	196
176	183	177	177	178	184	176	172	177	185
182	160	163	187	162	169	180	172	166	183

80,000	=VAR.S(A5:J11)
8,944	=STABW.S(A5:J11)
-0,461	=SCHIEFE(A5:J11)
0,002	=KURT(A5:J11)
173,843	=MITTELWERT(A5:J11)
175,835	=MEDIAN(A5:J11)
177,000	=MODUS.EINF(A5:J11)

	abs. Häufigkeit	rel. Häufigkeit	normalverteilt
155	2	3%	5%
160	5	7%	9%
165	5	7%	13%
170	8	11%	17%
175	14	20%	18%
180	20	29%	16%
185	13	19%	11%
190	2	3%	7%
195	0	0%	3%
200	1	1%	1%
	70	100%	100%

Abbildung 7.84 Beispiel für eine rechtssteile Verteilung einer Stichprobe

Für die Berechnung der Schiefe bei einer Grundgesamtheit verwenden Sie die in Excel 2013 neue Funktion `SCHIEFE.P()`.

SCHIEFE.P()

SKEW.P()

Syntax: `SCHIEFE.P(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=SCHIEFE.P(2;3;2;7;9;6;4;2)`

Ergebnis: `0,6301`

Die in Excel 2013 neu eingeführte Funktion `SCHIEFE.P()` liefert ein Maß für die Asymmetrie der Häufigkeitsverteilung einer Grundgesamtheit. Verglichen wird mit einer Normalverteilung mit gleichem Mittelwert und gleicher Streuung.

255 Argumente sind möglich. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie berücksichtigt.

Sind weniger als drei Datenelemente angegeben oder ist die Standardabweichung der Grundgesamtheit gleich `0`, gibt die Funktion den Fehler `#DIV/0!` zurück. Fehlerwerte in den Datenelementen führen dazu, dass die Funktion diesen Fehler als Ergebnis ausgibt.

Ist das Ergebnis größer als `0`, dann ist die linke Seite steiler, die Verteilung heißt linkssteil oder rechtsschief. Bei einem negativen Wert für die Schiefe ist die Verteilung linksschief oder rechtssteil. Der Mittelwert ist in diesem Fall kleiner als der Median und dieser kleiner oder gleich dem Modus. Vergleiche auch `KURT()`.

In Abbildung 7.85 wird die bei der Funktion `SCHIEFE()` verwendete Urliste mit Körpergrößen für 70 Schüler nicht als Stichprobe, sondern als Grundgesamtheit behandelt. Es ergibt sich auch hier, dass die Schiefe einen negativen Wert hat, was auf eine rechtssteile Verteilung schließen lässt. Das wird auch bestätigt durch die Tatsache, dass der Mittelwert kleiner ist als der Median.

Schiefe einer Verteilung

Urliste mit Körpergrößen von 70 Schülern

177	174	183	184	185	172	166	176	163	163
168	173	172	177	185	177	174	151	172	173
179	158	181	178	185	167	180	186	158	177
153	162	166	168	177	179	173	157	179	173
172	158	181	179	173	168	184	177	173	196
176	183	177	177	178	184	176	172	177	185
182	160	163	187	162	169	180	172	166	183

78,857	=VAR.P(A5:J11)
8,880	=STABW.N(A5:J11)
-0,451	=SCHIEFE.P(A5:J11)
0,002	=KURT(A5:J11)
173,843	=MITTELWERT(A5:J11)
175,835	=MEDIAN(A5:J11)
177,000	=MODUS.EINF(A5:J11)

	abs. Häufigkeit	rel. Häufigkeit	normalverteilt
155	2	3%	5%
160	5	7%	9%
165	5	7%	13%
170	8	11%	17%
175	14	20%	18%
180	20	29%	16%
185	13	19%	11%
190	2	3%	7%
195	0	0%	3%
200	1	1%	1%
	70	100%	100%

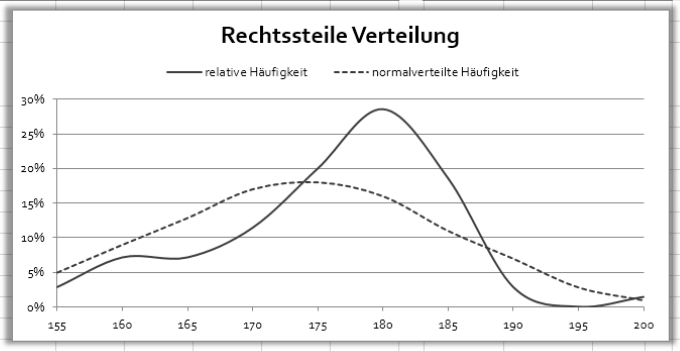


Abbildung 7.85 Beispiel für eine rechtssteile Verteilung einer Grundgesamtheit

Im Diagramm sind die relativen Häufigkeiten und die bei einer Normalverteilung zu erwartenden Häufigkeiten gegenübergestellt. Der Gipfel der Kurve der relativen Häufigkeiten ist deutlich nach rechts verschoben.

Für die Berechnung der Schiefe bei einer Stichprobe verwenden Sie die Funktion `SCHIEFE()`.

STABW.N()

STDEV.P()

Syntax: `STABW.N(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=STABW.N(33; 22; 28; 17; 23; 26)`
Ergebnis: `5,01`

Die Funktion `STABW.N()` berechnet für die Werte in der Argumentliste die Standardabweichung vom arithmetischen Mittelwert. Dabei stellen die in der Argumentliste angegebenen Daten eine Grundgesamtheit dar.

Die Funktion kann bis zu 255 Argumente verarbeiten. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie mitgerechnet.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler `#DIV/0!`.

Die Standardabweichung ist ein Maß dafür, wie weit die vorhandenen Daten um den Mittelwert streuen. Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichungen vom arithmetischen Mittel. Bei der Ermittlung des Mittels wird anders als bei den Funktionen `STABW.S()` und `STABWA()` mit `n` für den Umfang der Grundgesamtheit gerechnet.

Die verwendete Formel ist:

$$= \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

Die Funktion `STABW.N()` ersetzt seit Excel 2010 die bisherige Funktion `STABWN()`, die aber weiter in der Kategorie **Kompatibilität** angeboten wird. Handelt es sich bei den angegebenen Daten um eine Stichprobe, sollten Sie die Funktion `STABW.S()` verwenden.

Die Standardabweichung ist zugleich die Wurzel aus der Varianz, die für Grundgesamtheiten von der Funktion `VAR.P()` geliefert wird (siehe dort). Mehr zu Standardabweichung und Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

Abbildung 7.86 zeigt Werte für die Standardabweichung bei den Merkmalen »Alter« und »Gewicht«, wobei die Daten einmal als Grundgesamtheit und einmal als Daten ei-

ner Stichprobe behandelt werden. Wie zu sehen ist, berechnet `STABW.N()` einen geringeren Wert als `STABW.S()`, da ja einmal mit `n` und das andere Mal mit `n – 1` dividiert wird.

	A	B	C	D	E
1					
2	**Berechnen von Standardabweichung und Varianz**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	29	68	
8	4	m	33	72	
9	5	w	35	74	
10	6	w	35	63	
11	7	m	43	68	
12	8	w	43	79	
13	9	m	43	58	
14	10	w	43	79	
15		STABW.S()	7,74	6,65	
16		STABW.N()	7,35	6,31	
17		STABW()	7,74	6,65	
18		STABWN()	7,35	6,31	
19		VAR.S()	59,96	44,27	
20		VAR.P()	53,96	39,84	
21		VARIANZ()	59,96	44,27	
22		VARIANZEN()	53,96	39,84	
23					

Abbildung 7.86 Berechnung von Standardabweichung und Varianz

STABW.S()

STDEV.S()

Syntax: `STABW.S(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=STABW.S(33; 22; 28; 17; 23; 26)`

Ergebnis: `5,49`

Die Funktion `STABW.S()` schätzt die Standardabweichung einer Grundgesamtheit auf der Basis der durch die Argumente angegebenen Werte, die eine Stichprobe aus dieser Grundgesamtheit darstellen.

Die Funktion kann bis zu 255 Argumente verarbeiten. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie mitgerechnet.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!.

Die Standardabweichung ist ein Maß dafür, wie weit die vorhandenen Daten um den Mittelwert streuen. Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichungen vom arithmetischen Mittel. Bei der Ermittlung des Mittelwerts wird bei dieser Funktion aber nicht mit n für den gesamten Umfang der Stichprobe gerechnet, sondern mit n – 1.

Die verwendete Formel ist:

$$=\sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$$

Die Funktion STABW.S() ersetzt seit Excel 2010 die bisherige Funktion STABW(), die aber weiter in der Kategorie **Kompatibilität** angeboten wird. Handelt es sich bei den angegebenen Daten um eine Grundgesamtheit, sollten Sie die Funktion STABW.N() verwenden.

Die Standardabweichung ist zugleich die Wurzel aus der Varianz, die für Stichproben von der Funktion VAR.S() geliefert wird (siehe dort). Mehr zu Standardabweichung und Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

Siehe auch Abbildung 7.86 und STABW.N().

STABWA()

STDEVA()

Syntax:	STABWA(**Wert1**; Wert2; ...)
Beispiel:	=STABWA(33;22;FALSCH;17;WAHR;26)
Ergebnis:	13,45

Die Funktion STABWA() schätzt die Standardabweichung einer Grundgesamtheit auf der Basis der durch die Argumente angegebenen Werte, die eine Stichprobe aus dieser Grundgesamtheit darstellen.

Die Funktion kann seit Excel 2007 bis zu 255 Argumente verarbeiten, in den älteren Versionen nur bis zu 30. Das erste Argument ist erforderlich. Im Unterschied zur Funktion STABW.S() bzw. STABW() werden von dieser Funktion auch Textwerte und Wahrheitswerte mit ausgewertet. Textwerte zählen dabei 0. WAHR wird mit 1, FALSCH mit 0 gewer-

tet. Werden Zahlen als Text direkt als Argument angegeben, werden auch die jeweiligen Zahlenwerte berücksichtigt.

```
=STABWA("10"&"500";10000;"8000")
```

ergibt beispielsweise `1322,87566`.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler `#DIV/0!`.

Die Standardabweichung ist ein Maß dafür, wie weit die vorhandenen Daten um den Mittelwert streuen. Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichungen vom arithmetischen Mittel. Bei der Ermittlung des Mittelwerts wird bei dieser Funktion aber nicht mit `n` für den gesamten Umfang der Stichprobe gerechnet, sondern mit `n – 1`.

	A	B	C	D	E
1					
2	**Berechnen von Standardabweichung und Varianz (A)**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	29	Angabe fehlt	
8	4	m	Angabe fehlt	72	
9	5	w	35	74	
10	6	w	35	63	
11	7	m	43	Abgabe fehlt	
12	8	w	43	79	
13	9	m	43	58	
14	10	w	43	79	
15		STABWA()	13,60	30,24	
16		STABWNA()	12,90	28,69	
17		VARIANZA()	184,99	914,67	
18		VARIANZENA()	166,49	823,20	
19					

Abbildung 7.87 Die A-Variante der Funktionen zur Berechnung der Streuung

Handelt es sich bei den vorliegenden Daten um eine Grundgesamtheit, sollten Sie die entsprechende Funktion `STABWNA()` verwenden.

Abbildung 7.87 zeigt ein Beispiel, in dem für einige Personen noch keine Angabe zu Alter und Gewicht vorliegen. Die Funktion wertet die entsprechenden Texthinweise mit dem Wert `0`. Die Standardabweichung ist dadurch wesentlich höher, als wenn die Zellen mit den Texten nicht berücksichtigt würden.

STABWNA()

STDEVPA()

Syntax: STABWNA(**Wert1**; Wert2; ...)
Beispiel: =STABWNA(33;22;FALSCH;17;WAHR;26)
Ergebnis: 12,28

Die Funktion STABWNA() berechnet für die Werte in der Argumentliste die Standardabweichung vom arithmetischen Mittel. Dabei werden die in der Argumentliste gegebenen Daten als Grundgesamtheit behandelt.

Die Funktion kann seit Excel 2007 bis zu 255 Argumente verarbeiten, in den älteren Versionen nur bis zu 30. Das erste Argument ist erforderlich. Im Unterschied zur Funktion STABW.P() bzw. STABWN() werden von dieser Funktion auch Textwerte und Wahrheitswerte mit ausgewertet. Textwerte zählen dabei 0. WAHR wird mit 1, FALSCH mit 0 gewertet. Werden Zahlen als Text direkt als Argument angegeben, werden auch die jeweiligen Zahlenwerte berücksichtigt.

```
=STABWNA("10"&"500";10000;"8000")
```

ergibt beispielsweise 1080,12345.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!.

Die Standardabweichung ist ein Maß dafür, wie weit die vorhandenen Daten um den Mittelwert streuen. Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichungen vom arithmetischen Mittel. Bei der Ermittlung des Mittelwerts wird anders als bei den Funktionen STABW() und STABWA() mit n für den Umfang der Grundgesamtheit gerechnet.

Handelt es sich bei den vorliegenden Daten um eine Stichprobe, sollten Sie die entsprechende Funktion STABWA() verwenden.

Siehe auch Abbildung 7.87.

STANDARDISIERUNG()

STANDARDIZE()

Syntax: STANDARDISIERUNG(**x**; **Mittelwert**; **Standabwn**)
Beispiel: siehe Abbildung 7.88

Die Funktion STANDARDISIERUNG() berechnet Werte aus einer Normalverteilung, die durch ihren Mittelwert und die Standardabweichung gekennzeichnet ist, in Werte einer Standardnormalverteilung um. Dabei ist x der Wert, der standardisiert werden soll, Mittelwert ist das arithmetische Mittel der Verteilung, das Argument Standabwn liefert die Standardabweichung der Verteilung.

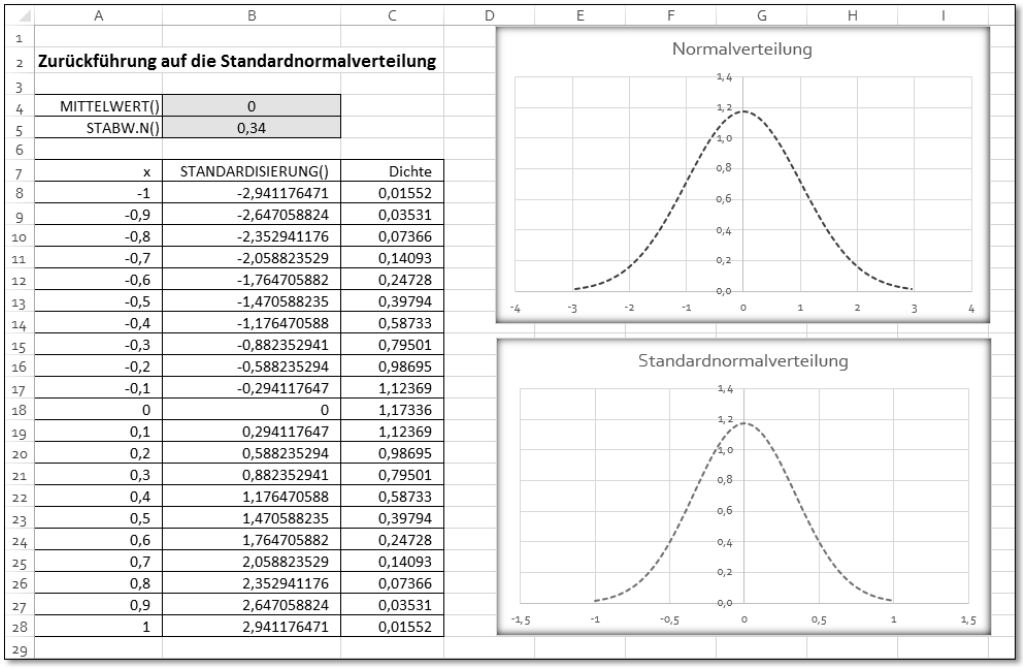
Die Funktion zieht dabei von dem jeweiligen x-Wert den Mittelwert ab und dividiert das Ergebnis durch die angegebene Standardabweichung:

$$z = \frac{x - \mu}{\sigma}$$

Mit Hilfe dieser Funktion lassen sich alle Normalverteilungen in Standardnormalverteilungen übersetzen, was nachfolgende Berechnungen häufig vereinfachen kann.

Abbildung 7.88 zeigt in Spalte A und C die Werte einer Normalverteilung mit den in Zelle B3 und B4 angegebenen Werten für den Mittelwert und die Standardabweichung. In Spalte B sind die Werte aus Spalte A standardisiert, die erste Formel lautet:

```
=STANDARDISIERUNG(A8;$B$4;$B$5)
```



	A	B	C
2	**Zurückführung auf die Standardnormalverteilung**		
4	MITTELWERT()	0	
5	STABW.N()	0,34	
7	x	STANDARDISIERUNG()	Dichte
8	-1	-2,941176471	0,01552
9	-0,9	-2,647058824	0,03531
10	-0,8	-2,352941176	0,07366
11	-0,7	-2,058823529	0,14093
12	-0,6	-1,764705882	0,24728
13	-0,5	-1,470588235	0,39794
14	-0,4	-1,176470588	0,58733
15	-0,3	-0,882352941	0,79501
16	-0,2	-0,588235294	0,98695
17	-0,1	-0,294117647	1,12369
18	0	0	1,17336
19	0,1	0,294117647	1,12369
20	0,2	0,588235294	0,98695
21	0,3	0,882352941	0,79501
22	0,4	1,176470588	0,58733
23	0,5	1,470588235	0,39794
24	0,6	1,764705882	0,24728
25	0,7	2,058823529	0,14093
26	0,8	2,352941176	0,07366
27	0,9	2,647058824	0,03531
28	1	2,941176471	0,01552

Abbildung 7.88 Ermitteln von standardisierten Werten

Der Bezug auf die Zellen B4 und B5 wird also absolut gesetzt, damit Sie die Formel die Spalte herunterkopieren können. Das obere Diagramm wertet die Werte in Spalte A und C aus, das zweite liefert mit den Werten in Spalte B und C die dazugehörige Standardnormalverteilung.

Eine Standardnormalverteilung ist eine Normalverteilung mit einem arithmetischen Mittel von `0` und einer Standardabweichung von `1`, vergleiche `NORM.VERT()` und `NORM.S.VERT()`.

STEIGUNG()

SLOPE()

Syntax: `STEIGUNG(`**`Y_Werte`**`;` **`X_Werte`**`)`
Beispiel: `=STEIGUNG({2;3;4};{4;6;8})`
Ergebnis: `0,5`

Die Funktion `STEIGUNG()` liefert die Steigung für die aus den Argumenten `Y_Werte` und `X_Werte` errechnete Regressionsgerade oder Trendlinie, vergleiche hierzu die Funktion `RGP()`. Für `Y_Werte` und `X_Werte` kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden.

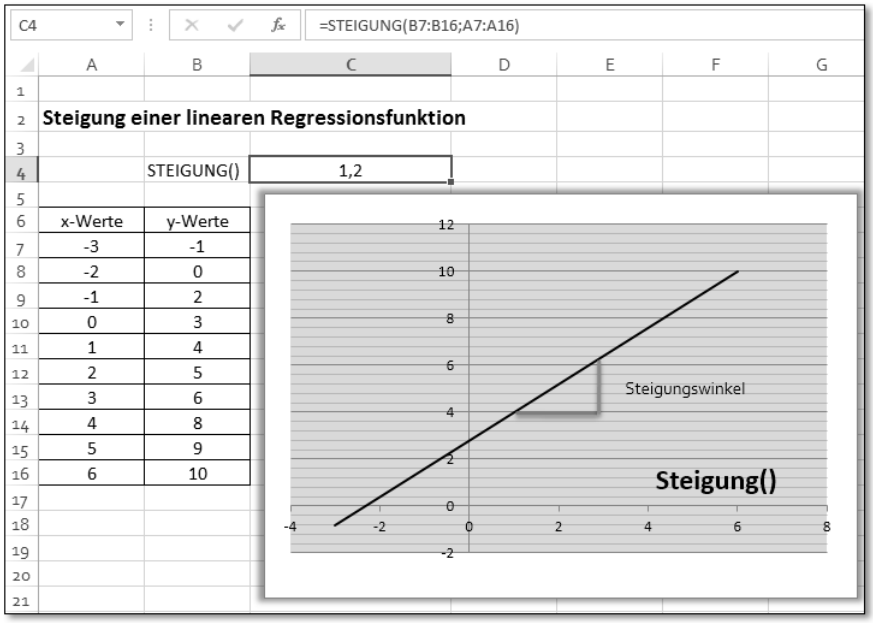


Abbildung 7.89 Berechnung der Steigung einer Regressionsgeraden

Die Regressionsgerade basiert auf der Gleichung:

```
y = b + m * x
```

wobei b der Schnittpunkt der Geraden mit der y-Achse ist und m die Steigung. Der Wert für b kann mit der Funktion ACHSENABSCHNITT() berechnet werden.

Die Argumente Y_Werte und X_Werte müssen jeweils die gleiche Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler #NV. Nicht numerische x-Werte ergeben den Fehler #WERT!.

Siehe auch die Erläuterungen zur Regressionsanalyse in Abschnitt 7.5, »Regressionsanalyse«.

STFEHLERYX()

STEYX()

Syntax:	STFEHLERYX(**Y_Werte**; **X_Werte**)
Beispiel:	=STFEHLERYX({2;3;4};{3;7;9})
Ergebnis:	0,2673

Die Funktion STFEHLERYX() liefert den Standardschätzfehler für die mittels linearer Regression aus den mit dem Argument X_Werte angegebenen Daten geschätzten Y_Werte, vergleiche hierzu RGP(). Dabei stellen die X_Werte die unabhängige, die Y_Werte die abhängige Variable dar.

Für Y_Werte und X_Werte kann jeweils ein Zellbereich oder eine Matrixkonstante angegeben werden. Texte, leere Zellen oder Wahrheitswerte werden ignoriert.

Enthalten Y_Werte und X_Werte unterschiedlich viele Datenelemente, liefert die Funktion den Fehler #NV. Enthalten beide nur ein oder gar kein Datenelement, erscheint der Fehler #DIV/0!.

Der errechnete Wert ist ein Maß für die Zuverlässigkeit der linearen Regression. Er gibt an, wie sehr der errechnete Stichprobenmittelwert um den tatsächlichen Mittelwert der Grundgesamtheit streut. Er sagt damit aus, wie zuverlässig die Schätzung eines y-Werts für einen gegebenen x-Wert ist. Je größer der Wert ist, umso stärker weichen die geschätzten Werte also von den tatsächlichen Werten ab.

Der Wert entspricht dem mit se(y) bezeichneten Kennwert, den die Funktion RGP() liefert, siehe dort.

D5 =STFEHLERYX(C5:C28;B5:B28)

	A	B	C	D	E
1					
2	**Berechnen des Standardfehlers**				
3					
4	Monat	Werbeaufwand	Umsatz	STFEHLERYX()	
5	Jan 12	10000	900000	65409	
6	Feb 12	12000	1090000		
7	Mrz 12	14000	1300000		
8	Apr 12	12000	1200000		
9	Mai 12	11000	900000		
10	Jun 12	10000	990000		
11	Jul 12	10000	980000		
12	Aug 12	11000	1050000		
13	Sep 12	12000	1020000		
14	Okt 12	13000	1200000		
15	Nov 12	13500	1230000		
16	Dez 12	14000	1300000		
17	Jan 13	11000	1090000		
18	Feb 13	12000	1100000		
19	Mrz 13	15000	1400000		
20	Apr 13	12000	1160000		
21	Mai 13	11000	1080000		
22	Jun 13	10000	950000		
23	Jul 13	10000	1000000		
24	Aug 13	11000	1090000		
25	Sep 13	13000	1240000		
26	Okt 13	14000	1360000		
27	Nov 13	13500	1290000		
28	Dez 13	16000	1700000		
29					

Abbildung 7.90 Berechnung des Standardfehlers bei linearer Regression

Siehe dazu auch den Abschnitt 7.5, »Regressionsanalyse«.

SUMQUADABW()

DEVSQ()

Syntax: `SUMQUADABW(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=SUMQUADABW(4;6;5;7;3;5)`

Ergebnis: `10`

Die Funktion `SUMQUADABW()` gibt die Summe der quadratischen Abweichungen der Einzelwerte vom arithmetischen Mittelwert einer Stichprobe an.

Als Argumente können Zahlen, Zell- oder Bereichsbezüge, Bereichsnamen oder Matrizen angegeben werden. Das erste Argument ist erforderlich, die anderen sind optional. Seit Excel 2007 sind insgesamt 255 Argumente möglich, in den älteren Versionen nur 30.

Wahrheitswerte und Zahlen in Textform werden nur berücksichtigt, wenn sie direkt als Argument angegeben werden. Innerhalb eines Bezugs oder einer Matrix werden Text- oder Wahrheitswerte oder leere Zellen ignoriert. Fehlerwerte führen dazu, dass die Funktion diesen Fehlerwert ausgibt.

Die Funktion arbeitet mit der Gleichung:

$$= \sum (x - \bar{x})^2$$

wobei das überstrichene x den Mittelwert angibt.

	A	B	C	D	E	F
1						
2	**Summe der quadrierten Abweichungen**					
3						
4	Monat	Umsatz	SUMQUADABW()		Abw. vom Mittelwert	Quadrat der Abw.
5	Jan 12	900000	780383333333		250833	62917361111
6	Feb 12	1090000			60833	3700694444
7	Mrz 12	1300000			-149167	22250694444
8	Apr 12	1200000			-49167	2417361111
9	Mai 12	900000			250833	62917361111
10	Jun 12	990000			160833	25867361111
11	Jul 12	980000			170833	29184027778
12	Aug 12	1050000			100833	10167361111
13	Sep 12	1020000			130833	17117361111
14	Okt 12	1200000			-49167	2417361111
15	Nov 12	1230000			-79167	6267361111
16	Dez 12	1300000			-149167	22250694444
17	Jan 13	1090000			60833	3700694444
18	Feb 13	1100000			50833	2584027778
19	Mrz 13	1400000			-249167	62084027778
20	Apr 13	1160000			-9167	84027778
21	Mai 13	1080000			70833	5017361111
22	Jun 13	950000			200833	40334027778
23	Jul 13	1000000			150833	22750694444
24	Aug 13	1090000			60833	3700694444
25	Sep 13	1240000			-89167	7950694444
26	Okt 13	1360000			-209167	43750694444
27	Nov 13	1290000			-139167	19367361111
28	Dez 13	1700000			-549167	301584027778
29	MITTELWERT()	1150833			SUMME()	780383333333
30						

Abbildung 7.91 Summe der quadrierten Abweichungen vom Mittelwert einer Stichprobe

Das Ergebnis dieser Funktion wird häufig in der Statistik verwendet, z. B. ist es Bestandteil und Ausgangspunkt der Berechnung von Varianz und Standardabweichung.

Siehe auch MITTELABW(), STABW.S() und STABW.N().

T.INV()

T.INV()

Syntax:	T.INV(**Wahrsch**; **Freiheitsgrade**)
Beispiel:	=T.INV(0,05; 4)
Ergebnis:	-2,13

Die Funktion T.INV() liefert linksseitige Quantile der t-Verteilung und ist damit die Umkehrung von T.VERT(). Die wiedergegebenen Werte sind in statistischen Tabellenwerken als t für einseitige Tests tabelliert.

Mit Wahrsch wird die zur t-Verteilung gehörige Wahrscheinlichkeit angegeben, der Wert für Freiheitsgrade wird aus den zu vergleichenden Größen ermittelt.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch ≤ 0 oder > 1 oder Freiheitsgrade < 1, erscheint der Fehler #ZAHL!.

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von TINV() angeboten werden. Die andere ist T.INV.2S().

Siehe auch T.INV.2S(), T.TEST(), T.VERT(), T.VERT.2S(), T.VERT.RE() und GAMMA().

T.INV.2S()

T.INV.2T()

Syntax:	T.INV.2S(**Wahrsch**; **Freiheitsgrade**)
Beispiel:	=T.INV.2S(0,05; 4)
Ergebnis:	2,776

Die Funktion T.INV.2S() liefert zweiseitige Quantile der t-Verteilung und ist damit die Umkehrung der ebenfalls zweiseitigen Funktion T.VERT.2S(). Die wiedergegebenen Werte sind in statistischen Tabellenwerken als t für zweiseitige Tests (Tests, bei denen die Werte nach beiden Seiten abweichen können) tabelliert.

Mit Wahrsch wird die zur t-Verteilung gehörige Wahrscheinlichkeit angegeben, der Wert für Freiheitsgrade wird aus den zu vergleichenden Daten, Anzahl Werte –1, ermittelt.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch < 0 oder > 1 oder Freiheitsgrade < 1, erscheint der Fehler #ZAHL!.

7.9.8 Einsatz der Funktion in Testverfahren

Für die Auswertung von t-verteilten Daten gibt es verschiedene Testverfahren. Der prinzipielle Ablauf des t-Tests umfasst folgende Schritte:

1. Aus den zu vergleichenden Größen wird ein rechnerischer t-Wert ermittelt.
2. Die Freiheitsgrade werden ermittelt.
3. Der errechnete t-Wert wird mit dem von T.INV.2S() gelieferten verglichen.

7.9.9 Vergleich zwischen Stichprobe und Grundgesamtheit

Ein Beispiel für einen solchen t-Test ist der Vergleich der Häufigkeit eines Merkmals in einer Stichprobe mit der Wahrscheinlichkeit dieses Merkmals in der Grundgesamtheit.

Angenommen, ein Hersteller stellt fest, dass 11% eines Produkts schadhaft sind. Die Wahrscheinlichkeit eines Produktfehlers ist also 0,11. Er prüft ein geändertes Herstellungsverfahren und testet 200 Produkte. Er stellt fest, dass 13 davon defekt sind. Die Nullhypothese ist in diesem Fall, dass sich die Stichprobe und die Grundgesamtheit in Bezug auf die Fehlerhäufigkeit nur zufällig unterscheiden.

Die Testgröße t kann mit folgender Formel berechnet werden:

```
t = ABS(z-n*p)/WURZEL(n*p*(1-p))
```

Dabei ist z = Häufigkeit des Merkmals in der Stichprobe, p = Wahrscheinlichkeit in der Grundgesamtheit und n = Größe der Stichprobe.

Das Ergebnis des Tests ist, dass der errechnete t-Wert größer ist als der mit T.INV.2S() ermittelte Wert für die Irrtumswahrscheinlichkeit 0,05. Die Nullhypothese wird also verworfen. Das neue Verfahren kann als etwas besser gelten, wobei die Wahrscheinlichkeit eines Irrtums kleiner als 0,05 (5%) ist. Die Wahrscheinlichkeit, dass die Nullhypothese doch richtig sein könnte, die mit T.VERT.2S ermittelt wird, beträgt nur 0,04, also 4%.

	A	B	C	D
1				
2	**Vergleich der Häufigkeit eines Merkmals**			
3				
4	Wahrscheinlichkeit Grundgesamtheit	**p**	0,11	
5	Häufigkeit in Stichprobe	**z**	13	
6	Größe der Stichprobe	**n**	200	
7				
8	t-Wert		2,033930766	**=ABS(_z-n*p)/WURZEL(n*p*(1-p))**
9	T.INV.2S(0,05)		1,971956544	**=T.INV.2S(0,05;n-1)**
10	T.VERT.2S(t)		0,043285814	**=T.VERT.2S(C8;n-1)**
11				

Abbildung 7.92 Vergleich der Häufigkeit in der Stichprobe mit der Wahrscheinlichkeit in der Grundgesamtheit

7

7.9.10 Vergleich zweier Stichproben

Abbildung 7.93 zeigt einen t-Test für zwei Stichproben aus Untersuchungen zur Knochendichte, mit dem geprüft wird, ob die Unterschiede als signifikant oder nur als zufällig einzustufen sind. Der mit `T.TEST()` errechnete Prüfwert ist deutlich kleiner als der mit `T.INV.2S()` errechnete kritische t-Wert, also kann davon ausgegangen werden, dass die Unterschiede der beiden Stichproben nicht signifikant sind.

	A	B	C	D	E
1					
2	**Berechnen des kritischen Werts der t-Verteilung**				
3					
4	Stichprobe 1	Knochendichte in mg/cm^3		Stichprobe 2	Knochendichte in mg/cm^3
5	p1	19,4		p1	19,2
6	p2	23		p2	19,3
7	p3	29,9		p3	61,9
8	p4	33		p4	69,9
9	p5	67,9		p5	71
10	p6	68		p6	78
11	p7	81		p7	96
12	p8	178		p8	101,4
13	p9	55		MITTELWERT()	64,5875
14	p10	34		VAR.S()	958,4784
15	MITTELWERT()	58,92		STABW.S()	30,9593
16	VAR.S()	2203,680		ANZAHL()	8
17	STABW.S()	46,943			
18	ANZAHL()	10			
19					
20	Freiheitsgrade	16			
21	Wahrsch	0,05			
22	T.TEST()	0,762707427	P-Wert zweiseitig		
23	T.INV.2S()	2,119905299	Kritischer t-Wert bei zweiseitigem t-Test		
24	T.TEST()	0,381353714	P-Wert einseitig		
25	T.INV()	1,745883676	Kritischer t-Wert bei einseitigem t-Test		
26					

Abbildung 7.93 Teststatistik bei t-Verteilung

Die Funktion ist eine der beiden Funktionen, die seit Excel 2010 anstelle von `TINV()` angeboten werden. Die andere ist `T.INV()`. `T.INV.2S()` liefert dasselbe Ergebnis wie die bisherige Funktion `TINV()`, die ebenfalls zweiseitig arbeitet.

Siehe auch `T.INV()`, `T.TEST()`, `T.VERT()`, `T.VERT.2S()`, `T.VERT.RE()` und `GAMMA()`.

T.TEST()

T.TEST()

Syntax:	`T.TEST(`**`Matrix1`**`; `**`Matrix2`**`; `**`Seiten`**`; `**`Typ`**`)`
Beispiel:	`=T.TEST({12; 19; 13; 14; 17}; {15; 17; 16; 15; 17}; 2; 2)`
Ergebnis:	`0,489`

Beim t-Test wird geprüft, ob zwei Stichproben sich in ihrem Mittelwert nur zufällig unterscheiden – das ist die Nullhypothese – oder ob sie sich nicht zufällig, sondern »echt« unterscheiden – das ist die Alternativhypothese. Im ersten Fall kann angenommen werden, dass die beiden Stichproben aus derselben Grundgesamtheit stammen können, im zweiten Fall stammen die Stichproben entweder aus verschiedenen Grundgesamtheiten oder sie sind nicht zufällig entnommen. Beim t-Test wird davon ausgegangen, dass die zu untersuchenden Merkmale wenigstens annähernd normalverteilt sind, wobei die Streuung aber nur geschätzt werden kann.

Die Funktion `T.TEST()` gestattet den direkten Vergleich zweier Stichproben, ohne dass so viele rechnerische Zwischenschritte nötig wären wie bei dem unter `T.INV()` geschilderten Verfahren. Sie liefert als Ergebnis die Wahrscheinlichkeit dafür, dass die beiden Stichproben übereinstimmen.

Die beiden Stichproben werden mit `Matrix1` und `Matrix2` angegeben. Mit `Seiten` wird vorgegeben, ob Abweichungen nach beiden Seiten (2) oder nur nach einer Seite (1) möglich sind.

Ein einseitiger Test betrifft beispielsweise die Frage, ob ein Mittelwert einen bestimmten Wert überschreitet – das wird auch eine gerichtete Hypothese genannt.

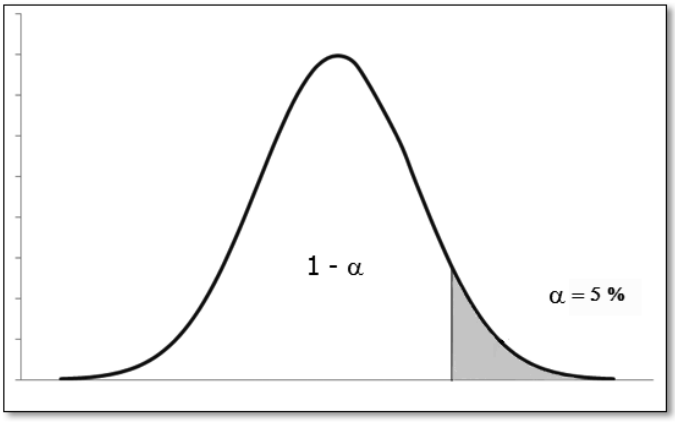


Abbildung 7.94 Einseitiger t-Test

Ein zweiseitiger Test basiert dagegen auf einer ungerichteten Hypothese, weil nur geprüft wird, ob ein Wert in der einen oder der anderen Richtung abweicht.

Die Irrtumswahrscheinlichkeit α ist die Wahrscheinlichkeit dafür, dass die Nullhypothese auch dann noch richtig sein kann, wenn sie durch das Testergebnis verworfen werden muss. Sie wird beim zweiseitigen Test gewissermaßen auf die beiden Endbereiche der Dichtekurve aufgeteilt.

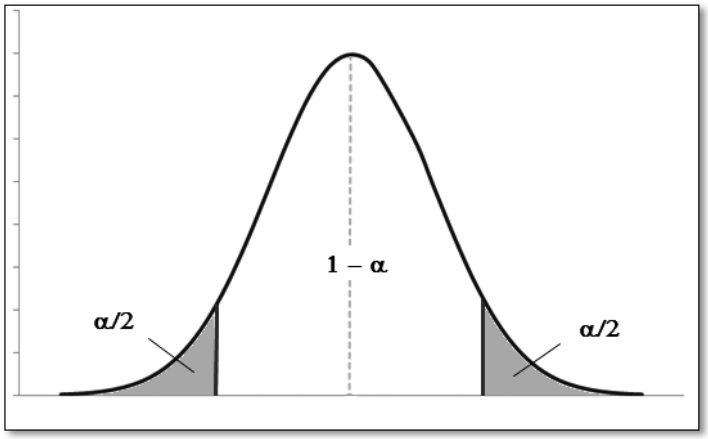


Abbildung 7.95 Zweiseitiger t-Test

Mit `Typ` wird zusätzlich der Charakter der Stichproben angegeben:

Typ	Charakter
1	gepaart – zwei Stichproben gleicher Größe
2	zwei Stichproben mit gleicher Varianz
3	zwei Stichproben mit unterschiedlicher Varianz

Tabelle 7.3 Werte für das Argument `Typ`

Enthalten die mit `Matrix1` und `Matrix2` angegebenen Datenreihen nicht numerische Elemente, werden sie ignoriert. Sind aber jeweils weniger als zwei numerische Elemente vorhanden, gibt die Funktion den Fehler `#DIV/0!` zurück. Ist `Typ = 1` (gepaart), müssen beide Datenreihen die gleiche Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Ist `Seiten` oder `Typ` nicht numerisch, erscheint der Fehler `#WERT!`. Bei ungültigen Werten für `Seiten` (`<> 1` oder `2`) oder `Typ` (`<> 1`, `2` oder `3`) erscheint der Fehler `#ZAHL!`.

	A	B	C	D	E	F
1						
2	**Vergleich der Mittelwerte zweier Stichproben**					
3						
4	Stichprobe 1	Knochendichte in mg/cm³		Stichprobe 2	Knochendichte in mg/cm³	
5	p1	19,4		p1	19,2	
6	p2	23		p2	19,3	
7	p3	29,9		p3	61,9	
8	p4	33		p4	69,9	
9	p5	67,9		p5	71	
10	p6	68		p6	78	
11	p7	81		p7	96	
12	p8	178		p8	101,4	
13	p9	55		MITTELWERT()	64,588	
14	p10	34		VAR.S()	958,478	
15	MITTELWERT()	58,92		ANZAHL()	8	
16	VAR.S()	2203,680				
17	ANZAHL()	10		T.TEST()	0,762707427	
18						

Abbildung 7.96 t-Test für zwei Stichproben

Die Funktion ersetzt seit Excel 2010 die bisherige Funktion `TTEST()`. Siehe auch `T.INV()`, `T.INV.2S()`, `T.VERT()`, `T.VERT.2S()`, `T.VERT.RE()` und `GAMMA()`.

T.VERT()

T.DIST()

Syntax: `T.VERT(x; Freiheitsgrade; Kumuliert)`

Beispiel: `=T.VERT(2;4;FALSCH)`

Ergebnis: `0,066`

Die Funktion `T.VERT()` liefert linksseitige Werte zur Wahrscheinlichkeit für eine t-verteilte Zufallsvariable.

Das Argument `x` ist das Quantil der Verteilung, dessen Wahrscheinlichkeit berechnet werden soll. Mit `Freiheitsgrade` wird die Anzahl der Freiheitsgrade angegeben. Mit `Kumuliert` lässt sich festlegen, ob die Dichtefunktion (`FALSCH`) oder die Verteilungsfunktion (`WAHR`) ausgegeben wird.

Ist eines der Argumente x oder Freiheitsgrade nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Freiheitsgrade = 0, erscheint der Fehler #DIV/0!, ist Freiheitsgrade negativ, erscheint der Fehler #ZAHL!.

T.VERT() ist die Umkehrung von T.INV(). Deshalb gilt: Wenn:

```
x = T.INV(W; df)
```

dann ist:

```
W = T.VERT(x; df; WAHR)
```

Die Verwendung einer t-Verteilung liegt immer dann nahe, wenn bei Stichproben zwar angenommen werden kann, dass sie normalverteilt sind, die Varianz der Grundgesamtheit, aus der sie gezogen sind, aber nur geschätzt werden kann. Das gilt insbesondere dann, wenn die Stichproben relativ klein sind, beispielsweise unter 30. In solchen Fällen ist der Schätzwert für die Varianz selbst eine Zufallsvariable. Die Statistik der Normalverteilung gilt hier nicht mehr. Die Verteilung wird dann definiert durch den Quotienten aus einer normalverteilten Zufallsvariablen, meist dem Stichprobenmittel, und einer χ^2-verteilten Zufallsvariablen, der geschätzten Varianz:

$$t_n = \frac{Z}{\sqrt{\chi_n{}^2}/n}$$

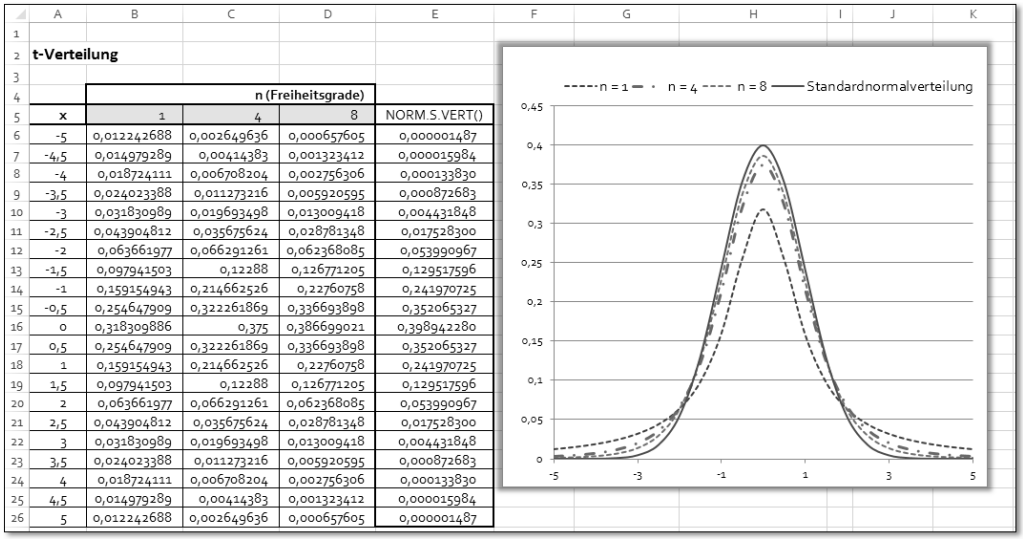
wobei die beiden Variablen Z und $\chi_n{}^2$ unabhängig voneinander sind. Dieses Verhältnis ist t-verteilt, wobei als Parameter mit n die Anzahl der Freiheitsgrade verwendet wird.

Für die Dichtefunktion der t-Verteilung gilt folgende Formel:

$$= \frac{\Gamma((n+1)/2)}{\sqrt{n\pi}\,\Gamma(n/2)(1+x^2/n)^{(n+1)/2}}$$

wobei n für Freiheitsgrade und Γ für die Funktion GAMMA() steht.

Abbildung 7.97 zeigt verschiedene Dichtefunktionen zu T.VERT(). Es wird deutlich, dass die Glockenkurve wie die der Normalverteilung symmetrisch ist. Außerdem nähert sie sich immer mehr der Kurve der Standardnormalverteilung an, je höher die Anzahl der Freiheitsgrade, also je größer der Stichprobenumfang ist.



t-Verteilung

x	n (Freiheitsgrade) 1	4	8	NORM.S.VERT()
-5	0,012242688	0,002649636	0,000657605	0,000001487
-4,5	0,014979289	0,00414383	0,001323412	0,000015984
-4	0,018724111	0,006708204	0,002756306	0,000133830
-3,5	0,024023388	0,011273216	0,005920595	0,000872683
-3	0,031830989	0,019693498	0,013009418	0,004431848
-2,5	0,043904812	0,035675624	0,028781348	0,017528300
-2	0,063661977	0,066291261	0,062368085	0,053990967
-1,5	0,097941503	0,12288	0,126771205	0,129517596
-1	0,159154943	0,214662526	0,22760758	0,241970725
-0,5	0,254647909	0,322261869	0,336693898	0,352065327
0	0,3183099886	0,375	0,3866999021	0,398942280
0,5	0,254647909	0,322261869	0,336693898	0,352065327
1	0,159154943	0,214662526	0,22760758	0,241970725
1,5	0,097941503	0,12288	0,126771205	0,129517596
2	0,063661977	0,066291261	0,062368085	0,053990967
2,5	0,043904812	0,035675624	0,028781348	0,017528300
3	0,031830989	0,019693498	0,013009418	0,004431848
3,5	0,024023388	0,011273216	0,005920595	0,000872683
4	0,018724111	0,006708204	0,002756306	0,000133830
4,5	0,014979289	0,00414383	0,001323412	0,000015984
5	0,012242688	0,002649636	0,000657605	0,000001487

Abbildung 7.97 t-Verteilungen im Vergleich zur Standardnormalverteilung

Die Funktion ist eine der drei Funktionen, die seit Excel 2010 anstelle von `TVERT()` angeboten werden. Die anderen sind `T.VERT.RE()` und `T.VERT.2S()`.

Siehe auch die Abschnitte in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, zu den t-Test-Tools.

Siehe auch `T.INV()`, `T.INV.2S()`, `T.TEST()`, `T.VERT.2S()`, `T.VERT.RE()`, `KONFIDENZ.T()` und `GAMMA()`.

T.VERT.2S()

T.DIST.2T()

Syntax: `T.VERT.2S(`**`x; Freiheitsgrade`**`)`

Beispiel: `=T.VERT.2S(2,57; 5; 2)`

Ergebnis: `0,05`

Die Funktion `T.VERT.2S()` liefert zweiseitige Werte zur Wahrscheinlichkeit einer t-verteilten Zufallsvariablen.

`T.VERT.2S()` ist die Umkehrung von `T.INV.2S()`. Wenn:

```
x = T.INV.2S(W; df)
```

dann ist:

```
W = T.VERT.2S(x; df)
```

Ist eines der Argumente x oder Freiheitsgrade nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Freiheitsgrade < 1, erscheint der Fehler #ZAHL!.

Die Funktion ist eine der drei Funktionen, die seit Excel 2010 anstelle von TVERT() angeboten werden. Die anderen sind T.VERT() und T.VERT.RE(). Die Funktion liefert dasselbe Ergebnis wie die Funktion TVERT() mit dem Argument Seiten = 2.

Siehe auch T.INV(), T.INV.2S(), T.TEST(), T.VERT(), T.VERT.RE() und GAMMA().

T.VERT.RE()

T.DIST.RT()

Syntax: T.VERT.RE(**x**; **Freiheitsgrade**)
Beispiel: =T.VERT.RE(2;4)
Ergebnis: 0,058

Die Funktion T.VERT.RE() liefert rechtsseitige Werte für die Wahrscheinlichkeit einer t-verteilten Zufallsvariablen.

Ist eines der Argumente x oder Freiheitsgrade nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Freiheitsgrade < 1, erscheint der Fehler #ZAHL!.

Die Funktion ist eine der drei Funktionen, die seit Excel 2010 anstelle von TVERT() angeboten werden. Die anderen sind T.VERT() und T.VERT.2S(). Die Funktion liefert dasselbe Ergebnis wie die Funktion TVERT() mit dem Argument Seiten = 1.

Siehe auch T.INV(), T.INV.2S(), T.TEST(), T.VERT(), T.VERT.2S() und GAMMA().

TREND()

TREND()

Syntax: TREND(**Y_Werte**; X_Werte; Neue_x_Werte;Konstante)
Beispiel: siehe Abbildung 7.98

Die Funktion TREND() liefert auf der Basis der linearen Regression geschätzte y-Werte für eine Trendberechnung. Die Funktion ist eines der Instrumente für die Prognose zukünftiger Entwicklungen auf der Basis der Auswertung vorliegender Daten über den Zusammenhang zwischen verschiedenen Werten.

Das Argument Y_Werte gibt schon bekannte y-Werte an; X_Werte sind bekannte x-Werte. Werden sie nicht angegeben, verwendet Excel stattdessen eine Matrix {1.2.3...}, die jedem y-Wert einen x-Wert zuordnet.

Mit Neue_x_Werte lassen sich zusätzlich zu den vorhandenen x-Werten weitere x-Werte angeben, für die y-Werte geschätzt werden sollen. Dabei wird davon ausgegangen, dass die Werte einem linearen Trend zugeordnet werden können, d.h., dass sich die Beziehung zwischen den x- und y-Werten durch die Gleichung

```
y = mx + b
```

darstellen lässt, wobei m die Steigung der Geraden und b ihren Schnittpunkt mit der y-Achse festlegt.

Werden für die x-Werte mehrere Spalten angegeben, dann wird als Gleichung für die Gerade angenommen:

```
y = m1*x1 + m2*x2 + ... + b
```

Wird für Y_Werte ein Bereich oder eine Matrix mit nur einer Spalte oder nur einer Zeile angegeben, wird jede Spalte bzw. Zeile in X_Werte als eigenständige Variable interpretiert.

Die Anzahl der Elemente muss in beiden Bereichen gleich sein, andernfalls liefert die Funktion den Fehler #BEZUG!. Nicht numerische Werte führen zu dem Fehler #WERT!. Umfasst X_Werte mehrere Variablengruppen, muss Y_Werte ein Vektor sein, der aus nur einer Zeile oder Spalte besteht.

Fehlt das Argument Neue_x-Werte, wird angenommen, dass die Funktion nur mit den Werten rechnen soll, die das Argument X_Werte liefert.

Die Funktion berechnet aus den gegebenen y- und x-Werten eine Trendgerade und gibt dann für Neue_x_Werte die y-Werte an, die auf dieser Geraden liegen. Dies wird sichtbar, wenn in das Diagramm aus den gesamten Daten eine lineare Trendlinie eingezogen wird.

Das Argument Konstante ist ein Wahrheitswert, der bestimmt, wie die in der Gleichung aufgeführte Konstante b gehandhabt werden soll. WAHR oder nicht angegeben bedeutet, dass b mit dem jeweiligen Wert berechnet wird, der sich aus den bekannten y-Werten

ableiten lässt, bei FALSCH wird b auf den Wert 0 gesetzt und m so angepasst, dass die Gleichung auf

```
y = mx
```

reduziert werden kann.

TREND() muss als Matrixfunktion eingegeben werden: Zunächst wird der Ausgabebereich markiert, und zwar in der Größe, die für die gewünschten Werte benötigt wird. Dann wird die Funktion eingegeben und mit Strg + ⇧ + ↵ beendet.

Abbildung 7.98 zeigt die Anwendung der Funktion bei der Schätzung zukünftiger Umsätze. Dabei wird ein lineares Wachstum angenommen.

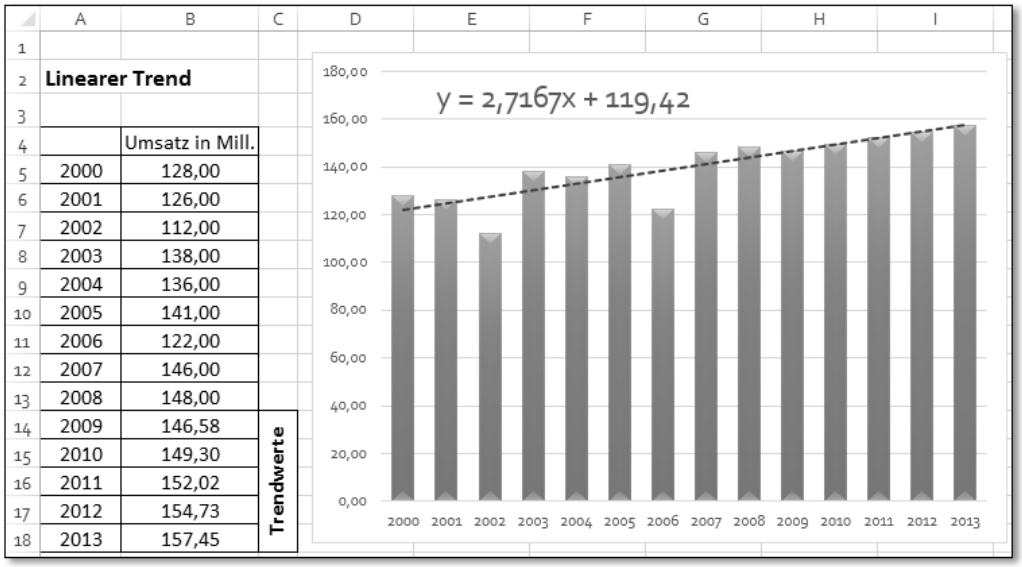


Abbildung 7.98 Lineare Vorausschätzung mit TREND()

7.9.11 Trendberechnung mit mehreren unabhängigen Variablen

Es ist eher selten, dass eine bestimmte Entwicklung nur durch einen einzigen Faktor beeinflusst wird. Das folgende Beispiel zeigt, wie mehrere unabhängige Variablen bei der Trendberechnung berücksichtigt werden können. In diesem Fall werden Daten für den Werbeaufwand getrennt nach Printwerbung und Onlinewerbung in eigenen Spalten angegeben. Die Formel für die Berechnung der Trendwerte muss dann entsprechend erweitert werden. Für X_Werte wird der Bereich B6:C14 angegeben, für Neue_x_Werte

ebenfalls ein zweispaltiger Bereich: B15:C19. Die Matrixformel für die zu schätzenden neuen y-Werte heißt dann entsprechend:

```
{=TREND(D6:D14;B6:C14;B15:C19;WAHR)}
```

	A	B	C	D	E
1					
2	**Linearer Trend mit zwei unabhängigen Variablen**				
3					
4		Ausgaben für			
5		Printwerbung	Onlinewerbung	Umsatz	
6	2000	230000	120000	12600000	
7	2001	250000	130000	12500000	
8	2002	244000	124000	11200000	
9	2003	266000	196000	13800000	
10	2004	270000	272000	13600000	
11	2005	272000	162000	14100000	
12	2006	260000	144000	12200000	
13	2007	282000	162000	14600000	
14	2008	296000	176000	14800000	
15	2009	295000	179000	14803665	
16	2010	300400	170003	15039743	
17	2011	305600	180300	15310487	
18	2012	314000	190400	15732885	
19	2013	318000	203000	15951863	
20					

Abbildung 7.99 Lineare Vorausschätzung mit mehreren Variablen

Siehe auch die Funktion `VARIATION()` zur Berechnung eines exponentiellen Trends und die Erläuterungen in Abschnitt 7.5, »Regressionsanalyse«.

VAR.P()

VAR.P()

Syntax:	`VAR.P(`**`Zahl1`**`; Zahl2; ...)`
Beispiel:	`=VAR.P(33; 22; 28; 17; 23; 26)`
Ergebnis:	`25,14`

Die Funktion `VAR.P()` berechnet für die Werte in der Argumentliste die Varianz. Dabei stellen die in der Argumentliste angegebenen Daten eine Grundgesamtheit dar.

Die Funktion kann bis zu 255 Argumente verarbeiten. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie mitgerechnet.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler `#DIV/0!`.

Die Varianz ist eine wichtige Kennzahl für die Streuung von metrischen Werten und zugleich Basis für die Ermittlung der Standardabweichung. Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte.

	A	B	C	D
1				
2	**Berechnen der Varianz**			
3				
4	Person	Alter	Gewicht	
5	1	23	69	
6	2	25	66	
7	3	29	68	
8	4	33	72	
9	5	35	74	
10	6	35	63	
11	7	43	68	
12	8	43	79	
13	9	43	58	
14	10	43	79	
15	11	45	68	
16	12	46	72	
17	13	47	77	
18	14	47	56	
19	15	51	60	
20	16	53	62	
21	17	53	58	
22	18	54	61	
23	19	55	55	
24	20	63	57	
25	VAR.S()	111,6	59,6	
26	VARIANZ()	111,6	59,6	
27	VAR.P()	106,0	56,6	
28	VARIANZEN()	106,0	56,6	
29	MITTELWERT()	43,3	66,1	
30				

Abbildung 7.100 Berechnung der Varianz

Die verwendete Formel ist:

$$= \frac{\sum (x - \bar{x})^2}{n}$$

Die Standardabweichung ist wiederum nichts anderes als die Wurzel der Varianz, womit dann wieder eine Größenordnung auf der Ebene der vorhandenen Abweichungen erreicht wird.

VAR.P() ersetzt seit Excel 2010 die bisherige Funktion VARIANZEN(). Mehr zur Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

Abbildung 7.100 zeigt Werte für die Varianz bei den Merkmalen »Alter« und »Gewicht«, wobei die Daten einmal als Grundgesamtheit und einmal als Daten einer Stichprobe behandelt werden. Wie zu sehen ist, berechnet VAR.P() einen etwas geringeren Wert als VAR.S(), da ja einmal mit n und das andere Mal mit n – 1 dividiert wird.

VAR.S()

VAR.S()

Syntax:	VAR.S(**Zahl1**; Zahl2; ...)
Beispiel:	=VAR.S(33; 22; 28; 17; 23; 26)
Ergebnis:	30,17

Die Funktion VAR.S() schätzt die Varianz einer Grundgesamtheit auf der Basis der durch die Argumente angegebenen Werte, die eine Stichprobe aus dieser Grundgesamtheit darstellen.

Die Funktion kann bis zu 255 Argumente verarbeiten. Das erste Argument ist erforderlich. Bei Bezügen auf Zellbereiche werden Texteinträge, Fehlerwerte, Wahrheitswerte und leere Zellen ignoriert. Wird eine Zahl in Textform oder ein Wahrheitswert dagegen direkt als Argument angegeben, werden sie mitgerechnet.

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!.

Die Varianz ist eine wichtige Kennzahl für die Streuung von metrischen Werten und zugleich Basis für die Ermittlung der Standardabweichung. Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte – 1.

Die verwendete Formel ist:

$$= \frac{\sum (x - \bar{x})^2}{n - 1}$$

Die Standardabweichung ist wiederum nichts anderes als die Wurzel der Varianz, womit dann wieder eine Größenordnung auf der Ebene der vorhandenen Abweichungen erreicht wird.

VAR.S() ersetzt seit Excel 2010 die bisherige Funktion VARIANZ(). Handelt es sich bei den angegebenen Daten um eine Grundgesamtheit, sollten Sie die Funktion VAR.P() verwenden. Siehe auch die Abbildung 7.100 zu VAR.P().

Mehr zur Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

VARIANZA()

VARA()

Syntax:	VARIANZA(**Wert1**; Wert2; ...)
Beispiel:	=VARIANZA(33;22;FALSCH;17;WAHR)
Ergebnis:	199,3

Die Funktion VARIANZA() schätzt für die Werte in der Argumentliste die vermutete Varianz der Grundgesamtheit. Dabei werden die in der Argumentliste angegebenen Daten als Stichprobe aus dieser Grundgesamtheit genommen.

Die Funktion kann seit Excel 2007 bis zu 255 Argumente verarbeiten, in den älteren Versionen nur bis zu 30. Das erste Argument ist erforderlich.

Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte – 1.

Im Unterschied zur Funktion VARIANZ() werden von dieser Funktion auch Textwerte und Wahrheitswerte mit ausgewertet. Textwerte zählen dabei 0. WAHR wird mit 1, FALSCH mit 0 gewertet. Direkt als Argument angegebene Zahlen in Textform werden als Zahlen berücksichtigt.

Abbildung 7.101 zeigt Werte für die Varianz bei den Merkmalen »Alter« und »Gewicht«, wobei die Daten einmal als Grundgesamtheit und einmal als Daten einer Stichprobe be-

handelt werden. Die Zellen mit dem Eintrag »keine Angabe« werden jeweils mit dem Wert 0 berücksichtigt.

	A	B	C	D
1				
2	**Berechnen der Varianz (A)**			
3				
4	Person	Alter	Gewicht	
5	1	23	69	
6	2	25	66	
7	3	29	68	
8	4	33	72	
9	5	35	74	
10	6	keine Angabe	keine Angabe	
11	7	43	68	
12	8	43	79	
13	9	43	58	
14	10	43	79	
15	11	45	68	
16	12	46	72	
17	13	keine Angabe	keine Angabe	
18	14	47	56	
19	15	51	60	
20	16	53	62	
21	17	53	58	
22	18	54	61	
23	19	55	55	
24	20	63	57	
25	VARIANZA()	286,9	461,1	
26	VARIANZENA()	272,6	438,1	
27	MITTELWERTA()	39,2	59,1	
28				

Abbildung 7.101 Berechnung der Varianz mit Berücksichtigung nicht numerischer Werte

Mehr zur Berechnung der Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

VARIANZENA()

VARPA()

Syntax: `VARIANZENA(`**`Wert1`**`; Wert2; ...)`

Beispiel: `=VARIANZENA(33;22;FALSCH;17;WAHR;26)`

Ergebnis: `150,91`

Die Funktion VARIANZENA() berechnet für die Werte in der Argumentliste die Varianz. Dabei werden die gegebenen Daten als Grundgesamtheit behandelt.

Die Argumentliste kann seit Excel 2007 bis zu 255 Werte enthalten, in den älteren Versionen nur bis zu 30. Das erste Argument ist erforderlich. Im Unterschied zu den Funktionen VAR.P() und VARIANZEN() werden von dieser Funktion auch Textwerte und Wahrheitswerte mit ausgewertet. Textwerte zählen dabei 0. WAHR wird mit 1, FALSCH mit 0 gewertet. Direkt als Argument angegebene Zahlen in Textform werden als Zahlen berücksichtigt.

```
=VARIANZENA("10"&"500";10000;"8000")
```

ergibt beispielsweise 1166666,67.

Enthält ein Bereich überhaupt keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!.

Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte.

Mehr zur Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

VARIATION()

GROWTH()

Syntax:	VARIATION(**Y_Werte**; X_Werte; Neue_x_Werte; Konstante)
Beispiel:	siehe Abbildung 7.102

Die Funktion VARIATION() liefert auf der Basis der exponentiellen Regression geschätzte y-Werte für eine Trendberechnung. Die Funktion ist eines der Instrumente für die Prognose zukünftiger Entwicklungen auf der Basis der Auswertung vorliegender Daten über den Zusammenhang zwischen verschiedenen Werten.

Das Argument Y_Werte gibt schon bekannte y-Werte an; X_Werte sind bekannte x-Werte. Werden sie nicht angegeben, verwendet Excel stattdessen eine Matrix {1.2.3...}, die jedem y-Wert einen x-Wert zuordnet.

Mit Neue_x_Werte lassen sich zusätzlich zu den vorhandenen x-Werten weitere x-Werte angeben, für die y-Werte geschätzt werden sollen. Dabei wird davon ausgegangen, dass

die Werte einem exponentiellen Trend zugeordnet werden können, d.h., dass sich die Beziehung zwischen den x- und y-Werten durch die Gleichung

```
y = b * m^x
```

darstellen lässt, wobei `b` den Schnittpunkt der Regressionskurve mit der y-Achse liefert und `m` die Basis ist, für die jeweils `x` als Exponent verwendet wird.

Werden mehrere Spalten mit x-Werten benutzt, dann wird eine Regression nach folgender Gleichung durchgeführt:

```
y = b * m1^x1 * m2^x2 ...
```

Wird für `Y_Werte` ein Bereich oder eine Matrix mit nur einer Spalte oder nur einer Zeile angegeben, wird jede Spalte bzw. Zeile in `X_Werte` als eigenständige Variable interpretiert. Umfasst `X_Werte` mehrere Variablengruppen, muss `Y_Werte` ein Vektor sein, der aus nur einer Zeile oder Spalte besteht.

Die Anzahl der Elemente muss in beiden Bereichen gleich sein, andernfalls liefert die Funktion den Fehler `#BEZUG!`. Ist einer der y-Werte ≤ `0`, liefert die Funktion den Fehler `#ZAHL!`. Nicht numerische Werte führen zu dem Fehler `#WERT!`.

Fehlt das Argument `Neue_x_Werte`, wird angenommen, dass die Funktion nur mit den Werten rechnen soll, die das Argument `X_Werte` liefert.

Das Argument `Konstante` ist ein Wahrheitswert, der bestimmt, wie die in der Gleichung aufgeführte Konstante `b` gehandhabt werden soll. `WAHR` oder nicht angegeben bedeutet, dass `b` mit dem jeweiligen Wert berechnet wird, der sich aus den bekannten y-Werten ableiten lässt, bei `FALSCH` wird `b` auf den Wert `1` gesetzt und `m` so angepasst, dass die Gleichung auf

```
y = m^x
```

reduziert werden kann.

`VARIATION()` muss als Matrixfunktion eingegeben werden: Zunächst wird der Ausgabebereich markiert, und zwar in der Größe, die für die gewünschten Werte benötigt wird. Dann wird die Funktion eingegeben und mit Strg + ⇧ + ↵ beendet.

Abbildung 7.102 zeigt die Anwendung der Funktion bei der Schätzung zukünftiger Umsätze. Dabei wird ein exponentielles Wachstum angenommen, d.h., es wird davon ausgegangen, dass sich die Umsätze jedes Jahr um einen bestimmten Faktor oder Prozent-

satz vergrößern. In dem beigefügten Säulendiagramm ist eine exponentielle Trendlinie eingefügt, um die Entwicklung zu verdeutlichen.

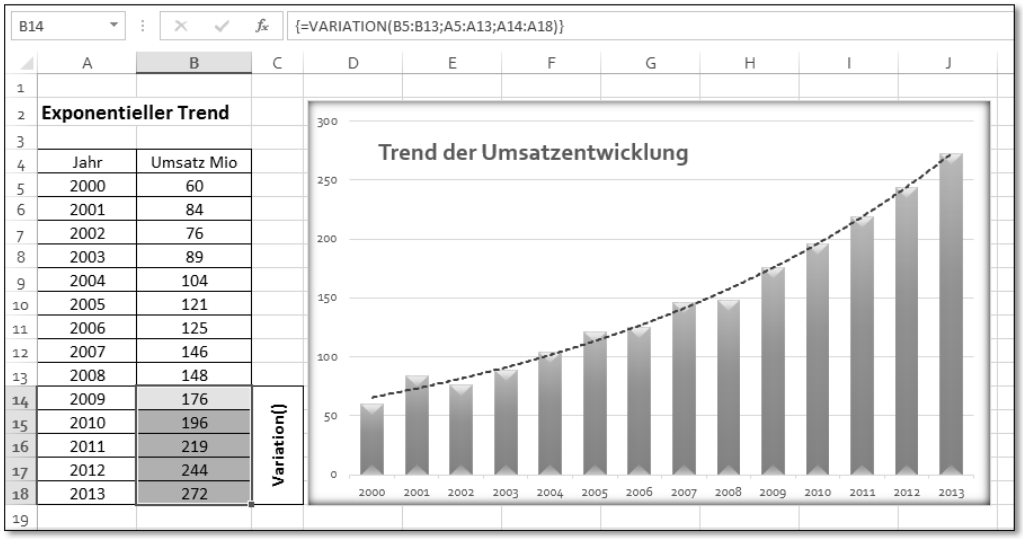


Abbildung 7.102 Exponentielle Vorausschätzung mit `VARIATION()`

Siehe auch die Funktion `TREND()` zur Berechnung eines linearen Trends und die Erläuterungen in Abschnitt 7.5, »Regressionsanalyse«.

VARIATIONEN()

PERMUT()

Syntax:	`VARIATIONEN(n; k)`
Beispiel:	`=VARIATIONEN(52;4)`
Ergebnis:	`6497400`

Die Funktion `VARIATIONEN()` berechnet die Reihe von geordneten Folgen, die mit den angegebenen Argumenten möglich sind. Dabei ist `k` die Anzahl der Elemente, die aus einer Menge von `n` Elementen gewählt werden. Die Funktion dividiert die Fakultät von `n` durch die Fakultät von `n – k`. Im Gegensatz zu der Funktion `KOMBINATIONEN()` werden in dieser Funktion Reihenfolgen berücksichtigt.

Beide Argumente werden auf die Ganzzahl gekürzt, wenn sie mit Dezimalstellen angegeben werden.

Nicht numerische Werte ergeben den Fehler #WERT!. Ist n ≤ 0 oder k < 0 oder n < k, liefert die Funktion den Fehler #ZAHL!.

Im Beispiel wird aus einem Patience-Kartenspiel eine bestimmte Folge von vier Karten gezogen. Wie viele mögliche Folgen ständen zur Verfügung? Die Wahrscheinlichkeit, die Folge von Kreuz-Ass, Pik-Ass, Herz-Ass und Karo-Ass (genau in dieser Reihenfolge) zu ziehen, liegt damit bei 1/6.497.400.

	A	B	C
1			
2	**Berechnung der Möglichkeiten, k Elemente aus einer Menge n zu ziehen**		
3			
4	Menge n	Auswahl k	VARIATIONEN()
5	6	2	30
6	12	4	11.880
7	18	6	13.366.080
8	24	8	29.654.190.720
9	30	10	109.027.350.432.000
10	36	12	599.555.620.984.320.000
11	42	14	4.608.264.443.634.950.000.000
12	48	16	47.177.730.049.421.200.000.000.000
13	54	18	620.558.705.531.887.000.000.000.000.000
14	60	20	10.198.346.916.138.400.000.000.000.000.000.000
15	66	22	204.774.013.400.329.000.000.000.000.000.000.000.000
16	72	24	4.932.727.141.604.180.000.000.000.000.000.000.000.000.000
17	78	26	140.398.430.217.471.000.000.000.000.000.000.000.000.000.000.000
18	84	28	4.661.387.788.121.680.000.000.000.000.000.000.000.000.000.000.000.000

Abbildung 7.103 Die Fülle der Möglichkeiten

Siehe auch VARIATIONEN2().

VARIATIONEN2()

PERMUTATIONA()

Syntax: VARIATIONEN2(**Zahl; Gewählte_Zahl**)
Beispiel: =VARIATIONEN2(36;6)
Ergebnis: 2176782336

Die in Excel 2013 neue Funktion VARIATIONEN2() berechnet die Reihe von geordneten Folgen, die mit den angegebenen Argumenten möglich sind, wenn Wiederholungen zugelassen sind. Dabei gibt Gewählte_Zahl die Anzahl der Elemente an, die aus einer mit

Zahl angegebenen Menge von Elementen gezogen werden. Die verwendete Formel ist sehr einfach:

$=Zahl^{Gewählte_Zahl}$

Beide Argumente werden auf die Ganzzahl gekürzt, wenn sie mit Dezimalstellen angegeben werden.

Nicht numerische Werte ergeben den Fehler #WERT!. Ist Zahl < 0 oder Gewählte_Zahl < 0, liefert die Funktion den Fehler #ZAHL!.

	A	B	C	D
1				
2	**Berechnung der Möglichkeiten, k Elemente aus einer Menge n zu ziehen, wenn Wiederholungen erlaubt sind.**			
3				
4	Menge n	Auswahl k	VARIATIONEN2()	
5	6	2	36	
6	12	4	20.736	
7	18	6	34.012.224	
8	24	8	110.075.314.176	
9	30	10	590.490.000.000.000	
10	36	12	4.738.381.338.321.620.000	
11	42	14	53.148.384.174.432.400.000.000	
12	48	16	794.071.845.499.379.000.000.000.000	
13	54	18	15.243.604.656.924.900.000.000.000.000.000	
14	60	20	365.615.844.006.298.000.000.000.000.000.000.000	
15	66	22	10.714.368.571.740.900.000.000.000.000.000.000.000.000	
16	72	24	376.686.377.243.952.000.000.000.000.000.000.000.000.000.000	
17	78	26	15.648.029.996.035.600.000.000.000.000.000.000.000.000.000.000.000	
18				

Abbildung 7.104 Die Fülle der Möglichkeiten ist bei zugelassenen Wiederholungen noch wesentlich größer.

Siehe auch VARIATIONEN().

WAHRSCHBEREICH()

PROB()

Syntax: WAHRSCHBEREICH(**Beob_Werte**; **Beob_Wahrsch**; **Untergrenze**; Obergrenze)

Beispiel: =WAHRSCHBEREICH({2;3;4;5;6};{0,1;0,2;0,4;0,2;0,1};5;6)

Ergebnis: 0,3

Die Funktion WAHRSCHBEREICH() berechnet auf der Basis der mit dem Argument Beob_Werte angegebenen bisher beobachteten Werte und ihrer mit Beob_Wahrsch aufgeführten Wahrscheinlichkeiten die Wahrscheinlichkeit dafür, dass ein neuer Beobachtungswert in ein bestimmtes Intervall fällt.

Für Beob_Wahrsch wird üblicherweise angenommen:

```
p(wert) = h(wert)/beobachtungen
```

also der Quotient aus der Häufigkeit, mit der ein Wert auftrat, und der Zahl der Beobachtungen. Die Summe aller Wahrscheinlichkeiten muss naturgemäß immer 1 bzw. 100% sein.

Das Intervall wird mit Untergrenze und Obergrenze (beide einschließlich) angegeben. Wird Obergrenze weggelassen, dann berechnet die Funktion die Wahrscheinlichkeit dafür, dass ein Beobachtungswert die Größe Untergrenze annimmt. Die Werte für beide Argumente müssen auch in dem Bereich der beobachteten Werte vorhanden sein, wenn die Auswertung sinnvoll sein soll. Werte außerhalb dieses Bereichs werden zwar akzeptiert, führen aber zu irreführenden Ergebnissen.

Nicht numerische Werte für Untergrenze oder Obergrenze führen zu dem Fehler #WERT!.

Abbildung 7.105 zeigt ein Beispiel zur Notenverteilung. In Spalte A und B sind die bisher beobachteten Werte und ihre Wahrscheinlichkeit angeordnet. In der Tabelle daneben wird die Funktion mit verschiedenen Intervallgrenzen benutzt. Die Zeile 8 bedeutet hier beispielsweise, dass 70% der Noten in dem Bereich von 2 bis 4 liegen. Die Funktion summiert die Wahrscheinlichkeiten einzelner Werte zu Intervallwahrscheinlichkeiten.

F5 =WAHRSCHBEREICH($A$5:$A$10;$B$5:$B$10;D5)

	A	B	C	D	E	F	G
1							
2	**Wahrscheinlichkeitsbereich**						
3							
4	Note	p(Wert)		UGrenze	OGrenze	p(Bereich)	
5	1	10%		1		10%	
6	2	20%		1	3	60%	
7	3	30%		2	3	50%	
8	4	20%		2	4	70%	
9	5	10%		3	6	70%	
10	6	10%					
11							

Abbildung 7.105 Berechnung von Intervallwahrscheinlichkeiten

WEIBULL.VERT()

WEIBULL.DIST()

Syntax: `WEIBULL.VERT(x; Alpha; Beta; Kumuliert)`
Beispiel: `=WEIBULL.VERT(2; 1; 0,5; WAHR)`
Ergebnis: `0,9817`

Die Funktion `WEIBULL.VERT()` liefert Wahrscheinlichkeiten für eine Zufallsvariable, die einer Weibull-Verteilung gehorcht. Diese Verteilung wird beispielsweise für Haltbarkeitsstatistiken im Bereich der Qualitätssicherung benutzt, aber auch für Berechnungen zur Häufigkeit von Windgeschwindigkeiten bei der Nutzung der Windenergie.

Benannt ist diese Verteilung nach dem schwedischen Ingenieur Waloddi Weibull, der sie im Zusammenhang mit Forschungen zur Materialermüdung definierte. Sie kann zur Beschreibung von steigenden, konstanten und fallenden Ausfallraten eingesetzt werden.

Weibull-Verteilung

Ausfallrate	0,25	1,00	3,40	0,25	1,00	3,40
charakt. Lebensdauer	70.000	70.000	70.000	70.000	70.000	70.000
	Überlebenswahrscheinlichkeit					
Zyklen	Verteilungsfunktion			Dichtefunktion		
10.000	0,4576403	0,1331221	0,0013378	0,0000084	0,0000124	0,0000005
20.000	0,5175319	0,2485227	0,0140315	0,0000044	0,0000107	0,0000024
30.000	0,5539868	0,3485609	0,0545452	0,0000030	0,0000093	0,0000060
40.000	0,5803083	0,4352819	0,1385741	0,0000023	0,0000081	0,0000109
50.000	0,6009048	0,5104583	0,2727904	0,0000019	0,0000070	0,0000158
60.000	0,6178062	0,5756272	0,4468249	0,0000015	0,0000061	0,0000186
70.000	0,6321206	0,6321206	0,6321206	0,0000013	0,0000053	0,0000179
80.000	0,6445210	0,6810934	0,7929114	0,0000012	0,0000046	0,0000139
90.000	0,6554475	0,7235470	0,9046429	0,0000010	0,0000039	0,0000085
100.000	0,6652039	0,7603490	0,9653526	0,0000009	0,0000034	0,0000040

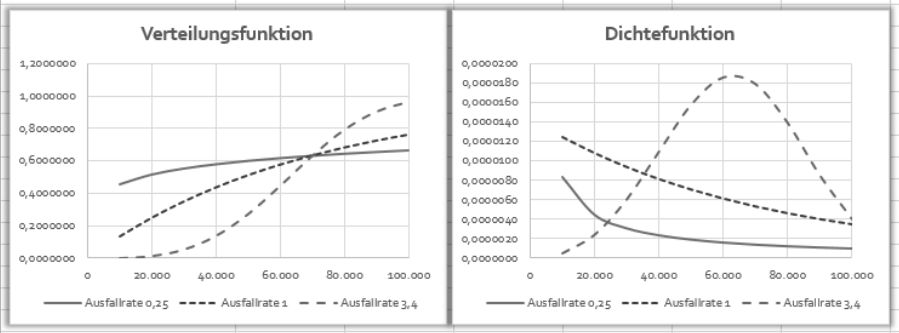


Abbildung 7.106 Weibull-Verteilung für verschiedene Parameter

Das Quantil, für das die Funktion ausgewertet werden soll, wird mit x angegeben, Alpha ist ein Form- oder Gestaltparameter der Verteilung, der die Ausfallrate bestimmt. Beta ist ein Skalenparameter, der die charakteristische Lebensdauer (oder die mittlere Festigkeit) angibt. (In Excel sind die Namen der Parameter allerdings gegenüber den meistens verwendeten Bezeichnungen – β für den Formparameter, α für den Skalenparameter – vertauscht worden.)

Mit Kumuliert lässt sich festlegen, ob die Dichtefunktion (FALSCH) oder die Verteilungsfunktion (WAHR) ausgegeben wird. Abbildung 7.106 zeigt ein fiktives Beispiel für ein Produkt, dessen Lebensdauer in Zyklen angegeben ist.

Ist x, Alpha oder Beta nicht numerisch, gibt die Funktion den Fehler #WERT! aus.
Ist x < 0 oder Alpha ≤ 0 oder Beta ≤ 0, erscheint der Fehler #ZAHL!.

Alpha = 1 zeigt eine konstante Ausfallrate an (in diesem Fall entspricht die Weibull-Verteilung der Exponentialverteilung).

Die Dichtefunktion der Weibull-Verteilung lautet:

$$= \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}}$$

wobei α für das Argument Alpha und β für das Argument Beta steht.

Den Einfluss der Parameter auf die Verteilung macht Abbildung 7.106 deutlich. Die Funktion ersetzt seit Excel 2010 die bisherige Funktion WEIBULL().

ZÄHLENWENN()

COUNTIF()

Syntax:	ZÄHLENWENN(**Bereich**; **Suchkriterien**)
Beispiel:	=ZÄHLENWENN(A10:A15;"<10")
Ergebnis:	4, wenn A10:A15 die Werte 5, 4, 9, 12, 11, 1 enthält

Die Funktion ZÄHLENWENN() gibt die Anzahl der Zellen in Bereich wieder, die Inhalte oder Bezüge auf Inhalte enthalten, die den mit dem Argument Suchkriterien angegebenen Kriterien entsprechen.

Bereich sind die auszuwertenden Daten. Das Argument kann ein Bereichsbezug, ein Bereichsname oder eine Matrixkonstante sein. Suchkriterien kann eine Zahl, ein Ausdruck oder eine Zeichenfolge sein oder ein Bezug auf eine Zelle, die solche Kriterien enthält. Ausdrücke mit logischen Operatoren wie "> M" oder "< 1000" müssen dabei in Anführungszeichen gesetzt werden. Die Groß-/Kleinschreibung der Kriterien wird nicht beachtet.

Außerdem können die Platzhalterzeichen – ? für ein beliebiges Zeichen, * für mehrere beliebige Zeichen – verwendet werden. Sollen diese Zeichen selbst gesucht werden, muss eine Tilde (~) vor sie gesetzt werden.

In Abbildung 1.107 werden in C5 die Besucherwerte, die über 120 pro Monat liegen, in C6 wird die Zahl der eingetragenen Monate nach dem Juni 2013 gesucht.

	A	B	C	D	E
1					
2	**Bedingtes Zählen mit einem Kriterium**				
3					
4	Monat	Besucher	ZÄHLENWENN()		
5	Jan 13	100	3	**=ZÄHLENWENN(B5:B16;">120")**	
6	Feb 13	120	6	**=ZÄHLENWENN(A5:A16;">Jun 13")**	
7	Mrz 13	70			
8	Apr 13	120			
9	Mai 13	110			
10	Jun 13	100			
11	Jul 13	90			
12	Aug 13	110			
13	Sep 13	120			
14	Okt 13	130			
15	Nov 13	135			
16	Dez 13	140			
17					

Abbildung 7.107 Beispiel für bedingtes Zählen

Mit dem Kriterium "" können auch die Zellen gezählt werden, die in Bereich leer sind. Auch Wahrheitswerte oder Fehlerwerte lassen sich mit entsprechenden Kriterien wie "FALSCH", "#NV" oder "#ZAHL!" in einem Zellbereich zählen. Das kann bei sehr großen Tabellen hilfreich sein, um zu prüfen, wie viele Werte noch fehlen oder fehlerhaft sind.

Bedingte Zusammenfassungen von Werten stellt Excel auch über andere Funktionen zur Verfügung: SUMMEWENN(), SUMMEWENNS(), ZÄHLENWENNS(), MITTELWERTWENN(), und MITTELWERTWENNS().

ZÄHLENWENNS()

COUNTIFS()

Syntax:	ZÄHLENWENNS(**Kriterienbereich1**; **Kriterien1**; Kriterienbereich2; Kriterien2 ...)
Beispiel:	=ZÄHLENWENNS(A10:A15,">4";A10:A15;"<10")
Ergebnis:	3, wenn A10:A15 die Werte 5, 4, 9, 12, 11, 1 enthält

Die Funktion ZÄHLENWENNS() erweitert die Möglichkeiten der Funktion ZÄHLENWENN() um zusätzliche Auswahlkriterien. Sie steht erst seit Excel 2007 zur Verfügung. Die Funktion gibt die Anzahl der Zellen in bis zu 127 Bereichen wieder, die Inhalte enthalten oder Bezüge auf Inhalte, die den angegebenen Suchkriterien entsprechen. Die Argumente für den Bereich und das darauf anzuwendende Kriterium werden jeweils paarweise angegeben. Sollen für einen Bereich mehrere Kriterien verwendet werden, muss der Bereich entsprechend oft angegeben werden.

	A	B	C	D
1				
2	**Bedingtes Zählen mit mehreren Kriterien**			
3				
4	Name	Lieblingsfarbe	Lieblingssport	Lieblingsmusik
5	Hans	rot	Fußball	Rock
6	Werner	blau	Tennis	Blues
7	Lotte	rot	Tennis	Pop
8	Lisa	gelb	Joggen	Pop
9	Franz	blau	Joggen	Pop
10	Rita	rot	Turnen	Pop
11	Henning	rot	Fußball	Rock
12	Wanda	blau	Tennis	Blues
13	Lori	rot	Tennis	Pop
14	Bella	rot	Tennis	Pop
15	Arne	blau	Joggen	Pop
16	Lukas	rot	Fußball	Blues
17	3			
18	**=ZÄHLENWENNS(B5:B16;"rot";C5:C16;"Tennis";D5:D16;"Pop")**			
19	4			
20	**=ZÄHLENWENNS(B5:B16;"rot";C5:C16;"T*";D5:D16;"Pop")**			

Abbildung 7.108 Bedingtes Zählen mit mehreren Kriterien

Kriterium kann eine Zahl, ein Ausdruck oder eine Zeichenfolge sein. Ausdrücke mit logischen Operatoren müssen in Anführungszeichen gesetzt werden. Außerdem können die Platzhalterzeichen – ? für ein beliebiges Zeichen, * für mehrere beliebige Zeichen –

verwendet werden. Sollen diese Zeichen selbst gesucht werden, muss eine Tilde (~) vor sie gesetzt werden. Die Groß-/Kleinschreibung der Kriterien wird nicht beachtet.

Das Beispiel in Abbildung 7.108 zeigt, wie mit der Funktion gearbeitet werden kann. Die erste Formel in A17 sucht in Spalte B die Einträge mit der Lieblingsfarbe »rot«, bei denen gleichzeitig in der Spalte für den Lieblingssport »Tennis« und in der Spalte zur Lieblingsmusik »Pop« eingetragen ist. Die Formel in A19 verwendet für die Sportart einen Platzhalter, so dass sowohl »Tennis« als auch »Turnen« gefunden werden. Die Ergebnisse zeigen, dass die drei ausgewerteten Bereiche in diesem Fall zeilenweise verglichen werden. Die Kriterien sind additiv zu verstehen, eine Zählung findet also nur statt, wenn alle Kriterien erfüllt sind.

Bedingte Zusammenfassungen von Werten stellt Excel auch über andere Funktionen zur Verfügung: `SUMMEWENN()`, `SUMMEWENNS()`, `ZÄHLENWENN()`, `MITTELWERTWENN()` sowie `MITTELWERTWENNS()`.

8 Kompatible statistische Funktionen

Funktion	Seite	Funktion	Seite
BETAINV()	618	NEGBINOMVERT()	640
BETAVERT()	619	NORMINV()	641
BINOMVERT()	620	NORMVERT()	642
CHIINV()	621	OBERGRENZE()	644
CHITEST()	622	POISSON()	645
CHIVERT()	624	QUANTIL()	646
EXPONVERT()	625	QUANTILSRANG()	647
FINV()	626	QUARTILE()	649
FTEST()	627	RANG()	650
FVERT()	628	STABW()	651
GAMMAINV()	629	STABWN()	652
GAMMAVERT()	629	STANDNORMINV()	652
GTEST()	631	STANDNORMVERT()	653
HYPGEOMVERT()	632	TINV()	654
KONFIDENZ()	633	TTEST()	656
KOVAR()	635	TVERT()	657
KRITBINOM()	636	UNTERGRENZE()	658
LOGINV()	637	VARIANZ()	659
LOGNORMVERT()	638	VARIANZEN()	660
MODALWERT()	639	WEIBULL()	661

In diesem Abschnitt finden Sie eine kompakte Referenz der Funktionen, die als kompatible Funktionen im Bereich der Statistik weiter gültig bleiben. Bei jeder Funktion ist vermerkt, welche aktuellen Funktionen dafür seit Excel 2010 zur Verfügung stehen. Bei diesen Funktionen finden Sie häufig ausführlichere Beschreibungen der in der Regel unveränderten Argumente. Nur bei den neuen Verteilungsfunktionen ist jetzt immer das Argument `Kumuliert` eingefügt, das den Typ der Funktion bestimmt.

In Tabelle 8.1 sind die umbenannten statistischen Funktionen den alten Namen gegenübergestellt, die jetzt in der Kategorie **Kompatibilität** geparkt sind.

Umbenannte Funktion	Kompatible Funktion
BETA.INV()	BETAINV()
BETA.VERT()	BETAVERT()
BINOM.INV()	KRITBINOM()
BINOM.VERT()	BINOMVERT()
CHIQU.INV()	CHIINV()
CHIQU.INV.RE()	CHIINV()
CHIQU.TEST()	CHITEST()
CHIQU.VERT()	CHIVERT()
CHIQU.VERT.RE()	CHIVERT()
EXPON.VERT()	EXPONVERT()
F.INV()	FINV()
F.INV.RE()	FINV()
F.TEST()	FTEST()
F.VERT()	FVERT()
F.VERT.RE()	FVERT()
G.TEST()	GTEST()
GAMMA.INV()	GAMMAINV()
GAMMA.VERT()	GAMMAVERT()
HYPGEOM.VERT()	HYPGEOMVERT()
KONFIDENZ.NORM()	KONFIDENZ()
KONFIDENZ.T()	KONFIDENZ()
KOVARIANZ.P()	KOVAR()
KOVARIANZ.S()	KOVAR()
LOGNORM.INV()	LOGINV()
LOGNORM.VERT()	LOGNORMVERT()
MODUS.EINF()	MODALWERT()
MODUS.VIELF()	MODALWERT()

Tabelle 8.1 Gegenüberstellung der aktuellen und der veralteten Funktionen

Umbenannte Funktion	Kompatible Funktion
NEGBINOM.VERT()	NEGBINOMVERT()
NORM.INV()	NORMINV()
NORM.S.INV()	STANDNORMINV()
NORM.S.VERT()	STANDNORMVERT()
NORM.VERT()	NORMVERT()
POISSON.VERT()	POISSON()
QUANTIL.EXKL()	QUANTIL()
QUANTIL.INKL()	QUANTIL()
QUANTILSRANG.EXKL()	QUANTILSRANG()
QUANTILSRANG.INKL()	QUANTILSRANG()
QUARTILE.EXKL()	QUARTILE()
QUARTILE.INKL()	QUARTILE()
RANG.GLEICH()	RANG()
STABW.N()	STABWN()
STABW.S()	STABW()
T.INV()	TINV()
T.INV.2S()	TINV()
T.TEST()	TTEST()
T.VERT()	TVERT()
T.VERT.2S()	TVERT()
T.VERT.RE()	TVERT()
VAR.P()	VARIANZEN()
VAR.S()	VARIANZ()
WEIBULL.VERT()	WEIBULL()

Tabelle 8.1 Gegenüberstellung der aktuellen und der veralteten Funktionen (Forts.)

Mit der Version 2013 sind noch die beiden Funktionen OBERGRENZE() und UNTERGRENZE() in die Kategorie der Kompatiblen verschoben worden, die vorher immer zu den mathematischen Funktionen gerechnet worden sind.

Damit es keine Probleme mit älteren Arbeitsmappen gibt, bleiben die umbenannten Funktionen weiterhin unter ihrem alten Namen verfügbar. Im Dialog **Funktion einfügen** sind diese Funktionen in der Kategorie **Kompatibilität** zu finden. Wenn Sie Funktionen direkt einfügen und die Option **AutoVervollständigen-Formel** über **Datei • Optionen • Formeln** eingeschaltet lassen, werden die kompatiblen Funktionen jeweils mit einem Symbol angezeigt, das ein kleines Warnzeichen enthält.

8.1 Referenz der kompatiblen statistischen Funktionen

BETAINV()

BETAINV()

Syntax: `BETAINV(Wahrsch; Alpha; Beta; A; B)`

Beispiel: `=BETAINV(0,1;3;4)`

Ergebnis: `0,2009`

Die Funktion `BETAINV()` liefert Quantile einer Betaverteilung und ist die Umkehrung von `BETAVERT()`.

Als notwendige Argumente sind mit `Wahrsch` die Wahrscheinlichkeit und die beiden Parameter `Alpha` und `Beta` einzutragen. `A` und `B` sind optionale Argumente, die die Intervallgrenzen bezeichnen. Werden sie nicht angegeben, dann wird `A = 0` und `B = 1` gesetzt (vergleiche `BETAVERT()`).

	A	B	C
1			
2	**Quantile der Betaverteilung**		
3			
4	Alpha	Beta	
5	3	4	
7	Wahrsch	BETAVERT()	BETAINV()
8	0	0	
9	0,1	0,01585	0,201
10	0,2	0,09888	0,269
11	0,3	0,25569	0,323
12	0,4	0,45568	0,373
13	0,5	0,65625	0,421
14	0,6	0,8208	0,471
15	0,7	0,92953	0,524
16	0,8	0,98304	0,585
17	0,9	0,99873	0,667
18	1	1	

Abbildung 8.1 Quantile der Betaverteilung

Für den Zusammenhang zwischen BETAINV() und BETAVERT() gilt:

Wenn:

```
Wert = BETAVERT(x;...)
```

dann ist:

```
x = BETAINV(Wert;...)
```

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Alpha ≤ 0 oder Beta ≤ 0, erscheint der Fehler #ZAHL!. Das gilt auch, wenn Wahrsch ≤ 0 oder > 1.

Für BETAINV() wird seit Excel 2010 die umbenannte Funktion BETA.INV() angeboten.

BETAVERT()

BETADIST()

Syntax:	BETAVERT(**x**; **Alpha**; **Beta**; A; B)
Beispiel:	=BETAVERT(0,5;3;4)
Ergebnis:	0,65625

Die Funktion BETAVERT() liefert die Wahrscheinlichkeitsverteilung für eine betaverteilte Zufallsvariable. Die Betaverteilung ist eine stetige Wahrscheinlichkeitsverteilung über dem Intervall [0,1] oder der Intervall [A,B]. Sie steht in engem Zusammenhang mit der Gammaverteilung und kann bei Berechnung der Verteilung von Größen aus beliebigen gleichmäßig stetig verteilten Grundgesamtheiten verwendet werden. Es wird berechnet, mit welcher Wahrscheinlichkeit die Zufallsvariable einen Wert zwischen A und x annimmt.

Das Argument x ist die Größe der Zufallsvariablen im Intervall A bis B; Alpha und Beta – beide müssen größer als 0 sein – sind Parameter der Verteilung. (In der Literatur werden normalerweise die Bezeichnungen p und q verwendet.)

A und B sind optionale Argumente und bezeichnen die untere und obere Grenze des Intervalls. Werden für A und B keine Werte angegeben, dann gilt die standardmäßige Betaverteilung (A = 0 und B = 1).

	A	B	C	D	E
1					
2	**Betaverteilung für verschiedene Parameterwerte**				
3					
4	Alpha -->	1	2	3	
5	Beta -->	1	1	2	
6					
7	X			Betaverteilung	
8	0	0	0	0	
9	0,1	0,1	0,01000	0,00370	
10	0,2	0,2	0,04000	0,02720	
11	0,3	0,3	0,09000	0,08370	
12	0,4	0,4	0,16000	0,17920	
13	0,5	0,5	0,25000	0,31250	
14	0,6	0,6	0,36000	0,47520	
15	0,7	0,7	0,49000	0,65170	
16	0,8	0,8	0,64000	0,81920	
17	0,9	0,9	0,81000	0,94770	
18	1	1	1	1	
19					

Abbildung 8.2 Beispiele für `BETAVERT()`

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Alpha` ≤ `0` oder `Beta` ≤ `0`, erscheint der Fehler `#ZAHL!`. Das gilt auch, wenn `x` < `A` oder `x` > `B` oder `A` = `B`.

BINOMVERT()

BINOMDIST()

Syntax: `BINOMVERT(`**`Zahl_Erfolge`**`;` **`Versuche`**`;` **`Erfolgswahrsch`**`;` **`Kumuliert`**`)`

Beispiel: `=BINOMVERT(3;10;1/6;FALSCH)`

Ergebnis: `15,5%`

Die Funktion `BINOMVERT()` liefert die Wahrscheinlichkeit von Zufallsvariablen bei einer Binomialverteilung. Sie gibt also die Wahrscheinlichkeit dafür an, dass bei voneinander unabhängigen diskreten Versuchsergebnissen bei einer mit `Versuche` angegebenen Anzahl von Versuchen ein bestimmtes Ergebnis mit einer durch `Zahl_Erfolge` angegebenen Häufigkeit auftritt.

Die (vorweg ermittelte) Wahrscheinlichkeit für das Einzelergebnis wird mit `Erfolgswahrsch` (zwischen `0` und `1`) angegeben. Es wird also vorausgesetzt, dass sie bekannt ist.

Beispiele sind etwa Münzwürfe (Erfolgswahrscheinlichkeit 1/2), Würfel (1/6) oder Kartenziehen (1/32), wobei aber nach jedem Versuch die Karte anschließend zurückgesteckt werden muss; es wird also jedes Mal der Ausgangszustand wieder hergestellt.

`Kumuliert` verlangt einen Wahrheitswert und beschreibt den Typ der Funktion. Wird das Argument mit `FALSCH` belegt, wird der Wert der Wahrscheinlichkeitsfunktion geliefert.

`Zahl_Erfolge` muss ≥ `0` und ≤ `Versuche` sein, ansonsten gibt die Funktion den Fehlerwert `#ZAHL!` zurück. Das gilt auch für `Erfolgswahrsch` < `0` oder > `1`. Ist eines der Argumente `Zahl_Erfolge`, `Versuche`, `Erfolgswahrsch` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück.

Das oben angeführte Beispiel liefert die Wahrscheinlichkeit dafür, dass bei zehn Würfen mit einem Würfel genau dreimal die Sechs gewürfelt wird. Wird das Argument mit `WAHR` belegt, wird die Verteilungsfunktion berechnet, im Beispiel die Wahrscheinlichkeit, dass die Sechs bis zu dreimal gewürfelt wird. Seit Excel 2010 steht anstelle dieser Funktion die umbenannte Funktion `BINOM.VERT()` zur Verfügung.

	A	B	C	D	E
1					
2	**Wahrscheinlichkeit einer binomialverteilten Zufallsvariablen**				
3					
4	Zahl_Erfolge	Versuche	Erfolgswahrscheinlichkeit	Kumuliert	BINOMVERT()
5	40	100	1/2	WAHR	2,8%
6	40	100	1/2	FALSCH	1,1%
7	20	100	1/50	WAHR	100,0%
8	3	10	1/6	FALSCH	15,5%

Abbildung 8.3 Berechnungen mit `BINOMVERT()`

CHIINV()

CHIINV()

Syntax: `CHIINV(`**`Wahrsch`**`; `**`Freiheitsgrade`**`)`

Beispiel: `=CHIINV(0,05;3)`

Ergebnis: `7,8147`

Die Funktion `CHIINV()` liefert Quantile der Chi-Quadrat-Verteilung. Sie wird verwendet, um einen Vergleichswert zu berechnen, mit dem Hypothesen über die Übereinstimmung von beobachteten und erwarteten Ergebnissen bewertet werden können.

Für das Argument Wahrsch erwartet die Funktion Wahrscheinlichkeitswerte aus einer Chi-Quadrat-Verteilung und dazu die Anzahl der Freiheitsgrade.

Ist Wahrsch < 0 oder > 1 oder Freiheitsgrade < 1 oder > 10^10, liefert die Funktion den Fehlerwert #ZAHL!, ist eines der Argumente nicht numerisch, den Fehler #WERT!.

Die Funktion CHIINV() ist zugleich die Umkehrfunktion von CHIVERT(). Es gilt also: Ist:

```
w = CHIINV(x;...)
```

dann ist:

```
CHIVERT(w;...) = x
```

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen CHIQU.INV() und CHIQU.INV.RE() zur Verfügung, wobei CHIQU.INV.RE() dasselbe Ergebnis liefert wie CHIINV().

CHITEST()

CHITEST()

Syntax:	CHITEST(**Beob_Messwerte**; **Erwart_Werte**)
Beispiel:	=CHITEST({9;11;9;12;10;9};{10;10;10;10;10;10})
Ergebnis:	0,977

Die Funktion CHITEST() liefert direkt den Wahrscheinlichkeitswert für den Chi-Quadrat-Test beim Vergleich zwischen beobachteten und erwarteten Größen. Als Argumente werden je ein Bereichsbezug oder eine Matrix für die beobachteten Werte Beob_Messwerte und die theoretisch erwarteten Werte Erwart_Werte eingetragen.

Beide Argumente müssen die gleiche Anzahl von Datenelementen enthalten, sonst liefert die Funktion den Fehler #NV. Sind die Argumente nicht numerisch, erscheint der Fehler #WERT!.

Abbildung 8.4 zeigt ein einfaches Beispiel für ein solches Testverfahren. Es soll geprüft werden, wie sehr sich bei 60-maligem Würfeln die beobachteten Ergebnisse, die in Spalte B abgelegt sind, an die Ergebnisse anpassen, die aufgrund der theoretischen Wahrscheinlichkeit zu erwarten sind. Deshalb wird auch von Anpassungstests gesprochen. Die theoretische Wahrscheinlichkeit ergibt sich aus der Formel 60 * 1/6, sie setzt also eine sogenannte Gleichverteilung voraus. Deshalb ist in Spalte C für alle Wurfergebnisse der Wert 10 abgelegt.

	A	B	C	D	E	F
1						
2	**Chi-Test Beispiel: Würfeleichung: 60 mal gewürfelt**					
3						
4	Wurf	Beob	Erw	Beob-Erw	(beob-erw)^2/erw	
5	1	8	10	-2	0,4	
6	2	12	10	2	0,4	
7	3	12	10	2	0,4	
8	4	7	10	-3	0,9	
9	5	12	10	2	0,4	
10	6	9	10	-1	0,1	
11	Summe	60	60	0	2,6	**<-- u Summe der relativierten quadrierten Abweichungen**
12						
13				CHITEST(B5:B10;C5:C10)	0,761	
14				CHIINV(0,05;5)	11,070	**<-- kritischer Wert**
15				CHIVERT(E11;5)	0,761	

Abbildung 8.4 Beispiel für `CHITEST()`

Die Nullhypothese, die durch den Chi-Quadrat-Test geprüft werden soll, lässt sich so formulieren: Die Differenzen zwischen den beobachteten und den theoretischen Häufigkeiten sind rein zufällig und nicht signifikant, die empirische Häufigkeitsverteilung passt sich in einem ausreichenden Maße der theoretischen Häufigkeitsverteilung an, es gibt also kein Indiz dafür, dass der Würfel beispielsweise gezinkt oder defekt ist.

Die Funktion `CHITEST()` rechnet nach folgendem Verfahren: Zunächst wird für alle Variablen die Differenz zwischen dem beobachteten und dem erwarteten Ergebnis gebildet und diese dann quadriert, so dass die negativen Vorzeichen keine Rolle mehr spielen. Das Ergebnis wird jedes Mal durch den erwarteten Wert geteilt, um die Abweichungen zu relativieren. Aus diesen Einzelergebnissen wird die Summe ermittelt, um den Wert der chi-quadrierten Verteilung zu erhalten, der auch als *Chi-Quadrat* oder mit dem Buchstaben `u` bezeichnet wird. Dieser Wert wird als Prüfgröße verwendet.

Dieser Wert würde bei einer perfekten Übereinstimmung zwischen dem erwarteten und dem beobachteten Ergebnis `0` sein. Je größer der Wert ist, umso fragwürdiger ist die Übereinstimmung. Im letzten Schritt ermittelt die Funktion nun die Wahrscheinlichkeit dafür, dass `u` den errechneten Wert annimmt. In diesem Fall ergibt sich der Wert 0,76 oder 76%. Dieser Wert liegt deutlich über dem vorgegebenen Signifikanzniveau. Mit der Funktion `CHIINV()` kann zum Vergleich mit dem Prüfwert ein kritischer Wert aus der Chi-Quadrat-Verteilung berechnet werden.

```
=CHIINV(0,05;5)
```

ergibt also mit einer Irrtumswahrscheinlichkeit von 5% und mit 5 Freiheitsgraden den Wert 11,070, der wesentlich höher als der Prüfwert ist. Der Prüfwert liegt also nicht im kritischen Bereich. Die Nullhypothese muss also nicht verworfen werden, die Abweichungen von den theoretisch erwarteten Ergebnissen können als rein zufällig eingestuft werden.

Seit Excel 2010 wird für CHITEST() die umbenannte Funktion CHIQU.TEST() angeboten.

CHIVERT()

CHIDIST()

Syntax:	CHIVERT(**x**; **Freiheitsgrade**)
Beispiel:	=CHIVERT(10;3)
Ergebnis:	0,018566

Die Funktion CHIVERT() berechnet aus dem Wert für x und für Freiheitsgrade die Wahrscheinlichkeit für die Übereinstimmung von beobachteten und erwarteten Werten, vergleiche CHITEST().

Ist x < 0, gibt die Funktion den Fehler #ZAHL! zurück. Das gilt auch, wenn Freiheitsgrade < 1 oder > 10^10 ist. Ist eines der beiden Argumente nicht numerisch, erscheint der Fehler #WERT!.

Der Wert x wird ermittelt als die Summe aus

```
(Beobachtungswert – Erwartungswert)^2 / Erwartungswert
```

für alle Werte. Die Variable wird auch als *Chi-Quadrat*, *c2* oder mit dem Buchstaben *u* bezeichnet.

Die Chi-Quadrat-Verteilung ist eine Wahrscheinlichkeitsverteilung, die sich über die Summe von n unabhängigen, quadrierten, standardnormalverteilten Variablen und einer Anzahl von Freiheitsgraden definiert. Die Funktion wird für den Chi-Quadrat-Test benötigt, der beim Vergleich von empirischen zu theoretisch erwarteten Häufigkeiten zum Einsatz kommt.

Je nach Anzahl der Freiheitsgrade ändert sich der Charakter der Verteilung. Mit steigender Anzahl wird die Funktion flacher und verschiebt sich nach rechts. Die Freiheitsgrade entsprechen der Anzahl der Möglichkeiten –1. Bei kontinuierlichen Größen wird

gerechnet mit der Anzahl der Klassen -1 bei einer Datenspalte oder Zeile; bei zweidimensionalen Wertetabellen gilt:

```
(Zeilenanzahl - 1) * (Spaltenanzahl - 1)
```

Dass immer ein Freiheitsgrad »verloren« geht, lässt sich an dem Beispiel mit den 60 Würfeltests aus Abbildung 8.4 zu CHITEST() leicht verstehen. Wenn nämlich für 5 mögliche Augenergebnisse die zufälligen Häufigkeiten feststehen, ist die Häufigkeit für das sechste mögliche Ergebnis nicht mehr zufällig, sondern vorgegeben als die Differenz der Summe der 5 Häufigkeiten zur Zahl der Würfe insgesamt.

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen CHIQU.VERT() und CHIQU.VERT.RE() zur Verfügung, wobei CHIQU.VERT.RE() dasselbe Ergebnis liefert wie CHIVERT().

EXPONVERT()

EXPONDIST()

Syntax: EXPONVERT(**x**; **Lambda**; **Kumuliert**)
Beispiel: =EXPONVERT(0,5;3;WAHR)
Ergebnis: 0,777

Die Funktion EXPONVERT() liefert Wahrscheinlichkeiten für eine exponentialverteilte Zufallsvariable. Eine Exponentialverteilung ist eine stetige Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen. Mit x wird das Quantil angegeben, für das der Wert ermittelt werden soll. Lambda ist ein Parameter, der bei der Dichtefunktion den Anfangswert bei $x = 0$ sowie den Grad des Abfalls bestimmt. Er wird auch als Ausfallrate interpretiert.

Kumuliert ist ein Wahrheitswert, mit dem der Typ der Funktion bestimmt wird. Ist Kumuliert mit WAHR belegt, wird der Wert der Verteilungsfunktion geliefert (die Fläche bis zum Quantil); mit FALSCH belegt, ergibt sich der Wert für die Dichtefunktion (der Wert auf der y-Achse).

Ist x oder Lambda nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist $x < 0$ oder Lambda ≤ 0, erscheint der Fehler #ZAHL!.

Normalerweise wird die Verteilungsfunktion benötigt, deren Wert aussagt, wie groß die Wahrscheinlichkeit ist, dass die Zufallsvariable einen Wert zwischen 0 und x annimmt. Abbildung 8.5 zeigt ein Beispiel für die Berechnung einer Ausfallrate.

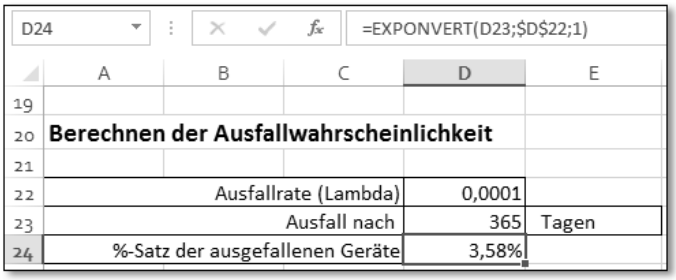


Abbildung 8.5 Berechnen der Ausfallrate

Seit Excel 2010 wird für `EXPONVERT()` die umbenannte Funktion `EXPON.VERT()` angeboten.

FINV()

FINV()

Syntax: `FINV(`**Wahrsch**`;` **Freiheitsgrade1**`;` **Freiheitsgrade2**`)`

Beispiel: `=FINV(0,05;7;7)`

Ergebnis: `3,787`

Die Funktion `FINV()` liefert Quantile der F-Verteilung (d.h. die Werte, die in statistischen Tabellenwerken tabelliert sind). Sie ist die Umkehrung von `FVERT()` (siehe dort). Die Funktion geht von einer zweiseitigen Verteilung aus.

Mit `Wahrsch` wird die Wahrscheinlichkeit angegeben. Als Werte für die Argumente `Freiheitsgrade1` und `Freiheitsgrade2` werden die Größen der beiden miteinander verglichenen Stichproben minus 1 angegeben. Sie lassen sich beispielsweise mit der Funktion `ANZAHL()` ermitteln.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` < `0` oder `Wahrsch` > `1` oder `Freiheitsgrade1` oder `Freiheitsgrade2` < `1`, erscheint der Fehler `#ZAHL!`.

Bei einem gegebenen Wert für `Wahrsch` sucht die Funktion durch ein Iterationsverfahren mit maximal 100 Schritten einen Wert `x`, so dass die Gleichung gilt:

```
Wahrsch = FVERT(x;..;..)
```

Siehe auch Abbildung 8.6 zu FTEST(). Seit Excel 2010 werden für FINV() die umbenannten Funktionen F.INV() und F.INV.RE() angeboten, wobei F.INV.RE() dasselbe Ergebnis liefert wie FINV().

FTEST()

FTEST()

Syntax: FTEST(**Matrix1**; **Matrix2**)

Beispiel: =FTEST({12;19;13;14;17};{15;17;16;15;17})

Ergebnis: 0,0618

Die Funktion FTEST() liefert unmittelbar die Wahrscheinlichkeit der Übereinstimmung zweier Stichproben hinsichtlich ihrer Varianzen. Mit dem F-Test lässt sich also ermitteln, ob sich zwei Stichproben in ihren Varianzen nur zufällig unterscheiden. Matrix1 und Matrix2 sind die Einzelwerte zweier Stichproben. Die Argumente müssen nicht denselben Umfang haben.

Enthalten die mit Matrix1 und Matrix2 angegebenen Datenreihen nicht numerische Elemente, werden sie ignoriert. Sind aber jeweils weniger als zwei numerische Elemente vorhanden, gibt die Funktion den Fehler #DIV/0! zurück.

	A	B	C	D	E
1					
2	**F-Test und F-Verteilung**				
3					
4	x-Werte	y-Werte		=FINV(0,05;7;7)	3,78704354
5	4	3		=FTEST(A5:A12;B5:B12)	0,730185877
6	2	2		=FVERT(E4;7;7)	0,05
7	3	5		=FVERT(E14;7;7)*2	0,730185877
8	5	6			
9	3	5			
10	6	5			
11	3	2			
12	5	3			
13	Varianz1	Varianz2			F-Wert
14	1,839285714	2,410714286		Varianz2/Varianz1	1,310679612

Abbildung 8.6 Beispiel für den F-Test

Im Beispiel in Abbildung 8.6 wird getestet, ob die beiden Stichproben aus derselben Grundgesamtheit stammen können. In A14 und B14 sind die Varianzen der einzelnen Stichproben mit der Funktion VARIANZ() berechnet.

Wenn Sie von diesen beiden Varianzwerten den Quotienten bilden, wobei der größere Wert in den Zähler gesetzt wird, ergibt sich ein F-Wert, der die Ausprägung einer Zufallsvariablen ist, die zu einer F-Verteilung gehört. Der in diesem Fall berechnete F-Wert 1,31 ist niedriger als der in E4 mit der Funktion FINV() berechnete kritische F-Wert mit einem Signifikanzniveau von 5% und den Freiheitsgraden der beiden Stichproben – jeweils Anzahl Werte -1. Die Nullhypothese muss also nicht verworfen werden.

Sie können nun für den F-Wert 1,31 die Überschreitungswahrscheinlichkeit berechnen, indem Sie für diesen Wert die Funktion FVERT() mit den beiden Werten für Freiheitsgrade berechnen und das Ergebnis mit 2 multiplizieren. Das ergibt hier den Wert 0,73. Dieser Wert ist größer als das Signifikanzniveau von 0,05, was noch einmal die Nullhypothese bestätigt. Denselben Wert erhalten Sie mit der Funktion FTEST() auch direkt, wenn Sie als Argumente die beiden Wertebereiche angeben, wie in Zelle E5 zu sehen ist.

Seit Excel 2010 wird für FTEST() die umbenannte Funktion F.TEST() angeboten.

FVERT()

FDIST()

Syntax:	FVERT(**x**; **Freiheitsgrade1**; **Freiheitsgrade2**)
Beispiel:	=FVERT(3,787;7;7)
Ergebnis:	0,05

Die Funktion FVERT() liefert Werte der Verteilungsfunktion (1-Alpha) für F-verteilte Zufallsvariable. Das Ergebnis gibt die Wahrscheinlichkeit, also das Signifikanzniveau, an. Die wichtigste Anwendung der F-Verteilung liegt in Signifikanztests für zwei unabhängige Stichproben.

Je nach der Anzahl der Freiheitsgrade1 (Größe der ersten Stichprobe -1) und der Anzahl der Freiheitsgrade2 (Größe der zweiten Stichprobe -1) unterscheiden sich die F-Verteilungen und nehmen verschiedene Gestalt an. Mit x wird das Quantil der Verteilung eingegeben.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist x < 0 oder Freiheitsgrade1 oder Freiheitsgrade2 < 1, erscheint der Fehler #ZAHL!.

Für den Zusammenhang zwischen FVERT() und FINV() gilt: Wenn:

x = FINV(p;...)

dann ist:

```
p = FVERT(x;...)
```

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `F.VERT()` und `F.VERT.RE()` zur Verfügung, wobei `FVERT()` dasselbe Ergebnis liefert wie `F.VERT.RE()`.

GAMMAINV()

GAMMAINV()

Syntax:	`GAMMAINV(`**`Wahrsch`**`;` **`Alpha`**`;` **`Beta`**`)`
Beispiel:	`=GAMMAINV(0,05;3;1)`
Ergebnis:	`0,8176`

Die Funktion `GAMMAINV()` liefert Quantile der Gammaverteilung. Mit dem Argument `Wahrsch` wird ein Wahrscheinlichkeitswert aus einer Gammaverteilung angegeben. `Alpha` und `Beta` sind Funktionsparameter (in der Literatur werden als Parameter meist `b` und `p` angegeben). `Beta` = `1` liefert die standardisierte Gammaverteilung.

Ist `Wahrsch`, `Alpha` oder `Beta` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` < `0` oder > `1` oder `Alpha` oder `Beta` ≤ `0`, erscheint der Fehler `#ZAHL!`.

Die Funktion ist die Umkehrfunktion von `GAMMAVERT()`. Wenn:

```
Wert = GAMMAINV(x;...)
```

dann ist:

```
x = GAMMAVERT(Wert;...WAHR)
```

Siehe auch Abbildung 8.7 zu `GAMMAVERT()`.

GAMMAVERT()

GAMMADIST()

Syntax:	`GAMMAVERT(`**`x`**`;` **`Alpha`**`;` **`Beta`**`;` **`Kumuliert`**`)`
Beispiel:	`=GAMMAVERT(1,5;2;1;WAHR)`
Ergebnis:	`0,44217`

Die Funktion `GAMMAVERT()` liefert Wahrscheinlichkeiten für eine gammaverteilte Zufallsvariable. Bei der Verteilungsfunktion ist dies die Wahrscheinlichkeit, dass eine Zufalls-

größe einen Wert zwischen 0 und x annimmt, bei der Dichtefunktion die Wahrscheinlichkeit für den Wert x. Die Gammaverteilung ist eine stetige Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen. Sie gilt als sehr anpassungsfähig, da sie auch die Untersuchung von schiefen Verteilungen erlaubt. Sie findet vor allem in der Warteschlangen- (oder Bedienungs-) und Zuverlässigkeitstheorie Anwendung.

Gammaverteilung mit verschiedenen Parametern

	Dichte		Verteilung			
Alpha (p)	2	5	2	5		
Beta (b)	1	1	1	1		
x	GAMMAVERT()				GAMMAINV()	GAMMALN()
0	0,000000	0,000000	0,000000	0,000000	0	
0,1	0,090484	0,000004	0,004679	0,000000	0,1	16,38
0,2	0,163746	0,000055	0,017523	0,000002	0,2	13,00
0,3	0,222245	0,000250	0,036936	0,000016	0,3	11,06
0,4	0,268128	0,000715	0,061552	0,000061	0,4	9,70
0,5	0,303265	0,001580				
0,6	0,329287	0,002964				
0,7	0,347610	0,004968				
0,8	0,359463	0,007669				
0,9	0,365913	0,011115				
1	0,367879	0,015328				
1,1	0,366158	0,020307				
1,2	0,361433	0,026023				
1,3	0,354291	0,032432				
1,4	0,345236	0,039472				
1,5	0,334695	0,047067				
1,6	0,323034	0,055131				
1,7	0,310562	0,063575				
1,8	0,297538	0,072302				
1,9	0,284180	0,081216	0,566251	0,044081	1,9	3,10
2	0,270671	0,090224	0,593994	0,052653	2	2,92

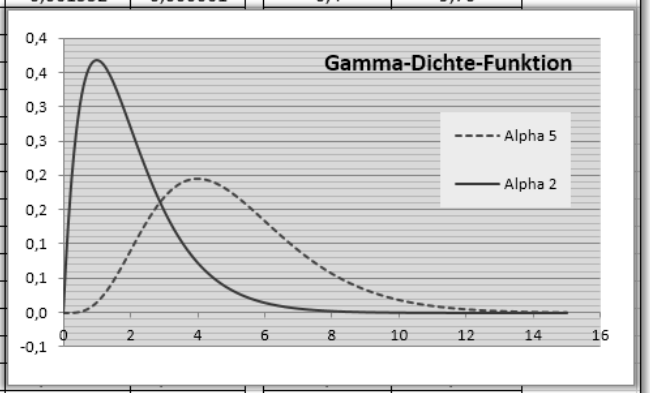


Abbildung 8.7 Gammaverteilungen und der Graph der Dichtefunktion

Von den Argumenten bezeichnet x das Quantil, für das die Wahrscheinlichkeit (1–Alpha) berechnet werden soll, Alpha und Beta sind Parameter der Verteilung (vergleiche GAMMAINV()). Das Argument Kumuliert bestimmt den Typ der Verteilung: Mit WAHR wird der Wert der Verteilungsfunktion berechnet, mit FALSCH der Wert der Dichtefunktion. Wird Beta = 1 gesetzt, ergibt dies die Werte für die standardisierte Gammaverteilung.

Ist x, Alpha oder Beta nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist x < 0 oder Alpha oder Beta ≤ 0, erscheint der Fehler #ZAHL!.

GTEST()

ZTEST()

Syntax: `GTEST(`**`Matrix`**`; `**`x`**`; Sigma)`
Beispiel: `=GTEST({11.19.18.21.13.17.9.14};12;4)`
Ergebnis: `0,01078`

Die Funktion `GTEST()` liefert die einseitige Wahrscheinlichkeit für einen Gaußtest bei normalverteilten Daten. Für einen Erwartungswert einer Zufallsvariablen, der als `x` bezeichnet wird, gibt `GTEST()` die Wahrscheinlichkeit zurück, mit der der Stichprobenmittelwert größer ist als der Durchschnitt der in `Matrix` angegebenen Werte. Mit diesem Test kann die Wahrscheinlichkeit dafür geschätzt werden, dass ein bestimmter Wert aus derselben (normalverteilten) Grundgesamtheit stammt wie eine angegebene Stichprobe.

Mit `Matrix` wird der Datenbereich der Stichprobe angegeben, mit der der Wert `x` als angenommener Erwartungswert einer Zufallsvariablen verglichen werden soll. Das optionale Argument `Sigma` bezeichnet die bekannte Standardabweichung der Grundgesamtheit. Wird `Sigma` nicht angegeben, dann verwendet die Funktion hilfsweise die Standardabweichung der Stichprobe als Schätzwert für `Sigma`.

	A	B	C	D
1				
2	**Gauß-Test**			
3				
4	**Stichprobe**		**Einzelwert:**	178
5	182	171	**=GTEST(A5:B20;D4)**	0,9799
6	170	178	**Mittelwert:**	176
7	165	186	**Standardabweichung**	5
8	174	169		
9	171	173		
10	178	180		
11	183	180		
12	169	179		
13	173	169		
14	175	173		
15	180	177		
16	179	180		
17	169	180		
18	180	178		
19	177	186		
20	180	169		

Abbildung 8.8 Beispiel für einen Gauß-Test

Ist Matrix leer, gibt die Funktion den Fehler #NV zurück. Nicht numerische Werte für x oder Sigma führen zu dem Fehler #WERT!, negative Werte für Sigma zu dem Fehler #ZAHL!.

In dem Beispiel in Abbildung 8.8 liefert der Test die Wahrscheinlichkeit 0,97 für die Hypothese, dass der Einzelwert zur gleichen Grundgesamtheit gehört.

Damit der Test zu brauchbaren Ergebnissen führt, sollte die Matrix wenigstens 30 Werte enthalten. Siehe auch Abschnitt 17.18, »Gaußtest«. Seit Excel 2010 wird für diese Funktion die umbenannte Funktion G.TEST() angeboten.

HYPGEOMVERT()

HYPGEOMDIST()

Syntax: HYPGEOMVERT(**Erfolge_S**; **Umfang_S**; **Erfolge_G**; **Umfang_G**)

Beispiel: =HYPGEOMVERT(6;6;6;49)

Ergebnis: 0,0000072%

Die Funktion HYPGEOMVERT() berechnet die Wahrscheinlichkeiten für hypergeometrisch verteilte Zufallsvariable. Die Funktion wird in Fällen angewendet, in denen es durch Entnahme aus der Grundgesamtheit – ohne anschließendes Zurücklegen – jedes Mal zu einer Änderung ihrer Zusammensetzung und damit der Erfolgswahrscheinlichkeit bei den nachfolgenden Entnahmen kommt, so dass hier die Binomialverteilung nicht eingesetzt werden kann. Vergleiche BINOMVERT().

	A	B	C	D	E	F	G
1							
2	**Wahrscheinlichkeit der hypergeometrischen Verteilung**						
3							
4					Gewinnchance im Lotto 6 aus 49:		
5	Erfolge_S	Umfang_S	Erfolge_G	Umfang_G	HYPGEOMVERT()		
6	6	6	6	49	0,0000072%	6 Richtige	
7	5	6	6	49	0,0018450%	5 Richtige	
8	4	6	6	49	0,0968620%	4 Richtige	
9	3	6	6	49	1,7650404%	3 Richtige	
10	2	6	6	49	13,2378029%	2 Richtige	
11	1	6	6	49	41,3019450%	1 Richtige	
12	0	6	6	49	43,5964976%	0 Richtige	
13							

Abbildung 8.9 Gewinnchancen im Lotto

Mit Umfang_S und Umfang_G werden die Größe der entnommenen Stichprobe und die Größe der Grundgesamtheit angegeben. Erfolge_G gibt an, wie oft das zu testende Ereignis in der Grundgesamtheit enthalten ist, Erfolge_S, wie oft es in der Stichprobe enthalten sein soll. Die Funktion liefert Werte für die Berechnung der Dichte, die entsprechende kumulierte Verteilungsfunktion muss durch Aufsummieren berechnet werden.

Ist Erfolge_S < 0 oder Erfolge_S größer als der kleinere der Werte von Umfang_S oder Erfolge_G, gibt die Funktion den Fehlerwert #ZAHL! zurück. Das gilt auch für folgende Fälle:

- Erfolge_S ist kleiner als der größere Wert von 0 bzw. (Umfang_S – Umfang_G + Erfolge_G).
- Umfang_S ist ≤ 0 oder Umfang_S > Umfang_G.
- Erfolge_G ist ≤ 0 oder Erfolge_G > Umfang_G.
- Umfang_G ist ≤ 0.

Ist eines der Argumente Erfolge_S, Umfang_S, Erfolge_G oder Umfang_G nicht numerisch, gibt die Funktion den Fehler #WERT! zurück.

Seit Excel 2010 wird für HYPGEOMVERT() die umbenannte Funktion HYPGEOM.VERT() angeboten, die beide Funktionstypen über das zusätzliche Argument Kumuliert erlaubt.

Im Beispiel wird die Wahrscheinlichkeit dafür ermittelt, dass beim Lotto 6 aus 49 6 Richtige erreicht werden. Da bei diesem Spiel jede Zahl höchstens einmal gezogen werden kann, ist die Fragestellung ein Fall für eine hypergeometrische Verteilung.

KONFIDENZ()

CONFIDENCE()

Syntax: KONFIDENZ(**Alpha**; **Standabwn**; **Umfang_S**)
Beispiel: =KONFIDENZ(0,05;2,6;200)
Ergebnis: 0,36033

Die Funktion KONFIDENZ() dient der Schätzung des Konfidenzintervalls (auch Vertrauensbereich, Mutungsintervall) für den Erwartungswert einer Zufallsvariable aus einer normalverteilten Grundgesamtheit anhand einer Stichprobe aus dieser Grundgesamtheit. Bei ein- wie zweiseitigen Fragestellungen wird ein bestimmter Prozentsatz (Alpha) extremer Fälle der Stichprobenverteilung als unwahrscheinlich ausgeschlossen. Diese Extremwerte liegen an den beiden Enden der Verteilung. Der Bereich zwischen den bei-

den Extremwerten beidseitig vom Mittelwert ist das Konfidenzintervall. Die entsprechende Wahrscheinlichkeit wird als Konfidenzniveau bezeichnet. Ein Wert von 90% ergibt sich über `1-Alpha`, wenn für `Alpha` 10% angenommen wird.

`Alpha` ist die Irrtumswahrscheinlichkeit (gewählt wird zumeist `0,05`, `0,01` oder `0,001`), das zweite Argument `Standabwn` gibt die als bekannt angenommene Standardabweichung der Grundgesamtheit (sie muss > `0` sein), `Umfang_S` die Größe der Stichprobe an. `Alpha` muss zwischen `0` und `1` ausschließlich liegen. Die Funktion ergibt das halbe Konfidenzintervall.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Alpha` ≤ `0` oder ≥ `1`, erscheint der Fehler `#ZAHL!`. Das gilt auch, wenn `Standabwn` ≤ `0` oder `Umfang_S` < `1`.

	A	B	C	D	E
1					
2	**Konfidenzintervall für den Erwartungswert einer t-verteilten Zufallsvariablen**				
3					
4	Seitenaufrufe pro Tag	Mittelwert (SP)	19,9		
5	10	Standardabweichung (GG)	5,00		
6	11	Irrtumswahrscheinlichkeit Alpha	5%		
7	11	Stichprobenumfang	30		
8	15				
9	16	**Berechnung des 95%-Konfidenzintervalls**			
10	16	KONFIDENZ()	1,79		
11	16	MITTELWERT()-KONFIDENZ()	18,11		
12	16	MITTELWERT()+KONFIDENZ()	21,69		
13	16				
14	16				
15	17				
16	18				
17	19				
18	19				
19	20				
20	20				
21	20				
22	20				
23	21				
24	21				
25	24				
26	24				
27	24				
28	25				
29	25				
30	26				

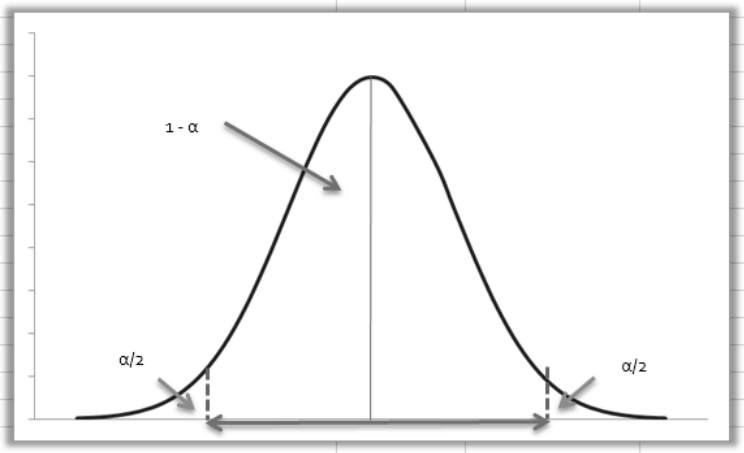


Abbildung 8.10 Schätzen des Konfidenzintervalls

Für den Mittelwert der Grundgesamtheit gilt:

```
Mgg = Mst +- k*(s/WURZEL(n))
```

wobei `Mgg` und `Mst` die Mittelwerte von Grundgesamtheit und Stichprobe sind, `k` der von der Funktion `KONFIDENZ()` ermittelte Wert, `s` die Standardabweichung der Stichprobe und `n` die Größe der Stichprobe.

Ergibt sich etwa im obigen Beispiel bei einer Werkstoffprüfung bei 200 Prüflingen eine durchschnittliche Länge von 102 mm mit einer Standardabweichung von `2,6`, dann liegt das arithmetische Mittel mit einer Wahrscheinlichkeit von 90% (`0,9 = 1 – 2*0,05`) im Bereich zwischen

```
102 - 0,3603 * 2,6/WURZEL(200)
```

und

```
102 + 0,3603 * 2,6/WURZEL(200)
```

also zwischen 101,934 und 102,066.

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `KONFIDENZ.NORM()` und `KONFIDENZ.T()` zur Verfügung, wobei `KONFIDENZ()` dasselbe Ergebnis liefert wie die Funktion `KONFIDENZ.NORM()`.

KOVAR()

COVAR()

Syntax:	`KOVAR(`**`Matrix1`**`; `**`Matrix2`**`)`
Beispiel:	`=KOVAR({2;4;6;8;10;12};{12;2;10;4;8;6})`
Ergebnis:	`-3`

Die Funktion `KOVAR()` liefert ähnlich wie die Funktion `KORREL()` ein Maß für den Zusammenhang zwischen den Daten zweier Datenreihen aus verbundenen Stichproben. Sie ermittelt, in welchem Maß die Daten der beiden Datenreihen gemeinsam von ihrem jeweiligen Mittelwert abweichen.

Beide Argumente müssen dieselbe Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Dabei werden in den angegebenen Zellbereichen oder Matrizen Texteinträge, Wahrheitswerte und leere Elemente ignoriert. Enthält eines oder beide Argumente kein numerisches Datenelement, erscheint der Fehler `#DIV/0!`.

Die Funktion ermittelt für jeden Datenpunkt die Differenz zum Mittelwert und bildet paarweise das Produkt aus den beiden Abweichungen. Anschließend wird der Mittelwert dieser Produkte berechnet. Dabei sind von der Größe her beliebige Ergebnisse möglich. Entscheidend ist, ob das Ergebnis positiv oder negativ ist. Positive Werte deuten auf einen linearen Zusammenhang der beiden Variablen hin – wenn x größer wird, wird auch y größer. Negative Werte deuten auf einen gegensinnigen Zusammenhang hin – wenn x größer wird, wird y kleiner. Null bedeutet, dass kein Zusammenhang existiert. Die Funktion zeigt also nur die Richtung an, in der zwei Variablen korrelieren.

	A	B	C	D
1				
2	**Kovarianz**			
3				
4	Matrix1	Matrix2	Matrix3	Matrix4
5	2	12	12	20
6	4	10	2	40
7	6	8	10	60
8	8	6	4	80
9	10	4	8	100
10	12	2	6	120
12		KOVARIANZ 1-2	-11,66666667	
13		KOVARIANZ 2-3	3	
14		KOVARIANZ 1-4	116,6666667	

Abbildung 8.11 Kovarianzen verbundener Stichproben

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `KOVARIANZ.P()` sowie `KOVARIANZ.S()` zur Verfügung, wobei `KOVAR()` dasselbe Ergebnis wie `KOVARIANZ.P()` liefert.

KRITBINOM()

CRITBINOM()

Syntax: `KRITBINOM(`**`Versuche`**`;` **`Erfolgswahrsch`**`;` **`Alpha`**`)`

Beispiel: `=KRITBINOM(200;0,9;0,01)`

Ergebnis: `170`

Die Funktion `KRITBINOM()` liefert die kleinste Anzahl erfolgreicher Versuche, für die die kumulierte Wahrscheinlichkeit größer oder gleich der mit `Alpha` angegebenen Irrtums- oder Grenzwahrscheinlichkeit ist. Voraussetzung ist, dass die Zufallsgröße binomialver-

teilt ist (vergleiche BINOMVERT()). Mit Versuche wird die Zahl der Versuche angegeben; mit Erfolgswahrsch die Wahrscheinlichkeit für den erfolgreichen Ausgang eines Versuchs.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Versuche < 0 oder Erfolgswahrsch < 0 oder > 1, erscheint der Fehler #ZAHL!; das gilt auch für Alpha < 0 oder > 1.

	A	B	C	D
1				
2	**KRITBINOM**			
3				
4	Versuche	Erfolgswahrsch	Alpha	KRITBINOM()
5	200	0,9	0,01	170
6	200	0,9	0,02	171
7	200	0,9	0,03	172

Abbildung 8.12 Beispiel für KRITBINOM()

Das Ergebnis der Funktion kann als Akzeptanzkriterium verwendet werden, um z. B. zu entscheiden, ob die Fehlerrate in einem Fertigungslos noch geduldet werden kann oder nicht. Im Beispiel wird angenommen, dass bei einer gegebenen Maschineneinstellung von 200 Prüflingen im Durchschnitt 180 (= 90%) korrekt sind, die Wahrscheinlichkeit für einen korrekten Prüfling also 0,9 ist. Die Fragestellung lautet: Mit wie vielen korrekten Prüflingen können Sie mit einer Irrtumswahrscheinlichkeit von 0,01 mindestens rechnen? Das Ergebnis lautet 170, d. h. in 99% aller 200-Stück-Lieferungen sind 170 korrekte Produkte zu erwarten.

Seit Excel 2010 wird für KRITBINOM() die umbenannte Funktion BINOM.INV() angeboten.

LOGINV()

LOGINV()

Syntax: LOGINV(**Wahrsch**; **Mittelwert**; **Standabwn**)
Beispiel: =LOGINV(0,01;0;1)
Ergebnis: 0,098

Die Funktion LOGINV() liefert Quantile einer logarithmischen Normalverteilung. Das Argument Wahrsch ist die Wahrscheinlichkeit, als Zweites wird der Mittelwert von

ln(x) und als Drittes mit Standabwn die Standardabweichung von ln(x) angegeben. Die Funktion hilft bei der Analyse von Daten, deren Logarithmus normalverteilt ist. Anwendungsbereiche sind Statistiken über Einkommensverteilungen oder Schadensfälle in der Versicherungsbranche. Die Funktion ist die Umkehrung von LOGNORMVERT().

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist Wahrsch ≤ 0 oder ≥ 1, erscheint der Fehler #ZAHL!; das gilt auch für Standabwn ≤ 0.

	A	B	C	D
1				
2	**Quantile einer logarithmischen Normalverteilung**			
3				
4	Wahrsch	Mittelwert	Standabwn	LOGINV()
5	0,01	0	1	0,097651733
6	0,05	3,5	1,2	4,600549016

Abbildung 8.13 Beispiele für LOGINV()

Seit Excel 2010 wird für LOGINV() die umbenannte Funktion LOGNORM.INV() angeboten.

LOGNORMVERT()

LOGNORMDIST()

Syntax: LOGNORMVERT(**x**; **Mittelwert**; **Standabwn**)

Beispiel: =LOGNORMVERT(1;0;1)

Ergebnis: 0,5

Die Funktion LOGNORMVERT() liefert die Wahrscheinlichkeitsverteilung für eine logarithmische Normalverteilung. Bei einigen Experimenten, z.B. über Reaktionszeiten, ergibt sich als Häufigkeitsverteilung ein asymmetrischer, linkssteiler Kurvenzug. Durch Logarithmieren lassen sich daraus häufig normalverteilte Messwerte erstellen.

Das Argument x bezeichnet den Wert des Quantils, Mittelwert ist das arithmetische Mittel, und drittes Argument ist Standabwn, die Standardabweichung der Stichprobe.

Sind x, Mittelwert oder Standabwn nicht numerisch, gibt die Funktion den Fehler #WERT! zurück. Ist x ≤ 0 oder Standabwn ≤ 0, erscheint der Fehler #ZAHL!.

Seit Excel 2010 wird für LOGNORMVERT() die umbenannte Funktion LOGNORM.VERT() angeboten.

	A	B	C
1			
2	**Wahrscheinlichkeit bei logarithmischer Normalverteilung**		
3			
4	Standabwn	1	0,5
5	Mittelwert	0	0
6		LOGNORMVERT()	
7	x	Verteilung	
8	0,2	0,05376031	0,000643471
9	0,4	0,179757214	0,033432418
10	0,6	0,304736583	0,153472997
11	0,8	0,411711892	0,327694936
12	1	0,5	0,5
13	1,2	0,572334809	0,642311086
14	1,4	0,631742608	0,749508691
15	1,6	0,680823788	0,826393081
16	1,8	0,721662252	0,880117291
17	2	0,755891404	0,917171481
18	2,2	0,784785385	0,942592426
19	2,4	0,809340543	0,96002166
20	2,6	0,83034044	0,97199918
21	2,8	0,848405652	0,980264318

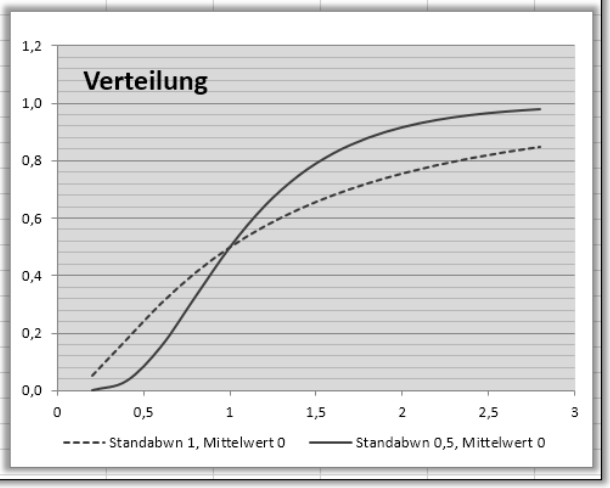


Abbildung 8.14 Verteilung logarithmischer Werte

MODALWERT()

MODE()

Syntax: `MODALWERT(`**`Zahl1`**`; Zahl2; ...)`

Beispiel: `=MODALWERT(2;6;3;6;1;5;6)`

Ergebnis: `6`

Die Funktion `MODALVERT()` liefert den in einer Datenreihe am häufigsten vorkommenden Wert. Damit gehört die Funktion zu den grundlegenden statistischen Kennwerten der Maße der zentralen Tendenz. Mit dem Modalwert oder Modus lassen sich schnell Informationen über den Schwerpunkt der Verteilung gewinnen. Die Funktion kann seit Excel 2007 bis zu 255 Argumente enthalten, in den älteren Versionen bis zu 30.

Enthält eines der Argumente einen Fehlerwert, gibt auch die Funktion diesen Fehlerwert aus. Kann die Funktion keinen Modalwert angeben, weil keiner der Werte zumindest zweimal vorkommt, wird der Fehlerwert `#NV` ausgegeben.

Betrachten Sie eine Verteilung, so ist das Maximum der Verteilung gleich dem Modalwert. Der Modalwert einer Häufigkeitsverteilung (siehe `HÄUFIGKEIT()`) liegt in der Kategorienmitte der am häufigsten besetzten Kategorie.

	A	B	C	D	E
1					
2	**Modalwerte berechnen**				
3					
4	Person	Geschlecht	Alter	Gewicht	
5	1	m	23	69	
6	2	m	25	66	
7	3	w	35	68	
8	4	m	35	72	
9	5	w	34	74	
10	6	w	35	63	
11	7	m	43	68	
12	8	w	22	79	
13	9	m	43	58	
14	10	w	43	79	
15	11	w	45	68	
16	12	m	47	72	
17	13	m	47	77	
18	14	m	47	56	
19	15	w	47	60	
20	16	w	33	62	
21	17	w	53	58	
22	18	m	53	61	
23	19	w	53	55	
24	20	m	63	57	
25		**MODALWERT()**	47	68	
26					

Abbildung 8.15 Modalwert für zwei Merkmale

Kann die Funktion keinen Modalwert angeben, weil keiner der Werte zumindest zweimal vorkommt, wird ein Fehlerwert ausgegeben. Bei gleich häufigem Vorkommen verschiedener Werte wird der in der Liste zuerst vorkommende ausgegeben. Anstelle dieser Funktion werden seit Excel 2010 die beiden Funktionen `MODUS.EINF()` sowie `MODUS.VIELF()` angeboten.

NEGBINOMVERT()

NEGBINOMDIST()

Syntax: `NEGBINOMVERT(Zahl_Misserfolge; Zahl_Erfolge; Erfolgswahrsch)`
Beispiel: `=NEGBINOMVERT(5;1;1/6)`
Ergebnis: `0,0669`

Die Funktion `NEGBINOMVERT()` benutzt als Grundlage ihrer Berechnungen ebenso wie `BINOMVERT()` die Binomialverteilung und wird auch als negative Binomialverteilung bezeichnet.

Sie berechnet, mit welcher Wahrscheinlichkeit ein zusammengesetztes Ereignis auftritt. Als Argumente werden `Zahl_Misserfolge` und `Zahl_Erfolge` angegeben. Zusammen mit der Angabe von `Erfolgswahrsch` ermittelt die Funktion die Wahrscheinlichkeit dafür, dass das zusammengesetzte Ereignis (erst die angegebene Zahl an Misserfolgen, dann die angegebene Zahl der Erfolge) auftritt. Die Funktion liefert Werte für die Berechnung der Dichte, die entsprechende kumulierte Verteilungsfunktion muss durch Aufsummieren berechnet werden.

Ist `Zahl_Erfolge` < `0` oder `Zahl_Erfolge` < `1`, gibt die Funktion den Fehlerwert `#ZAHL!` zurück. Das gilt ebenfalls für `Erfolgswahrsch` < `0` oder > `1`. Ist eines der Argumente `Zahl_Misserfolge`, `Zahl_Erfolge`, `Erfolgswahrsch` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück.

Seit der Version Excel 2010 wird anstelle dieser Funktion die umbenannte Funktion `NEGBINOM.VERT()` angeboten, die beide Funktionstypen über das zusätzliche Argument `Kumuliert` erlaubt.

Im Beispiel oben wird die Wahrscheinlichkeit ermittelt, hintereinander genau fünfmal nicht die Sechs und dann die Sechs zu werfen. Das Ergebnis liegt bei etwa 7%.

	A	B	C	D
1				
2	**Wahrscheinlichkeit einer negativbinomialverteilten Zufallsvariablen**			
3				
4	Zahl_Misserfolge	Zahl_Erfolge	Erfolgswahrsch	NEGBINOMVERT()
5	0	5	0,5	3,13%
6	3	5	0,5	13,67%
7	6	5	0,5	10,25%
8	10	5	0,5	3,05%
9	20	5	0,5	0,03%

Abbildung 8.16 Wahrscheinlichkeit bei zusammengesetzten Ereignissen

NORMINV()

NORMINV()

Syntax: `NORMINV(`**`Wahrsch`**`;` **`Mittelwert`**`;` **`Standabwn`**`)`

Beispiel: `=NORMINV(0,5;20;30)`

Ergebnis: `20`

Die Funktion `NORMINV()` liefert Quantile der Normalverteilung und ist die Umkehrfunktion von `NORMVERT()`.

	A	B	C	D
1				
2	**Berechnen von Quantilen der Normalverteilung**			
3				
4	Wahrsch	Mittelwert	Standabwn	NORMINV()
5	10%	1200	450	623,30
6	20%	1200	450	821,27
7	30%	1200	450	964,02
8	40%	1200	450	1085,99
9	50%	1200	450	1200,00
10	60%	1200	450	1314,01
11	70%	1200	450	1435,98
12	80%	1200	450	1578,73
13	90%	1200	450	1776,70

Abbildung 8.17 Rückrechnen von der Wahrscheinlichkeit in einer Normalverteilung auf das Quantil

Als Argumente werden `Wahrsch` (die Wahrscheinlichkeit, zu der das Quantil gesucht wird) sowie `Mittelwert` und mit `Standabwn` die Standardabweichung der Verteilung angegeben.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` < `0` oder > `1`, erscheint der Fehler `#ZAHL!`; das gilt auch für `Standabwn` ≤ `0`.

Wie bei der Normalverteilung gilt auch hier, dass bei Mittelwert = `0` und Standardabweichung = `1` eine Standardnormalverteilung vorliegt. In diesem Fall kann auch `STANDNORMINV()` eingesetzt werden. Seit Excel 2010 wird für `NORMINV()` die umbenannte Funktion `NORM.INV()` angeboten.

NORMVERT()

NORMDIST()

Syntax: `NORMVERT(`**`x`**`;` **`Mittelwert`**`;` **`Standabwn`**`;` **`Kumuliert`**`)`

Beispiel: `=NORMVERT(9;9;4;WAHR)`

Ergebnis: `0,5`

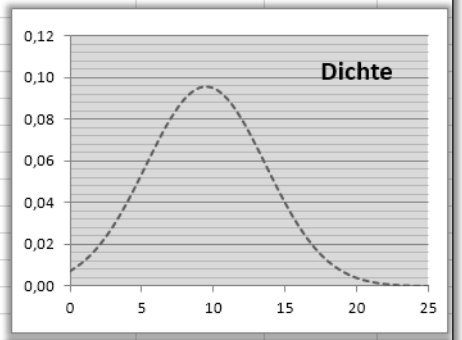
Die Funktion `NORMVERT()` liefert die Werte für eine Normalverteilung. `x` bezeichnet den Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit berechnet werden soll. Wird die Funktion grafisch dargestellt, ist das der Wert auf der x-Achse. Dabei ergibt sich immer ein glockenförmiger Verlauf. Wie er im Einzelnen ausfällt, hängt von den Argumenten `Mittelwert` und `Standabwn` ab. Der Mittelwert (Erwartungswert) gibt die Lage der Funktion auf der x-Achse an und markiert dabei den Gipfel dieser Funktion. Die Stan-

dardabweichung gibt die Streuung an und bestimmt damit, wie flach oder steil die Funktion verläuft. Mit `Kumuliert = WAHR` erhalten Sie die Verteilungsfunktion, also die Wahrscheinlichkeit dafür, dass die Zufallsvariable einen Wert von `x` oder kleiner annimmt. Mit `FALSCH` erhalten Sie die Werte der Dichtefunktion.

Mit `Mittelwert = 0` und `Standabwn = 1` erhalten Sie die Standardnormalverteilung, die Sie auch direkt mit der Funktion `STANDNORMVERT()` abfragen können.

Sind `x`, `Mittelwert` oder `Standabwn` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Standabwn` ≤ 0, erscheint der Fehler `#ZAHL!`.

	A	B	C	D	E
2	**Normalverteilung**				
3					
4	x	Verteilung	Dichte	MITTELWERT()	9,5
5	0	0,01157548	0,00723655	STABW()	4,183300133
6	1	0,02108247	0,01210272		
7	2	0,03649902	0,01911691		
8	3	0,0601164	0,02851908		
9	4	0,09429672	0,04018246		
10	5	0,14102944	0,05347129		
11	6	0,20139185	0,06720289		
12	7	0,27504866	0,07976979		
13	8	0,35995893	0,08942771		
14	9	0,45243056	0,09468669		
15	10	0,54756944	0,09468669		
16	11	0,64004107	0,08942771		
17	12	0,72495134	0,07976979		
18	13	0,79860815	0,06720289		
19	14	0,85897056	0,05347129		
20	15	0,90570328	0,04018246		
21	16	0,9398836	0,02851908		
22	17	0,96350098	0,01911691		
23	18	0,97891753	0,01210272		
24	19	0,98842452	0,00723655		
25	20	0,9939631	0,00408661		
26	21	0,99701137	0,00217961		
27	22	0,99859628	0,00109793		
28	23	0,99937478	0,00052235		
29	24	0,99973603	0,0002347		
30	25	0,99989439	9,9602E-05		



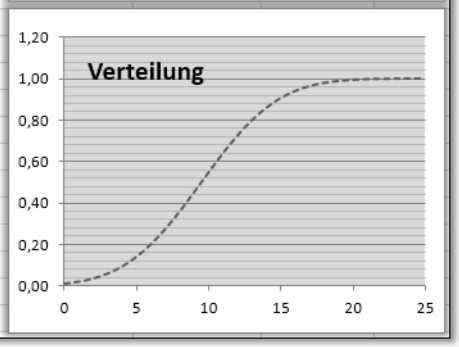


Abbildung 8.18 Normalverteilungsfunktion und Dichtefunktion

Für die Normalverteilung gelten folgende Eigenschaften:

Die Verteilung ist glockenförmig und eingipflig. Sie nähert sich asymptotisch der x-Achse. Zugleich ist sie symmetrisch. Der höchste Wert ist zugleich der Mittelwert, wobei der arithmetische Mittelwert mit dem Median zusammenfällt. 50% der Fläche

liegen beidseitig vom Mittelwert. Die Wendepunkte liegen bei Mittelwert + Standardabweichung bzw. Mittelwert – Standardabweichung.

Die Fläche unter der Dichtekurve hat immer den Wert 1. Die Wahrscheinlichkeit, dass eine Zufallsvariable einen Wert zwischen x1 und x2 annimmt, wird ermittelt, indem die entsprechende Fläche unter der Dichtekurve berechnet wird. Folglich hat der Mittelwert die Wahrscheinlichkeit von 50%.

Seit Excel 2010 wird für NORMVERT() die umbenannte Funktion NORM.VERT() angeboten.

OBERGRENZE()

CEILING()

Syntax: OBERGRENZE(**Zahl**; **Schritt**)
Beispiel: =OBERGRENZE(2,2434;0,05)
Ergebnis: 2,25

Die Funktion OBERGRENZE() rundet den mit dem Argument Zahl angegebenen Wert auf das nächste Vielfache von Schritt auf und ist damit komplementär zu UNTERGRENZE(). Sie erlaubt also das Aufrunden auf bestimmte Intervallgrenzen.

Mit dem Wert 0,05 für Schritt kann z.B. bestimmt werden, dass die Hundertstelstelle beim Aufrunden immer nur eine 5 oder eine 0 sein kann. Mit einem Wert 0,05 für Schritt wird dafür gesorgt, dass z.B. nicht mehr in Cent, sondern nur noch für 5-Cent-Stücke ausgepreist wird.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler #WERT!.

	A	B	C	D
1				
2	**Aufrunden auf das kleinste Vielfache**			
3				
4	Wert	Schritt	OBERGRENZE()	
5	14,25	0,5	14,5	
6	14,33	1	15	
7	123,56	2	124	
8	124,33	5	125	
9	126,33	5	130	
10	124,33	10	130	
11	-112	-10	-120	
12	-112	10	-110	
13				

Abbildung 8.19 Aufrunden auf bestimmte Obergrenzen

Aufrunden bedeutet im Sinne dieser Funktion, dass immer von der Null weg gerundet wird, z.B. ergibt `=OBERGRENZE(-4,2546;-0,5)` den Wert `-4,5`. Bei unterschiedlichen Vorzeichen für `Zahl` und `Schritt` wird eine Fehlermeldung ausgegeben.

POISSON()

POISSON()

Syntax:	`POISSON(`**`x`**`;` **`Mittelwert`**`;` **`Kumuliert`**`)`
Beispiel:	`=POISSON(50;60;WAHR)`
Ergebnis:	`0,1077`

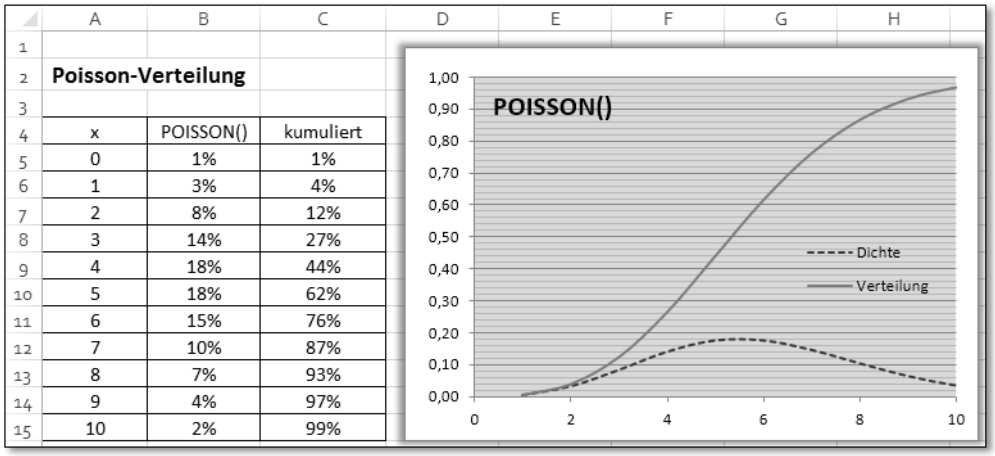
Die Funktion `POISSON()` liefert die Wahrscheinlichkeiten für Zufallsvariable, die einer Poisson-Verteilung angehören. Die Poisson-Verteilung ist wie die Binomial- und die hypergeometrische Verteilung eine Verteilung, die nur diskrete Werte annehmen kann. Die Poisson-Verteilung ist für große Zahlen eine gute Näherung an die Binomialverteilung. Sie ist die Grenzverteilung der Binomialverteilung für den Fall, dass die Anzahl der Ereignisse insgesamt gegen unendlich und die Anzahl der Ausnahmeereignisse gegen null geht.

An Argumenten verlangt die Funktion `x` für die Anzahl der Fälle und `Mittelwert` für den Erwartungswert. `Kumuliert` ist ein Wahrheitswert. Mit `Kumuliert = FALSCH` wird die Wahrscheinlichkeit dafür berechnet, dass die Zufallsvariable genau den Wert `x` annimmt, mit `Kumuliert = WAHR` die Wahrscheinlichkeit dafür, dass die Zufallsvariable einen Wert von `x` oder kleiner annimmt.

Ist `x` oder `Mittelwert` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `x < 0` oder `Mittelwert < 0`, erscheint der Fehler `#ZAHL!`.

Die Funktion kann angewendet werden, wenn für eine sehr große Zahl von Fällen die Wahrscheinlichkeit von seltenen Ausnahmeereignissen geschätzt werden soll. Dabei muss nur bekannt sein, wie häufig im Durchschnitt das Ausnahmeereignis auftritt.

Die Tabelle in Abbildung 8.20 kann beispielsweise benutzt werden, um die Wahrscheinlichkeit von Bitübertragungsfehlern im Netz zu schätzen. Angenommen, bei einer Übertragung von 1 Million Bits treten durchschnittlich 5 Fehler auf, dann kann die `POISSON()`-Funktion die Frage beantworten, mit welcher Wahrscheinlichkeit bei der nächsten Million die Fehleranzahl `x` auftritt.



	A	B	C
1			
2	**Poisson-Verteilung**		
3			
4	x	POISSON()	kumuliert
5	0	1%	1%
6	1	3%	4%
7	2	8%	12%
8	3	14%	27%
9	4	18%	44%
10	5	18%	62%
11	6	15%	76%
12	7	10%	87%
13	8	7%	93%
14	9	4%	97%
15	10	2%	99%

Abbildung 8.20 Wahrscheinlichkeit von Variablen, die eine Poisson-Verteilung darstellen

Da die Poisson-Verteilung normalerweise dazu verwendet wurde, die bei großen Zahlen schwer zu handhabende Binomialverteilung anzunähern, gibt es selten einen Grund, sie noch zu verwenden. Schließlich bietet Excel auch jene Funktion an.

Seit Excel 2010 wird für `POISSON()` die umbenannte Funktion `POISSON.VERT()` angeboten.

QUANTIL()

PERCENTILE()

Syntax:	`QUANTIL(`**`Matrix; Alpha`**`)`
Beispiel:	siehe Abbildung 8.21

Die Funktion `QUANTIL()` liefert denjenigen Wert einer Datenreihe, die über das Argument `Matrix` geliefert wird, unterhalb dessen ein mit `Alpha` angegebener Bruchteil der Daten liegt. Mit dieser Funktion wird eine Verteilung nach einer Skala unterteilt, deren unterster und oberster Punkt den tiefsten und den höchsten Wert der Daten bildet.

`Matrix` sind die zu unterteilenden Daten. Durch das Argument `Alpha` wird ein Lagemaß (Quantil) angegeben. Das Maß `0,25` (25%) bezeichnet z. B. den Punkt, unterhalb dessen ein Viertel aller Beobachtungen liegt. Einige Quantile, die besonders oft verwendet werden, haben eigene Bezeichnungen wie Quartil für 25%-Abschnitte, Dezil für 10%-Abschnitte. Das zweite Quartil oder ein Quantil von 0,5 bezeichnet dann den Median. Das Argument `Alpha` kann jeden Wert zwischen `0` und `1` annehmen; liegt ein Quantil zwischen

zwei Beobachtungen, wird durch Interpolation der entsprechende Wert ermittelt. Enthält `Matrix` mehr als 8.191 Datensätze, wird eine Fehlermeldung ausgegeben.

Nicht numerische Werte oder Wahrheitswerte in `Matrix` werden ignoriert. Enthält `Matrix` aber gar keine numerischen Werte, liefert die Funktion den Fehler `#ZAHL!`. Das gilt auch, wenn `Alpha` < `0` oder > `1` ist. Ist `Alpha` nicht numerisch, erscheint der Fehler `#WERT!`.

	A	B	C	D	E
1					
2	**Quantile berechnen**				
3					
4	Person	Gewicht	Alpha	QUANTIL()	
5	1	69	0,0	55	
6	2	66	0,1	57	
7	3	68	0,2	58	
8	4	72	0,3	61	
9	5	74	0,4	63	
10	6	63	0,5	67	**=MEDIAN()**
11	7	68	0,6	68	
12	8	79	0,7	70	
13	9	58	0,8	72	
14	10	79	0,9	77	
15	11	68	1,0	79	
16	12	72			
17	13	77			
18	14	56			
19	15	60			
20	16	62			
21	17	58			
22	18	61			
23	19	55			
24	20	57			

Abbildung 8.21 Einteilung von Daten in Quantile

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `QUANTIL.EXKL()` und `QUANTIL.INKL()` zur Verfügung, wobei `QUANTIL.INKL()` dasselbe Ergebnis liefert wie `QUANTIL()`.

QUANTILSRANG()

PERCENTRANK()

Syntax: `QUANTILSRANG(`**`Matrix`**`; `**`x`**`; Genauigkeit)`

Beispiel: siehe Abbildung 8.22

Die Funktion QUANTILSRANG() liefert die Angabe des Anteils von Daten, die unterhalb des angegebenen Werts liegen.

Das Argument x bezeichnet den Wert, dessen relative Position ermittelt werden soll; Matrix sind die Daten. Wenn x selbst als Wert nicht in der Matrix auftaucht, wird der entsprechende Wert interpoliert. Mit Genauigkeit lässt sich die Anzahl der Stellen für die Ausgabe des Ergebnisses bestimmen. Wird Genauigkeit nicht angegeben, wird 3 angenommen.

Enthält Matrix gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Ist x größer oder kleiner als der größte oder kleinste Wert, liefert die Funktion den Fehler #NV.

Der Zusammenhang mit QUANTIL() sieht so aus: Wenn:

```
x = QUANTIL(Matrix;0,2)
```

dann ist:

```
0,2 = QUANTILSRANG(Matrix;x)
```

	A	B	C
1			
2	**Quantilsrang berechnen**		
3			
4	Person	Gewicht	QUANTILSRANG()
5	1	69	0,68
6	2	66	0,47
7	3	68	0,53
8	4	72	0,74
9	5	74	0,84
10	6	63	0,42
11	7	68	0,53
12	8	79	0,95
13	9	58	0,16
14	10	79	0,95
15	11	68	0,53
16	12	72	0,74
17	13	77	0,89
18	14	56	0,05
19	15	60	0,26
20	16	62	0,37
21	17	58	0,16
22	18	61	0,32
23	19	55	0,00
24	20	57	0,11

Abbildung 8.22 Berechnen des Quantilsrangs

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen QUANTILSRANG.EXKL() und QUANTILSRANG.INKL() zur Verfügung, wobei QUANTILSRANG() dasselbe Ergebnis liefert wie QUANTILSRANG.INKL().

QUARTILE()

QUARTILE()

Syntax: QUARTILE(**Matrix**; **Quartil**)
Beispiel: siehe Abbildung 8.23

Die Funktion QUARTILE() unterteilt die Daten von Matrix in Bereiche mit je gleichen Anteilen von Daten und ist damit ein Spezialfall von QUANTIL() (siehe dort). Für Quartil sind fünf Belegungen möglich: 0 (liefert den niedrigsten Wert); 1 (25%-Quantil), 2 (50%-Quantil = Median); 3 (75%-Quantil) und 4 (für den höchsten Wert).

	A	B	C	D	E
1					
2	**Einteilung in Quartile**				
3					
4	Person	Gewicht	QUARTILE()		
5	1	69,00	58,00	Kleinster Wert	=MIN()
6	2	66,00	65,25	25%-Quartil	
7	3	68,00	68,50	50%-Quartil	=MEDIAN()
8	4	72,00	72,50	75%-Quartil	
9	5	74,00	79,00	Größter Wert	=MAX()
10	6	63,00			
11	7	68,00			
12	8	79,00			
13	9	58,00			
14	10	69,00			
15	11	59,00			
16	12	79,00			

Abbildung 8.23 Einteilung der Daten in Quartile

Enthält Matrix gar keine numerischen Werte, liefert die Funktion den Fehler #ZAHL!. Das gilt auch, wenn Quartile < 0 oder > 4 ist. Ist Quartile nicht numerisch, erscheint der Fehler #WERT!.

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen QUARTILE.EXKL() und QUARTILE.INKL() zur Verfügung, wobei QUARTILE() dasselbe Ergebnis liefert wie die Funktion QUARTILE.INKL().

RANG()

RANK()

Syntax: `RANG(`**`Zahl`**`; `**`Bezug`**`; Reihenfolge)`

Beispiel: siehe Abbildung 8.24

Die Funktion `RANG()` liefert den Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt. Mit `Zahl` wird der Wert angegeben, dessen Rang bestimmt werden soll; `Bezug` ist die Datenreihe, wobei nicht numerische Werte bei der Rangberechnung ignoriert werden bzw. zu Fehlern führen, wenn der Rang dieses Werts angegeben werden soll. Mit `Reihenfolge` wird angegeben, ob in fallender oder steigender Ordnung gezählt wird. Vorgegeben ist die fallende Ordnung, die dann verwendet wird, wenn das Argument nicht oder mit `0` belegt ist. Bei jedem anderen Wert zählt Excel in steigender Ordnung.

Nicht numerische Werte in `Bezug` werden bei der Rangberechnung ignoriert. Soll aber der Rang eines solchen Werts angegeben werden, liefert die Funktion den Fehler `#WERT!`. Ist eine Zelle in `Bezug` leer, liefert die Funktion für diese Zelle den Fehler `#NV`.

	A	B	C
1			
2	**Rang von Laufergebnissen berechnen**		
3			
4	Teilnehmer	Zeit	RANG()
5	5	10,60	1
6	9	10,70	2
7	4	10,80	3
8	18	10,80	3
9	3	10,90	5
10	14	10,90	5
11	17	11,30	7
12	13	11,40	8
13	2	11,50	9
14	15	disqual.	#WERT!

Abbildung 8.24 Rangordnung von Laufzeiten

Excel 2010 stellt für Rangberechnungen die beiden Funktionen `RANG.GLEICH()` und `RANG.MITTELW()` zur Verfügung.

STABW()

STDEV()

Syntax: `STABW(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=STABW(33;22;28;17;23;26)`
Ergebnis: `5,49`

Die Funktion `STABW()` schätzt für die Werte in der Argumentliste die vermutete Standardabweichung vom arithmetischen Mittelwert der Grundgesamtheit. Dabei werden die in der Argumentliste gegebenen Daten als Stichprobe aus dieser Grundgesamtheit genommen. Die Funktion kann seit Excel 2007 bis zu 255 Argumente enthalten, in den älteren Versionen bis zu 30.

Die Standardabweichung ist ein Maß dafür, wie weit die vorhandenen Daten um den Mittelwert streuen. Sie ist die Quadratwurzel aus dem arithmetischen Mittelwert der quadrierten Abweichungen vom arithmetischen Mittelwert. Bei der Ermittlung des Mittelwerts wird bei dieser Funktion aber nicht mit `n` für den gesamten Umfang der Stichprobe gerechnet, sondern mit `n – 1`.

	A	B	C	D
1				
2	**Berechnen von Standardabweichung und Varianz**			
3				
4	Person	Geschlecht	Alter	Gewicht
5	1	m	23	69
6	2	m	25	66
7	3	w	29	68
8	4	m	33	72
9	5	w	35	74
10	6	w	35	63
11	7	m	43	68
12	8	w	43	79
13	9	m	43	58
14	10	w	43	79
15		STABW()	7,74	6,65
16		STABWN()	7,35	6,31
17		VARIANZ()	59,96	44,27
18		VARIANZEN()	53,96	39,84

Abbildung 8.25 Standardabweichung und Varianz der Angaben zum Gewicht

Enthält ein Bereich gar keine numerischen Werte, liefert die Funktion den Fehler `#DIV/0!`.

Seit Excel 2010 wird für `STABW()` die umbenannte Funktion `STABW.S()` angeboten.

STABWN()

STDEVP()

Syntax: STABWN(**Zahl1**; Zahl2; ...)
Beispiel: =STABWN(33;22;28;17;23;26)
Ergebnis: 5,01

Die Funktion STABWN() berechnet für die Werte in der Argumentliste die Standardabweichung vom arithmetischen Mittelwert. Dabei werden die in der Argumentliste gegebenen Daten als Grundgesamtheit genommen. Die Funktion kann seit Excel 2007 bis zu 255 Argumente enthalten, in den älteren Versionen bis zu 30.

Enthält ein Bereich überhaupt keine numerischen Werte, liefert die Funktion den Fehler #DIV/0!.

Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittelwert der quadrierten Abweichungen vom arithmetischen Mittelwert. Bei der Ermittlung des Mittelwerts wird anders als bei den Funktionen STABW() und STABWA() mit n für den Umfang der Grundgesamtheit gerechnet. Für diese Funktion ist seit Excel 2010 die umbenannte Funktion STABW.N() vorgesehen.

STANDNORMINV()

NORMSINV()

Syntax: STANDNORMINV(**Wahrsch**)
Beispiel: =STANDNORMINV(0,90)
Ergebnis: 1,28

Die Funktion STANDNORMINV() liefert bei einer Standardnormalverteilung für die mit Wahrsch angegebene Wahrscheinlichkeit – ein Wert zwischen 0 und 1 einschließlich – den Wert auf der x-Achse (Quantil). Die Funktion ist die Umkehrung der Funktion STANDNORMVERT().

Ist Wahrsch < 0 oder > 1, erscheint der Fehler #ZAHL!, ist der Wert nicht numerisch, der Fehler #WERT!.

Die Standardnormalverteilung ist eine Variante der Normalverteilung und dadurch gekennzeichnet, dass der Mittelwert (Erwartungswert) gleich 0 ist und die Standardabweichung gleich 1.

	A	B	C
1			
2	**Quantile der Standardnormalverteilung**		
3			
4	z	STANDNORMINV()	
5	5%	-1,64	
6	10%	-1,28	
7	15%	-1,04	
8	20%	-0,84	
9	25%	-0,67	
10	30%	-0,52	
11	35%	-0,39	
12	40%	-0,25	
13	45%	-0,13	
14	50%	0,00	
15	55%	0,13	
16	60%	0,25	
17	65%	0,39	
18	70%	0,52	
19	75%	0,67	
20	80%	0,84	
21	85%	1,04	
22	90%	1,28	
23	95%	1,64	

Abbildung 8.26 Quantile der Standardnormalverteilung

Seit Excel 2010 wird anstelle dieser Funktion die Funktion `NORM.S.INV()` angeboten.

STANDNORMVERT()

NORMSDIST()

Syntax: `STANDNORMVERT(`**z**`)`
Beispiel: `=STANDNORMVERT(0)`
Ergebnis: `0,5`

Die Funktion `STANDNORMVERT()` gibt die Wahrscheinlichkeit dafür, dass eine Zufallsvariable aus einer Standardnormalverteilung den Wert `z` oder kleiner annimmt. Aus der Tabelle in Abbildung 8.27 kann also beispielsweise abgelesen werden, dass ein Wert `z` von höchstens `1` mit 84%-iger Wahrscheinlichkeit auftritt, der Wert `1` selbst mit 24%-iger Wahrscheinlichkeit.

	A	B	C
1			
2	**Standardnormalverteilung**		
3			
4	z	STANDNORMVERT()	NORMVERT(z;0;1;0)
5	-3	0,0013	0,0044
6	-2,75	0,0030	0,0091
7	-2,5	0,0062	0,0175
8	-2,25	0,0122	0,0317
9	-2	0,0228	0,0540
10	-1,75	0,0401	0,0863
11	-1,5	0,0668	0,1295
12	-1,25	0,1056	0,1826
13	-1	0,1587	0,2420
14	-0,75	0,2266	0,3011
15	-0,5	0,3085	0,3521
16	-0,25	0,4013	0,3867
17	0	0,5000	0,3989
18	0,25	0,5987	0,3867
19	0,75	0,7734	0,3011
20	1	0,8413	0,2420
21	1,25	0,8944	0,1826
22	1,5	0,9332	0,1295
23	1,75	0,9599	0,0863
24	2	0,9772	0,0540
25	2,25	0,9878	0,0317
26	2,5	0,9938	0,0175
27	2,75	0,9970	0,0091
28	3	0,9987	0,0044

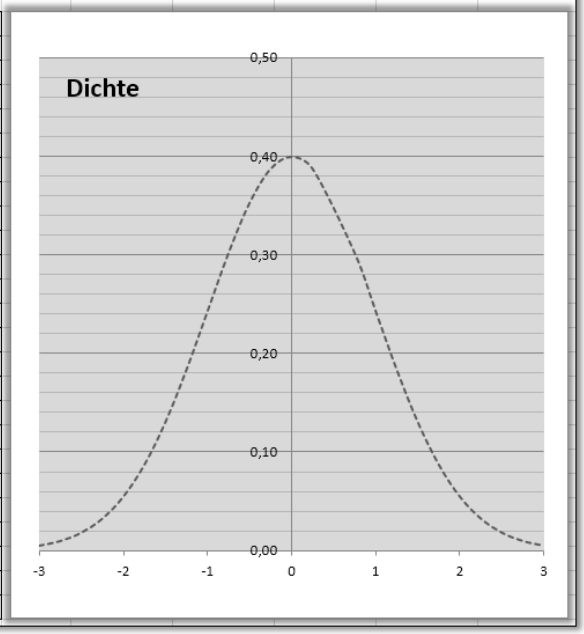


Abbildung 8.27 Die Standardnormalverteilung und der Glocken-Graph

Ist `z` nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück.

Die Standardnormalverteilung ist eine Variante der Normalverteilung und dadurch gekennzeichnet, dass der Mittelwert (Erwartungswert) gleich `0` ist und die Standardabweichung gleich `1`. Die von dieser Funktion ermittelten Werte lassen sich auch über `=NORMVERT(z; 0; 1; WAHR)` berechnen. Um die Dichtefunktion zu berechnen, wird im abgebildeten Beispiel mit `=NORMVERT(z;0;1;FALSCH)` gearbeitet.

Seit Excel 2010 wird anstelle dieser Funktion die Funktion `NORM.S.VERT()` angeboten.

TINV()

TINV()

Syntax: `TINV(`**`Wahrsch`**`; `**`Freiheitsgrade`**`)`

Beispiel: `=TINV(0,05;4)`

Ergebnis: `2,776`

Die Funktion TINV() liefert den t-Wert der t-Verteilung und ist damit die Umkehrung von TVERT() mit dem Parameter 2 für Seiten. Die wiedergegebenen Werte sind in statistischen Tabellenwerken als t-Wert für zweiseitige Tests (Tests, bei denen die Werte nach beiden Seiten abweichen können) tabelliert. Mit Wahrsch wird die zur t-Verteilung gehörige zweiseitige Wahrscheinlichkeit angegeben, der Wert für Freiheitsgrade ergibt sich aus der Gesamtzahl der Stichprobenelemente – 2.

Der prinzipielle Ablauf des t-Tests umfasst folgende Schritte:

1. Aus den zu vergleichenden Größen wird ein rechnerischer t-Wert ermittelt (im Folgenden tr).
2. Die Freiheitsgrade (im Folgenden df) werden ermittelt.
3. Der errechnete tr-Wert wird mit dem von TINV() gelieferten verglichen. Soll der Test einseitig sein, muss für die Funktion das Maß der Wahrscheinlichkeit halbiert werden.

Benötigt wird der von TINV() gelieferte Wert u. a. bei folgenden Tests:

8.1.1 Vergleich der Mittelwerte von Stichprobe und Grundgesamtheit

```
tr = WURZEL(n) * ABS(Ms-Mg)/Ss
df = n-1
```

mit n = Stichprobengröße; Ms = Mittelwert Stichprobe; Mg = Mittelwert Grund-gesamtheit; Ss = Standardabweichung Stichprobe.

8.1.2 Vergleich der Mittelwerte zweier Stichproben

```
tr = (M1-M2)/Sg
Sg^2 = ((n1-1)*S1^2 + (n2-1)*S2^2) * (n1+n2) /((n1+n2-2)*(n1*n2))
df = n1 + n2 - 2
```

mit M1 und M2 für die Mittelwerte der beiden Stichproben, S1 und S2 für die Standardabweichungen, n1 und n2 für die Stichprobengrößen. Ist der so errechnete tr-Wert kleiner als der von TINV() gelieferte, kann davon ausgegangen werden, dass die Unterschiede zwischen den zu testenden Größen zufällig sind. Die Wahrscheinlichkeit, dass diese Annahme falsch ist, wird mit dem Argument Wahrsch angegeben.

Ist eines der Argumente nicht numerisch, gibt die Funktion den Fehler `#WERT!` zurück. Ist `Wahrsch` ≤ `0` oder `Wahrsch` > `1` oder `Freiheitsgrade` < `1`, erscheint der Fehler `#ZAHL!`.

Abbildung 8.28 zeigt einen t-Test für zwei Stichproben aus Untersuchungen zur Knochendichte, mit dem geprüft wird, ob die Unterschiede als signifikant oder nur als zufällig einzustufen sind. Der mit `TTEST()` errechnete Prüfwert ist deutlich kleiner als der mit `TINV()` errechnete kritische t-Wert, also kann davon ausgegangen werden, dass die Unterschiede der beiden Stichproben nicht signifikant sind.

	A	B	C	D	E	F
1						
2	**Berechnen des kritischen Werts der t-Verteilung**					
3						
4	Stichprobe 1	Knochendichte in mg/cm³		Stichprobe 2	Knochendichte in mg/cm³	
5	p1	19,4		p1	19,2	
6	p2	23		p2	19,3	
7	p3	29,9		p3	61,9	
8	p4	33		p4	69,9	
9	p5	67,9		p5	71	
10	p6	68		p6	78	
11	p7	81		p7	96	
12	p8	178		p8	101,4	
13	p9	55		MITTELWERT()	64,5875	
14	p10	34		STABW()	30,9593022	
15	MITTELWERT()	58,92		ANZAHL()	8	
16	STABW()	46,94336541				
17	ANZAHL()	10		TTEST()	0,762707427	
18				TVERT()	0,05	
19				Freiheitsgrade	16	
20				Wahrsch	0,05	
21				TINV()	2,119905299	t-Wert

Abbildung 8.28 Berechnen des Quantils der t-Verteilung

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `T.INV()` und `T.INV.2S()` zur Verfügung, wobei `TINV()` dasselbe Ergebnis liefert wie `T.INV.2S()`.

TTEST()

TTEST()

Syntax: `TTEST(`**Matrix1**`;` **Matrix2**`;` **Seiten**`;` **Typ**`)`

Beispiel: `=TTEST({12;19;13;14;17};{15;17;16;15;17};2;2)`

Ergebnis: `0,489`

Die Funktion TTEST() gestattet den direkten Vergleich zweier Stichproben in Bezug auf den Mittelwert der entsprechenden Grundgesamtheiten, ohne dass so viele rechnerische Zwischenschritte nötig wären wie bei dem unter TINV() geschilderten Verfahren. Seit Excel 2010 wird für TTEST() die umbenannte Funktion T.TEST() angeboten.

Die beiden Stichproben werden mit Matrix1 und Matrix2 angegeben. Mit Seiten wird vorgegeben, ob Abweichungen nach beiden Seiten (2) oder nur nach einer Seite (1) möglich sind. Mit Typ wird der Charakter der Stichproben angegeben:

Typ	Charakter
1	gepaart – zwei Stichproben gleicher Größe
2	zwei Stichproben mit gleicher Varianz
3	zwei Stichproben mit unterschiedlicher Varianz

Tabelle 8.2 Typ-Codes und ihre Bedeutung

Siehe auch die Abschnitte in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, zu den t-Test-Tools.

TVERT()

TDIST()

Syntax: TVERT(**x**; **Freiheitsgrade**; **Seiten**)
Beispiel: =TVERT(2;4;1)
Ergebnis: 0,058

Die Funktion TVERT() liefert die Wahrscheinlichkeit für eine t-verteilte Zufallsvariable. Das Argument x ist das Quantil der Verteilung, dessen Wahrscheinlichkeit berechnet werden soll. Mit Freiheitsgrade wird die Anzahl der Freiheitsgrade angegeben. Das Argument Seiten gibt mit den möglichen Werten 1 und 2 an, ob die Funktion Werte für einen einseitigen oder einen zweiseitigen t-Test liefern soll. Siehe dazu TTEST().

TVERT() ist die Umkehrung zu TINV(). Wenn:

```
x = TINV(Wahrsch;...)
```

dann gilt:

```
Wahrsch = TVERT(x;...;2)
```

Ein Beispiel für die Anwendung ist der Vergleich der Häufigkeit eines Merkmals in einer Stichprobe mit der Wahrscheinlichkeit dieses Merkmals in der Grundgesamtheit. Die Testgröße `t` ist

```
t = ABS(z-n*p)/WURZEL(n*p*(1-p))
```

mit `z` = Häufigkeit des Merkmals in der Stichprobe, `p` = Wahrscheinlichkeit in der Grundgesamtheit und `n` = Größe der Stichprobe. Die Zahl der Freiheitsgrade beträgt `df` = n – 1.

Setzen Sie diese beiden Größen (`t` und `df`) in die Funktion ein, dann erhalten Sie direkt die Wahrscheinlichkeit dafür, dass der Unterschied zwischen Stichprobe und Grundgesamtheit zufällig ist.

Enthalten die mit `Matrix1` und `Matrix2` angegebenen Datenreihen nicht numerische Elemente, werden sie ignoriert. Sind aber jeweils weniger als zwei numerische Elemente vorhanden, gibt die Funktion den Fehler `#DIV/0!` zurück. Ist `Typ` = `1` (gepaart), müssen beide Datenreihen die gleiche Anzahl von Elementen enthalten, sonst liefert die Funktion den Fehler `#NV`. Ist `Seiten` oder `Typ` nicht numerisch, erscheint der Fehler `#WERT!`. Bei ungültigen Werten für `Seiten` (<> `1` oder `2`) oder `Typ` (<> `1`, `2` oder `3`) erscheint der Fehler `#ZAHL!`.

Seit Excel 2010 stehen anstelle dieser Funktion die Funktionen `T.VERT()`, `T.VERT.2S()` und `T.VERT.RE()` zur Verfügung, wobei `TVERT()` mit dem Parameter `Seiten` = `1` dasselbe Ergebnis liefert wie `T.VERT.RE()` und mit dem Parameter `Seiten` = `2` dasselbe Ergebnis wie `T.VERT.2S()`.

Siehe auch die Abschnitte in Kapitel 17, »Zusätzliche Tools für die Datenanalyse«, zu den t-Test-Tools.

UNTERGRENZE()

FLOOR()

Syntax:	`UNTERGRENZE(`**`Zahl`**`;` **`Schritt`**`)`
Beispiel:	`=UNTERGRENZE(3,085;0,1)`
Ergebnis:	`3`

Die Funktion `UBTERGRENZE()` rundet den mit dem Argument `Zahl` angegebenen Wert auf das nächste Vielfache von `Schritt` ab und ist damit komplementär zu `OBERGRENZE()`. Sie erlaubt also die Abrundung auf bestimmte Intervallgrenzen.

Dadurch ist es möglich, Kalkulationsergebnisse so abzurunden, dass nicht nur der Wert der letzten Stelle, die angegeben wurde, gerundet wird. Mit dem Wert `0,05` für `Schritt` kann z.B. bestimmt werden, dass die Hundertstelstelle beim Abrunden immer nur eine 5 oder eine 0 sein kann. Mit einem Wert `0,5` für `Schritt` wird z.B. dafür gesorgt, dass nicht mehr in Cent, sondern nur noch für 5-Cent-Stücke ausgepreist wird.

Aufrunden meint im Sinne dieser Funktion, dass immer zur Null hin gerundet wird, z.B. wird `=UNTERGRENZE(-2,54542;0,05)` gerundet zu `-2,5`. Bei unterschiedlichen Vorzeichen für `Zahl` und `Schritt` wird eine Fehlermeldung ausgegeben.

Ist eines der Argumente nicht numerisch, liefert die Funktion den Fehler `#WERT!`.

	A	B	C	D
1				
2	**Abrunden auf das kleinste Vielfache**			
3				
4	Wert	Schritt	UNTERGRENZE()	
5	14,25	0,5	14	
6	14,33	1	14	
7	123,56	2	122	
8	124,33	5	120	
9	126,33	5	125	
10	124,33	10	120	
11	-112	-10	-110	
12	-112	10	-120	
13				

Abbildung 8.29 Abrunden auf das kleinste Vielfache des als `Schritt` angegebenen `Werts`

VARIANZ()

VAR()

Syntax: `VARIANZ(`**`Zahl1`**`; Zahl2; ...)`
Beispiel: `=VARIANZ(33;22;28;17;23;26)`
Ergebnis: `30,17`

Die Funktion `VARIANZ()` schätzt für die Werte in der Argumentliste die vermutete Varianz der Grundgesamtheit. Dabei werden die in der Argumentliste gegebenen Daten als Stichprobe aus dieser Grundgesamtheit genommen. Die Argumentliste kann seit Excel 2007 bis zu 255 Werte enthalten, in den älteren Versionen bis zu 30. Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte – `1`. Die Standardabweichung ist wie-

derum nichts anderes als die Wurzel der Varianz, womit dann wieder eine Größenordnung auf der Ebene der vorhandenen Abweichungen erreicht wird.

	A	B	C
1			
2	**Berechnen der Varianz**		
3			
4	Person	Alter	Gewicht
5	1	23	69
6	2	25	66
7	3	29	68
8	4	33	72
9	5	35	74
10	6	35	63
11	7	43	68
12	8	43	79
13	9	43	58
14	10	43	79
15	11	45	68
16	12	46	72
17	13	47	77
18	14	47	56
19	15	51	60
20	16	53	62
21	17	53	58
22	18	54	61
23	19	55	55
24	20	63	57
25	VARIANZ()	111,6	59,6
26	VARIANZEN()	106,0	56,6
27	MITTELWERT()	43,3	66,1

Abbildung 8.30 Berechnen der Varianz

Seit Excel 2010 wird anstelle dieser Funktion die umbenannte Funktion `VAR.S()` angeboten.

VARIANZEN()

VARP()

Syntax: `VARIANZEN(`**Zahl1**`; Zahl2; ...)`

Beispiel: `=VARIANZEN(33;22;28;17;23;26)`

Ergebnis: `25,14`

Die Funktion VARIANZEN() berechnet für die Werte in der Argumentliste die Varianz. Dabei werden die gegebenen Daten als Grundgesamtheit genommen. Die Argumentliste kann seit Excel 2007 bis zu 255 Werte enthalten, in den älteren Versionen bis zu 30. Die Funktion ermittelt die Differenz der einzelnen Werte zum arithmetischen Mittelwert, quadriert diese und teilt das Ergebnis durch die Anzahl der Werte. Seit Excel 2010 wird anstelle dieser Funktion die umbenannte Funktion VAR.P() angeboten.

Mehr zur Varianz finden Sie in Abschnitt 7.3, »Zufallsvariable und Wahrscheinlichkeit«.

WEIBULL()

WEIBULL()

Syntax: WEIBULL(**x**; **Alpha**; **Beta**; **Kumuliert**)

Beispiel: =WEIBULL(20000;0,25;70000;WAHR)

Ergebnis: 0,5186

Die Funktion WEIBULL() liefert Wahrscheinlichkeiten für eine Zufallsvariable, die einer Weibull-Verteilung gehorcht. Diese Verteilung wird beispielsweise für Haltbarkeitsstatistiken im Bereich der Qualitätssicherung benutzt.

Das Quantil, für das die Funktion ausgewertet werden soll, wird mit x angegeben, Alpha ist ein Skalenparameter, Beta ein Form- oder Gestaltparameter der Verteilung, der die Ausfallrate bestimmt. Mit Kumuliert lässt sich festlegen, ob die Dichtefunktion (FALSCH) oder die Verteilungsfunktion (WAHR) ausgegeben wird. Seit Excel 2010 wird anstelle dieser Funktion die umbenannte Funktion WEIBULL.VERT() angeboten.

9 Nachschlage- und Verweisfunktionen

Funktion	Seite	Funktion	Seite
ADRESSE()	668	SPALTE()	684
BEREICH.VERSCHIEBEN()	670	SPALTEN()	686
BEREICHE()	673	SVERWEIS()	687
FORMELTEXT()	675	VERGLEICH()	688
HYPERLINK()	676	VERWEIS()	690
INDEX()	678	WAHL()	691
INDIREKT()	680	WVERWEIS()	694
MTRANS()	681	ZEILE()	695
PIVOTDATENZUORDNEN()	682	ZEILEN()	696
RTD()	684		

9.1 Einsatzbereiche für Nachschlage- und Verweisfunktionen

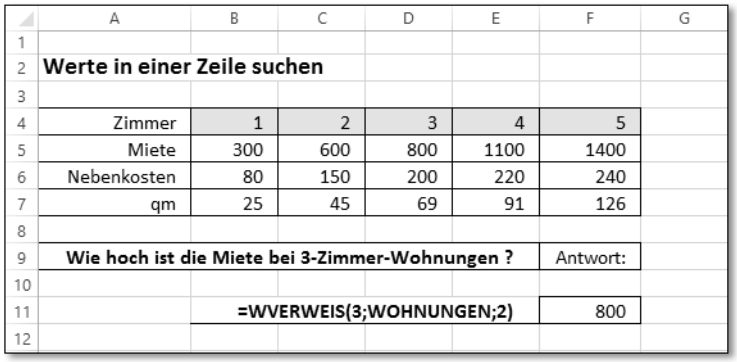
Die unter dieser Rubrik aufgeführten Funktionen dienen in erster Linie dem Durchsuchen von Bereichen nach bestimmten Werten. Meist werden diese Werte innerhalb von Berechnungen benötigt, die auf diese Werte angewiesen sind. Die Funktionen tauchen deshalb häufig innerhalb komplexerer Formeln auf, um den Wert eines bestimmten Arguments darin zu gewinnen. Neben ihrer Anwendung in Tabellen werden diese Funktionen häufig auch innerhalb von Makros verwendet, um gezielt bestimmte Werte darin abzugreifen.

Außerdem finden Sie in dieser Kategorie eine Reihe von Funktionen zur Behandlung von Bezügen, zur Ermittlung der Adressen von Zellen oder der Abfrage der Größe von Bereichen. Neu in Excel 2013 ist nur die Funktion `FORMELTEXT()`, die Ihnen jetzt endlich eine bequeme Methode anbietet, um die Formeln einer Zelle innerhalb des Tabellenblatts zu dokumentieren.

9.2 Arbeiten mit Verweisfunktionen

Zu den Funktionen, die in ganz unterschiedlichen Situationen von großem Nutzen sind, gehören die Verweisfunktionen `VERWEIS()`, `WVERWEIS()` und `SVERWEIS()`. Sie werden im-

mer dann benutzt, wenn aus einer vorhandenen Tabelle, deren erste Zeile oder Spalte in aufsteigender Reihenfolge sortiert ist, gezielt Informationen extrahiert werden sollen. Typische Beispiele für solche Tabellen sind Steuertabellen, Inventarlisten, Verzeichnisse oder Kataloge.



	A	B	C	D	E	F	G
1							
2	**Werte in einer Zeile suchen**						
3							
4	Zimmer	1	2	3	4	5	
5	Miete	300	600	800	1100	1400	
6	Nebenkosten	80	150	200	220	240	
7	qm	25	45	69	91	126	
8							
9	**Wie hoch ist die Miete bei 3-Zimmer-Wohnungen ?**					Antwort:	
10							
11		**=WVERWEIS(3;WOHNUNGEN;2)**				800	
12							

Abbildung 9.1 Waagerechte Verweistabelle

Abbildung 9.1 zeigt ein Beispiel für eine horizontale Verweistabelle mit der Funktion `WVERWEIS()`. Dort sind durchschnittliche Werte für Wohnungen aufgelistet. In der ersten Zeile ist die Zimmeranzahl in aufsteigender Reihenfolge eingetragen. Für jede Zimmeranzahl sind nun in den nachfolgenden Spalten bestimmte Daten aufgeführt. Das heißt, die Daten von Zelle B5 bis B7 gehören zu den Einzimmerwohnungen, die Daten in den Zellen C5 bis C7 gehören zu den Zweizimmerwohnungen usw.

Der gesamte Datenbereich der Tabelle – B4 bis F7 – ist in diesem Beispiel mit dem Namen `Wohnungen` bezeichnet. Möchten Sie nun die Miete für die Dreizimmerwohnungen abgreifen, lautet die Formel:

```
=WVERWEIS(3; Wohnungen; 2)
```

Die Funktion verwendet drei Argumente:

1. das `Suchkriterium`, im Beispiel die Anzahl der Zimmer
2. die `Matrix` mit den zu durchsuchenden Daten, hier der Bereich `Wohnungen`
3. der `Zeilenindex`, im Beispiel 2 für 2. Zeile

`Zeilenindex` gibt an, die wievielte Zeile in `Matrix` den Ergebniswert liefern soll. Der hier angegebene Wert hängt davon ab, welche Information jeweils gewünscht wird. Interessieren Sie sich für die Nebenkosten, wäre der anzugebende Index `= 3`.

Die Funktion geht also mit dem angegebenen Suchkriterium in die erste Zeile des als Matrix angegebenen Bereichs, sucht von links nach rechts nach einem Wert, der zum Suchkriterium passt. Ist die entsprechende Spalte gefunden, wandert der Zellzeiger zu der Zeile abwärts, deren Index angegeben wurde. Die Funktion liefert schließlich den dort gefundenen Zellinhalt als ihr Ergebnis. Dieser Inhalt kann ein numerischer Wert, aber auch ein Text sein. Gibt es keinen mit dem Suchkriterium identischen Wert in der Zeile, wird der Wert genommen, der vor dem nächsthöheren liegt. Wäre das Suchkriterium z. B. 2,5 Zimmer, liefert die Funktion den Wert für 2 Zimmer.

Die Funktion SVERWEIS() wird bei Tabellen benutzt, die senkrecht aufgebaut sind. Typisches Beispiel ist eine Steuertabelle. Dort sind in der ersten Spalte in aufsteigender Reihenfolge die verschiedenen Einkommensstufen aufgeführt. Zu jeder Einkommensstufe finden sich in den Spalten rechts neben der ersten Spalte die Steuerbeträge für die verschiedenen Steuerklassen.

Wer nachschauen will, was er zu zahlen hat, liest zunächst die erste Spalte von oben nach unten, bis er einen Betrag gefunden hat, der seinem Einkommen entspricht. Dann geht er in der entsprechenden Tabellenzeile so viele Spalten nach rechts, bis er in der Spalte angekommen ist, die seiner Steuerklasse entspricht. Nichts anderes macht die SVERWEIS()-Funktion.

9.3 Einsatz der INDEX()-Funktion

Auch die Funktion INDEX() dient dazu, bestimmte Werte aus einer Tabelle abzugreifen. Excel bietet die INDEX()-Funktion in zwei Versionen an. Die Bezugsversion mit der Syntax =INDEX(Bezug; Zeile; Spalte; Bereich) dient der Abfrage von Datentabellen, die Matrixversion mit der Syntax =INDEX(Matrix; Zeile; Spalte) ist speziell für die Abfrage der Werte von Matrizen ausgelegt.

Die INDEX()-Funktion für Datentabellen beantwortet die Frage, was in der Zelle eingetragen ist, die im Schnittpunkt der Zeile n und der Spalte m des mit Bezug angegebenen Bereichs liegt. Das letzte Argument Bereich kann angegeben werden, wenn Bezug eine Mehrfachauswahl enthält. Sind also z. B. drei Zellblöcke in der Mehrfachauswahl enthalten, bedeutet eine 2 für Bereich, dass sich die Zeilen- und Spaltennummern auf den zweiten Zellblock beziehen. Wird Bereich nicht angegeben, wird immer der erste Block genommen.

Im Beispiel, das durch Abbildung 9.2 verdeutlicht wird, ist ein Bereich für die Produktionsergebnisse mehrerer Werke der letzten fünf Jahre mit dem Namen Produktion de-

finiert. Die Zelle, die in der dritten Zeile und der vierten Spalte dieses Bereichs liegt, enthält den Betrag 1.300.000 €.

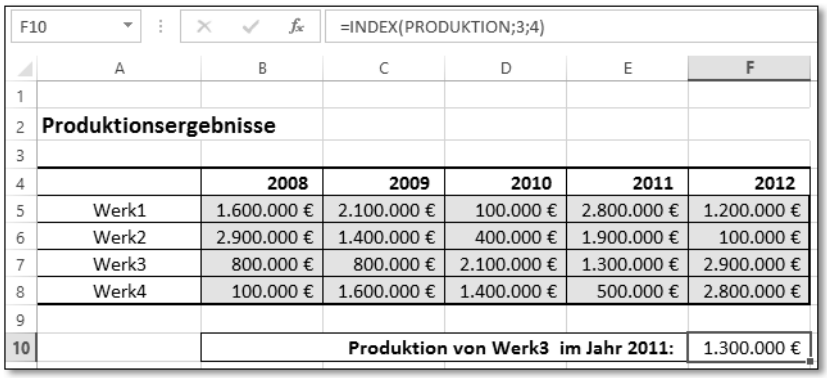


Abbildung 9.2 Ein Beispiel für die Funktion `INDEX()`

Diese `INDEX()`-Funktion arbeitet also nicht mit der Zelladresse, sondern mit der relativen Position einer Zelle in einem definierten Bereich. Die Spalten- oder Zeilennummer kann natürlich auch selbst wiederum das Ergebnis einer Formel sein.

Die `INDEX()`-Funktion wird auch benutzt, um Werte aus einer Matrix auszulesen. Dabei kann es sich um einen einzelnen Wert, aber auch um eine Matrix handeln. Es ist möglich, das Argument `Matrix` als Bereichsbezug oder als Matrixkonstante einzugeben.

Um nicht nur einen einzelnen Wert auszulesen, sondern eine Matrix, wird der Wert für Zeile und/oder Spalte auf null gesetzt oder weggelassen.

```
=INDEX({3.5.9.7;34.54.23.98};0;0)
=INDEX({3.5.9.7;34.54.23.98};;)
```

liefern als Ergebnis die gesamte Matrix.

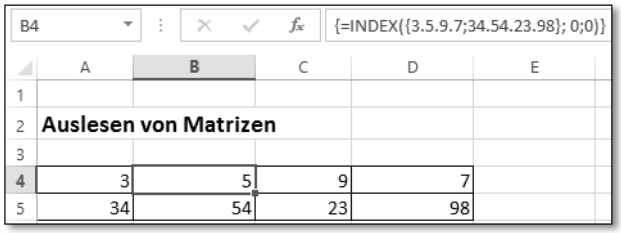


Abbildung 9.3 Werte einer Matrixkonstanten auslesen

```
=INDEX({3.5.9.7;34.54.23.98};2)
```

ergibt eine Matrix mit den Werten der zweiten Zeile.

```
=INDEX({3.5.9.7;34,54.23.98};;2)
```

ergibt eine Matrix mit den Werten der zweiten Spalte.

Soll das Ergebnis der Formel wiederum eine Matrix sein, muss wie bei der Eingabe von Matrixformeln verfahren werden. Sie markieren also zunächst einen entsprechend großen Bereich für die Ergebnismatrix und geben in einer beliebigen Zelle die INDEX()-Funktion ein. Die Formel muss mit Strg + ⇧ + ↵ abgeschlossen werden.

9.4 Einsatz der WAHL()-Funktion

Die einfachste Nachschlagefunktion ist wohl die Funktion WAHL(). Sie liefert einen beliebigen Wert aus einer gegebenen Liste von Werten. Die Syntax ist:


```
=WAHL(Index; Wertl; Wert2; ... ; Wertn)
```

Das erste Argument der Funktion gibt an, das wievielte Element der Liste gesucht wird. Index kann als Zahl, Bezug oder Formel eingegeben werden. Als Werte können Zahlen, Zellbezüge, Namen, Texte und Formeln verwendet werden. Bereichsbezüge sind dagegen nicht erlaubt. (Um Werte aus Bereichen zu ziehen, wird die Funktion INDEX() verwendet.)

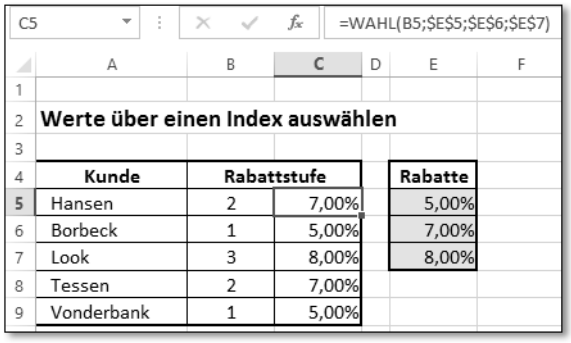


Abbildung 9.4 Beispiel für die WAHL()-Funktion

Abbildung 9.4 zeigt ein praktisches Beispiel für die Nutzung der WAHL()-Funktion: Kunden sind in verschiedene Rabattstufen eingeteilt. Die verschiedenen Rabatte werden in der Spalte E aufgelistet. Der Kunde mit der Rabattstufe 1 soll den ersten Rabatt aus dieser Liste erhalten, der Kunde mit Stufe 2 den zweiten etc. Die Rabattstufen sind in der Spalte B ab der Zelle B5 eingetragen.

Die Formel wird zunächst in der Zelle C5 entwickelt. Sie lautet:

```
=WAHL(B5;$E$5;$E$6;$E$7)
```

Das Ergebnis ist ohne weitere Probleme kopierbar, da die Adressen der Rabatte absolut sind. Anstelle der Zellverweise in der Argumentliste könnten die Prozentsätze auch direkt als Konstanten in die Formel eingetragen werden, also:

```
=WAHL(C9;5% ;7% ;8%)
```

Dadurch ginge aber einiges an Flexibilität verloren. Änderte sich der Rabattsatz, müssten auch die Formeln geändert werden.

Besonders interessant ist die WAHL()-Funktion auch für Berechnungsalternativen. In diesem Fall können verschiedene Formeln als Argumente benutzt werden. Hier ein Beispiel:

```
=WAHL(D1; SUMME(POSTEN)*0,70; SUMME(POSTEN)*0,75; SUMME(POSTEN)*0, 80+PORTO)
```

Abhängig von dem in D1 abgestellten Wert wird die Summe der Posten mit einem anderen Faktor multipliziert. Im dritten Fall werden noch Portokosten addiert. Während die WENN()-Funktion nur eine einfache Verzweigung zwischen zwei Möglichkeiten zulässt – es sei denn, die Formel arbeitet mit verschachtelten WENN()-Funktionen –, erlaubt die WAHL()-Funktion seit Excel 2007 die Auswahl aus bis zu 254 Möglichkeiten, in den älteren Versionen sind es 29.

9.5 Referenz der Nachschlage- und Verweisfunktionen

ADRESSE()

ADDRESS()

Syntax: ADRESSE(**Zeile**; **Spalte**; Abs; A1; Tabellenname)
Beispiel: =ADRESSE(2;5;4;FALSCH)
Ergebnis: Z(2)S(5)

Die Funktion ADRESSE() liefert die Adresse der mit den Argumenten Zeile und Spalte angegebenen Zelle. Mit dem optionalen Argument Abs können Sie den Bezugstyp angeben:

1 oder keine Angabe: absoluter Bezug

2 absolute Zeile, relative Spalte

3 relative Zeile, absolute Spalte

4 relativer Bezug

Das Argument `A1` ist ein Wahrheitswert und bestimmt die Schreibweise des gewünschten Bezugs. Wenn `A1` den Wert `1` oder `WAHR` hat oder ausgelassen wird, werden die Bezüge in der üblichen A1-Schreibweise zurückgegeben. Wenn `A1` `FALSCH` ist, werden die Bezüge in der alternativen Z1S1-Schreibweise zurückgegeben.

Bei dieser Schreibweise werden statt der üblichen Spalten- und Zeilenbeschriftungen die Nummern der Zeilen bzw. Spalten angegeben. Sie können diese Schreibweise aktivieren, wenn Sie über **Datei • Optionen** auf der Seite **Formeln** die Option **Z1S1-Bezugsart** aktivieren. Diese Schreibweise wird häufiger innerhalb von Makros genutzt, weil sie die Berechnung von Zelladressen vereinfacht. Das Ergebnis der Funktion ist aber unabhängig davon, ob diese Option eingeschaltet ist oder nicht.

`Tabellenname` bestimmt den Tabellenblattnamen und wird zur Abtrennung von den Zelladressen durch ein Ausrufezeichen ergänzt. Fehlt `Tabellenname`, wird die Adresse als lokale Blattadresse behandelt.

	A	B	C	D	E	F
1						
2	**ADRESSE**					
3						
4	Zeile	Spalte	Abs	A1	Tabellenname	ADRESSE()
5	2	1	1	WAHR	Matrixfunktionen	Matrixfunktionen!$A$2
6	2	1	2	WAHR	Matrixfunktionen	Matrixfunktionen!A$2
7	2	1	3	WAHR	Matrixfunktionen	Matrixfunktionen!$A2
8	2	1	4	WAHR	[Matrixfunktionen]BEREICHE	[Matrixfunktionen]BEREICHE!A2
9	2	1	4	FALSCH	[Matrixfunktionen]BEREICHE	[Matrixfunktionen]BEREICHE!Z(2)S(1)

Abbildung 9.5 Beispiele für Adressen

Die Argumente `Zeile` und `Spalte` müssen numerische Werte ergeben, sonst liefert die Funktion den Fehler `#WERT!`. Das gilt auch, wenn die Werte die maximale Anzahl von Zeilen – `1048576` – oder Spalten – `16384` – überschreiten.

Die Funktion liefert als Ergebnis eine Zeichenfolge, die wie eine gültige Adresse aussieht. Ob ein angegebenes Tabellenblatt existiert, wird nicht geprüft. Um mit diesem Ergebnis etwas anfangen zu können, kann die Funktion beispielsweise zum Aufbau von

Zeichenfolgen verwendet werden, die von der Funktion INDIREKT() ausgewertet werden, um den Bezug zu der betreffenden Zell- oder Bereichsadresse herzustellen.

```
=INDIREKT(ADRESSE(3;1))
```

liefert den Inhalt der Zelle A3 im aktuellen Blatt.

Anstatt die Werte für Zeile und Spalte als Zahl einzugeben, kann ihr Wert auch durch andere Funktionen errechnet werden. In der Formel

```
=INDIREKT(ADRESSE(ANZAHL(A:A)+4);1)
```

wird mit ANZAHL(A:A) die Zahl der Werte in Spalte A ermittelt. Anschließend wird der Wert 4 als Vorschub für die vorhandenen Überschriftenzeilen addiert. Die Funktion liefert dann den letzten Wert in der Spalte.

	A	B	C	D
1				
2	**ADRESSE() indirekt verwenden**			
3				
4	Werte		**=INDIREKT(ADRESSE(ANZAHL(A:A)+4;1))**	
5	124			
6	125		129	
7	126			
8	127			
9	128			
10	129			

Abbildung 9.6 ADRESSE() in Kombination mit INDIREKT()

BEREICH.VERSCHIEBEN()

OFFSET()

Syntax: BEREICH.VERSCHIEBEN(**Bezug**; **Zeilen**; **Spalten**; Höhe; Breite)
Beispiel: =BEREICH.VERSCHIEBEN($A$5;5;0)
Ergebnis: Wert aus Zelle A10

Die Funktion BEREICH.VERSCHIEBEN() liefert einen Bereichsbezug, der um eine mit den Argumenten Zeilen und Spalten festgelegte Zahl von Zeilen und Spalten gegenüber dem mit Bezug angegebenen Bereich verschoben ist. Als Ausgangspunkt dient dabei die linke obere Eckzelle dieses Bereichs. Bei positiven Werten werden Zeilen und Spalten nach unten bzw. nach rechts versetzt, bei negativen in die umgekehrte Richtung.

`Höhe` und `Breite` sind optionale Argumente und bestimmen die Größe des neuen Bezugs durch die Anzahl der Zeilen und Spalten. Wenn die Zeilen- und Spaltenanzahl mit dem ursprünglichen Bezug übereinstimmen, müssen für diese Argumente keine Werte eingetragen werden.

Das Argument `Bezug` muss einen gültigen Bezug ergeben. Die anderen Argumente müssen numerische Werte ergeben, die die Grenzen der maximalen Zeilenzahl – `1048576` – und der maximalen Spaltenzahl – `16384` – nicht überschreiten, ansonsten liefert die Funktion den Fehler `#WERT!`.

In dem Beispiel in Abbildung 9.7 werden in den Zellen F5 bis F8 Werte aus dem Zellblock A10:D14 ausgegeben. In der Zelle F10 wird eine Summenfunktion verwendet, die die Funktion `BEREICH.VERSCHIEBEN()` für die Angabe des zu summierenden Datenbereichs benutzt. Die ersten Argumente bestimmen die Ausgangszelle C10, die folgenden Argumente für `Höhe` und `Breite` legen dann die Größe des Bereichs fest, aus dem Daten in die Summe einfließen sollen – C10 bis D11.

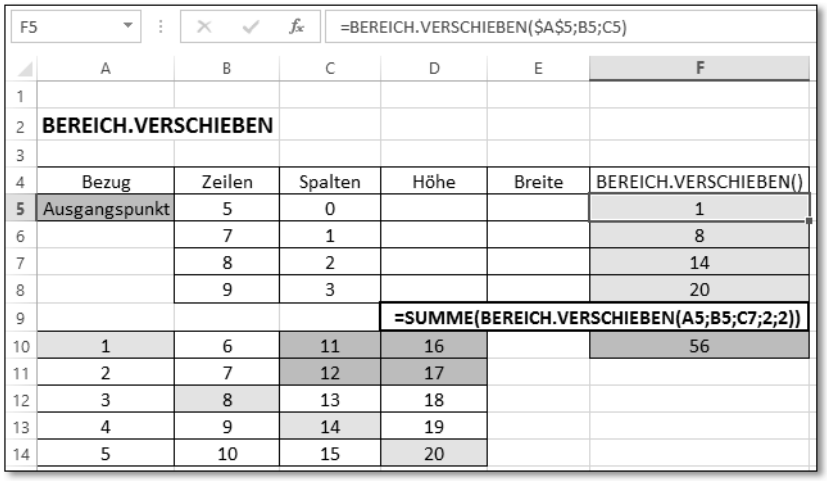


Abbildung 9.7 Versetzte Bezüge

9.5.1 Dynamische Bereiche

Die Funktion `BEREICH.VERSCHIEBEN()` kann unter anderem dazu verwendet werden, dynamische Bereiche zu erzeugen. Die Notwendigkeit dynamischer Bereiche ergibt sich immer dann, wenn sich Formeln auf Bereiche beziehen, deren Abgrenzung sich im Laufe der Zeit ändert.

Wird beispielsweise eine SVERWEIS()-Funktion benutzt, um bestimmte Werte aus einer Artikelliste abzufragen, kommt es natürlich darauf an, dass die Bereichsdefinition der Liste aktuell ist. Wird die Liste erweitert, ohne dass die Bereichsdefinition angepasst wird, kommt es vor, dass die SVERWEIS()-Funktion einen Artikel nicht findet, der nachträglich angehängt worden ist. Um dies zu vermeiden, kann die Funktion BEREICH.VERSCHIEBEN() genutzt werden.

	A	B	C	D	E
1	Dynamische Bereiche definieren				BEREICH.VERSCHIEBEN()
2	Artikelnr	Bezeichnung	Preis		=BEREICH.VERSCHIEBEN($A$3;;;ANZAHL($A:$A);3)
3	1001	Bohrer A	12,00 €		#WERT!
4	1002	Schleifer A	23,00 €		
5	1003	Bohrschrauber A	17,00 €		SVERWEIS()
6	1004	Bohrer B	16,00 €		=SVERWEIS(1004;Artikelliste;2;WAHR)
7	1005	Schleifer B	13,00 €		Bohrschrauber A
8	1006	Bohrschrauber B	23,00 €		
9	1007	Bohrer C	12,00 €		

Abbildung 9.8 Mit BEREICH.VERSCHIEBEN() lassen sich Bereichsdefinitionen dynamisieren.

In dem Beispiel in Abbildung 9.8 wird zunächst in Zelle E3 die Formel

```
=BEREICH.VERSCHIEBEN($A$3;;;ANZAHL($A:$A);3)
```

abgelegt. Das erste Argument gibt an, dass der Ausgangsbereich bei A3 beginnt. Die beiden folgenden Argumente bleiben leer oder enthalten den Wert 0, weil der Bereich nicht verschoben, sondern nur dynamisch erweitert werden soll. Die Erweiterung findet in diesem Fall nach unten statt, deshalb wird für die Angabe des Arguments Höhe die Funktion ANZAHL() herangezogen, die prüft, wie viele Artikelnummern in der Spalte A aktuell eingetragen sind. Die Lösung muss natürlich voraussetzen, dass, wenn in die Spalte A eine neue Artikelnummer eingetragen wird, auch die übrigen Daten für die Zeile eingetragen werden. Der Wert für Breite kann hier auf die Anzahl der verwendeten Spalten fixiert werden.

Ist die Formel angelegt, wird sie in die Zwischenablage kopiert und anschließend in dem Dialog für die Definition eines neuen Bereichsnamens Artikelliste in das Feld **Bezieht sich auf:** kopiert. Dieser Bereichsname kann dann in einer SVERWEIS()-Funktion verwendet werden, um etwa zu einer Artikelnummer die jeweilige Bezeichnung oder den Preis zu suchen:

```
=SVERWEIS(1004;Artikelliste;2;WAHR)
```

Elegant ist diese Lösung nicht, insbesondere bei der Übernahme von externen Daten ist sie aber in vielen Fällen eine unverzichtbare Lösung. Werden die Daten dagegen direkt in das Tabellenblatt eingegeben, lässt sich das Verfahren vermeiden. Es muss dazu nur

sichergestellt werden, dass ein einmal definierter Tabellenbereich immer auf eine Weise um weitere Daten ergänzt wird, bei der die Bereichsdefinition automatisch mitwächst.

In den älteren Excel-Funktionen war dazu eine gewisse Disziplin beim Einfügen notwendig, mit den neuen Tabellenbereichen seit Excel 2007 wird das Verfahren nun aber deutlich einfacher. Wird an einen solchen Bereich eine Zeile angehängt, wird automatisch eine Bereichserweiterung über die AutoFormat-Funktion vorgenommen, jedenfalls solange die Option **Neue Zeilen und Spalten in die Tabelle einschließen** im Dialog **AutoKorrektur** nicht abgewählt wird. Diese erreichen Sie über **Datei • Optionen • Dokumentprüfung**.

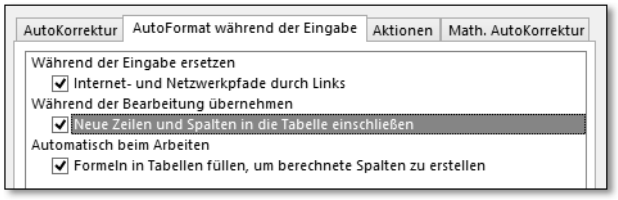


Abbildung 9.9 Optionen für die Erweiterung von Tabellenbereichen im Dialog **AutoKorrektur**

Das Beispiel kann aufgrund dieses Verhaltens nun wesentlich einfacher realisiert werden (Abbildung 9.10):

	A	B	C	D	E
14	**Artikelnr**	**Bezeichnung**	**Preis**		SVERWEIS()
15	1001	Bohrer A	12,00 €		=SVERWEIS(1008;Artikelliste2;2;WAHR)
16	1002	Schleifer A	23,00 €		Schleifer C
17	1003	Bohrschrauber A	17,00 €		
18	1004	Bohrer B	16,00 €		
19	1005	Schleifer B	13,00 €		
20	1006	Bohrschrauber B	23,00 €		
21	1007	Bohrer C	12,00 €		
22	1008	Schleifer C	13,00 €		

Abbildung 9.10 Das Beispiel nutzt die dynamischen Tabellenbereiche.

BEREICHE()

AREAS()

Syntax: `BEREICHE(`**Bezug**`)`

Beispiel: `=BEREICHE((C18:D18;E10:F12))`

Ergebnis: `2`

Die Funktion BEREICHE() ermittelt die Anzahl der Bereiche, die in Bezug angegeben sind. Wird mehr als ein Bereich angegeben, muss ein zusätzliches Klammerpaar wie im Beispiel in Abbildung 9.11 verwendet werden. Bezug kann Zell- und Bereichsadressen oder Bereichsnamen enthalten. Allerdings ist die Funktion nicht in der Lage zu erkennen, wie viele Bereiche beispielsweise in einem größeren Zellbereich als benannte Bereiche vorkommen.

Das Argument Bezug muss einen gültigen Bezug ergeben, sonst liefert die Funktion den Fehler #NAME?.

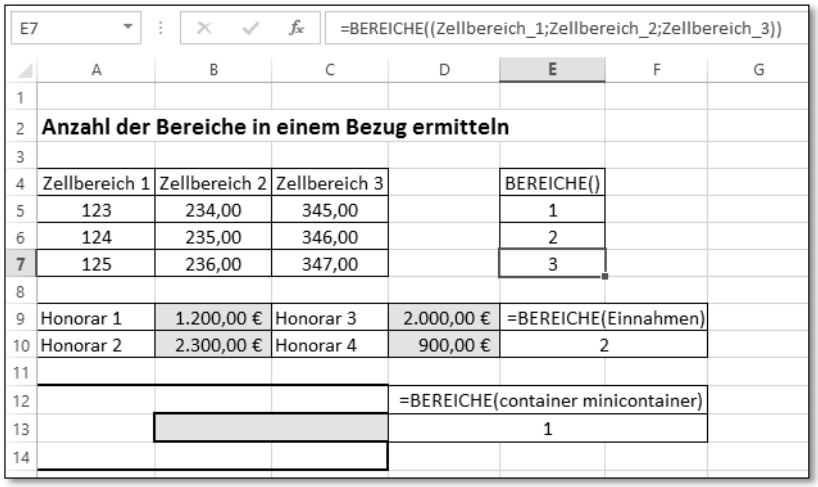


Abbildung 9.11 Zählen der in einem Bezug verwendeten Bereiche

Sinnvolle Anwendungen der Funktion sind eher im Bereich der Makroprogrammierung angesiedelt. Die Funktion kann beispielsweise verwendet werden, wenn Bereichsnamen für separate Zellbereiche vergeben sind. Werden etwa die Bereiche B3:B20 und D3:D20 mit dem Bereichsnamen Einnahmen belegt, liefert

```
=BEREICHE(Einnahmen)
```

das Ergebnis 2.

Soll sichergestellt werden, dass ein verwendeter Bereich, etwa der Standardbereich Druckbereich, tatsächlich ein geschlossener Bereich ist, kann mit

```
=WENN(BEREICHE(Druckbereich)=1;"Anzahl Druckzeilen " &
 ZEILEN(Druckbereich);"Achtung: Separate Druckbereiche!")
```

in einer freien Zelle eine Warnung installiert werden.

Umgekehrt kann auch festgestellt werden, ob Bereiche eine Schnittmenge haben. Dazu wird der Schnittmengenoperator verwendet. Ist der Bereich `minicontainer` wenigstens teilweise oder ganz in dem Bereich `container` angesiedelt, ergibt

```
=BEREICHE(container minicontainer)
```

den Wert `1`, ansonsten den Fehlerwert `#NULL!`.

FORMELTEXT()

FORMULATEXT ()

Syntax: `FORMELTEXT(`**Bezug**`)`

Beispiel: `=FORMELTEXT(B4)`

Die in Excel 2013 neu eingeführte Funktion `FORMELTEXT()` liefert die Formel, die in der mit `Bezug` angegebenen Zelle enthalten ist, als Zeichenfolge zurück, so wie sie sonst in der Bearbeitungsleiste angezeigt wird, wenn die Zelle ausgewählt ist. Verweist `Bezug` auf einen Bereich von mehreren Zellen, wird nur die linke obere Eckzelle ausgewertet.

Es sind auch Bezüge auf andere Arbeitsmappen möglich, falls diese geöffnet sind, ansonsten erscheint der Fehlerwert `#NV`. Denselben Fehlerwert liefert die Funktion, wenn die angegebene Zelle keine Formel enthält, wenn die Formel mehr als 8.192 Zeichen enthält oder wenn das Blatt, das die Formel enthält, geschützt ist. Wird die in `Bezug` angegebene Zelle nachträglich gelöscht, liefert die Formel den Fehler `#BEZUG!`.

Ob die bezogene Zelle eine Formel enthält, kann in Excel 2013 mit der ebenfalls neuen Funktion `ISTFORMEL()` geprüft werden.

	A	B	C	D
1				
2	**Formeln von Zellen anzeigen**			
3				
4	Jahr	Beschäftigte	190	**=VERWEIS(2005;A5:A15;B5:B15)**
5	2000	208	210	**=VERWEIS(2010;A5:B15)**
6	2001	201	WAHR	**=ISTFORMEL(C4)**
7	2002	222		**#NV**
8	2003	223	FALSCH	**=ISTFORMEL(C7)**
9	2004	205		
10	2005	190		
11	2006	187		
12	2007	175		
13	2008	179		
14	2009	200		
15	2010	210		

Abbildung 9.12 Anzeige der Formeln von Zellen

HYPERLINK()

HYPERLINK()

Syntax: `HYPERLINK(Hyperlink_Adresse; Freundlicher_Name)`
Beispiel: `=HYPERLINK(http://www.galileo-press.de;"Galileo")`

Die Funktion `HYPERLINK()` liefert einen Link zu der mit `Hyperlink_Adresse` angegebenen Adresse. Neben vollständigen Webadressen – das Protokoll muss mit angegeben werden, also `http://www.***` und nicht bloß `www.***` – können auch Pfade zu Dokumenten im Dateisystem angegeben werden. Bei Excel-Dokumenten kann direkt zu einer bestimmten Zelle oder einem benannten Bereich gesprungen werden, bei Word-Dokumenten zu einer Textmarke, die mit dem Zeichen # gekennzeichnet wird.

Das Argument `Freundlicher_Name` erlaubt es, einen beschreibenden Hinweis auf das Verknüpfungsziel anzugeben. Dieser wird in der Zelle anstelle der Adresse angezeigt, wenn er angegeben wird.

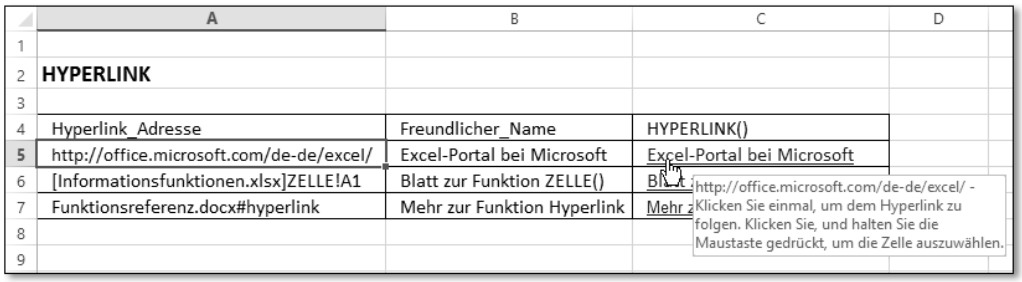


Abbildung 9.13 Beispiel für Hyperlinks in einer Tabelle

Wenn die Tabelle einen Hyperlink enthält, ändert sich der Mauszeiger, sobald er die Zelle berührt. Ein einfacher Klick oder Tipp führt zum Aufruf des verknüpften Dokuments. Soll die Funktion noch einmal bearbeitet werden, wählen Sie die Zelle am besten mit den Richtungstasten aus und benutzen F2, um die Zelle zum Editieren zu öffnen.

Wird die Hyperlink-Adresse direkt eingegeben, sind Anführungszeichen erforderlich. Bei Eingabe über den Funktions-Assistenten werden diese automatisch hinzugefügt. Es ist auch der Bezug auf eine Zelle möglich, die die Zieladresse als Text enthält oder als Ergebnis einer Formel liefert.

Kann das Hyperlink-Ziel nicht gefunden werden, erscheint die Meldung, dass die Ressource mit der angegebenen Adresse nicht geöffnet werden kann. Fehlt die Angabe des

Protokolls am Anfang der Adresse, erhalten Sie die Fehlermeldung, dass die angegebene Datei nicht geöffnet werden kann.

9.5.2 Dynamische Links

Die HYPERLINK()-Funktion erlaubt auch variable Sprungziele. Sie können beispielsweise in Zelle H1 den Namen eines Ordners und in Zelle H2 den Namen einer Datei ablegen, übrigens jeweils ohne Anführungszeichen. In Zelle B3 wird dann folgende Funktion eingetragen:

```
=HYPERLINK(H1&H2; "Details finden Sie in der folgenden Tabelle")
```

Die Adresse wird also in diesem Fall aus zwei Textteilen zusammengesetzt. Der Vorteil dieser Vorgehensweise lässt sich folgendermaßen beschreiben: Nehmen wir an, Sie haben in Ihrer Arbeitsmappe eine ganze Reihe solcher Links, die sich alle auf Dateien im selben Ordner beziehen.

Nun ordnen Sie Ihre Festplatte neu oder kopieren die Dateien auf eine andere Festplatte. Dann brauchen Sie nur noch den Eintrag in der Zelle H1 zu ändern, um alle Verknüpfungen wiederherzustellen.

9.5.3 Bedingte Links

Besonders flexibel ist die Verwendung bedingter Hyperlinks. Angenommen, Sie kalkulieren ein Angebot mal für einen deutschen Kunden und mal für einen französischen Kunden. Die beiden dafür verwendeten Preislisten sind in separaten Tabellen gespeichert.

Sie legen in der Tabelle für das Angebot in der Zelle K2 ein Kennzeichen ab, um die deutschen Angebote von den französischen zu unterscheiden. Dann kann in eine andere Zelle eine Hyperlink-Formel zu den Preistabellen in Abhängigkeit von diesem Kennzeichen eingegeben werden. Die Formel würde lauten:

```
=WENN(K2="Fr"; HYPERLINK("PLISTEFR.XLSX";"französische Preisliste");
HYPERLINK("PLISTEDE.XLSX";"deutsche Preisliste"))
```

Je nachdem, was in Zelle K2 steht, erscheint nicht nur ein anderer Text, beim Klick oder Tipp auf den Link wird auch eine andere Datei geöffnet.

INDEX()

INDEX()

Syntax1:	INDEX(**Matrix**; Zeile; Spalte)
Beispiel:	=INDEX({1.2.3;4.5.6};2;2)
Ergebnis:	5
Syntax2:	INDEX(**Bezug**; Zeile; Spalte; Bereich)
Beispiel:	=INDEX(D18:F21;3;3)
Ergebnis:	liefert den Inhalt der Zelle F20

Die Funktion INDEX() liegt, wie oben schon angesprochen, in zwei unterschiedlichen Ausprägungen vor. In der ersten Form dient sie der Abfrage der Werte einer Matrix, in der zweiten der Ermittlung eines Bezugs aus Zellbereichen, um die entsprechenden Daten abgreifen zu können.

9.5.4 Matrizenwerte abfragen

Mit der ersten Form der Funktion lassen sich Werte aus einer Matrix abfragen. Das Ergebnis kann ein einzelner Wert sein oder wiederum eine Matrix. Als Argument Matrix wird ein Zellbereich, der eine Wertematrix enthält, oder eine Matrixkonstante erwartet.

Die Angaben von Zeile und Spalte dürfen nicht negativ sein. Sie beziehen sich auf die Zeilen und Spalten der Matrix. So wird die Stelle innerhalb der Matrix festgelegt, deren Wert ermittelt werden soll. Die Werte für Zeile und Spalte dürfen auch nicht größer sein als die Anzahl der Zeilen und Spalten der betreffenden Matrix, sonst liefert die Funktion den Fehler #BEZUG!.

Wollen Sie nicht nur einen einzelnen Wert ermitteln, so lässt sich auch eine Matrix in Form einer Spalte oder einer Zeile oder die komplette Matrix auslesen. Allerdings muss in diesen Fällen die Funktion selbst wie eine Matrixformel eingegeben werden (Ausgabebereich markieren, Funktion eingeben, mit Strg + ⇧ + ↵ beenden).

Wollen Sie beispielsweise für den markierten Bereich B16:B19 eine Spalte auslesen, muss die Angabe für Zeile auf 0 gesetzt oder weggelassen werden.

```
{=INDEX(B5:B8;0;1)}
```

zeigt die Werte aus B5 bis B8 in dem Bereich an, der die Matrixformeln enthält. Wollen Sie eine Zeile auslesen, gilt das entsprechende Verfahren:

```
{=INDEX(B5:F5;1;0)}
```

9.5.5 Bereichsbezüge liefern

Die zweite Syntax der Funktion verlangt mit dem ersten Argument `Bezug` die Angabe eines Bezugs auf einen oder auch mehrere Zellbereiche. Mit der Nummer von `Zeile` und `Spalte` wird die Auswahl bezeichnet, deren Bezug ermittelt werden soll. `Bereich` als optionales Argument wird dann verwendet, wenn `Bezug` eine Mehrfachauswahl enthält. In diesem Fall muss die Liste der Bereiche in `Bezug` in Klammern gesetzt werden, etwa:

```
=INDEX((B15:B18;E15:E18);1;1;1)
```

Es sei denn, die Mehrfachauswahl ist mit einem Namen belegt worden. Mit `Bereich` wird durch die Eingabe der entsprechenden Nummer auf diesen Bereich für die Abfrage verwiesen. Wird `Bereich` nicht angegeben, wird immer der erste Teilbereich genommen. Die Anzahl der Bereiche kann bei Bedarf mit Hilfe der Funktion `BEREICHE()` ermittelt werden.

Abbildung 9.14 zeigt ein Beispiel für einen zweiteiligen Bereich. Die Werte in B5:B8 und E5:E8 sind alle unter dem gemeinsamen Bereichsnamen `WERKSERGEBNISSE` zusammengefasst. Es handelt sich also um einen Bereich, der aus zwei Teilbereichen besteht, wie die folgende Formel bestätigt:

```
=BEREICHE(WERKSERGEBNISSE)
```

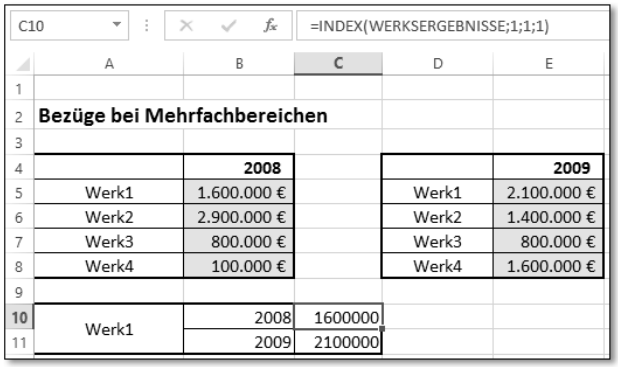


Abbildung 9.14 Zugriff auf Teilbereiche über die `INDEX()`-Funktion

In Zelle C20 und C21 werden nun Formeln verwendet, um Werte aus diesen Teilbereichen auszulesen:

```
=INDEX(WERKSERGEBNISSE;1;1;1)
=INDEX(WERKSERGEBNISSE;1;1;2)
```

Jedes Mal wird die erste Spalte und Zeile des Teilbereichs ausgewertet.

Das Ergebnis der INDEX()-Funktion ist zunächst immer ein Bezug. Je nach der Formel, in der die Funktion verwendet wird, wird das Ergebnis als Bezug interpretiert, oder es wird der Wert geliefert, auf den dieser Bezug verweist. Funktionen, die einen Bezug verlangen, werten ein durch INDEX() geliefertes Argument also als Bezug aus, etwa die Funktion ZELLE():

```
=ZELLE("Typ";INDEX(WERKSERGEBNISSE;1;1;1))
```

liefert w als Hinweis auf das Währungsformat der Zelle. Alle Funktionen, die einen Wert verlangen, interpretieren das Ergebnis als Wert.

Siehe auch Abschnitt 9.3, »Einsatz der INDEX()-Funktion«.

INDIREKT()

INDIRECT()

Syntax:	INDIREKT(**Bezug**; A1)
Beispiel:	=INDIREKT(A2)
Ergebnis:	1994, wenn in A2 z.B. A128 eingetragen ist und in der Zelle A128 1994 steht.

Die Funktion INDIREKT() ermittelt indirekt den Inhalt einer Zelle, auf die mit dem Argument Bezug verwiesen wird. Der Wert von Bezug muss sich dabei als Zeichenfolge auswerten lassen, die einem Bezug entspricht. Diese Zeichenfolge wandelt die Funktion in einen aktuellen Bereichsbezug um und kann dann den dort abgelegten Wert als ihr Ergebnis liefern.

Eine solche Zeichenfolge liefert beispielsweise die Funktion ADRESSE(). Das Argument Bezug kann aber auch eine direkt eingegebene Zeichenfolge sein, etwa:

```
=INDIREKT("A1")
```

oder ein Bezug auf eine Zelle, die eine Adresse als Zeichenfolge enthält.

Mit A1 wird angegeben, wie der Eintrag in Bezug steht: in der A1-Schreibweise (WAHR oder weggelassen) oder in der Z1S1-Schreibweise (FALSCH). Siehe dazu auch die Erläuterungen zu der Funktion ADRESSE().

Nützlich ist das indirekte Verfahren, wenn Zellbezüge dynamisch erzeugt werden sollen. In einer Arbeitsmappe, die Blätter für jeden Monat enthält, benannt mit dem kurzen Monatsnamen, kann mit Hilfe der Funktion ein Zellbezug aufgebaut werden, der vom Monat des aktuellen Datums abhängig ist.

```
=WAHL(MONAT(HEUTE());"Jan";"Feb";"Mrz";"Apr";
"Mai";"Jun";"Jul";"Aug";"Sep";"Okt";
"Nov";"Dez") & "!A10
```

Diese Formel in Zelle A1 erzeugt im Mai den Eintrag `Mai!A10`. Die Formel

```
=INDIREKT(A1)
```

liefert dann den Inhalt der Zelle A10 auf dem Blatt für den Monat Mai.

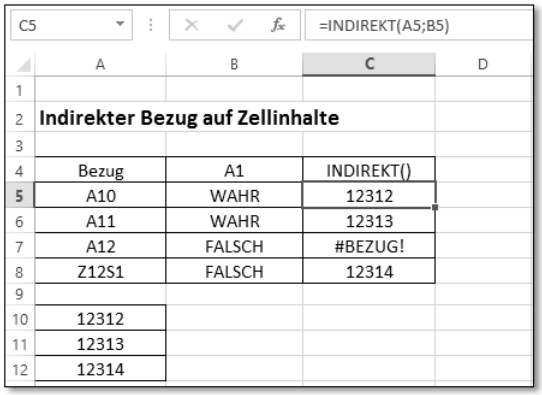


Abbildung 9.15 Bezug über eine anderswo als Text hinterlegte Adresse

Ungültige Bezüge führen zu dem Fehler `#BEZUG!`, das gilt auch für Bezüge auf Zellen in anderen Arbeitsmappen, wenn diese nicht geöffnet sind. Anders als bei den Versionen vor Excel 2007 wird dieser Fehler auch angezeigt, wenn eine Adresse angegeben wird, die über die maximale Spalten- oder Zeilenanzahl hinausreicht (`XFD` für `16384` oder `1048576`).

MTRANS()

TRANSPOSE()

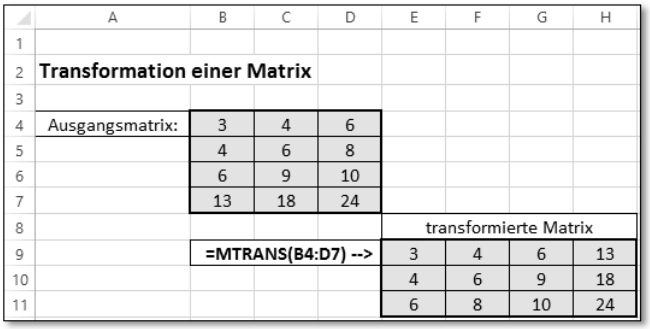
Syntax:	`MTRANS(`**Matrix**`)`
Beispiel:	siehe Abbildung 9.16

Die Funktion `MTRANS()` tauscht die Zeilen und Spalten in einer Matrix, was auch als »Transponieren« bezeichnet wird. Die erste Zeile der bisherigen Matrix wird zur ersten Spalte der neuen Matrix usw. Eine einspaltige Matrix lässt sich mit Hilfe dieser Funktion auch sehr einfach in eine einzeilige überführen.

Das Argument `Matrix` kann als Bezug auf einen Zellbereich abgegeben werden, der eine Matrix enthält, oder als Matrixkonstante. Die Werte der Matrixkonstanten müssen in geschweiften Klammern eingeschlossen werden, wobei Punkte die Werte trennen und Semikola die Zeilen.

Die Funktion muss als Matrixfunktion eingegeben werden. Sie müssen also zunächst den Ausgabebereich markieren, die Funktion eingeben und mit Strg + ⇧ + ↵ abschließen. Leere Zellen in der Ausgangsmatrix werden mit `0` ausgegeben.

Ist der markierte Bereich kleiner als der Ausgangsbereich, werden die überzähligen Zellen ignoriert. Ist er größer, erscheint in den Zellen, für die Werte fehlen, der Fehlerwert `#NV`.



	A	B	C	D	E	F	G	H
1								
2	**Transformation einer Matrix**							
3								
4	Ausgangsmatrix:	3	4	6				
5		4	6	8				
6		6	9	10				
7		13	18	24				
8					transformierte Matrix			
9		**=MTRANS(B4:D7) -->**			3	4	6	13
10					4	6	9	18
11					6	8	10	24

Abbildung 9.16 Transponieren einer Matrix

Zwei Hinweise sind angebracht: Anders als beim Transponieren über den Dialog, Befehl **Inhalte einfügen • Transponieren**, bleibt die transponierte Matrix von der Ursprungsmatrix abhängig, d.h., sie macht alle Veränderungen der Ursprungsmatrix mit. In der transponierten Matrix sind keine Veränderungen einzelner Elemente möglich.

Die Platzierung dieser Funktion in der Kategorie **Nachschlage- und Verweisfunktionen** ist etwas eigenartig, die verwandten Funktionen zum Umgang mit Matrizen finden Sie in der Kategorie **Math. & Trigonom.**: `MDET()`, `MEINHEIT()`, `MMULT()`, `MINV()`.

PIVOTDATENZUORDNEN()

GETPIVOTDATA()

Syntax: `PIVOTDATENZUORDNEN(`**`Datenfeld`**`; `**`PivotTable`**`; Feld1; Element1; Feld2; Element2; ...)`

Beispiel: `=PIVOTDATENZUORDNEN("Bestand";H4;"Land";"Italien")`, siehe Abbildung 9.17

Die Funktion `PIVOTDATENZUORDNEN()` gibt einzelne Werte oder die Werte eines angegebenen Bereichs aus einer Pivot-Tabelle zurück. Diese Funktion wird verwendet, um Daten aus einer Pivot-Tabelle in andere Zellbereiche zu übernehmen.

I12 =PIVOTDATENZUORDNEN("Bestand";H4;"Land";"Italien")

Zugriff auf Daten aus einer Pivot-Tabelle

Name	Land	Anbaugebiet	Farbe	Jahrgang	Bestand
Barolo	Italien	Piemont	rot	2005	300
Beaujolais	Frankreich	Burgund	rot	2005	200
Chablis	Frankreich	Burgund	weiß	2006	230
Ravello	Italien	Salerno	rosé	2005	200
Valpolicella	Italien	Verona	rot	2005	300
Bourgogne	Frankreich	Burgund	weiß	2006	230
Chianti	Italien	Toskana	rot	2005	120
Frascati	Italien	Rom	weiß	2006	230
Freisa	Italien	Piemont	rot	2005	120
Grignolino	Italien	Piemont	rot	2005	230
Médoc	Frankreich	Bordeaux	rot	2005	300
Müller-Turgau	Deutschland	Mosel	weiß	2006	340
Oppenheimer	Deutschland	Rheinhessen	weiß	2006	300
Ortenau	Deutschland	Baden	rot	2005	340
Sauternes	Frankreich	Bordeaux	weiß	2006	170
Riesling	Deutschland	Mosel	weiß	2006	200
Brolio	Italien	Toskana	rot	2005	230

Land	Summe von Bestand
Deutschland	1180
Frankreich	1130
Italien	1730
Gesamtergebnis	**4040**

Der Bestand für Weine aus Italien:

	1730
Gesamtbestand	4040

Abbildung 9.17 Übernahme von Werten aus einer Pivot-Tabelle

Bei der Eingabe der Funktion kann ein Schnellverfahren verwendet werden: Sie geben ein Gleichheitszeichen ein und klicken dann in die Zelle der Pivot-Tabelle, deren Daten übernommen werden sollen. Seit Excel 2007 funktioniert dies, solange über **Excel-Optionen** auf der Seite **Formeln** die Option **GetPivotTable-Funktionen für PivotTable-Bezüge verwenden** aktiviert ist.

Das Argument `Datenfeld` gibt den Namen des Datenfelds an, von dem Daten übernommen werden sollen. `PivotTable` ist ein Bezug auf eine Zelle oder einen Zellbereich in der Pivot-Tabelle, als Vorgabe die linke obere Eckzelle. Es folgen dann Paare von Feld- und Elementnamen, die die gewünschten Daten identifizieren. Bis zu 126 Paare sind seit Excel 2007 möglich, in den älteren Versionen 14. Wird kein Paar mit Feld- und Elementnamen angegeben, wird das Gesamtergebnis ausgegeben.

Der Vorteil der Funktion gegenüber einem einfachen Zellbezug auf einen Wert in einer Pivot-Tabelle ist insbesondere der, dass die Funktion auch dann noch den korrekten Wert liefert, wenn die Daten in der Pivot-Tabelle inzwischen umgeordnet worden sind.

Ist PivotTable kein Bereich, in dem sich eine Pivot-Tabelle befindet, liefert die Funktion den Fehler #BEZUG!. Das gilt auch, wenn die anderen Argumente kein in der Pivot-Tabelle sichtbares Feld beschreiben.

RTD()

RTD()

Syntax:	RTD(**ProgID**; **Server**; **Topic1**; Topic2; ...)
Beispiel:	=RTD("MeinProgramm.ProgID";"MeinServer";"Preis")
Ergebnis:	Daten aus dem Programm

Die Funktion RTD() – die Abkürzung kommt von *Real Time Data* – empfängt Echtzeitdaten eines registrierten Add-Ins, das die COM-Automatisierung unterstützt. Benötigt wird dazu ein RTD-Server, der die Daten fortlaufend liefert. Das Ergebnis der Funktion wird immer dann aktualisiert, wenn neue Daten vom Server kommen. ProgID ist eine Programmkennung des auf dem lokalen System installierten RTD-Servers.

Server ist der Name des Servers, auf dem der RTD-Server ausgeführt wird. Er kann weggelassen werden, wenn der RTD-Server lokal betrieben wird. Mindestens das Argument Topic1 muss noch angegeben werden. Es identifiziert die abgerufenen Daten. Seit Excel 2007 sind bis zu 253 Themen möglich, in den früheren Versionen 28.

Ist der angegebene Server nicht verfügbar, liefert die Funktion den Fehler #NAME!.

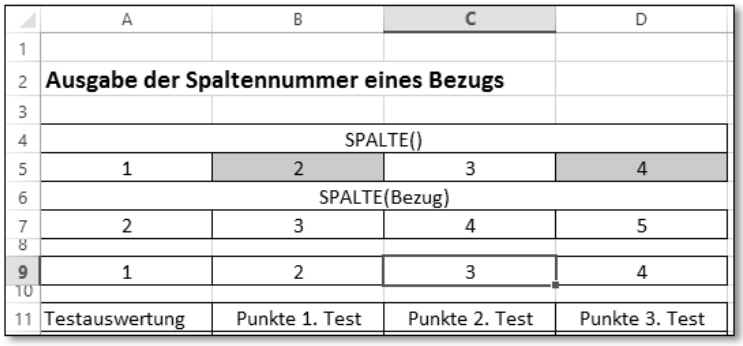
SPALTE()

COLUMN()

Syntax:	SPALTE(**Bezug**)
Beispiel:	=SPALTE(C2)
Ergebnis:	2

Die Funktion SPALTE() liefert die Spaltennummer des mit Bezug angegebenen Bereichs. Reicht die Bereichsadresse über mehrere Spalten, wird die Nummer der ersten Spalte ausgegeben. Wird Bezug nicht angegeben, ist das Ergebnis die Spaltennummer der Zelle, in der die Funktion steht.

Wird die Funktion als horizontale Matrix für einen vorher markierten Zellbereich eingegeben, dann werden die entsprechenden Spaltennummern in dem markierten Bereich ausgegeben. Wenn also in dem Beispiel in Abbildung 9.18 die Spalten A5 bis D5 markiert werden und die Formel {=SPALTE()} mit Strg + ⇧ + ↵ bestätigt wird, erscheinen die Spaltennummern in allen Zellen des Bereichs.



	A	B	C	D
1				
2	**Ausgabe der Spaltennummer eines Bezugs**			
3				
4	SPALTE()			
5	1	2	3	4
6	SPALTE(Bezug)			
7	2	3	4	5
8				
9	1	2	3	4
10				
11	Testauswertung	Punkte 1. Test	Punkte 2. Test	Punkte 3. Test

Abbildung 9.18 Abfrage der Spaltennummer

Wird wie in B9 ein Zellbereich angegeben, der größer ist als der markierte Bereich, werden die überzähligen Werte ignoriert. Umgekehrt werden die Fehlerwerte #NV für markierte Zellen ausgegeben, wenn der angegebene Bereich weniger Spalten enthält.

Eine der vielen Anwendungsmöglichkeiten der Funktion ist die Erzeugung von nummerierten Spaltenbeschriftungen. Zwar kann hier schon vieles mit Hilfe der Reihenbildung mit dem Ausfüllkästchen erreicht werden, aber dabei muss die fortlaufende Zahl immer am Anfang oder Ende der Zeichenfolge stehen.

Mit einer Formel wie

```
="Punkte " & SPALTE()-1 & ". Test"
```

kann leicht eine Kombination aus Text und Spaltennummer erzeugt werden, wie in der Zeile 11 der Abbildung zu sehen ist.

Die Funktion kann auch für bedingte Formate genutzt werden. Soll jede zweite Spalte in einem Bereich einen anderen Hintergrund erhalten, hilft ein bedingtes Format mit der Formel

```
=ISTGERADE(SPALTE())
```

wie in Zeile 5 der Abbildung 9.18.

SPALTEN()

COLUMNS()

Syntax: `SPALTEN(`**`Matrix`**`)`
Beispiel: `=SPALTEN({1.2.3;4.5.6})`
Ergebnis: `3`

Die Funktion `SPALTEN()` liefert die Anzahl der Spalten eines Bereichs oder einer Matrix. Das Argument `Matrix` kann eine Bereichsadresse oder eine Matrixkonstante enthalten. Die Matrixkonstante muss in geschweiften Klammern eingeschlossen sein. Punkte trennen die Spaltenwerte, Semikola die Zeilen.

Ist `Matrix` ein ungültiger Bezug, liefert die Funktion den Fehler `#NAME!`.

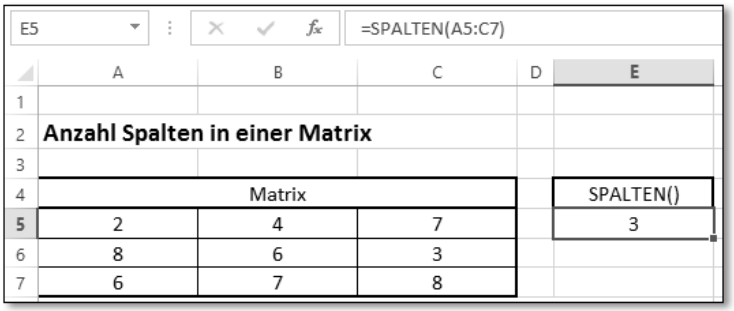


Abbildung 9.19 Abfrage der Spaltenanzahl einer Matrix

9.5.6 Kombination mit der Funktion INDEX()

Effektive Lösungen erlaubt die Funktion `SPALTEN()` in Kombination mit der Funktion `INDEX()`. Die Formel

```
=INDEX(Testbereich;1;SPALTEN(Testbereich))
```

liefert beispielsweise den ersten Wert in der letzten Spalte eines mit dem Namen `Testbereich` belegten Bereichs.

```
=INDEX(Testbereich;ZEILEN(Testbereich);SPALTEN(Testbereich))
```

ergibt den letzten Wert in dem benannten Bereich überhaupt.

	A	B	C	D
1				
2	**Die Spaltenanzahl als Argument für INDEX()**			
3				
4	**Testauswertung**	**Punkte 1. Test**	**Punkte 2. Test**	**Punkte 3. Test**
5	Mo	10,99	10,38	10,33
6	Di	10,39	10,88	10,64
7	Mi	10,23	10,75	10,48
8				
9		**=INDEX(Testbereich;1;SPALTEN(Testbereich))**		10,33
10		**=INDEX(Testbereich;2;SPALTEN(Testbereich))**		10,64
11	**=INDEX(Testbereich;ZEILEN(Testbereich);SPALTEN(Testbereich))**			10,48

Abbildung 9.20 `SPALTE()` als Argument der Funktion `INDEX()`

SVERWEIS()

VLOOKUP()

Syntax: `SVERWEIS(`**Suchkriterium**`;` **Matrix**`;` **Spaltenindex**`; Bereich_Verweis)`

Beispiel: `=SVERWEIS(12;{1.2.3.4.5;12.13.14.15.16};3)`

Ergebnis: `14`

Die Funktion `SVERWEIS()` ermittelt ausgehend von einer Zelle in der ersten Spalte eines Zellbereichs oder einer Matrix den Inhalt der Zelle in derselben Zeile einer anderen Spalte.

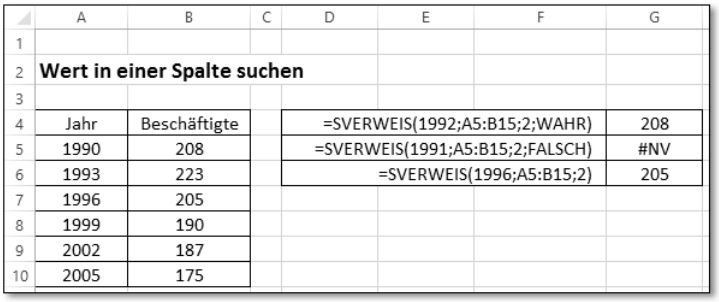
Hierbei durchsucht die Funktion die erste Spalte eines mit `Matrix` angegebenen Bereichs oder einer Matrixkonstanten nach dem mit `Suchkriterium` angegebenen Wert. Dabei wird die Groß- und Kleinschreibung ignoriert, wenn es sich um Zeichenfolgen handelt. Falls der angegebene Wert nicht gefunden werden kann, benutzt die Funktion den nächstkleineren Wert in der Spalte.

Von der gefundenen Position in dieser ersten Spalte aus wird innerhalb derselben Zeile die mit `Spaltenindex` angegebene Spalte aufgesucht (`1` für die erste Spalte, also die Spalte, in der der gesuchte Wert steht, `2` für die zweite).

Ist der mit `Suchkriterium` angegebene Wert kleiner als der kleinste Wert in der betreffenden Spalte, erscheint der Fehler `#NV`. Wird für `Spaltenindex` eine Zahl < `1` angegeben, erscheint der Fehler `#WERT!`. Ist die Zahl größer als die Spaltenbreite der Matrix, liefert die Funktion den Fehlerwert `#BEZUG`.

Das Argument `Bereich_Verweis` ist ein Wahrheitswert. `WAHR` bedeutet, dass eine genaue oder ungefähre Entsprechung mit `Suchkriterium` akzeptiert wird. Das ist die Vorgabe. Damit dies brauchbare Ergebnisse liefert, sollten die Werte in der Tabelle nach der ers-

ten Spalte in aufsteigender Ordnung sortiert sein. Wollen Sie nur eine exakte Übereinstimmung mit dem Suchkriterium akzeptieren, muss der Wert FALSCH gesetzt werden. In diesem Fall ist es nicht nötig, dass die erste Spalte sortiert ist.



	A	B	C	D	E	F	G
1							
2	**Wert in einer Spalte suchen**						
3							
4	Jahr	Beschäftigte		=SVERWEIS(1992;A5:B15;2;WAHR)			208
5	1990	208		=SVERWEIS(1991;A5:B15;2;FALSCH)			#NV
6	1993	223		=SVERWEIS(1996;A5:B15;2)			205
7	1996	205					
8	1999	190					
9	2002	187					
10	2005	175					

Abbildung 9.21 Werte aus einer Spalte auslesen

Weitere Beispiele finden Sie in Abschnitt 9.2, »Arbeiten mit Verweisfunktionen«.

VERGLEICH()

MATCH()

Syntax: VERGLEICH(**Suchkriterium**; **Suchmatrix**; Vergleichstyp)

Beispiel: siehe Abbildung 9.22

Die Funktion VERGLEICH() durchsucht eine Suchmatrix bzw. einen Bereich nach einem Suchkriterium und gibt die relative Position aus. Bei Zeichenfolgen wird die Groß- und Kleinschreibung ignoriert.

Das optionale Argument Vergleichstyp gibt an, auf welche Art nach dem Suchkriterium gesucht werden soll:

Typ	Suchart
1	Ist der voreingestellte Wert, mit dem die Funktion arbeitet, wenn kein Wert angegeben wird. Die Suchmatrix muss in steigender Folge sortiert sein. Wird kein passender Wert gefunden, wird der nächstkleinere gewählt.
-1	Die Matrix muss in fallender Folge sortiert sein; wird kein passender Wert gefunden, wird der nächstgrößere gewählt.
0	Die Matrix muss nicht sortiert sein, die Position des ersten passenden Werts wird ausgegeben.

Tabelle 9.1 Codes für das Argument Vergleichstyp

	A	B	C	D	E	F	G	H	I	J
1										
2	**Position eines Werts in einem Bereich**									
3										
4	Jahr	Beschäftigte		Jahr	Beschäftigte		=VERGLEICH(203;B5:B15;-1)			5
5	1990	223		2000	175		=VERGLEICH(205;B5:B15;0)			5
6	1991	222		1999	179					
7	1992	210		1998	187		=VERGLEICH(207;E5:E15;1)			7
8	1993	208		1997	190					
9	1994	205		1996	200		=VERGLEICH(1995;A5:A15)			6
10	1995	201		1995	201					
11	1996	200		1994	205					
12	1997	190		1993	208					
13	1998	187		1992	210					
14	1999	179		1991	222					
15	2000	175		1990	223					

Abbildung 9.22 Suchen nach Übereinstimmungen

Wird kein übereinstimmender Wert gefunden, wird der Fehlerwert `#NV` ausgegeben.

9.5.7 Zweidimensionales Lookup

Nützlich wird das Ergebnis der Funktion in der Regel erst, wenn das gelieferte Ergebnis an die Funktion `INDEX()` weitergereicht wird, um die Zeilen- oder Spaltennummer zu gewinnen, in der die Daten zu finden sind, die interessieren. Oder beides gleichzeitig, wie das folgende Beispiel demonstrieren soll.

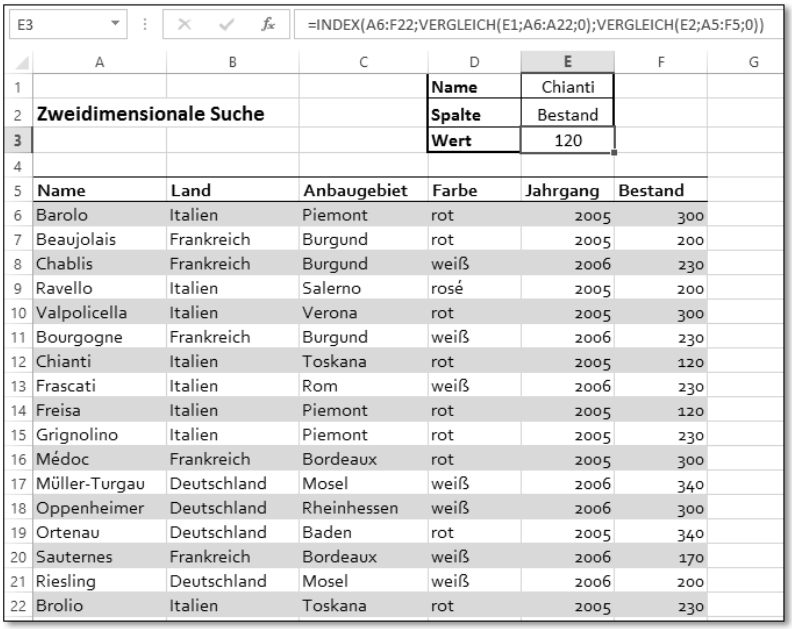
In der Tabelle soll die erste Spalte nach dem Namen durchsucht werden und die erste Zeile nach der interessierenden Spaltenüberschrift. Liefern soll die Formel den Wert der Zelle, in der sich die betreffende Zeile und Spalte kreuzen.

Die beiden Suchkriterien können hier in die Zellen E1 und E2 eingegeben oder aus einer Liste ausgesucht werden, wenn die Zellen mit einer entsprechenden Gültigkeitsregel versehen sind, die die erste Spalte und die erste Zeile von A5:F22 als Quelle auswertet. Siehe dazu den Abschnitt »Prüfung der Dateneingabe durch Gültigkeitsregeln« in Kapitel 1, »Einstieg in Berechnungen mit Excel«.

Die Formel in E3 kann dann lauten:

```
=INDEX(A6:F22; VERGLEICH(E1;A6:A22;1);VERGLEICH(E2;A5:F5;1))
```

Wichtig in diesem Fall ist, dass das Argument `Vergleichstyp` jeweils auf `0` gesetzt wird, so dass nur exakte Übereinstimmungen berücksichtigt werden.



E3 =INDEX(A6:F22;VERGLEICH(E1;A6:A22;0);VERGLEICH(E2;A5:F5;0))

	A	B	C	D	E	F	G
1				**Name**	Chianti		
2	**Zweidimensionale Suche**			**Spalte**	Bestand		
3				**Wert**	120		
4							
5	**Name**	**Land**	**Anbaugebiet**	**Farbe**	**Jahrgang**	**Bestand**	
6	Barolo	Italien	Piemont	rot	2005	300	
7	Beaujolais	Frankreich	Burgund	rot	2005	200	
8	Chablis	Frankreich	Burgund	weiß	2006	230	
9	Ravello	Italien	Salerno	rosé	2005	200	
10	Valpolicella	Italien	Verona	rot	2005	300	
11	Bourgogne	Frankreich	Burgund	weiß	2006	230	
12	Chianti	Italien	Toskana	rot	2005	120	
13	Frascati	Italien	Rom	weiß	2006	230	
14	Freisa	Italien	Piemont	rot	2005	120	
15	Grignolino	Italien	Piemont	rot	2005	230	
16	Médoc	Frankreich	Bordeaux	rot	2005	300	
17	Müller-Turgau	Deutschland	Mosel	weiß	2006	340	
18	Oppenheimer	Deutschland	Rheinhessen	weiß	2006	300	
19	Ortenau	Deutschland	Baden	rot	2005	340	
20	Sauternes	Frankreich	Bordeaux	weiß	2006	170	
21	Riesling	Deutschland	Mosel	weiß	2006	200	
22	Brolio	Italien	Toskana	rot	2005	230	

Abbildung 9.23 Gleichzeitige Suche in einer Zeile und einer Spalte

VERWEIS()

LOOKUP()

Syntax1: `VERWEIS(`**`Suchkriterium; Suchvektor; Ergebnisvektor`**`)`

Syntax2: `VERWEIS(`**`Suchkriterium; Matrix`**`)`

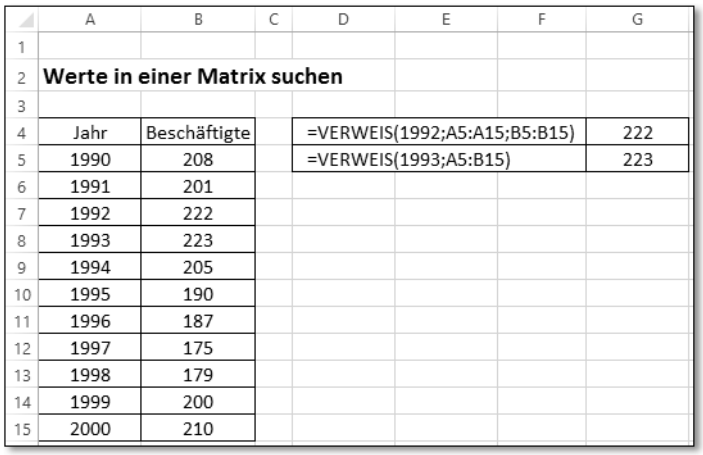
Beispiel: siehe Abbildung 9.24

Die Funktion `VERWEIS()` liefert auf der Grundlage eines Suchkriteriums den Inhalt einer korrespondierenden Zelle, vergleiche `SVERWEIS()` und `WVERWEIS()`.

Die Funktion liegt in zwei Versionen vor. Bei der ersten werden zwei getrennte Bereiche (Spalten oder Zeilen) benutzt, bei der zweiten ein zusammenhängender Bereich.

In der ersten Version benötigt die Funktion neben `Suchkriterium` einen `Suchvektor` und einen `Ergebnisvektor`. Der `Suchvektor` ist eine Spalte oder eine Zeile, die nach dem mit `Suchkriterium` angegebenen Wert durchsucht werden soll. Von der Fundstelle aus wird der Wert, der die gleiche Position in `Ergebnisvektor` einnimmt, als Ergebnis von der Funktion ausgegeben. `Suchvektor` und `Ergebnisvektor` sollten deshalb die gleiche Größe haben.

Die Einträge in Suchvektor müssen in steigender Folge sortiert sein. Kann kein dem Suchkriterium genau entsprechender Wert gefunden werden, wird der nächstkleinere Wert übernommen.



	A	B	C	D	E	F	G
1							
2	**Werte in einer Matrix suchen**						
3							
4	Jahr	Beschäftigte		=VERWEIS(1992;A5:A15;B5:B15)			222
5	1990	208		=VERWEIS(1993;A5:B15)			223
6	1991	201					
7	1992	222					
8	1993	223					
9	1994	205					
10	1995	190					
11	1996	187					
12	1997	175					
13	1998	179					
14	1999	200					
15	2000	210					

Abbildung 9.24 Die zwei Varianten der Funktion VERWEIS()

Für die zweite Version der Funktion wird die erste Zeile oder Spalte der angegebenen Matrix nach dem Suchkriterium durchsucht. Ob eine Zeile oder eine Spalte durchsucht wird, ist von der Dimensionierung der Matrix abhängig. Hat eine Matrix mehr Spalten als Zeilen oder ist deren Anzahl gleich, so wird die erste Zeile durchsucht. Besitzt eine Matrix mehr Zeilen, so wird die erste Spalte durchsucht.

Ist die Funktion auf der Suche nach dem Kriterium beispielsweise in der ersten Spalte fündig geworden, dann geht sie in dieser Zeile nach rechts bis zur letzten Spalte und gibt den Wert der dortigen Zelle zurück. Analog wird bei der Suche in der ersten Zeile verfahren.

Ist der Wert von Suchkriterium kleiner als der kleinste Wert in Suchvektor, erscheint der Fehler #NV.

WAHL()

CHOOSE()

Syntax: WAHL(**Index**; **Wert1**; Wert2; ...)

Beispiel: =WAHL(3;"orange";"rot";"gelb";"grün";"blau")

Ergebnis: gelb

Die Funktion WAHL() liefert einen Wert aus einer Liste von Werten. Mit Index wird festgelegt, der wievielte Wert als Ergebnis zurückgegeben werden soll. Seit Excel 2007 sind 254 Werte möglich, in älteren Versionen sind es 29.

Entsprechend ist Index auf Werte zwischen 1 und 254 bzw. 1 und 29 begrenzt. Ist der Wert < 1 oder größer als die Anzahl der Werte in der Werteliste oder nicht numerisch, liefert die Funktion den Fehler #WERT!. Falls Index keine ganze Zahl ist, wird die nächstkleinere Ganzzahl verwendet. Werden für die Werteliste nicht vorhandene Bereichsnamen verwendet, liefert die Funktion den Fehler #NAME?.

Die Argumente in der Werteliste können Zahlen, Zeichenfolgen, Zell- oder Bereichsbezüge, definierte Namen oder Formeln sein.

9.5.8 Monatsnamen oder Wochentage ausgeben

Ein beliebtes Beispiel für eine Anwendung der Funktion WAHL() ist die Erzeugung der Monatsnamen oder der Tagesnamen aus gegebenen Datumswerten in einer Tabelle. Allerdings lässt sich die etwas umständliche Vorgehensweise häufig vermeiden, indem mit benutzerdefinierten Datumsformaten gearbeitet wird, die aus dem Datum nur diese Namen herausziehen, etwa MMMM oder MMM bzw. TTTT oder TTT.

Abbildung 9.25 zeigt zwei Spalten, in denen mit der Funktion WAHL() aus dem Datum in der Spalte A die abgekürzten Monats- oder Tagesnamen abgeleitet werden.

Die erste Formel in Spalte D arbeitet mit der Funktion MONAT(), um den Wert für Index zu gewinnen:

```
=WAHL(MONAT(A5);"Jan";"Feb";"Mrz";"Apr";"Mai";
"Jun";"Jul";"Aug";"Sep";"Okt";"Nov";"Dez")
```

Die zweite Formel in Spalte F benutzt die Funktion WOCHENTAG(), um den Wert für Index zu finden:

```
=WAHL(WOCHENTAG(A5);"So";"Mo";"Di";"Mi";"Do";"Fr";"Sa")
```

Firmen, die das Geschäftsjahr nicht am 1. Januar beginnen lassen, können die Funktion benutzen, um den jeweiligen Monat des abweichenden Geschäftsjahres auszugeben. Wenn das Geschäftsjahr beispielsweise immer am 1. April beginnt, gibt die folgende Formel immer die Monatszahl für den aktuellen Monat an:

```
=WAHL(MONAT(HEUTE());10;11;12;1;2;3;4;5;6;7;8;9)
```

	A	B	C	D	E	F
1						
2	**Monats- oder Tagesnamen mit der Funktion WAHL()**					
3						
4	**Datum**	**MMMM**	**TTTT**	**WAHL()**	**WOCHENTAG()**	**WAHL()**
5	01. Jan	Januar	Dienstag	Jan	3	Di
6	01. Feb	Februar	Freitag	Feb	6	Fr
7	01. Mrz	März	Freitag	Mrz	6	Fr
8	01. Apr	April	Montag	Apr	2	Mo
9	01. Mai	Mai	Mittwoch	Mai	4	Mi
10	01. Jun	Juni	Samstag	Jun	7	Sa
11	01. Jul	Juli	Montag	Jul	2	Mo
12	01. Aug	August	Donnerstag	Aug	5	Do
13	01. Sep	September	Sonntag	Sep	1	So
14	01. Okt	Oktober	Dienstag	Okt	3	Di
15	01. Nov	November	Freitag	Nov	6	Fr
16	01. Dez	Dezember	Sonntag	Dez	1	So

Abbildung 9.25 Beispiel für die Funktion `WAHL()` bei Datumswerten

9

9.5.9 Bereichsbezüge, wahlweise

Wenn Sie eine Werteliste aus Bereichsadressen oder Bereichsnamen zusammenstellen, liefert die isolierte Funktion `WAHL()` den Fehler `#WERT!`, die Funktion kann also mit den Werten in den Bereichen selbst nichts anfangen. Eine Werteliste mit Bereichsbezügen lässt sich aber verwenden, um einer übergeordneten Funktion Bereichsargumente in Abhängigkeit von dem mit `Index` angegebenen Wert zu übergeben.

	A	B	C	D	E	F	G
1							
2	**Bereiche über einen Index auswählen**						
3							
4	**Kunde**	2008	2009	2010	2011	2012	2013
5	Hansen	412	656	345	487	607	464
6	Borbeck	739	338	298	662	595	676
7	Look	441	230	425	404	523	692
8	Tessen	246	636	644	556	536	430
9	Vonderbank	496	377	443	688	528	738
10							
11	Summe für welches Jahr?		2008	2334			
12			2008				
13			2009				
14			2010				
15			2011				
			2012				
			2013				

Abbildung 9.26 Beispiel für die Bereichsauswahl für eine Summe

In Abbildung 9.26 sind Verkaufszahlen pro Kunde für mehrere Jahre gelistet. Diese Jahreszahlen werden in Zelle C11 über eine entsprechende Gültigkeitsregel zur Auswahl angeboten. Die Formel in D11 kann dann lauten:

```
=SUMME(WAHL(C11-MIN(B4:G4)+1;B5:B9;C5:C9;D5:D9;E5:E9;F5:F9;G5:G9))
```

Siehe auch Abschnitt 9.4, »Einsatz der WAHL()-Funktion«.

WVERWEIS()

HLOOKUP()

Syntax:	`=WVERWEIS(`**Suchkriterium**`;` **Matrix**`;` **Zeilenindex**`;` `Bereich_Verweis)`
Beispiel:	`=WVERWEIS(4;{1.2.3.4.5;12.13.14.15.16};2)`
Ergebnis:	`14`

Die Funktion `WVERWEIS()` entspricht exakt der Funktion `SVERWEIS()`; lediglich Zeilen und Spalten sind vertauscht. Die Funktion ermittelt ausgehend von einer Zelle in der ersten Zeile eines Zellbereichs oder einer Matrix den Inhalt der Zelle in derselben Spalte einer anderen Zeile.

Hierbei durchsucht die Funktion die erste Zeile eines mit `Matrix` angegebenen Bereichs oder einer Matrixkonstanten nach dem mit `Suchkriterium` angegebenen Wert. Dabei wird die Groß- und Kleinschreibung ignoriert, wenn es sich um Zeichenfolgen handelt. Falls der angegebene Wert nicht gefunden werden kann, benutzt die Funktion den nächstkleineren Wert in der Spalte.

Von der gefundenen Position in dieser ersten Zeile aus wird innerhalb derselben Spalte die mit `Zeilenindex` angegebene Zeile aufgesucht (`1` für die erste Zeile, also die Spalte, in der der gesuchte Wert steht, `2` für die zweite).

Ist der mit `Suchkriterium` angegebene Wert kleiner als der kleinste Wert in der betreffenden Zeile, erscheint der Fehler `#NV`. Wird für `Zeilenindex` eine Zahl < `1` angegeben, erscheint der Fehler `#WERT!`. Ist die Zahl größer als die Zeilenbreite der Matrix, liefert die Funktion den Fehlerwert `#BEZUG`. Wird für `Zeilenindex` eine Zahl größer als die Zeilenanzahl der Matrix angegeben, liefert die Funktion den Fehlerwert `#BEZUG`.

Das Argument `Bereich_Verweis` ist ein Wahrheitswert. `WAHR` bedeutet, dass eine genaue oder ungefähre Entsprechung mit `Suchkriterium` akzeptiert wird. Das ist die Vorgabe. Damit dies brauchbare Ergebnisse liefert, sollten die Werte in der Tabelle nach der ersten Zeile in aufsteigender Ordnung sortiert sein.

Wollen Sie nur eine exakte Übereinstimmung mit dem Suchkriterium akzeptieren, muss der Wert FALSCH gesetzt werden. In diesem Fall ist es nicht nötig, dass die erste Zeile sortiert ist.

Siehe auch Abschnitt 9.2, »Arbeiten mit Verweisfunktionen«.

ZEILE()

ROW()

Syntax: ZEILE(**Bezug**)
Beispiel: =ZEILE(B105)
Ergebnis: 105

Die Funktion ZEILE() liefert die Zeilennummer des unter Bezug angegebenen Bereichs. Reicht die Bereichsadresse über mehrere Zeilen, wird die Nummer der ersten Zeile ausgegeben. Wird für Bezug keine Angabe gemacht, wird als Ergebnis die Zeilennummer der Zelle ausgegeben, in der die Funktion steht.

Wird die Funktion als vertikale Matrix für einen vorher markierten Zellbereich eingegeben, dann werden die entsprechenden Zeilennummern in dem markierten Bereich ausgegeben. Wenn also in dem Beispiel in Abbildung 9.27 A5 bis A7 markiert und die Formel {=ZEILE()} mit Strg + ⇧ + ↵ bestätigt wird, erscheinen die Zeilennummern in allen Zellen des Bereichs.

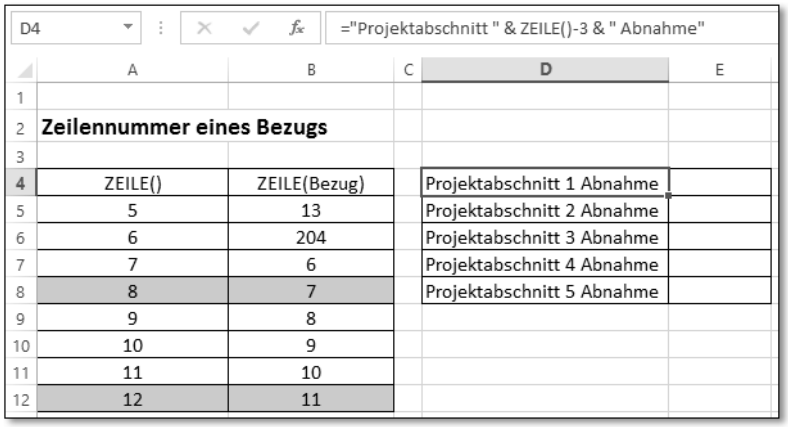


Abbildung 9.27 Abfrage der Zeilennummer

Wird bei einer Matrixformel ein Zellbereich angegeben, der größer ist als der markierte Bereich, werden die überzähligen Werte ignoriert. Umgekehrt werden die Fehlerwerte `#NV` für markierte Zellen ausgegeben, wenn der angegebene Bereich weniger Zeilen enthält. Ungültige Bezüge ergeben den Fehler `#NAME!`.

Eine typische Anwendungsmöglichkeit ist die Generierung von nummerierten Zeilenbeschriftungen. Zwar kann hier vieles mit Hilfe der Reihenbildung mit dem Ausfüllkästchen erreicht werden, aber dabei muss die fortlaufende Zahl immer am Anfang oder Ende der Zeichenfolge stehen.

Mit einer Formel wie

```
="Projektabschnitt " & ZEILE()-3 & " Abnahme"
```

kann leicht eine Kombination aus Text und Zeilennummer erzeugt werden.

Die Funktion kann auch für bedingte Formate genutzt werden. Soll beispielsweise jede vierte Zeile in einem Bereich einen anderen Hintergrund erhalten, hilft ein bedingtes Format mit der Formel

```
=REST(ZEILE();4)=0
```

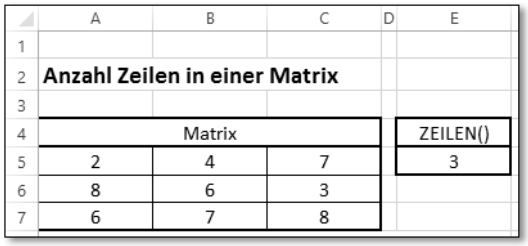
wie in Abbildung 9.27 zu sehen.

ZEILEN()

ROWS()

Syntax:	`ZEILEN(`**`Matrix`**`)`
Beispiel:	`=ZEILEN(Inhalt)`
Ergebnis:	`4`, wenn der Bereich »Inhalt« 4-zeilig ist

Die Funktion `ZEILEN()` liefert die Anzahl der Zeilen eines Bereichs oder einer Matrix. Das Argument `Matrix` kann eine Bereichsadresse oder Matrixkonstante enthalten. Die Konstante muss in geschweiften Klammern eingeschlossen sein. Punkte trennen die Spaltenwerte, Semikola die Zeilen.



	A	B	C	D	E
1					
2	**Anzahl Zeilen in einer Matrix**				
3					
4	Matrix				ZEILEN()
5	2	4	7		3
6	8	6	3		
7	6	7	8		

Abbildung 9.28 Ermitteln der Zeilenanzahl in einem Bereich oder einer Matrix

Effektive Lösungen erlaubt die Funktion in Kombination mit der Funktion `INDEX()`.

Die Formel

```
=INDEX(Testbereich;ZEILEN(Testbereich);1)
```

liefert beispielsweise den ersten Wert in der letzten Zeile eines mit dem Namen `Testbereich` belegten Bereichs.

Ungültige Bezüge ergeben den Fehler `#WERT!`.

10 Datenbankfunktionen

Funktion	Seite	Funktion	Seite
DBANZAHL()	707	DBPRODUKT()	712
DBANZAHL2()	708	DBSTDABW()	713
DBAUSZUG()	708	DBSTDABWN()	714
DBMAX()	709	DBSUMME()	714
DBMIN()	710	DBVARIANZ()	715
DBMITTELWERT()	711	DBVARIANZEN()	716

10.1 Einsatzbereiche für Datenbankfunktionen

Die Zusammenstellung von Daten in Form von tabellarischen Listen – Adresslisten, Artikellisten, Bestelllisten etc. – gehört zu den Standardaufgaben, die bei der Arbeit mit Excel anfallen. Um solche gleichmäßig strukturierten Tabellen anzulegen, zu pflegen und auszuwerten, stellt Excel leistungsfähige Verfahren zur Verfügung, die teilweise durchaus mit den Fähigkeiten spezieller Datenbankprogramme vergleichbar sind. Das gilt jedenfalls für relationale Datenbanken, die ihre Datenbestände in Form von Tabellen ordnen.

10.1.1 Datenbanken, Datenlisten und Tabellen

Excel kann solche Datenbanktabellen ohne Umstände aus externen Datenbanken in seine Tabellenblätter importieren, die ja eine solche tabellarische Struktur schon vorgeben. Genauso gut ist es möglich, im Tabellenblatt direkt solche Datenstrukturen aufzubauen.

Die in diesem Kapitel beschriebenen Funktionen werden als Datenbankfunktionen bezeichnet, weil sie für solche Datenstrukturen spezielle Auswertungsmöglichkeiten anbieten, die den bei Datenbanken üblichen Abfragemethoden ähnlich sind.

Dabei spielt es keine Rolle, ob es sich um einfache Zellbereiche handelt, die so geordnet sind, dass sie einer relationalen Datentabelle entsprechen können, oder ob zusätzlich noch die speziellen Möglichkeiten genutzt werden, die Excel anbietet, wenn ein solcher

Zellbereich seit Excel 2007 mit **Start • Formatvorlagen • Als Tabelle formatieren** als Tabelle definiert oder in den älteren Versionen mit **Daten • Liste • Liste erstellen** als Datenliste bestimmt wird.

10.1.2 Tabellenstrukturen

Voraussetzung für die Anwendung der Datenbankfunktionen ist nur, dass ein paar Minimalbedingungen gewährleistet sind. Tabellen in diesem Sinne können immer dann eingesetzt werden, wenn Informationen sich gleichmäßig zeilenweise anordnen lassen. Die Zellen in einer Zeile enthalten dabei jeweils die Daten für eine bestimmte Einheit. Das kann eine Person oder Personengruppe, eine räumliche, zeitliche oder eine sachliche Einheit sein. Während z. B. eine Tabellenzeile einer solchen Tabelle verschiedene Informationen über eine bestimmte Person enthält, wird in eine Tabellenspalte jeweils dieselbe Art von Information zu den verschiedenen aufgeführten Personen eingetragen, etwa der Name.

Der Inhalt der Spalte wird durch Feldnamen in Form der Spaltenbeschriftungen in der ersten Zeile der Tabelle angezeigt. Welche Spaltennamen notwendig sind, hängt natürlich vom Gegenstand der Tabelle ab, insbesondere aber auch von den Fragen, die mit Hilfe der Tabelle zu beantworten sind. Die Einheit kann eine Person in einer bestimmten Eigenschaft sein, z. B. als Kunde oder Mitarbeiter. Sie kann eine Sache sein, wie ein Artikel oder ein Bausatz für ein Produkt, ein Vorgang wie eine Dienstleistung oder auch ein Tatbestand wie die Zuordnung von Personen zu verschiedenen Projekten.

Lagerdaten

Artnr	Bezeichnung	Warengruppe	Material	Mindestbestand	Bestand	Umsatz_lfd_Jahr
7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.1 Beispieltabelle, die mit Hilfe der Datenbankfunktionen ausgewertet werden soll

Für die folgenden Funktionsbeschreibungen soll eine einfache Lagertabelle verwendet werden, die neben den Daten, die die einzelnen Artikel beschreiben, noch die Informationen zum Bestand und zum Umsatz im laufenden Jahr enthält.

Die Reihenfolge der Spalten ist im Prinzip beliebig, aber es ist natürlich sinnvoll, Informationen, die sachlich zusammengehören, auch nebeneinander anzuordnen. Solange die Tabelle nur wenige Spalten enthält, ist das alles nicht weiter tragisch. Bei Tabellen mit zahlreichen Spalten aber ist es sinnvoll, in den ersten Spalten die Informationen unterzubringen, die die Sache oder die Person, um die es geht, eindeutig identifizieren.

10.1.3 Datentypen und Feldlängen

Niemand kann Sie daran hindern, ein bestimmtes Feld in der einen Zeile mit einer Zahl und in der nächsten mit einer Zeichenfolge zu füllen. Diese Freiheit kann aber auch Nachteile haben, z. B. bei einer späteren Sortierung, weil Excel Zahlen immer vor Zeichenfolgen einordnet. Wird also eine Artikelnummer einmal als Zahl und einmal als Zeichenfolge aus Zahlen eingegeben, gerät die Reihenfolge durcheinander. Zudem treten Probleme bei der Formulierung von Suchkriterien auf.

Für die einzelnen Felder stellt sich also die Frage, welcher Datentyp in der jeweiligen Spalte verwendet werden soll. Ist es ein numerischer Wert, ein Datums- oder Zeitwert, ein Wahrheitswert oder eine Zeichenfolge? Im Unterschied zu einem speziellen Datenbankprogramm wie Access findet bei der Dateneingabe in eine Excel-Tabelle zunächst keine Prüfung des Datentyps oder der Länge der Eingabe statt, es sei denn, Sie steuern die Dateneingabe über ein entsprechendes Makro oder verwenden Gültigkeitsregeln. Diese Funktion wurde in Abschnitt 1.19, »Verfahren der Fehlervermeidung«, ja bereits beschrieben. Sie bietet sehr schöne Möglichkeiten zur Kontrolle der korrekten Dateneingabe, sie muss aber nicht verwendet werden.

Die Spaltennamen sollten normalerweise eindeutig sein. Vermeiden Sie Kommata, Punkte, Leerzeichen, Bindestriche und Semikola. Verwenden Sie keine Namen, die wie Zellbezüge aussehen. Was die Länge der Spaltennamen betrifft, ist ein brauchbarer Kompromiss sinnvoll zwischen einem Namen, der sprechend ist, also genau erkennen lässt, worum es in der Spalte geht, und einem Namen, der möglichst kurz ist. Wenn der Name sehr lang ist, haben Sie nicht nur mehr Arbeit bei der Formulierung von Abfragekriterien, die Wahrscheinlichkeit von Tippfehlern wächst ebenfalls mit der Länge des Namens.

10.1.4 Gemeinsame Merkmale der Datenbankfunktionen

Neben den speziellen Programmfunktionen, die Excel für die Auswertung und Bearbeitung von Datenlisten und Tabellen zur Verfügung stellt, bietet das Programm seit Langem auch eine Anzahl von Funktionen, die in erster Linie der statistischen Auswertung von Datenbeständen dient. Dabei geht es um die Methoden der beschreibenden Statistik, die in Kapitel 7, »Statistische Funktionen«, bereits angesprochen sind.

Der wesentliche Unterschied zu den dort behandelten statistischen Funktionen ist dabei der, dass für die Auswertung bestimmte Kriterien definiert werden können, die die Daten einschränken, die in der Berechnung der Funktion berücksichtigt werden. Diese Kriterien werden außerhalb der Datentabelle in einem speziellen Bereich abgelegt, der als *Kriterienbereich* bezeichnet wird.

Alle Datenbankfunktionen verwenden drei Argumente:

1. Das erste – `Datenbank` – ist erforderlich und gibt den Zellbereich an, der die Tabelle oder Liste enthält. Es ist zu empfehlen, den entsprechenden Bereich vorher zu benennen, damit die Formeln auch bei einer Erweiterung weiterhin korrekt arbeiten. Wenn Sie mit benannten Tabellenbereichen arbeiten, können Sie auch mit den strukturierten Bezügen arbeiten, die in Abschnitt 1.16, »Einsatz von strukturierten Bezügen in Tabellen«, beschrieben worden sind.
2. Das zweite Argument – `Datenbankfeld` – ist bei einigen Funktionen optional, meist aber erforderlich. Es gibt an, auf welches Feld die jeweilige Auswertung angewendet werden soll. Das Argument kann als Zellbezug auf den Feldnamen im Kriterienbereich angegeben werden, als Spaltenname in Anführungszeichen oder als Spaltennummer bezogen auf die verwendete Tabelle. `1` ist dann die erste Spalte der Tabelle.
3. Als drittes Argument muss `Suchkriterien` angegeben werden, das ist der Zellbezug oder auch der vergebene Name für einen Kriterienbereich.

10.1.5 Kriterienbereiche

Der Kriterienbereich ist ein spezieller Zellbereich für die Festlegung von Abfragekriterien. Er muss mindestens zwei Zeilen umfassen. In der ersten Zeile befinden sich ein oder mehrere Spaltennamen. Die zweite Zeile oder weitere Zeilen darunter sind für die Eingabe von Kriterien reserviert.

Die Spaltennamen im Kriterienbereich müssen buchstabengetreu mit den Spaltennamen der Tabelle übereinstimmen, mit Ausnahme zusätzlicher Namen für berechnete Kriterien, von denen noch die Rede sein wird. Das erreichen Sie am einfachsten, indem

Sie die Namenszeile aus der Tabelle oder Teile der Namenszeile in die erste Zeile des Kriterienbereichs kopieren.

Es ist dabei nicht notwendig, dass alle Feldnamen im Kriterienbereich auftauchen, Sie können einzelne Feldnamen weglassen. Für kombinierte Kriterien kann ein Spaltenname sogar mehrfach benutzt werden. Achten Sie darauf, den Kriterienbereich im Tabellenblatt so zu platzieren, dass er nicht unbeabsichtigt von Änderungen wie dem Einfügen oder Löschen von Spalten oder Zeilen zerstört werden kann.

Wird der Kriterienbereich innerhalb des Tabellenblatts angelegt, in dem sich die Tabelle befindet, ist es am sichersten, dafür neue Zeilen oberhalb der Tabelle einzufügen. Die Tabelle kann dann ungehindert nach unten wachsen, ohne dass Gefahr besteht, mit dem Kriterienbereich zu kollidieren. Eine andere Lösung ist, den Kriterienbereich gleich auf ein eigenes Blatt zu legen.

10.1.6 Syntax der Auswahlkriterien

Mit Hilfe eines Auswahlkriteriums legen Sie fest, welche Daten Excel aus einer Tabelle bei der Berechnung einer Datenbankfunktion berücksichtigen soll. Das Kriterium wird immer positiv formuliert, und zwar als die Bedingung, die ein Datensatz erfüllen muss, wenn er den Filter passieren soll; ein Kriterium wie `Warengruppe = Rollo` lässt alle Artikel durch den Filter, die zu dieser Warengruppe gehören.

Die einfachste Form eines Kriteriums im Kriterienbereich ist eines für eine Spalte. Entweder wird unter dem Spaltennamen ein konstanter Wert als Kriterium angegeben oder die Kombination von einem der Vergleichsoperatoren mit einem konstanten Wert.

	A	B	C	D	E	F	G
1							
2	**Vorkommen prüfen**				Artnr	8443	**Artikel vorhanden**
3							
4	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Farbe**	**Absatz**	**Umsatz**
5	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1232	26181
6	7778	Jalousie Ccs	Jalousie	Metall	rot	1463	63360
7	7578	Jalousie Ccw	Jalousie	Metall	rot	1353	62810
8	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1243	25631
9	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	770	12650
10	8443	Jalousie V	Jalousie	Metall	silbergrau	660	44000
11	8243	Jalousie VV	Jalousie	Metall	silbergrau	253	43450

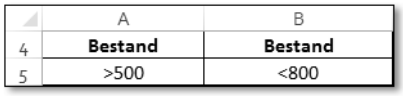
Abbildung 10.2 Einfaches Kriterium für ein Feld

Wenn Sie nicht nur in einem, sondern in mehreren Feldern der ersten Zeile des Kriterienbereichs eine Eintragung vornehmen, behandelt das Programm dies als ein kombiniertes Kriterium. Gesucht wird in einem solchen Fall ein Datensatz, in dem es bei allen benutzten Feldern einen genau entsprechenden Inhalt gibt.

Wenn Sie also im Kriterienbereich unter Warengruppe `Jalousie` eintragen und unter Material `Kunststoff`, sucht das Programm die Artikel, die beide Kriterien gleichzeitig erfüllen. Das entspricht einer logischen `UND()`-Funktion.

Achten Sie vor der Eingabe eines neuen Kriteriums immer darauf, dass nicht Eintragungen von alten Abfragen stehen geblieben sind, die dann als ungewolltes Zusatzkriterium wirken.

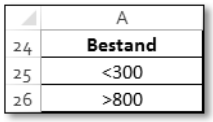
Müssen zwei Bedingungen bei derselben Spalte erfüllt sein, können Sie Spaltennamen auch mehrfach verwenden.



	A	B
4	**Bestand**	**Bestand**
5	>500	<800

Abbildung 10.3 Doppeltes Kriterium für ein Feld

Sollen Kriterien alternativ verwendet werden, müssen sie untereinander in eine Spalte des Kriterienbereichs geschrieben werden. Im folgenden Fall würden also alle Artikel gefunden, deren Bestand entweder kleiner als 300 oder größer als 800 ist.



	A
24	**Bestand**
25	<300
26	>800

Abbildung 10.4 Alternatives Kriterium für ein Feld

Für diese Abfrage erweitern Sie den Kriterienbereich einfach um eine zusätzliche Zeile.

Bei vergleichenden Suchkriterien kann der Inhalt einer Zelle in der Tabelle mit einer Zeichenfolge, einer Zahl, einem Wahrheitswert oder auch einem Fehlerwert verglichen werden. Wird als Kriterium nur das Gleichheitszeichen eingetragen, werden alle Datensätze gesucht, die in dem kritischen Feld keinen Eintrag vorweisen, also leer sind. <> dagegen sucht alle Datensätze, die in der betreffenden Spalte über irgendeinen Eintrag verfügen, egal welcher Art.

Wird nur eine Zeichenfolge als Kriterium verwendet, werden jeweils die Datensätze gesucht, in denen Übereinstimmung zwischen dem Feld in der Tabelle und dem Eintrag im

Kriterienbereich besteht. Das Kriterium `Rollo` ist dabei eine verkürzte Schreibweise für `="Rollo"`.

Textvergleiche unterscheiden normalerweise nicht zwischen Groß- und Kleinschreibung. Mit einem Kriterium wie `Rollo` würde außerdem nicht nur `Rollo`, sondern auch `Rollohalter` gefunden. Soll die Übereinstimmung exakt sein, müssen Sie `="=Rollo"` eintragen.

Statt der Übereinstimmung mit einer Zeichenfolge kann auch die Stellung in Bezug auf die alphabetische Reihenfolge als Kriterium benutzt werden. `>M` filtert beispielsweise alle Einträge in einer Spalte aus, die mit den Buchstaben A bis M beginnen.

Gerade bei Zeichenfolgen, bei Namen und Bezeichnungen steht der Suchende aber oft vor dem Problem, dass ihm gerade die exakte Schreibweise nicht bekannt ist. In so einem Fall helfen Stellvertreterzeichen:

Zeichen	Wirkung
*	Steht für beliebig viele Zeichen in einer Zeichenfolge.
?	Steht für genau ein Zeichen in einer Zeichenfolge.
~	Dieses Zeichen kann vor das Fragezeichen oder den Stern gesetzt werden, wenn diese Zeichen selbst gesucht werden sollen. Die Tilde verhindert also, dass die beiden Zeichen als Stellvertreterzeichen wirken.

Tabelle 10.1 Liste der Joker-Zeichen

Wenn Sie in ein Feld des Kriterienbereichs eine Zahl eintragen, muss der Wert dieser Zahl mit dem Wert in dem gesuchten Datensatz exakt übereinstimmen. Nicht übereinstimmen muss das Format. Wenn also in den Kriterienbereich eine Zahl im Standardformat eingegeben wird, in der Tabelle aber diese Spalte mit dem Währungszeichen formatiert ist, hat das für die Suchoperation keine Bedeutung.

Probleme kann es allerdings geben, wenn die Zahl in der Tabelle in der Anzeige gerundet ist, intern aber mit mehreren Nachkommastellen geführt wird. Das kommt z. B. häufig vor, wenn die Zahl das Ergebnis einer Division ist. Wenn im Kriterienbereich `3,33` steht und in der Tabelle das Ergebnis von `10/3` angezeigt wird, wird der Datensatz nicht gefunden. In solch einem Fall kann ein berechnetes Suchkriterium mit einer Formel wie

```
=RUNDEN(F9;2)=3,33
```

helfen. Ansonsten können Sie beim Vergleich mit Zahlen mit den schon aufgeführten Vergleichsoperatoren arbeiten.

Die Bedeutung der statistischen Auswertungsfunktionen wird bei der folgenden Beschreibung nur kurz angedeutet. Ausführlichere Hinweise finden Sie bei den entsprechenden statistischen Funktionen in Kapitel 7, »Statistische Funktionen«.

10.1.7 Statistische Übersicht über eine Datentabelle

Neben einzelnen Abfragen, die Sie mit Hilfe der Datenbankfunktionen für eine Datentabelle einrichten können, besteht auch die Möglichkeit, eine statistische Übersicht über die Eigenschaften der Tabelle einzurichten. Das folgende Beispiel zeigt eine Auswertung der Tabelle in Abbildung 10.1, die alle wichtigen statistischen Kennzahlen en bloc anbietet. Um die Formeln zu vereinfachen, sind der Datenbereich und der Kriterienbereich – D1:D2 – vorher benannt worden. Die Formeln in den Zeilen 4 bis 6 sehen dann so aus:

```
=DBANZAHL2(Lagerdaten;;Kriterien)
=DBSUMME(Lagerdaten;6;Kriterien)
=DBMITTELWERT(Lagerdaten;6;Kriterien)

=DBSUMME(Lagerdaten;7;Kriterien)
=DBMITTELWERT(Lagerdaten;7;Kriterien)

=DBMAX(Lagerdaten;7;Kriterien)
=DBMIN(Lagerdaten;7;Kriterien)
```

Zelle D2 kann noch mit einer Gültigkeitsprüfung verknüpft werden, die die möglichen Warengruppen zur Auswahl anbietet. Bleibt das Feld leer, werden die Daten für die gesamte Tabelle berechnet.

	A	B	C	D	E	F	G
1				Warengruppe			
2	Statistische Daten für Warengruppe:						
3				Jalousie Rollo			
4		Anzahl Artikel	10	Summe Umsatz	295.091,80	Maximum Umsatz	87690
5		Summe Bestand	5310	Mittelwert Umsatz	29.509,18	Minimum Umsatz	6000
6		Mittelwert Bestand	531				
7							
8	Artn	Bezeichnung	Warengruppe	Material	Mindestbestanc	Bestand	Umsatz_lfd_Jah
9	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
10	7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
11	8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
12	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
13	5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
14	8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
15	5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
16	6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
17	5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
18	7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.5 Statistische Werte zu einer Tabelle

10.2 Referenz der Datenbankfunktionen

DBANZAHL()

DCOUNT()

Syntax: `DBANZAHL(Datenbank; Datenbankfeld; Suchkriterien)`

Beispiel: `=DBANZAHL(Lagerdaten;A4;A4:A5)`

Ergebnis: in dem Beispiel in Abbildung 10.6 der Wert `10`, wenn als Kriterium `Bestand >0` eingetragen ist

Die Funktion `DBANZAHL()` ermittelt die Anzahl der Zellen in der mit `Datenbankfeld` angegebenen Spalte, die numerische Werte enthalten und mit den über das Argument `Suchkriterien` festgelegten Kriterien übereinstimmen. Wird für `Datenbankfeld` keine Angabe gemacht, werden alle Datensätze gezählt, die die Kriterien erfüllen.

A8 =DBANZAHL(Lagerdaten;A4;A4:A5)

Anzahl der Datenzeilen, die die angegebenen Kriterien erfüllen

Bestand		Warengruppe	Warengruppe
>0		Rollo	Rollo
DBANZAHL()		**DBANZAHL()**	**DBANZAHL2()**
10		0	5

Artnr	Bezeichnung	Warengruppe	Material	Mindestbestand	Bestand	Umsatz_lfd_Jahr
7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.6 Ermittlung der Anzahl

In dem Beispiel in Abbildung 10.6 wird in A8 die Anzahl der numerischen Zellen in der Bestandsspalte ermittelt. In Zelle C8 wird in der Spalte »Warengruppe« gezählt, was `0` ergibt, weil keine numerischen Werte vorhanden sind. Dagegen findet die Funktion `DBANZAHL2()`, die auch nicht numerische Werte zählt, dort mit demselben Kriterium 5 Werte. Siehe auch die Beschreibung der Funktion `ANZAHL()` in Kapitel 7, »Statistische Funktionen«.

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

DBANZAHL2()

DCOUNTA()

Syntax:	DBANZAHL2(**Datenbank; Datenbankfeld; Suchkriterien**)
Beispiel:	=DBANZAHL2(Lagerdaten;"Warengruppe";D4:D5)
Ergebnis:	in dem Beispiel in Abbildung 10.6 der Wert 5, wenn im Kriterienbereich unter Warengruppe der Wert Rollo eingetragen ist.

Die Funktion DBANZAHL2() ermittelt die Anzahl der Zellen in der mit Datenbankfeld angegebenen Spalte, die Werte enthalten – also nicht leer sind – und mit den über das Argument Suchkriterien festgelegten Kriterien übereinstimmen. Wird für Datenbankfeld keine Angabe gemacht, werden alle Datensätze gezählt, die die Kriterien erfüllen.

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch Abbildung 10.6 und die Beschreibung der Funktion ANZAHL2() in Kapitel 7, »Statistische Funktionen«.

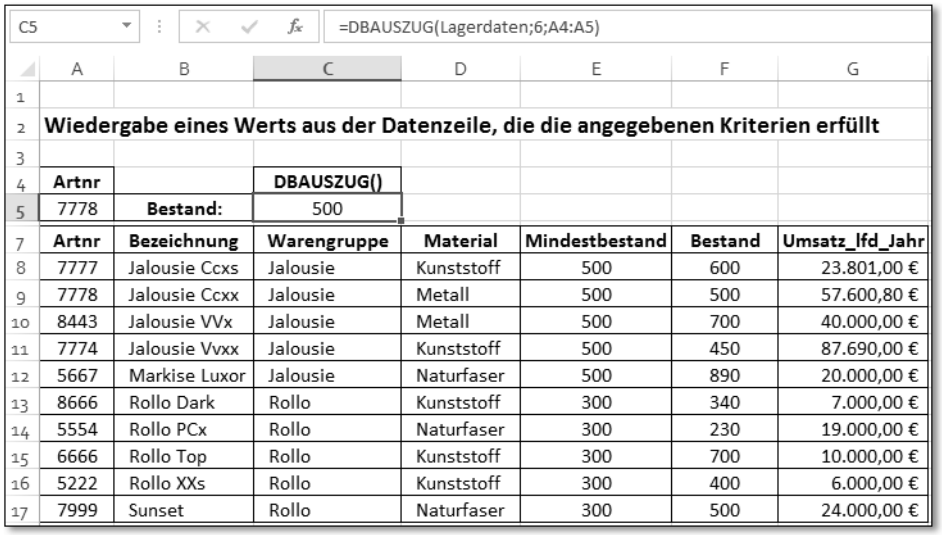
DBAUSZUG()

DGET()

Syntax:	DBAUSZUG(**Datenbank; Datenbankfeld; Suchkriterien**)
Beispiel:	=DBAUSZUG(Lagerdaten;6;A4:A5)
Ergebnis:	in dem in Abbildung 10.7 gezeigten Beispiel ein Bestand von 500 für die im Kriterienbereich angegebene Artikelnummer

Die Funktion DBAUSZUG() nimmt keine statistische Auswertung vor, sondern liefert einfach einen einzelnen Wert für das mit Datenbankfeld angegebene Feld des Datensatzes, der mit den Suchkriterien übereinstimmt.

Liefern die Argumente `Datenbank`, `Datenbankfeld` und `Suchkriterien` keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler `#WERT!`. Findet die Funktion keinen Datensatz, der mit dem Suchkriterium übereinstimmt, wird die Fehlermeldung `#WERT!` ausgegeben. Etwas gewöhnungsbedürftig ist das Verhalten der Funktion, wenn sie mehr als einen Datensatz findet, der mit dem Kriterium übereinstimmt. In diesem Fall wird die Fehlermeldung `#Zahl!` ausgegeben.



C5 =DBAUSZUG(Lagerdaten;6;A4:A5)

Wiedergabe eines Werts aus der Datenzeile, die die angegebenen Kriterien erfüllt

Artnr		DBAUSZUG()
7778	Bestand:	500

Artnr	Bezeichnung	Warengruppe	Material	Mindestbestand	Bestand	Umsatz_lfd_Jahr
7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.7 Daten eines Felds aus einer Datenzeile ausgeben

DBMAX()

DMAX()

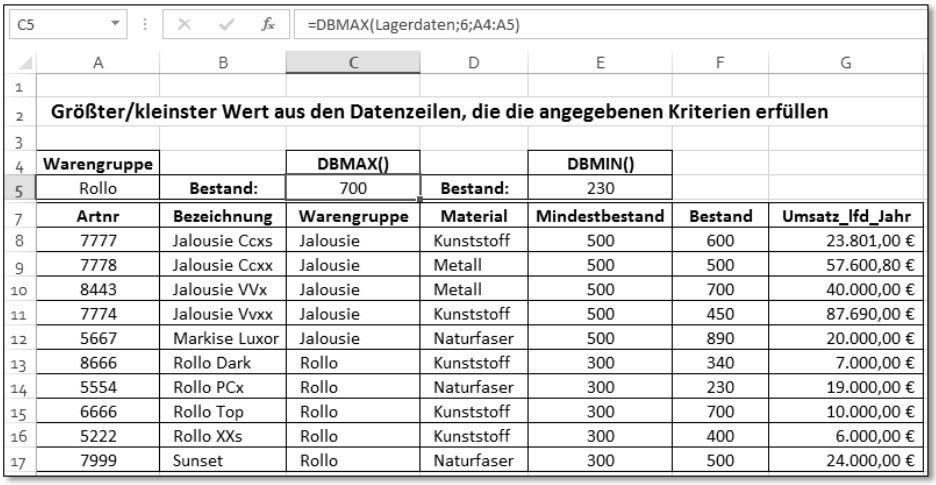
Syntax: `DBMAX(`**`Datenbank; Datenbankfeld; Suchkriterien`**`)`

Beispiel: `=DBMAX(Lagerdaten;6;A4:A5)`

Ergebnis: in dem in Abbildung 10.8 gezeigten Beispiel `700` als höchster Bestand eines Artikels aus der Warengruppe `Rollo`

Die Funktion `DBMAX()` liefert den größten Wert in der mit `Datenbankfeld` angegebenen Spalte, der den angegebenen Kriterien entspricht. In dem abgebildeten Beispiel wird der Artikel aus der Warengruppe `Rollo` gefunden, der den höchsten Bestand hat.

Soll einfach der größte Wert in einer Spalte ausgegeben werden, lassen Sie im Kriterienbereich die Zelle unter dem Spaltennamen leer.



C5 =DBMAX(Lagerdaten;6;A4:A5)

	A	B	C	D	E	F	G
1							
2	**Größter/kleinster Wert aus den Datenzeilen, die die angegebenen Kriterien erfüllen**						
3							
4	**Warengruppe**		**DBMAX()**		**DBMIN()**		
5	Rollo	**Bestand:**	700	**Bestand:**	230		
7	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Mindestbestand**	**Bestand**	**Umsatz_lfd_Jahr**
8	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
9	7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
10	8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
11	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
12	5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
13	8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
14	5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
15	6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
16	5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
17	7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.8 Maximal- und Minimalwerte suchen

Liefern die Argumente `Datenbank`, `Datenbankfeld` und `Suchkriterien` keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler `#WERT!`.

Siehe auch die Beschreibung der Funktion `MAX()` in Kapitel 7, »Statistische Funktionen«.

DBMIN()

DMIN()

Syntax:	`DBMIN(`**Datenbank**`;` **Datenbankfeld**`;` **Suchkriterien**`)`
Beispiel:	`=DBMIN(Lagerdaten;6;A4:A5)`
Ergebnis:	in dem in Abbildung 10.8 gezeigten Beispiel `230` als niedrigster Bestand eines Artikels aus der Warengruppe `Rollo`

Die Funktion `DBMIN()` liefert den kleinsten Wert in der mit `Datenbankfeld` angegebenen Spalte, der den angegebenen Kriterien entspricht. In dem Beispiel in Abbildung 10.8 wird der Artikel aus der Warengruppe `Rollo` gefunden, der den geringsten Bestand hat.

Soll einfach der kleinste Wert in einer Spalte ausgegeben werden, lassen Sie im Kriterienbereich die Zelle unter dem Spaltennamen leer.

Liefern die Argumente `Datenbank`, `Datenbankfeld` und `Suchkriterien` keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler `#WERT!`.

Siehe auch Abbildung 10.8 und die Beschreibung der Funktion MIN() in Kapitel 7, »Statistische Funktionen«.

DBMITTELWERT()

DAVERAGE()

Syntax:	DBMITTELWERT(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)
Beispiel:	=DBMITTELWERT(Lagerdaten;7;A4:A5)
Ergebnis:	Liefert in dem in Abbildung 10.9 gezeigten Beispiel den mittleren Umsatz für Artikel der Warengruppe Jalousie.

Die Funktion DBMITTELWERT() liefert den Mittelwert aller Werte in der Spalte Datenbankfeld, deren Datensätze die Suchkriterien erfüllen. Leere Zellen bleiben dabei unberücksichtigt.

C5 =DBMITTELWERT(Lagerdaten;7;A4:A5)

Mittlerer Wert aus den Datenzeilen, die die angegebenen Kriterien erfüllen

Warengruppe		DBMITTELWERT()
Jalousie	**Umsatz:**	45.818,36 €

Artnr	Bezeichnung	Warengruppe	Material	Mindestbestand	Bestand	Umsatz_lfd_Jahr
7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.9 Mittelwert der Umsätze für eine Warengruppe

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

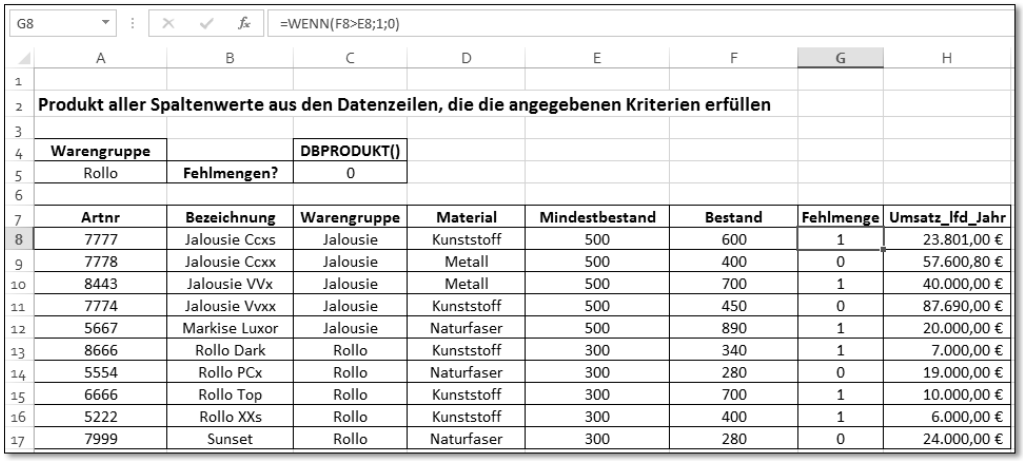
Siehe auch die Beschreibung der Funktion MITTELWERT() in Kapitel 7, »Statistische Funktionen«.

DBPRODUKT()

DPRODUCT()

Syntax:	DBPRODUKT(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)
Beispiel:	=DBPRODUKT(Lagerdaten2;7;A4:A5)
Ergebnis:	Liefert in dem in Abbildung 10.10 gezeigten Beispiel einen Hinweis, dass es Fehlmengen gibt.

Die Funktion DBPRODUKT() multipliziert alle Werte in der Spalte Datenbankfeld derjenigen Datensätze, die die Suchkriterien erfüllen. In dem Beispiel in Abbildung 10.10 ist die Lagerdatenbank um ein berechnetes Feld »Fehlmenge« erweitert worden, das mit einer WENN()-Funktion prüft, ob der Mindestbestand unterschritten ist. Wenn ja, erhält die Zelle den Wert 0, sonst den Wert 1. Werden nun alle Werte dieses Felds für die angegebene Warengruppe miteinander multipliziert, ergibt die Funktion immer den Wert 0, wenn mindestens für einen Artikel Fehlmengen existieren.



G8 =WENN(F8>E8;1;0)

	A	B	C	D	E	F	G	H
1								
2	**Produkt aller Spaltenwerte aus den Datenzeilen, die die angegebenen Kriterien erfüllen**							
3								
4	**Warengruppe**		**DBPRODUKT()**					
5	Rollo	**Fehlmengen?**	0					
6								
7	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Mindestbestand**	**Bestand**	**Fehlmenge**	**Umsatz_lfd_Jahr**
8	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	1	23.801,00 €
9	7778	Jalousie Ccxx	Jalousie	Metall	500	400	0	57.600,80 €
10	8443	Jalousie VVx	Jalousie	Metall	500	700	1	40.000,00 €
11	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	0	87.690,00 €
12	5667	Markise Luxor	Jalousie	Naturfaser	500	890	1	20.000,00 €
13	8666	Rollo Dark	Rollo	Kunststoff	300	340	1	7.000,00 €
14	5554	Rollo PCx	Rollo	Naturfaser	300	280	0	19.000,00 €
15	6666	Rollo Top	Rollo	Kunststoff	300	700	1	10.000,00 €
16	5222	Rollo XXs	Rollo	Kunststoff	300	400	1	6.000,00 €
17	7999	Sunset	Rollo	Naturfaser	300	280	0	24.000,00 €

Abbildung 10.10 Prüfen, ob Fehlmengen existieren

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch die Beschreibung der Funktion PRODUKT() in Kapitel 5, »Mathematische und trigonometrische Funktionen«.

DBSTDABW()

DSTDEV()

Syntax: DBSTDABW(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)

Beispiel: =DBSTDABW(Lagerdaten;6;A7:A8)

Ergebnis: Liefert in dem in Abbildung 10.11 gezeigten Beispiel die Standardabweichung vom Mittelwert des Bestands für Artikel der Warengruppe Jalousie.

Die Funktion DBSTDABW() berechnet die Standardabweichung der Spalte Datenbankfeld für diejenigen Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Stichprobe behandelt.

C5 =DBSTDABW(Lagerdaten;7;$A$4:$A$5)

	A	B	C	D	E	F	G
1							
2	**Standardabweichung der Werte aus den Datenzeilen, die die angegebenen Kriterien erfüllen**						
3							
4	**Warengruppe**		**DBSTDABW()**	**DBSTDABWN()**			
5	Rollo	**Umsatz_lfd_Jahr:**	7918,33	7082,37			
7	**Warengruppe**	**Bestand:**	175	157			
8	Jalousie	**Basis:**	Stichprobe	Grundgesamtheit			
10	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Mindestbestand**	**Bestand**	**Umsatz_lfd_Jahr**
11	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
12	7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
13	8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
14	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
15	5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
16	8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
17	5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
18	6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
19	5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
20	7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.11 Berechnung der Standardabweichung vom Mittelwert

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch die Beschreibung der Funktion STABW.S() in Kapitel 7, »Statistische Funktionen«.

DBSTDABWN()

DSTDEVP()

Syntax:	DBSTDABWN(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)
Beispiel:	=DBSTDABWN(Lagerdaten;6;A7:A8)
Ergebnis:	Liefert in dem in Abbildung 10.11 gezeigten Beispiel die Standardabweichung vom Mittelwert des Bestands für Artikel der Warengruppe Jalousie.

Die Funktion DBSTDABWN() berechnet die Standardabweichung der Spalte Datenbankfeld für diejenigen Datensätze, die die Suchkriterien erfüllen. Dabei werden die Daten als Grundgesamtheit behandelt.

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch die Beschreibung der Funktion STABW.N() in Kapitel 7, »Statistische Funktionen«.

DBSUMME()

DSUM()

Syntax:	DBSUMME(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)
Beispiel:	=DBSUMME(Lagerdaten;7;A4:A5)
Ergebnis:	Liefert in dem in Abbildung 10.12 gezeigten Beispiel die Summe der Umsätze für Artikel der Warengruppe Jalousie.

Die Funktion DBSUMME() summiert alle Werte derjenigen Datensätze in der Spalte Datenbankfeld, die die Suchkriterien erfüllen.

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch die Beschreibung der Funktion SUMME() in Kapitel 5, »Mathematische und trigonometrische Funktionen«.

	A	B	C	D	E	F	G
1							
2	**Summe der Werte aus den Datenzeilen, die die angegebenen Kriterien erfüllen**						
3							
4	**Warengruppe**		**DBSUMME()**				
5	Jalousie	**Umsatz:**	229.091,80 €				
8	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Mindestbestand**	**Bestand**	**Umsatz_lfd_Jahr**
9	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
10	7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
11	8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
12	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
13	5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
14	8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
15	5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
16	6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
17	5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
18	7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.12 Summe der Umsätze einer Warengruppe

DBVARIANZ()

DVAR()

Syntax: `DBVARIANZ(`**`Datenbank`**`; `**`Datenbankfeld`**`; `**`Suchkriterien`**`)`

Beispiel: `=DBVARIANZ(Lagerdaten;7;A4:A5)`

Ergebnis: Liefert in dem in Abbildung 10.13 gezeigten Beispiel die Varianz der Umsätze für Artikel der Warengruppe `Jalousie`.

	A	B	C	D	E	F	G
1							
2	**Varianz der Werte aus den Datenzeilen, die die angegebenen Kriterien erfüllen**						
3							
4	**Warengruppe**		**DBVARIANZ()**	**DBVARIANZEN()**			
5	Jalousie	**Umsatz:**	769316324,05	615453059,24			
7		**Basis:**	Stichprobe	Grundgesatzheit			
9	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Mindestbestand**	**Bestand**	**Umsatz_lfd_Jahr**
10	7777	Jalousie Ccxs	Jalousie	Kunststoff	500	600	23.801,00 €
11	7778	Jalousie Ccxx	Jalousie	Metall	500	500	57.600,80 €
12	8443	Jalousie VVx	Jalousie	Metall	500	700	40.000,00 €
13	7774	Jalousie Vvxx	Jalousie	Kunststoff	500	450	87.690,00 €
14	5667	Markise Luxor	Jalousie	Naturfaser	500	890	20.000,00 €
15	8666	Rollo Dark	Rollo	Kunststoff	300	340	7.000,00 €
16	5554	Rollo PCx	Rollo	Naturfaser	300	230	19.000,00 €
17	6666	Rollo Top	Rollo	Kunststoff	300	700	10.000,00 €
18	5222	Rollo XXs	Rollo	Kunststoff	300	400	6.000,00 €
19	7999	Sunset	Rollo	Naturfaser	300	500	24.000,00 €

Abbildung 10.13 Varianz der Umsätze einer Warengruppe

Die Funktion DBVARIANZ() berechnet die Varianz der Spalte Datenbankfeld für diejenigen Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Stichprobe behandelt.

Siehe auch die Beschreibung der Funktion VAR.S() in Kapitel 7, »Statistische Funktionen«.

DBVARIANZEN()

DVARP()

Syntax:	DBVARIANZEN(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)
Beispiel:	=DBVARIANZEN(Lagerdaten;7;A4:A5)
Ergebnis:	Liefert in dem in Abbildung 10.13 gezeigten Beispiel die Varianz der Umsätze für Artikel der Warengruppe Jalousie.

Die Funktion DBVARIANZEN() berechnet die Varianz der Spalte Datenbankfeld für diejenigen Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Grundgesamtheit behandelt.

Liefern die Argumente Datenbank, Datenbankfeld und Suchkriterien keine gültigen Angaben zu den entsprechenden Bereichen im Tabellenblatt, liefert die Funktion den Fehler #WERT!.

Siehe auch Abbildung 10.13 und die Beschreibung der Funktion VAR.P() in Kapitel 7, »Statistische Funktionen«.

11 Cube-Funktionen

Funktion	Seite	Funktion	Seite
CUBEELEMENT()	739	CUBEMENGENANZAHL()	743
CUBEELEMENTEIGENSCHAFT()	740	CUBERANGELEMENT()	744
CUBEKPIELEMENT()	740	CUBEWERT()	744
CUBEMENGE()	742		

Seit Excel 2007 gibt es die Kategorie der Cube-Funktionen. Dabei geht es hauptsächlich um die Möglichkeit, von Excel aus auf Daten zuzugreifen, die in mehrdimensionalen Datenstrukturen auf einem Datenbankserver zusammengestellt sind. Diese werden *Cubes* genannt.

Neu in Excel 2013 ist, dass sich diese Funktionen nun auch auf Datenmodelle anwenden lassen, die in Excel intern aus mehreren Tabellen zusammengestellt worden sind, um daraus Pivot-Tabellen zu erzeugen. Dies wird in Kapitel 16, »Analyse mit Pivot-Tabellen und -Diagrammen«, beschrieben.

11.1 Einsatzbereiche für Cube-Funktionen

Solche »Datenwürfel« fassen Daten nach ganz unterschiedlichen Dimensionen zusammen: personenbezogen, sachlich, räumlich oder zeitlich. Dadurch wird es möglich, Querbezüge zwischen den Daten herzustellen und Fragen zu beantworten wie: Welche Artikel werden von einer bestimmten Altersgruppe besonders bevorzugt? Entwickeln sich die Verkäufe in allen Vertriebsgebieten gleich stark, oder gibt es regionale Unterschiede? Welche Produktlinie liegt im Trend? Außerdem besteht die Möglichkeit, bestimmte Indikatoren einzurichten, die kritische Werte kontrollieren oder frühzeitig auf Fehlentwicklungen oder auch auf besondere Chancen aufmerksam machen.

Der Einsatz von Cubes ähnelt dem Einsatz von Pivot-Tabellen in Excel, und es ist deshalb auch nicht verwunderlich, dass solche *OLAP*-Objekte direkt als Datenquelle für Pivot-Tabellen verwendet werden können. Wenn es sich aber um sehr große Datenmengen handelt, ist es häufig sinnvoll, das Pivotieren schon auf dem Datenbankserver vorzunehmen oder wenigstens vorzubereiten, so dass die Weiterverarbeitung in Excel anschließend mit kleineren Datenmengen vorgenommen werden kann.

Anders als bei der Ausgabe von Cube-Daten in einer Pivot-Tabelle, bei der bestimmte Strukturen vorgegeben sind, erlauben die Cube-Funktionen eine freie Platzierung der aus einem Cube übernommenen Daten in ein Arbeitsblatt. Sie ergänzen insofern die Möglichkeiten, die die Pivot-Tabellen bieten.

11.2 Voraussetzungen für den Einsatz von Cube-Funktionen

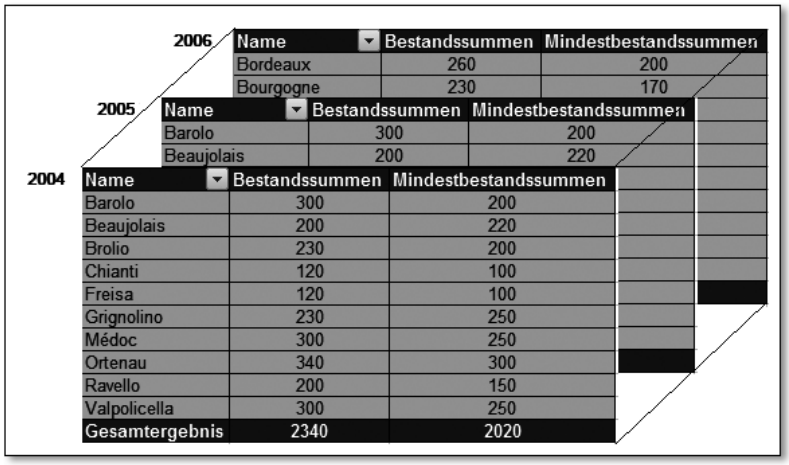
Um die Cube-Funktionen im Zusammenspiel mit einem Microsoft SQL Server nutzen zu können, müssen einige Voraussetzungen geschaffen werden, die hier zunächst kurz skizziert werden sollen. Die SQL-Server von Microsoft unterstützen seit der Version 7 mit den *Analysis Services* das *Online Analytical Processing*, kurz *OLAP*. Dabei handelt es sich um Analysemethoden für die Auswertung von umfangreichen Unternehmensdaten.

Im Kern geht es dabei darum, über die bei den relationalen Datentabellen zwangsläufig vorgegebene Zweidimensionalität hinauszukommen. Dazu werden, getrennt von den normalen Datentransaktionen in den SQL-Datenbanken, vorhandene Daten aus relationalen Datentabellen in einer Art Meta-Datenbank zu mehrdimensionalen Datenstrukturen zusammengeführt, für die die Bezeichnung *Cube* verwendet wird.

11.2.1 OLAP-Cubes

Für *OLAP*-Lösungen ist typisch, dass sie Informationen aus mehreren relationalen Datenquellen zusammenführen, um Datenbestände unter flexibel wählbaren Gesichtspunkten zusammenfassen (aggregieren) und auswerten zu können.

Das technische Konzept für die *OLAP*-Datenbank basiert dabei auf dem Modell der Cubes, der mehrdimensionalen Datenwürfel. Natürlich handelt es sich bei diesem Begriff um ein Bild, das allerdings bei dreidimensionalen Datenstrukturen noch gut grafisch dargestellt werden kann, wie Abbildung 11.1 andeutet.



2006

Name	Bestandssummen	Mindestbestandssummen
Bordeaux	260	200
Bourgogne	230	170

2005

Name	Bestandssummen	Mindestbestandssummen
Barolo	300	200
Beaujolais	200	220

2004

Name	Bestandssummen	Mindestbestandssummen
Barolo	300	200
Beaujolais	200	220
Brolio	230	200
Chianti	120	100
Freisa	120	100
Grignolino	230	250
Médoc	300	250
Ortenau	340	300
Ravello	200	150
Valpolicella	300	250
Gesamtergebnis	2340	2020

Abbildung 11.1 Kleiner Datenwürfel, bei dem die Zeit die dritte Dimension darstellt

11.2.2 SQL Server Data Tools

Für den Umgang mit *OLAP* und Cubes steht unter dem Namen *Microsoft SQL Server Data Tools* eine spezielle Variante der Entwicklungsumgebung *Microsoft Visual Studio 2012* zur Verfügung, die entsprechende Vorlagen für BI-Projekte anbietet und Tools für ihr Design. Sie finden die Tools über die Seite *msdn.microsoft.com/de-de/data/hh297027*. Im Zentrum steht dabei der Cube-Designer. Es soll an dieser Stelle wenigstens angedeutet werden, welche Konzepte bei der Erstellung von Cubes in der Visual-Studio-Umgebung maßgeblich sind, weil so die Syntax der im Folgenden beschriebenen Funktionen verständlich wird.

11.2.3 Dimensionen

Beim Entwurf von Cubes werden insbesondere die Begriffe *Dimension* und *Measure* unterschieden. Dimensionen sind maßgeblich für die Anordnung der Daten. Dabei kann es sich um sachliche Kriterien handeln, etwa die Anordnung von Produkten, wobei verschiedene Warengruppen oder Produktlinien als Zwischenebene dieser Dimension eingerichtet werden können. Eine zweite Dimension könnte die räumliche Verteilung der jeweiligen Geschäftstätigkeiten, eine dritte die zeitliche Verteilung darstellen, wobei wiederum mehrstufige Hierarchien von Elementen möglich sind, beispielsweise Jahre <- Quartale <- Monate <- Wochen. Eine weitere Dimension könnte die personelle Anordnung sein: Kunden, Lieferanten etc.

11.2.4 Measures

Der Begriff *Measure*, der in den deutschen Texten manchmal auch als *Kennzahl* oder *Maßzahl* wiedergegeben wird, bezeichnet die Werte, auf die aus der Perspektive der jeweiligen Dimensionen zugegriffen wird, beispielsweise die Umsatz- oder Absatzdaten in einem Zeitraum oder in einer Region. Diese Werte können als Einzelwerte oder schon als aggregierte Werte vorliegen.

Die als Measures bezeichneten Werte werden innerhalb der Analysis Services von SQL Server 2012 als Faktentabellen geführt, während die Elementwerte jeder Dimension jeweils in Dimensionentabellen vorgehalten werden. Die Verknüpfung erfolgt über Schlüssel, wobei der Primärschlüssel aus einer Dimensionentabelle als Fremdschlüssel in der Faktentabelle auftaucht. Dabei sind verschiedene Strukturschemas möglich, die einfachste Struktur wird als Sternschema bezeichnet. Dabei liegt für jede Dimension eine Tabelle vor, und alle Dimensionentabellen sind mit der Faktentabelle verknüpft. Neben Dimensionen und Measures, die direkt aus vorhandenen Tabellen verfügbar sind, lassen sich zusätzlich berechnete Dimensionen und Measures erzeugen.

Ausnahmsweise wird in den folgenden Abschnitten auf die von Microsoft zur Verfügung gestellte Beispieldatenbank *Adventure Works DW2012* zurückgegriffen, um nicht zu viel Zeit mit der Beschreibung der Einrichtung von eigenen Datentabellen zu verlieren. Diese Beispieldaten sind komplex genug, um erste Erfahrungen im Einsatz von Cube-Funktionen zu sammeln. Sie lassen sich über *http://msftdbprodsamples.codeplex.com* herunterladen und separat installieren.

11.2.5 Ein Cube-Projekt anlegen

Ist die Datenbank installiert, kann über die angesprochenen Microsoft SQL Server Data Tools ein Projekt mit dem Projekttyp **Analysis Services-Projekt für mehrdimensionale und Data Mining-Modelle** angelegt werden, um darin einen Cube zu definieren und bereitzustellen, auf den die unten beschriebenen Cube-Funktionen zugreifen können. Im **Projektmappen-Explorer** erscheint ein vorgegebener Ordnerbaum für die Komponenten eines solchen Projekts.

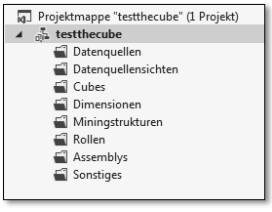


Abbildung 11.2 Komponenten für ein Cube-Projekt

Die Abfolge bis zur Bereitstellung eines Cubes soll hier nur angedeutet werden. Der erste Schritt ist in der Regel die Definition mindestens einer Datenquelle, die das Material für den vorgesehenen Cube liefert. Dabei wird die Verbindungszeichenfolge festgelegt, die bei der Eröffnung der Verbindung benötigt wird. Im Dialog **Verbindungs-Manager** wird unter **Anbieter** der Datenprovider abgefragt. Hier übernehmen Sie die Einstellung **OLE DB systemeigen\SQL Server Native Client 11.0**.

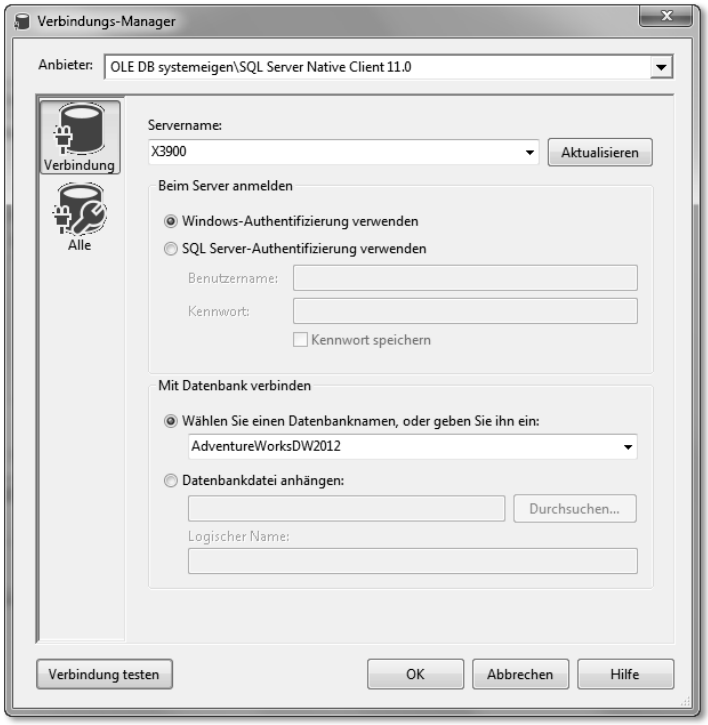


Abbildung 11.3 Herstellen der Verbindung zur Beispieldatenbank

Unter **Servername** kann »localhost«, die IP-Adresse oder der Name des SQL-Servers eingegeben werden. Für das Anmeldeverfahren wird die Windows-Authentifizierung verwendet. Wenn die Verbindung zum Server gelungen ist, kann die Beispieldatenbank »Adventure Works DW2012« ausgewählt werden.

Der nächste Schritt ist die Festlegung einer brauchbaren Datenquellensicht. Dazu wird über das Kontextmenü von **Datenquellensichten** die Option **Neue Datenquellensicht** aufgerufen. Im Dialog **Datenquellensicht-Assistent** können die für den Cube vorgesehen Dimensionentabellen und Faktentabellen bestimmt werden. Die Ersteren sind an dem vorgegebenen Präfix *Dim* zu erkennen, Letztere an dem Präfix *Fact*.

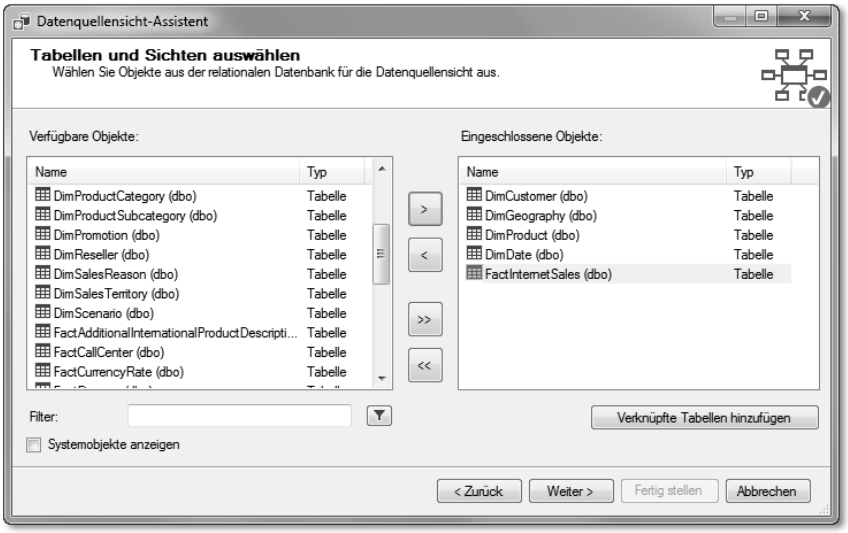


Abbildung 11.4 Auswahl der für den Cube vorgesehenen Tabellen

Nach diesen Vorbereitungen lässt sich aus der ausgewählten Sicht mit der Option **Neuer Cube**, die über das Kontextmenü von **Cubes** angeboten wird, ein Cube erstellen. Dabei wählen Sie zunächst die Erstellungsmethode. In diesem Fall lassen sich die vorhandenen Tabellen aus der vorher definierten Ansicht verwenden.

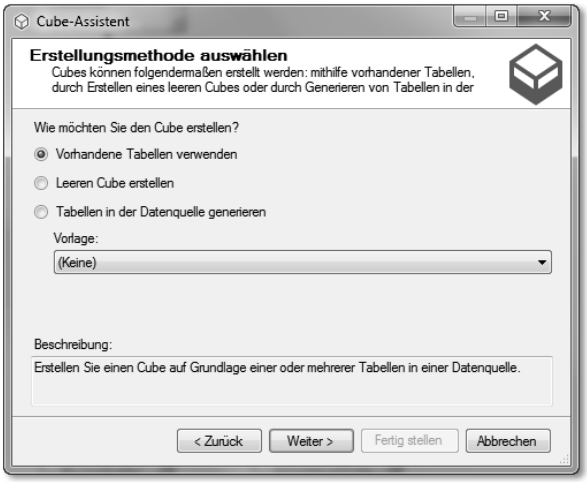


Abbildung 11.5 Wahl der Erstellungsmethode für den Cube

Für einen Cube muss mindestens eine Faktentabelle und eine Dimensionentabelle ausgewählt werden.

Abbildung 11.6 Der Cube-Assistent kann die Verwendung bestimmter Tabellen vorschlagen.

Im nächsten Schritt lassen sich aus den Spalten der angegebenen Faktentabelle diejenigen auswählen, die tatsächlich als Measures für den Cube verwendet werden sollen. Zum Schluss muss noch ein Name für den Cube vergeben werden. Im **Projektmappen-Explorer** sind schließlich alle für das Projekt benötigten Komponenten sichtbar.

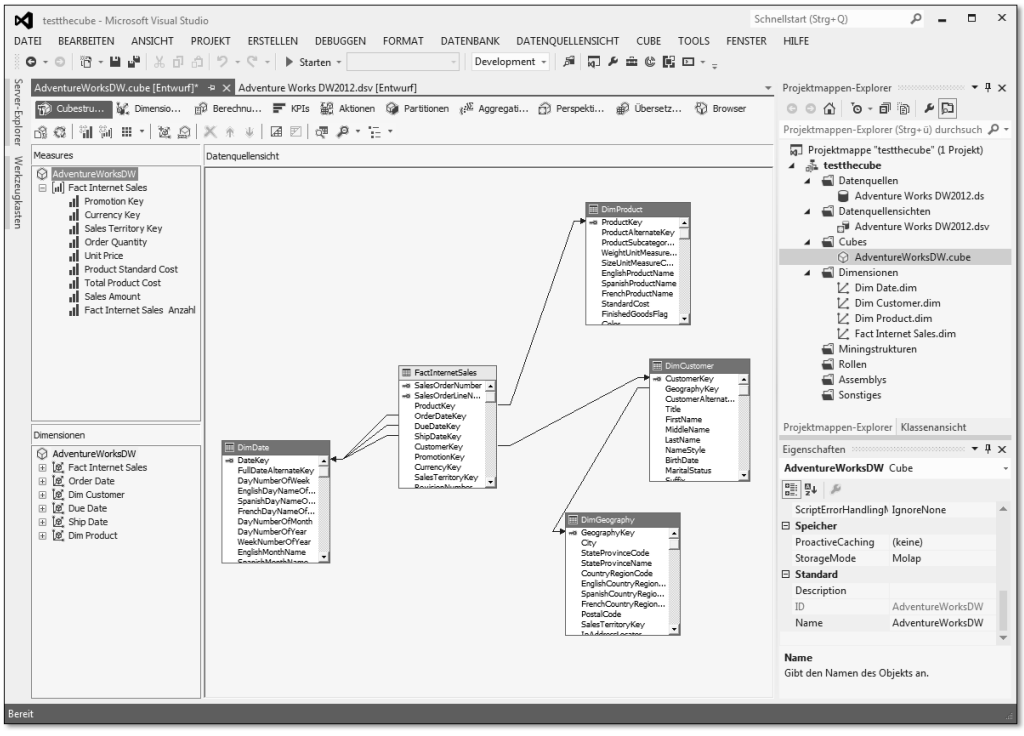


Abbildung 11.7 Das Diagramm zeigt die Struktur des Cubes.

Abbildung 11.7 zeigt, dass beispielsweise drei verschiedene Zeitbeziehungen zwischen der »Internet Sales«-Tabelle und der »Date«-Tabelle vorhanden sind, weil jedem Verkauf ein Bestelldatum, ein Fälligkeitsdatum und ein Lieferdatum zugeordnet sind.

11.2.6 Bereitstellen des Cubes

Der letzte Schritt ist die Erstellung und die Bereitstellung des Cubes. Beim Erstellen werden die für Dimensionen und Measures ausgewählten Daten zusammengestellt bzw. berechnet, in den Cube kopiert und entsprechend gespeichert, so dass der Cube für Anwendungen zur Verfügung steht, die auf die entsprechende Instanz des SQL-Servers zugreifen wollen. Beim Bereitstellen werden die definierten Objekte in einer Instanz von Analysis Services verfügbar gemacht, so dass beispielsweise von Excel darauf zugegriffen werden kann.

Gestartet werden die Erstellung sowie Bereitstellung über **Erstellen • <Projektname> erstellen** und **<Projektname> bereitstellen**. Nach der Bereitstellung des Cubes kann auch das vom Cube-Designer zur Verfügung gestellte Register **Browser** genutzt werden, um innerhalb der Visual-Basic-Umgebung Daten im Cube in einer Form anzuzeigen, die große Ähnlichkeit mit der Darstellung der Pivot-Tabellen in Excel hat.

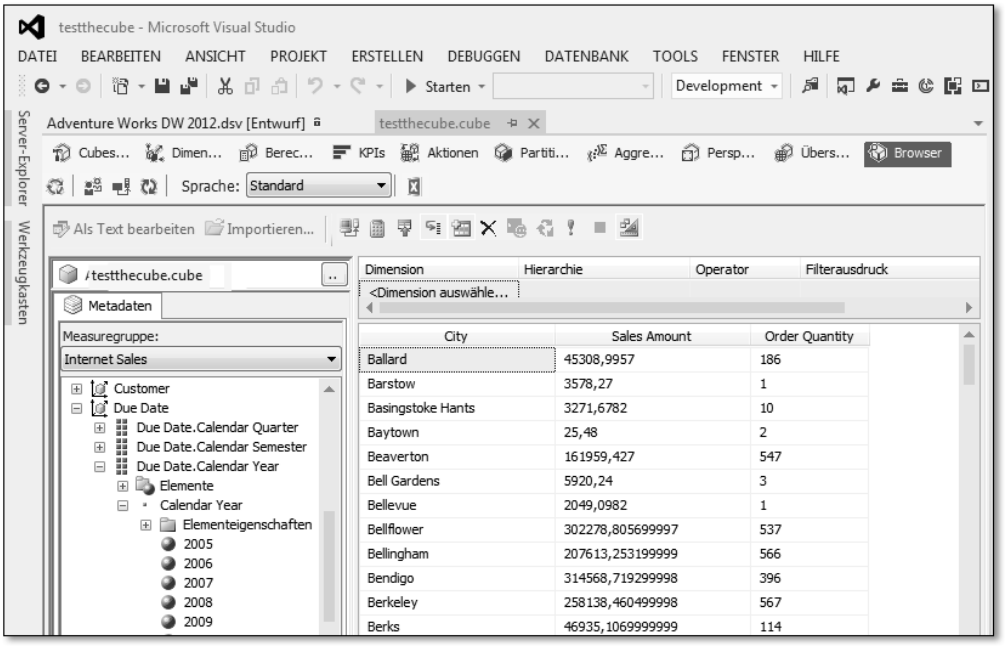


Abbildung 11.8 Anzeige von Daten aus dem Cube auf dem Register **Browser**

Abbildung 11.8 zeigt beispielsweise, wie sich die Internetverkäufe auf die verschiedenen Städte verteilen.

11.2.7 Die Abfragesprache MDX

Während für die Abfrage von relationalen Datentabellen üblicherweise SQL verwendet wird, gilt für die Handhabung von mehrdimensionalen Datenstrukturen eine spezielle Abfragesprache, die als *Multidimensional Expressions* (*MDX*) bezeichnet wird. MDX ist eine von Microsoft geförderte Abfragesprache für multidimensionale Datenbanken.

Die Syntax einer `SELECT`-Anweisung in MDX gleicht der SQL-Syntax, geht aber von einem anderen Datenmodell aus, da ja statt mit zweidimensionalen Tabellen mit mehrdimensionalen Cubes gearbeitet wird. Die generelle Syntax sieht so aus:

```
SELECT [<Achsenspezifikation>
[,<Achsenspezifikation>...]]
FROM [<Cubespezifikation>]
[WHERE [<Slicerspezifikation>]]
```


Die Anweisung arbeitet anstelle der Spalten in SQL mit Abfrageachsen, wobei die ersten fünf Achsen über Aliasnamen angesprochen werden können. Jede Achsenspezifikation definiert eine Dimension. Die Syntax ist `<Menge> ON <Name der Achse>`. Für die ersten fünf Achsen werden die Aliasnamen `COLUMNS | ROWS | PAGES | SECTIONS | CHAPTERS` verwendet. Die folgende Anweisung benutzt beispielsweise `COLUMNS` und `ROWS` als Aliasnamen für Achsen:

```
SELECT {[Measures].[Sales Amount],
[Measures].[Tax Amount] } ON COLUMNS,
{ [Date].[Fiscal].[Fiscal Year].&[2009],
[Date].[Fiscal].[Fiscal Year].&[2010] } ON ROWS
FROM [Adventure Works]
WHERE ( [Sales Territory].[Southwest] )
```

Alternativ können die Achsen auch über ihre Position angesprochen werden; dazu wird mit einem nullbasierten Index gearbeitet: `AXIS(0)` ist also die erste Achse. Die Numerierung muss immer fortlaufend sein:

```
SELECT {[Measures].[Unit Sales], [Measures].[Store Sales] }
ON AXIS(0),
{ [Time].[2009], [Time].[2010] } ON AXIS(1)
```

In der FROM-Klausel kann jeweils ein Cube angegeben werden. Die FROM-Klausel bestimmt den Cube-Kontext. Dabei kann es sich um einen kompletten Cube oder auch um einen Teilcube handeln.

Während in SQL die WHERE-Klausel im Sinne eines Filters darüber entscheidet, welche Zeilen angezeigt werden, wird die WHERE-Klausel in MDX verwendet, um die Daten über eine bestimmte Dimension oder ein Mitglied derselben einzuschränken. Das wird auch als *Slicing* bezeichnet. Die WHERE-Klausel legt dabei die Slicerachse fest und liefert so in jedem Fall eine klar abgegrenzte Untermenge.

11.2.8 Attribute und Attributhierarchien

Dimensionen bestehen aus Gruppen von Attributen, die auf Spalten in den verwendeten Dimensionentabellen basieren. Diese Gruppen sind in Form von Hierarchien geordnet.

Um Elemente auf den verschiedenen Ebenen einer Dimension innerhalb einer Anweisung ansprechen zu können, wird eine Schreibweise verwendet, bei der Bezeichner durch eckige Klammern gekennzeichnet werden. Punkte trennen Ebenen innerhalb einer Hierarchie. Dabei können untergeordnete Elemente entweder explizit angegeben werden, etwa:

```
[Date].[Calendar Year].[CY 2010]
```

oder mit Hilfe von MDX-Funktionen:

```
[Date].[Calendar Year].LastChild
```

Die eckigen Klammern sind notwendig, wenn der Name eines Elements eine Zahl oder ein Leerzeichen enthält. Die Klammern dürfen aber auch bei einfachen Namen verwendet werden. Der Bezug auf Elemente einer Dimension kann statt über den Namen auch über entsprechende Schlüssel hergestellt werden. Das ist sinnvoll, wenn die Namen innerhalb mehrerer Dimensionen nicht eindeutig sind. Schlüssel werden durch ein vorangestelltes Ampersand-Zeichen (&) gekennzeichnet, etwa:

```
[Time].[1nd half].&[Q2].
```

Elemente können auch durch Berechnungen aus vorhandenen Werten neu gebildet werden. Zur Definition solcher Elemente für eine Abfrage wird das Schlüsselwort WITH verwendet:

```
WITH MEMBER [Measures].[Special Discount] AS [Measures].[Discount Amount] * 1.5
```

In diesem Fall wird das neue Element `[Measures].[Special Discount]`, dessen voll qualifizierter Name zuerst angegeben wird, auf der Basis eines bereits vorhandenen Elements `[Measures].[Discount Amount]` durch Multiplikation mit dem Faktor gebildet. Das berechnete Element gilt nur für die so spezifizierte Abfrage.

11.2.9 Tupel und Mengen

Ein Tupel innerhalb einer multidimensionalen Datenstruktur ist eine geordnete Kollektion von je einem Element für jede Dimension, die in einer Abfrage verwendet wird. Dabei können auch vorgegebene Elemente einbezogen werden. Tupel, die nur aus einem Element einer Dimension bestehen, können in einer vereinfachten Syntax angesprochen werden, etwa:

```
Time.[1st half]
```

Bei Tupeln, die Elemente verschiedener Dimensionen enthalten, wird mit Klammern gearbeitet:

```
(Time.[2nd half], Route.nonground.air)
```


Mehrere Tupel können eine Menge als geordnete Kollektion bilden. Dies geschieht bei der Spezifikation von Achsen. Mengenausdrücke werden durch geschweifte Klammern gekennzeichnet:

```
{(Time.[1st half], Route.nonground.air), (Time.[2nd half], Route.nonground.sea)}
```

Um einen Bereich von Elementen anzugeben, kann mit dem Bereichsoperator gearbeitet werden:

```
{[1st quarter]:[4th quarter]}
```

Mengen können aber auch durch MDX-Funktionen geliefert werden.

```
{Time.Children}
```

liefert beispielsweise die untergeordneten Elemente zu `Time`.

MDX ist eine umfangreiche Sprache, die hier nicht im Detail beschrieben werden kann. Neben zahlreichen Operatoren und Funktionen steht noch eine Reihe von Anweisungen zur Verfügung. Die möglichen Anweisungen sind in drei Gruppen unterteilt: Anweisungen für Skripte, Anweisungen für die Datendefinition und Anweisungen für die Datenbearbeitung.

Skripte werden verwendet, um innerhalb des Cubes bestimmte Berechnungsprozesse durchzuführen. Dabei werden die folgenden Anweisungen verwendet:

Anweisung	Beschreibung
CALCULATE	Berechnet einen Teilcube.
CASE	Erlaubt die bedingte Rückgabe von Werten aus mehreren Vergleichen.
EXISTING	Erzwingt die Auswertung der angegebenen Menge im aktuellen Kontext.
FREEZE	Fixiert die Werte eines Teilcubes auf die aktuell vorhandenen Werte.
IF	bedingte Anweisung
SCOPE	Beschränkt eine Anweisung auf einen Teil des Cubes.

Tabelle 11.1 Skriptanweisungen in MDX

Die vollständige, deutschsprachige MDX-Referenz finden Sie auf folgender Microsoft-Webseite: *http://msdn.microsoft.com/de-de/library/ms145970.aspx*.

11.2.10 Key Performance Indicators

Eine spezielle Funktion, die in einem Cube eingerichtet werden kann, ist ein mit grafischen Hinweisen verbundener Indikator, dessen Anzeige über einen Statusausdruck gesteuert wird. Sinn und Zweck solcher *Key Performance Indicators* (*KPIs*) ist es, auffällige Hinweise auf relevante betriebswirtschaftliche Kennzahlen zu generieren, die über Erfolg und Misserfolg von Geschäftsprozessen entscheiden. Für die Definition von KPIs steht ein spezielles Register zur Verfügung, wie Abbildung 11.9 zeigt.

Die Schaltfläche **Neuer KPI** blendet ein Formular ein, das zunächst einen Namen und die Measuregruppe abfragt, mit der der Indikator verknüpft werden soll. Unter **Wertausdruck** wird ein Wert oder ein Ausdruck angegeben, der beobachtet und mit einem Zielausdruck verglichen werden soll. Ein einfaches Beispiel wäre etwa das Verhältnis der Produktkosten zum Preis der Einheit. Unter **Zielausdruck** wird der angestrebte Planwert oder ein Ausdruck, der diesen Wert liefert, eingegeben.

Für die Anzeige des erreichten Status kann dann ein grafischer **Statusindikator** gewählt werden. Die Anzeige in diesem Indikator wird über einen **Statusausdruck** gesteuert, der Werte zwischen 1 und -1 liefert. 1 steht für sehr gut, -1 für sehr schlecht. Der Ausdruck kann ein MDX-Skriptfragment sein mit verschiedenen Prüfpunkten, die bestimmten Positionen des Indikators entsprechen.

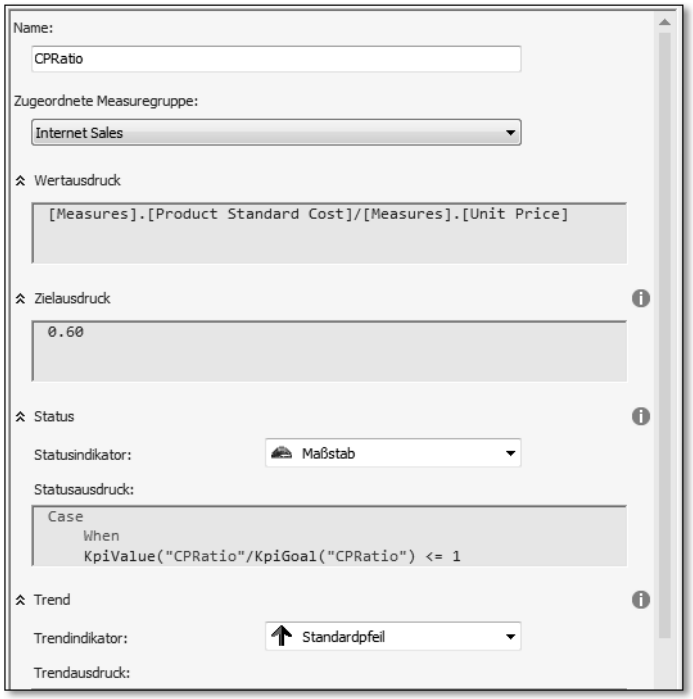


Abbildung 11.9 Definition von KPIs

Ein einfaches Beispiel:

```
Case
 When
  KpiValue("CPRatio")/KpiGoal("CPRatio")<= 1
   Then 1
 When
  KpiValue("CPRatio ")/KpiGoal("CPRatio")>=1
   And
  KpiValue("CPRatio ")/KpiGoal("CPRatio")<=1.1
   Then 0
  Else -1
End
```

Zusätzlich können noch ein **Trendindikator** (ebenfalls Werte zwischen `1` und `-1`) und ein **Trendausdruck** eingegeben werden. Die so erzeugten Daten können über die unten beschriebene Funktion `CUBEKPIELEMENT()` abgefragt werden.

11.2.11 Definieren einer Verbindung zu einem Analysis Services Server

Wird auf einem Microsoft SQL Server ein Cube bereitgestellt, kann von Excel 2013 aus eine Verbindung zu diesem Cube eingerichtet werden, die anschließend für die Cube-Funktionen zur Verfügung steht.

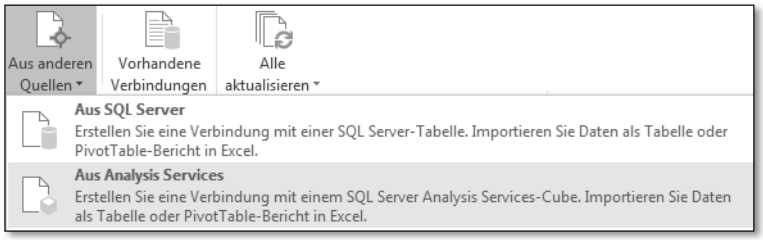


Abbildung 11.10 Zugriff auf Analysis Services

Dazu wird auf dem Register **Daten** in der Gruppe **Externe Daten abrufen** die Schaltfläche **Aus anderen Quellen** verwendet. In der Menüleiste wird die Option **Aus Analysis Services** angeboten, die den Dialog **Datenverbindungs-Assistent** öffnet.

Abbildung 11.11 Aufbau einer Verbindung zu einem SQL-Server

Im ersten Dialog werden der **Servername** und die **Anmeldeinformationen** eingegeben. Kommt die Verbindung zustande, lassen sich anschließend die Analysis Services-Datenbank und der darin enthaltene Cube auswählen. Der Datenbankname entspricht in diesem Fall dem Namen des Projekts, das mit dem BI-Tool angelegt wurde.

Abbildung 11.12 Auswahl des Cubes

Die Daten zu der Verbindung werden in einer *.odc*-Datei gespeichert, so dass immer wieder darauf zurückgegriffen werden kann. Der Verbindungsname wird in fast allen Cube-Funktionen als erstes Argument benötigt.

Abbildung 11.13 Speichern der Datenverbindungsdatei

Sollen die Daten des Cubes für eine Pivot-Tabelle verwendet werden, benutzen Sie die entsprechende Option in dem abschließend angebotenen Dialog **Daten importieren**.

Abbildung 11.14 Dialog zur Übernahme der Daten in eine Pivot-Tabelle

Über **Daten • Verbindungen • Verbindungen** kann jederzeit der Dialog **Arbeitsmappenverbindungen** geöffnet werden, in dem die mit der Arbeitsmappe verknüpften Verbindungsdefinitionen verwaltet werden. Die Schaltfläche **Eigenschaften** zeigt die Detaildaten zu einer ausgewählten Verbindung.

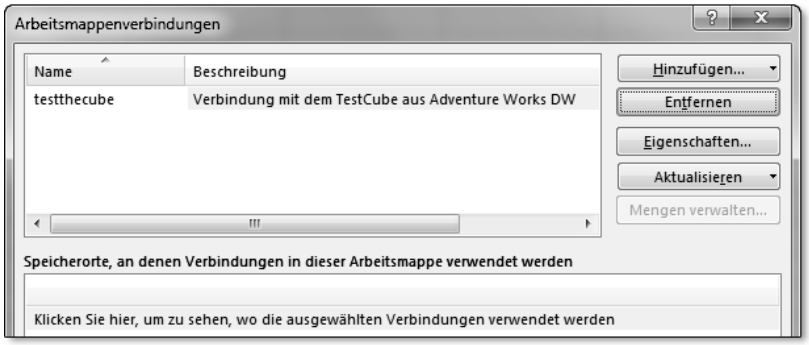


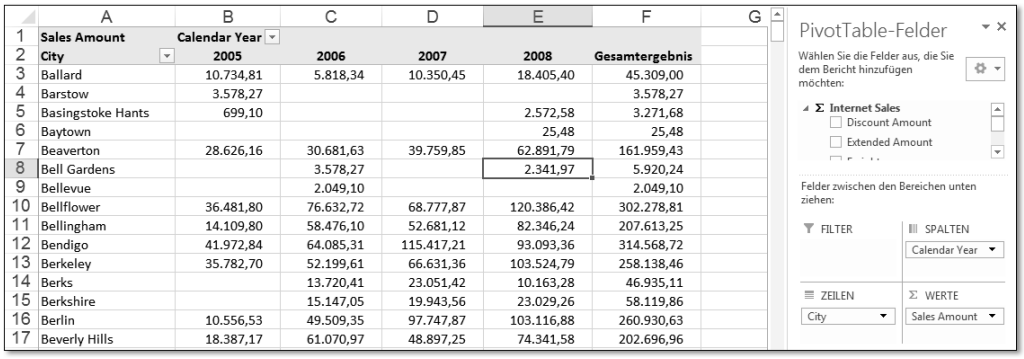
Abbildung 11.15 Verbindungen in einer Arbeitsmappe werden über diesen Dialog verwaltet.

11.2.12 Offline-Cubes

Alternativ zur Onlineverbindung mit einem Cube auf dem Server können Daten auch offline aus *CUB*-Dateien übernommen werden, die vorher aus den Serverdaten generiert wurden. Mit Hilfe einer solchen *CUB*-Datei lassen sich Daten lokal verfügbar ma-

chen, insbesondere für den Fall, dass die Verbindung zum Server, auf dem die Daten gepflegt werden, unterbrochen ist. Die *CUB*-Datei enthält in der Regel eine Untermenge der Quelldaten aus einer *OLAP*-Datenbank.

Um eine *CUB*-Datei zu erstellen, muss zunächst über eine Onlineverbindung eine Pivot-Tabelle mit den Daten aus einem *OLAP*-Cube erstellt werden. Dazu wird eine Pivot-Tabelle über **Einfügen • Tabellen • PivotTable** mit der Option **Externe Datenquelle verwenden** angelegt. Über die Schaltfläche **Verbindung auswählen** wird die vorher definierte Verbindung angegeben. Abbildung 11.16 zeigt eine Pivot-Tabelle mit Daten aus der Beispieldatenbank.



	A	B	C	D	E	F	G
1	**Sales Amount**	**Calendar Year**					
2	**City**	**2005**	**2006**	**2007**	**2008**	**Gesamtergebnis**	
3	Ballard	10.734,81	5.818,34	10.350,45	18.405,40	45.309,00	
4	Barstow	3.578,27				3.578,27	
5	Basingstoke Hants	699,10			2.572,58	3.271,68	
6	Baytown				25,48	25,48	
7	Beaverton	28.626,16	30.681,63	39.759,85	62.891,79	161.959,43	
8	Bell Gardens		3.578,27		2.341,97	5.920,24	
9	Bellevue		2.049,10			2.049,10	
10	Bellflower	36.481,80	76.632,72	68.777,87	120.386,42	302.278,81	
11	Bellingham	14.109,80	58.476,10	52.681,12	82.346,24	207.613,25	
12	Bendigo	41.972,84	64.085,31	115.417,21	93.093,36	314.568,72	
13	Berkeley	35.782,70	52.199,61	66.631,36	103.524,79	258.138,46	
14	Berks		13.720,41	23.051,42	10.163,28	46.935,11	
15	Berkshire		15.147,05	19.943,56	23.029,26	58.119,86	
16	Berlin	10.556,53	49.509,35	97.747,87	103.116,88	260.930,63	
17	Beverly Hills	18.387,17	61.070,97	48.897,25	74.341,58	202.696,96	

Abbildung 11.16 *OLAP*-Daten in einer PivotTable

Um nun aus den online verfügbaren Daten eine *CUB*-Datei zu erstellen, wird, wenn die Pivot-Tabelle ausgewählt ist, über das Register **Analysieren** in der Gruppe **Berechnungen** die Schaltfläche **OLAP-Tools** benutzt.

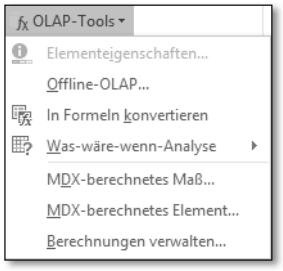


Abbildung 11.17 Die Schaltfläche **OLAP-Tools**

Aus dem Menü wird die Option **Offline-OLAP** verwendet und im Dialog die Schaltfläche **Offlinedatei erstellen**.

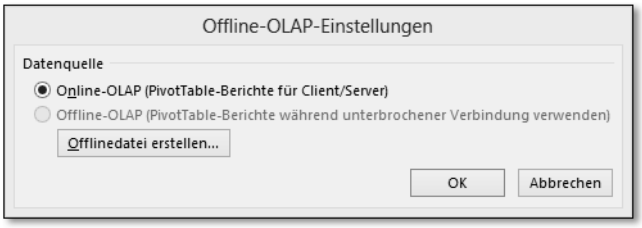


Abbildung 11.18 Dialog für die Erstellung eines Offline-Cubes

Im nächsten Schritt werden die Dimensionen des Cubes angegeben, die in den Offline-Cube mit aufgenommen werden sollen. In der Regel sollten alle Dimensionen übernommen werden, für die gruppierte Elemente vorhanden sind. Innerhalb der Dimensionen lassen sich die Ebenen auswählen. Zwar können Zwischenebenen nicht übersprungen werden, aber es ist möglich, die unteren Ebenen wegzulassen, wenn der Umfang des Cubes eingeschränkt werden soll.

Abbildung 11.19 Auswahl der Dimensionen und Ebenen

Nur bei den Dimensionen, die nicht mit einem Plusfeld ausgestattet sind, müssen alle Ebenen übernommen werden. Diese Dimensionen können also nur ganz ein- oder ausgeschlossen werden.

Im nächsten Schritt können die gewünschten Elemente zu den Measures gewählt werden. Hierbei handelt es sich in der Regel um die Datenfelder, die numerische Werte enthalten. Mindestens ein Measure-Eintrag muss ausgewählt werden, weil sonst die zugeordneten Dimensionen keine Daten aufweisen. Die Eigenschaftenfelder, die den Elementen eines *OLAP*-Cubes zugeordnet sind, werden automatisch in die Cubedatei

aufgenommen. Zuletzt müssen nur noch der Name und der Speicherort der *CUB*-Datei angegeben werden.

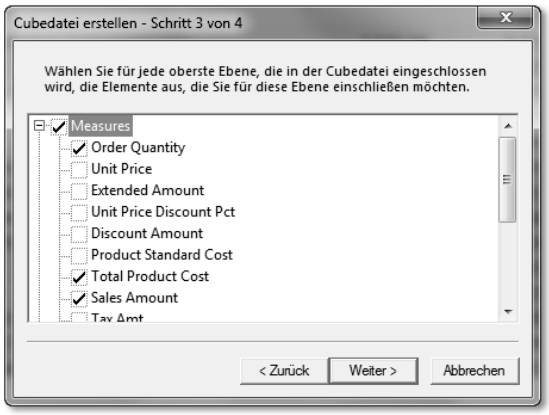


Abbildung 11.20 Auswahl der Measures

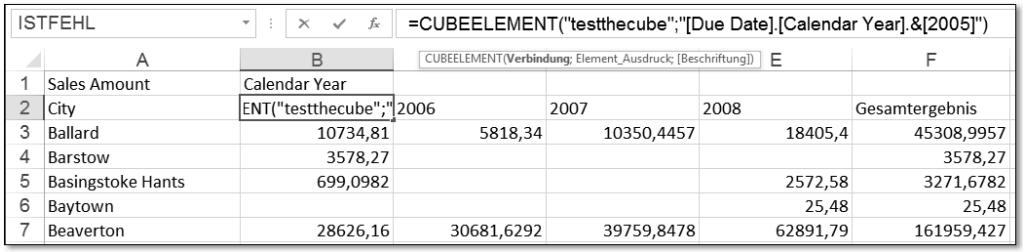
CUB-Dateien können einerseits direkt als Quelldaten für PivotTable-Berichte und PivotCharts verwendet werden, andererseits besteht seit Excel 2007 auch die Möglichkeit, Daten daraus direkt über die neuen Cube-Funktionen auszuwerten. Die Verwendung vorhandener *CUB*-Dateien für Pivot-Berichte ist einfach. Sie brauchen lediglich die *CUB*-Datei als externe Datenquelle anzugeben.

11.2.13 Cube-Formeln automatisch erzeugen

Wenn Sie aus *OLAP*-Daten eine Pivot-Tabelle angelegt haben, gibt es auch noch die Möglichkeit, diese Tabelle in einen normalen Zellbereich zurückzuverwandeln und dabei gleichzeitig automatisch Cube-Funktionen zu generieren, die dafür sorgen, dass die aktuell angezeigten Daten als Ergebnis der entsprechenden Formeln ausgegeben werden.

Solange eine Zelle in der Pivot-Tabelle ausgewählt ist, wird dazu über die schon genannte Schaltfläche **OLAP-Tools** die Option **In Formeln konvertieren** verwendet. Enthält die Pivot-Tabelle einen Berichtsfilter, muss, falls die gesamte Pivot-Tabelle konvertiert werden soll, die dann angebotene Option **Berichtsfilter konvertieren** aktiviert werden, ansonsten werden nur die Spalten- und Zeilenbereiche konvertiert, und der Berichtsfilter kann weiter wie in einer Pivot-Tabelle genutzt werden.

Abbildung 11.21 zeigt die oben noch als PivotTable angezeigte Tabelle nun als normalen Zellbereich mit lauter Cube-Funktionen.



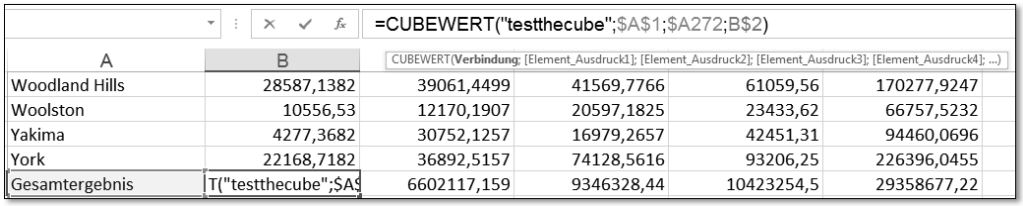
ISTFEHL | =CUBEELEMENT("testthecube";"[Due Date].[Calendar Year].&[2005]")

CUBEELEMENT(**Verbindung**; Element_Ausdruck; [Beschriftung])

	A	B	E	F		
1	Sales Amount	Calendar Year				
2	City	ENT("testthecube";"	2006	2007	2008	Gesamtergebnis
3	Ballard	10734,81	5818,34	10350,4457	18405,4	45308,9957
4	Barstow	3578,27				3578,27
5	Basingstoke Hants	699,0982			2572,58	3271,6782
6	Baytown				25,48	25,48
7	Beaverton	28626,16	30681,6292	39759,8478	62891,79	161959,427

Abbildung 11.21 Die konvertierte PivotTable

Abbildung 11.22 zeigt die generierte Formel für das Gesamtergebnis einer Spalte. Die CUBEWERT()-Funktionen verwenden jeweils eine absolute Zeilen- und Spaltenadresse, d.h., es handelt sich um eine Überkreuzauswertung wie auch in der Pivot-Tabelle.



=CUBEWERT("testthecube";$A$1;$A272;B$2)

CUBEWERT(**Verbindung**; [Element_Ausdruck1]; [Element_Ausdruck2]; [Element_Ausdruck3]; [Element_Ausdruck4]; ...)

A	B				
Woodland Hills	28587,1382	39061,4499	41569,7766	61059,56	170277,9247
Woolston	10556,53	12170,1907	20597,1825	23433,62	66757,5232
Yakima	4277,3682	30752,1257	16979,2657	42451,31	94460,0696
York	22168,7182	36892,5157	74128,5616	93206,25	226396,0455
Gesamtergebnis	T("testthecube";A	6602117,159	9346328,44	10423254,5	29358677,22

Abbildung 11.22 Die Formel für das Gesamtergebnis

11.2.14 Besonderheiten der Cube-Funktionen

Anders als bei den sonstigen Excel-Funktionen liefern Cube-Funktionen mit Ausnahme von CUBEMENGENANZAHL() jeweils zwei Ergebnisse: ein Ergebnis, das im jeweiligen Zellbereich ausgegeben wird, und ein internes Ergebnis, das verwertet wird, wenn eine Cube-Funktion als Argument für eine andere Cube-Funktion verwendet wird. Beispielsweise liefert die Funktion CUBEMENGE() einen internen Wert, der dann von der Funktion CUBEMENGENANZAHL() ausgewertet werden kann. Wieder mit Ausnahme von CUBEMENGENANZAHL() verwenden alle Funktionen als erstes Argument den Namen der Verbindung zu dem für die Abfrage vorgesehenen Cube.

Die in Kapitel 1, »Einstieg in Berechnungen mit Excel«, bereits beschriebene Funktion **AutoVervollständigen bei Formeln** ist bei der Eingabe von Cube-Funktionen besonders hilfreich. Zuerst wird die Eingabe des Verbindungsnamens unterstützt, wobei immer nur die für die Anwendung verfügbaren Verbindungen angeboten werden, sobald hinter der öffnenden Klammer die Anführungszeichen eingegeben sind.

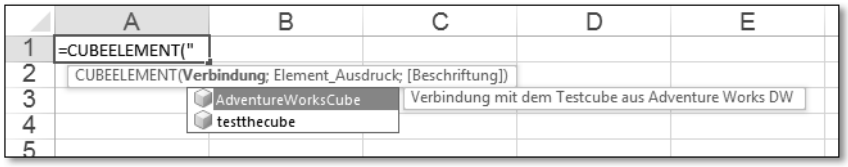


Abbildung 11.23 Die definierten Verbindungen werden angeboten.

Besonders nützlich ist dann die Unterstützung bei der Zusammenstellung der als Argument benötigten Ausdrücke. Nach der Eingabe von Anführungszeichen oder einem Punkt in einem solchen Ausdruck werden die aktuell möglichen Optionen angeboten. Die innerhalb der Ausdrücke möglichen Funktionsnamen wie `Children`, `Parent` oder `Crossjoin` müssen allerdings manuell eingetragen werden.

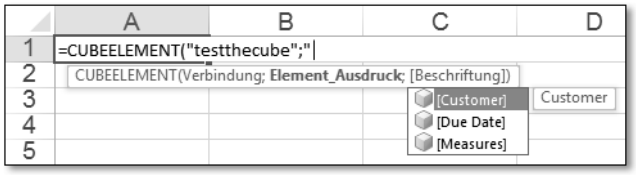


Abbildung 11.24 Auswahl von Argumenten für eine Cube-Funktion

11.2.15 Beispiel für eine Lösung mit Cube-Funktionen

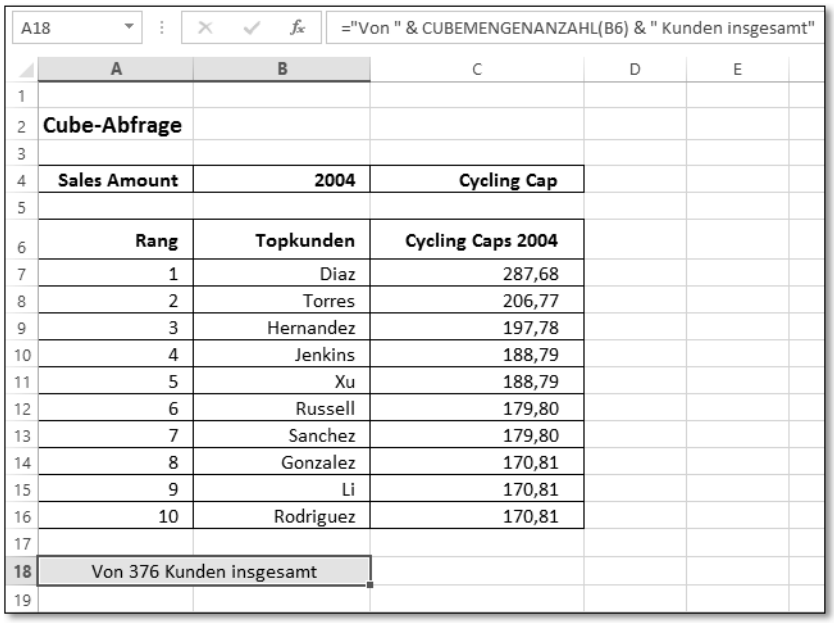
Da die Cube-Funktionen selten einzeln eingesetzt werden, soll an einem übersichtlichen Beispiel das Zusammenspiel der wichtigsten Funktionen demonstriert werden. Abbildung 11.25 zeigt eine Tabelle, in der für ein bestimmtes Jahr die Verkaufssumme bei einem Produkt für die zehn besten Kunden ausgegeben wird.

Um diese Lösung in die Tat umzusetzen, wird zunächst also in den drei Zellen in Zeile 4 mit `CUBEELEMENT()`-Funktionen gearbeitet. Die Formeln sehen so aus:

```
=CUBEELEMENT("testthecube";"[Measures].[SalesAmount]")
=CUBEELEMENT("testthecube";"[Order Date].[Fiscal Year].[All].[2004]")
=CUBEELEMENT("testthecube";"[Product].[Model Name].[All].[Cycling Cap]")
```

Die erste Funktion liefert das `[Measures]`-Element `[Sales Amount]`. Die zweite Funktion liefert ein untergeordnetes Element der Zeitdimension, hier ein bestimmtes Jahr. Die dritte Funktion liefert ein Element aus der Dimension `[Product]`. Alle drei Funktionen sorgen dafür, dass in der jeweiligen Zelle der Name des Elements als Text angezeigt wird. Gleichzeitig aber stellen diese Funktionen die Metadaten für die mit Hilfe der

MDX-Ausdrücke angesprochenen Elemente zur Verfügung, so dass andere Cube-Funktionen diese Daten nutzen können.



	A	B	C	D	E
1					
2	**Cube-Abfrage**				
3					
4	**Sales Amount**	**2004**	**Cycling Cap**		
5					
6	**Rang**	**Topkunden**	**Cycling Caps 2004**		
7	1	Diaz	287,68		
8	2	Torres	206,77		
9	3	Hernandez	197,78		
10	4	Jenkins	188,79		
11	5	Xu	188,79		
12	6	Russell	179,80		
13	7	Sanchez	179,80		
14	8	Gonzalez	170,81		
15	9	Li	170,81		
16	10	Rodriguez	170,81		
17					
18	Von 376 Kunden insgesamt				
19					

Abbildung 11.25 Beispiel für eine Cube-Abfrage

Die Werte in der Spalte mit der Bezeichnung »Rang« werden in diesem Fall einfach manuell eingetragen. In der Zelle B6 ist eine Formel abgelegt, die eine Menge von Elementen spezifiziert:

```
=CUBEMENGE("testthecube";"[Customer].[Last Name].[All].children";
"Top Kunden";2;C6)
```

Diese Formel liefert den Zugang zu der Liste der Nachnamen der Kunden. Die Anzeige in der Zelle wird als drittes Argument der Funktion eingegeben. Das vierte Argument sorgt für eine absteigende Sortierung, das fünfte Argument gibt über den Zellbezug auf C6 an, auf welche Elemente sich die Sortierung bezieht.

In der Zelle C6 wird ein Tupel bestimmt, das die Ergebnisse der in den Zellen A4:C4 abgelegten `CUBEELEMENT()`-Funktionen nun kombiniert und die darin angesprochenen Dimensionen zusammenführt. Die Sortierung bezieht sich also auf die Verkaufssumme für das ausgewählte Produkt im Jahr 2004.

```
=CUBEELEMENT("testthecube";A4:C4;"Cycling Caps 2004")
```

Die Bereichsangabe A4:C4 wird also automatisch als eine korrekte Tupel-Definition in MDX übersetzt.

Um nun die Namen aus der Kundenliste im Tabellenblatt anzuzeigen, wird mit der Funktion CUBERANGELEMENT() gearbeitet. Die Formel für den ersten Kunden lautet:

```
=CUBERANGELEMENT("testthecube";$B$6;$A7)
```

Das zweite Argument ist ein absoluter Bezug auf die Mengendefinition. Das dritte Argument übernimmt mit einem teilabsoluten Bezug die Rangzahl auf Spalte A. Diese Formel kann also ohne Weiteres nach unten kopiert werden, um die restlichen Topkunden anzuzeigen.

Nun fehlt noch die Funktion für die Anzeige der einzelnen Werte pro Kunde. Dies besorgt die CUBEWERT()-Funktion. Diese Funktion liefert als Measuredaten die zusammengefassten Verkaufswerte, wobei das zweite und das dritte Argument die jeweilige Schnittmenge bestimmen. Das zweite Argument ist ja ein Bezug auf die Funktion CUBERANGELEMENT(), die den einzelnen Kunden festlegt. Das dritte Argument ist jedes Mal ein Bezug auf die Tupel-Definition in C6.

```
=CUBEWERT("testthecube";$B7;C$6)
```

Zur Ergänzung ist in A18 noch eine Formel eingetragen, die anzeigt, wie viele Kunden die Formel in B6 insgesamt geliefert hat:

```
="Von " & CUBEMENGENANZAHL(B6) & " Kunden insgesamt"
```

11.3 Referenz der Cube-Funktionen

CUBEELEMENT()

CUBEMEMBER()

Syntax: CUBEELEMENT(**Verbindung**; **Element_Ausdruck**; Beschriftung)

Beispiel: =CUBEELEMENT("testthecube";"[Measures].[Sales Amount]")

Die Funktion CUBEELEMENT() gibt ein Element oder Tupel aus einem Cube zurück. Verbindung ist der Name der Verbindung zum Cube. Element_Ausdruck ist ein gültiger MDX-Ausdruck, der ein eindeutiges Element im Cube ergibt. Stattdessen kann auch ein Zellbereich oder ein Array angegeben werden, das ein gültiges Tupel darstellt. Beschriftung ist eine Zeichenfolge, die anstelle der sonst in der Zelle angezeigten Ele-

mentnamen erscheinen soll. (Bei Tupeln erscheint als Vorgabe immer die Beschriftung des letzten Elements.)

Mit Hilfe dieser Funktion wird zugleich überprüft, ob das mit Hilfe von Element_Ausdruck angegebene Element oder Tupel im Cube überhaupt vorhanden ist.

Wird ein ungültiger Name für Verbindung angegeben, liefert die Funktion den Fehler #NV. Ist das mit Element_Ausdruck angegebene Element nicht im Cube zu finden, erscheint ebenfalls der Fehler #NV. Ist Element_Ausdruck länger als 255 Zeichen, erscheint der Fehler #WERT!.

CUBEELEMENTEIGENSCHAFT()

CUBEMEMBERPROPERTY()

Syntax: `CUBEELEMENTEIGENSCHAFT(`**`Verbindung`**`;`
`Element_Ausdruck``;` **`Eigenschaft`**`)`

Beispiel: `=CUBEELEMENTEIGENSCHAFT("testthecube";`
`"[Order Date].[Fiscal Year].[All].[2005];B10)`

Die Funktion CUBEELEMENTEIGENSCHAFT() liefert den Wert einer Eigenschaft des über Element_Ausdruck angegebenen Elements im Cube. Damit wird geprüft, ob ein entsprechender Elementname im Cube vorhanden ist. Wenn ja, wird die für dieses Element angegebene Eigenschaft zurückgegeben. Das Argument Eigenschaft enthält den Namen der Eigenschaft oder einen Zellbezug, der diesen Namen enthält.

Wird ein ungültiger Name für Verbindung angegeben, liefert die Funktion den Fehler #NV. Ist das mit Element_Ausdruck angegebene Element nicht im Cube zu finden, erscheint ebenfalls der Fehler #NV. Ist Element_Ausdruck länger als 255 Zeichen, erscheint der Fehler #WERT!.

CUBEKPIELEMENT()

CUBEKPIMEMBER()

Syntax: `CUBEKPIELEMENT(`**`Verbindung`**`;` **`KPI_Name`**`;` **`KPI_Eigenschaft`**`;`
`Beschriftung)`

Beispiel: `=CUBEKPIELEMENT("testthecube";`
`"CPRatio";3;"Kosten/Preis-Relation")`

Die Funktion CUBEKPIELEMENT() gibt die angegebene KPI_Eigenschaft eines Key Performance Indicator (KPI) zurück und zeigt KPI_Name und Eigenschaft in der Zelle an. Wird ein Text für Beschriftung angegeben, wird dieser stattdessen in der Zelle angezeigt.

Ein solcher Indikator ist ein quantifizierbares Maß, wie z.B. der Quartalsgewinn oder die vierteljährliche Mitarbeiterfluktuation, mit dem die Leistungsfähigkeit eines Unternehmens beurteilt werden kann. Für das Argument KPI_Eigenschaft sind folgende Werte möglich:

Zahl	Eigenschaft	Beschreibung
1	KPIWert	aktueller Wert
2	KPIZiel	Zielwert
3	KPIStatus	Zustand des KPI zu einem bestimmten Zeitpunkt
4	KPITrend	Measure des Werts über einen Zeitraum
5	KPIStärke	Gewichtung des Indikators
6	KPICurrentTimeMember	Zeitrahmen des Indikators

Tabelle 11.2 Werte für das Argument KPI_Eigenschaft

Wird KPIWert verwendet, wird nur der KPI_Name in der Zelle angezeigt. Die Eingabe der Zahlenwerte wird durch die Funktion **AutoVervollständigen** unterstützt.

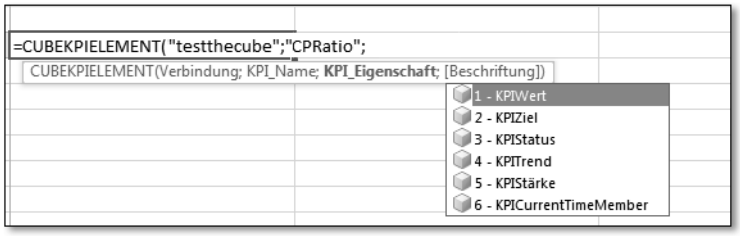


Abbildung 11.26 Die möglichen Werte für KPI_Eigenschaft werden bei der Eingabe angeboten.

Enthält der Cube einen solchen KPI, kann mit einer Kombination von CUBEKPIELEMENT() und CUBEVALUE() gearbeitet werden, wie Abbildung 11.27 zeigt.

In der Spalte A wird die CUBEKPIELEMENT()-Funktion mit unterschiedlichen Werten für das Argument KPI_Eigenschaft verwendet. Dabei wird jeweils eine passende Beschriftung angegeben. In der Spalte B wird mit Hilfe von CUBEWERT()-Funktionen auf die Funktionen in Spalte A Bezug genommen, um die den verschiedenen Eigenschaften entsprechenden Werte auszulesen.

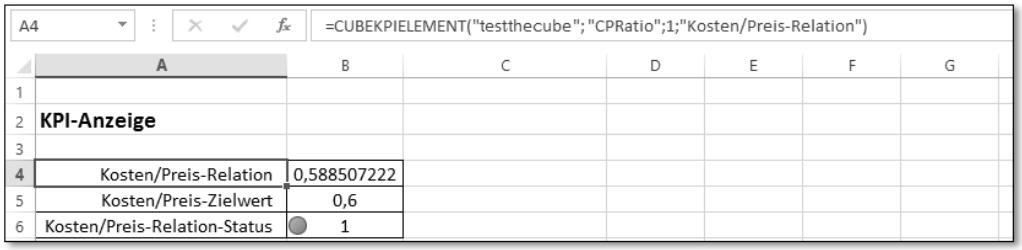


Abbildung 11.27 Anzeige von KPI-Daten in Excel 2013

Wird ein ungültiger Name für `Verbindung` angegeben, liefert die Funktion den Fehler `#NV`. Das gilt auch, wenn `KPI_Name` und `KPI_Eigenschaft` ungültig sind.

CUBEMENGE()

CUBESET()

Syntax: `CUBEMENGE(`**`Verbindung`**`;` **`Menge_Ausdruck`**`; Beschriftung; Sortierreihenfolge; Sortieren_nach)`

Beispiel:
```
=CUBEMENGE("testthecube";"[Customer].
[Last Name].[All].children";"Top Kunden";2;C6)
```

Die Funktion `CUBEMENGE()` sendet einen Mengenausdruck für einen bestimmten Satz von Elementen oder Tupeln an den Cube, der die entsprechenden Daten zusammenstellt und an Excel liefert. `Menge_Ausdruck` ist entweder ein MDX-Ausdruck oder ein Zellbezug auf einen solchen. Das Argument kann auch durch den Bezug auf einen Zellbereich zusammengestellt werden.

Mit `Beschriftung` kann eine andere Beschriftung gewählt werden als die im Cube vorhandene Beschriftung. Außerdem können eine Sortierreihenfolge und ein Sortierfeld angegeben werden.

`Sortierreihenfolge` bestimmt die Art der auszuführenden Sortierung und kann folgende Werte annehmen:

Zahl	Eigenschaft	Beschreibung
0	Ohne	keine Änderung der Reihenfolge
1	Aufsteigend	Sortieren in aufsteigender Reihenfolge

Tabelle 11.3 Werte für das Argument `Sortierreihenfolge`

Zahl	Eigenschaft	Beschreibung
2	Absteigend	Sortieren in absteigender Reihenfolge
3	Alphabetisch aufsteigend	Sortieren in alphabetisch aufsteigender Reihenfolge
4	Alphabetisch absteigend	Sortieren in alphabetisch absteigender Reihenfolge
5	Natürlich aufsteigend	Sortieren in natürlich aufsteigender Reihenfolge
6	Natürlich absteigend	Sortieren in natürlich absteigender Reihenfolge

Tabelle 11.3 Werte für das Argument `Sortierreihenfolge` (Forts.)

Der Standardwert ist `0`. Wird `1` oder `2` verwendet, muss das Argument `Sortieren_nach` angegeben werden, in allen anderen Fällen ist es nicht erforderlich. Bei einer alphabetischen Sortierung für eine Menge mit Tupeln wird jeweils anhand des letzten Elements in jedem Tupel sortiert. `Sortieren_nach` ist eine Textzeichenfolge des Werts, nach dem sortiert werden soll.

Wird ein ungültiger Name für `Verbindung` angegeben, liefert die Funktion den Fehler `#NV`. Sind die mit `Menge_Ausdruck` angegebenen Daten nicht im Cube zu finden, erscheint ebenfalls der Fehler `#NV`. Ist `Menge_Ausdruck` länger als 255 Zeichen, erscheint der Fehler `#WERT!`.

CUBEMENGENANZAHL()

CUBESETCOUNT()

Syntax: `CUBEMENGENANZAHL(`**Menge**`)`

Beispiel:
```
=CUBEMENGENANZAHL(CUBEMENGE("testthecube";
"[Customer].[Last Name].[All].children";"Top Kunden";2;C6))
```

Die Funktion `CUBEMENGENANZAHL()` liefert die Anzahl der Elemente in einer Menge, die durch das Argument `Menge` bestimmt ist. Dieses Element besteht aus einer `CUBEMENGE()`-Funktion oder einem Zellbezug auf eine solche Funktion.

CUBERANGELEMENT()

CUBERANKEDMEMBER()

Syntax: `CUBERANGELEMENT(Verbindung; Menge_Ausdruck; Rang; Beschriftung)`
Beispiel: `=CUBERANGELEMENT("testthecube";$B$6;$A7)`

Die Funktion `CUBERANGELEMENT()` gibt das mit dem Wert für `Rang` angegebene Element in einer Menge zurück. `1` liefert den besten Wert, `2` den zweitbesten Wert etc. Wird verwendet, um mindestens ein Element in einer Menge zurückzugeben, die meist durch die Funktion `CUBEMENGE()` definiert ist oder durch einen Bezug auf eine Zelle, die eine solche Funktion enthält.

Wird ein ungültiger Name für `Verbindung` angegeben, liefert die Funktion den Fehler `#NV`. Sind die mit `Menge_Ausdruck` angegebenen Daten nicht im Cube zu finden, erscheint ebenfalls der Fehler `#NV`. Ist `Menge_Ausdruck` länger als 255 Zeichen, erscheint der Fehler `#WERT!`. Dies geschieht auch, wenn `Rang` nicht numerisch ist.

CUBEWERT()

CUBEVALUE()

Syntax: `CUBEWERT(Verbindung; Element_Ausdruck1; Element_Ausdruck2...)`
Beispiel: `=CUBEWERT("testthecube";$A4;D$3)`

Die Funktion `CUBEEWERT()` liefert den aggregierten Wert für die durch einen oder mehrere Elementausdrücke eingeschränkten Elemente.

Wird ein ungültiger Name für `Verbindung` angegeben, liefert die Funktion den Fehler `#NV`. Sind die mit `Element_Ausdruck1` oder `Element_Ausdruck2` etc. angegebenen Elemente nicht im Cube zu finden, erscheint ebenfalls der Fehler `#NV`. Ist einer der Elementausdrücke länger als 255 Zeichen, erscheint der Fehler `#WERT!`.

11.3.1 Weblinks zum Thema

- SQL Server Developer Center von Microsoft
  http://msdn.microsoft.com/de-de/sqlserver/default.aspx
- ADOMD.NET
  http://msdn.microsoft.com/en-us/library/ms123483.aspx
- MDX-Referenz
  http://msdn.microsoft.com/de-de/library/ms145506.aspx

12 Textfunktionen

Funktion	Seite	Funktion	Seite
BAHTTEXT()	747	SÄUBERN()	764
CODE()	748	SUCHEN()	764
DM()	750	T()	765
ERSETZEN()	751	TEIL()	766
FEST()	752	TEXT()	768
FINDEN()	753	UNICODE()	769
GLÄTTEN()	754	UNIZEICHEN()	770
GROSS()	755	VERKETTEN()	771
GROSS2()	756	WECHSELN()	772
IDENTISCH()	757	WERT()	774
KLEIN()	758	WIEDERHOLEN()	775
LÄNGE()	759	ZAHLENWERT()	776
LINKS()	759	ZEICHEN()	777
RECHTS()	761		

12.1 Einsatzbereiche der Textfunktionen

Die Textfunktionen gehören zu den häufig benötigten Standardfunktionen von Excel und decken die Aufgaben ab, die beim Umgang mit Zeichenfolgen anfallen. Dabei geht es häufig darum, Bestandteile von Texten in einer Tabelle zu manipulieren, Zeichen in Texten auszutauschen oder zu entfernen. Häufig ist es auch nötig, Textdaten in numerische Daten zu konvertieren und umgekehrt.

Excel betrachtet alle Eingaben als Zeichenfolgen, die sich nicht als numerische Werte, als Datums- oder Zeitwerte, als logische Werte oder als Formel interpretieren lassen. Zeichenfolgen – der Argumentname in den Funktionen dafür ist üblicherweise `Text` – können auch das Ergebnis von Formeln oder Funktionen sein. Wenn in Zelle A8 die Zeichenfolge `"Monat: "` und in B8 die Zeichenfolge `"Mai"` steht, liefert eine Formel in C8 wie `=A8 & B8` wiederum eine Zeichenfolge `"Monat: Mai"`. C8 kann dann auch als Bezug für ein Textargument verwendet werden.

Es ist häufiger sinnvoll, auf diese Weise eine Zeichenfolge in einer Zelle durch eine Formel zu erzeugen, die verschiedene Zeichen oder Zeichenfolgen verknüpft, etwa eine Beschreibung für einen Termin und den Termin selbst. Wenn der Termin wiederum von anderen Werten in einem Tabellenblatt abhängig ist, lassen sich so leicht dynamische Beschriftungen erzeugen, die automatisch angepasst werden. Hier zunächst noch einige oft vorkommende Operationen bei der Arbeit mit Texten. Sie betreffen die Umwandlung von Zahlen in Zeichenfolgen, spezielle Sortiermöglichkeiten über Teile einer Zeichenfolge, die Verknüpfung mit logischen Werten und die Kombination von Texten und Datumswerten.

12.1.1 Zahlen in Text umwandeln

Sie wollen Artikelnummern um zwei Zeichen erweitern, die die Warengruppe beinhalten. Wenn Sie beispielsweise in die Zelle C3

```
=VERKETTEN(A3;B3)
```

eintragen, wird der Inhalt von Zelle A3 mit dem Inhalt von Zelle B3 verkettet. Wenn A3 das Warengruppenkennzeichen `PX` enthält und B3 die Artikelnummer `3370086`, ist das Ergebnis in Zelle C3 `PX3370086`. Wenn die bisherige Artikelnummer in B3 als Zahl und nicht als Zeichenfolge eingetragen worden ist, wird die Zahl in eine Zeichenfolge umgewandelt. Anschließend können Sie diese Formel noch mit F9 in ihr Ergebnis verwandeln. Wollen Sie zwischen den beiden Textelementen eine Leertaste sehen, müssen Sie Folgendes schreiben:

```
=VERKETTEN(A3;" ";B3)
```

12.1.2 Sortiermöglichkeiten durch Textfunktionen

Sollen umgekehrt bei Artikelbezeichnungen, die immer auf den zwei ersten oder letzten Stellen ein Kennzeichen der Warengruppe enthalten, diese abgelesen werden, etwa um eine Liste danach zu sortieren, können Sie mit einer Textfunktion die beiden Stellen herausziehen:

```
=RECHTS(C9;2)
```

liefert die beiden letzten Stellen,

```
=LINKS(C9;2)
```

die beiden ersten Stellen.

12.1.3 Logische Werte in Texte aufnehmen

Auch eine Zelle mit einem logischen Wert kann mit dem Inhalt einer anderen Zelle zu einem Text verkettet werden.

```
=VERKETTEN("Die Behauptung, der Umsatz steigt, ist ";C9)
```

ergibt den Satz »Die Behauptung ... ist FALSCH«, wenn die entsprechende Bedingung in C9 nicht erfüllt ist.

12.1.4 Verknüpfung von Text mit einem Datum

Etwas schwieriger ist die Verknüpfung von Text mit einem Datum. Wenn Sie einen Bezug auf ein Datumsfeld benutzen, liefert Excel anstelle des Datums die serielle Zahl, mit der das Datum intern dargestellt wird. Das Problem ist lösbar, wenn Sie mit den anderen Datumsfunktionen arbeiten. Hier ein Beispiel: Wenn Sie ein Tagesformular verwenden, in dessen Kopf stehen soll: »Abrechnung vom ...« , können Sie in eine Zelle eintragen:

```
="Abrechnung vom "&TAG(HEUTE())&"."&MONAT(HEUTE())&"."& JAHR(HEUTE())
```

Das Formular wird dann automatisch mit dem Tagesdatum gedruckt.

12.1.5 Hinweis zu Umwandlungen mit der Blitzvorschau

Wenn Sie mit Excel 2013 arbeiten, haben Sie in zahlreichen Fällen, die bisher Lösungen mit Hilfe von Textfunktionen gefordert haben, die Alternative, die neue Füllfunktion zu nutzen, die bereits in Kapitel 1 im Abschnitt über die Blitzvorschau, »Umwandlungen per Mustererkennung«, beschrieben worden ist. Die Funktion macht in vielen Fällen komplizierte Textumwandlungen mit Hilfe von Makros und Textfunktionen überflüssig.

12.2 Referenz der Textfunktionen

BAHTTEXT()

BAHTTEXT()

Syntax:	`BAHTTEXT(`**`Zahl`**`)`
Beispiel:	siehe Abbildung 12.1

Die Funktion BAHTTEXT() wandelt den Wert von Zahl in Text im thailändischen Währungsformat um. Die Umwandlung von Zahlen in Zahlwörter wird beispielsweise bei Betragsangaben auf Schecks oder in Verträgen benötigt. Anders als bei dem thailändischen Zahlenformat, das über den Dialog **Zellen formatieren** ausgewählt werden kann, wird kein Währungssymbol verwendet, sondern dem Text das Suffix »Baht« angehängt. Warum diese Funktion nur für die thailändische Währung angeboten wird, darüber kann nur spekuliert werden.

Ist Zahl nicht numerisch, liefert die Funktion den Fehlerwert #WERT!.

	A	B
1		
2	**Beispiele für die BAHTTEXT()-Funktion**	
3		
4	**Zahl**	**BAHTTEXT()**
5	123	หนึ่งร้อยยี่สิบสามบาทถ้วน
6	8	แปดบาทถ้วน
7	-8	ลบแปดบาทถ้วน
8	12,23	สิบสองบาทยี่สิบสามสตางค์
9	-12,23	ลบสิบสองบาทยี่สิบสามสตางค์

Abbildung 12.1 Zahlworte in Thai-Text

CODE()

CODE()

Syntax: CODE(**Text**)
Beispiel: =CODE("T")
Ergebnis: 84

Die Funktion CODE() liefert den Code des ersten Zeichens in der Zeichenfolge, die mit dem Argument Text angegeben wird. Die ausgegebene Codezahl bezieht sich dabei auf den Zeichensatz, mit dem das System arbeitet, unter Windows also auf den ANSI-Code. Dieser Code läuft von 1 bis 255, die ersten 31 Codes stehen für nicht druckbare Zeichen wie dem Zeilenvorschub. Die Großbuchstaben belegen die Nummern 65 bis 90, die Kleinbuchstaben 97 bis 122, die Zahlen liegen zwischen 48 und 57. Eine komplette Übersicht finden Sie in Abbildung 12.37.

Ist Text ein Verweis auf eine leere Zelle oder das Ergebnis einer Formel, die eine leere Zeichenfolge zurückgibt, liefert die Funktion den Fehler #WERT!.

	A	B	C	D
1				
2	**ANSI-Codes**			
3				
4	Text	CODE()	Text	CODE()
5	A	65	T	84
6	B	66	U	85
7	C	67	V	86
8	D	68	W	87
9	E	69	X	88
10	F	70	Y	89
11	G	71	Z	90
12	H	72	a	97
13	I	73	b	98
14	J	74	c	99
15	K	75	1	49
16	L	76	2	50
17	M	77	3	51
18	N	78		32
19	O	79	!	33
20	P	80	?	63
21	Q	81	*	42
22	R	82	#	35
23	S	83	\	92

Abbildung 12.2 Ausgewählte `CODE()`-Beispiele

Die Umkehrfunktion von `CODE()` ist die Funktion `ZEICHEN()`, sie liefert das Zeichen zu einem gestimmten Codewert.

12.2.1 Gruppenbildung über CODE()

Die Funktion kann von Nutzen sein, wenn es beispielsweise darum geht, für eine Namensliste eine Gruppenbildung nach den Anfangsbuchstaben vorzunehmen. Abbildung 12.3 zeigt ein kleines Beispiel, in dem die Namen über eine Nummer in zwei Gruppen gegliedert werden. Die im Alphabet unterhalb des Buchstabens mit dem Code 77 – das ist das M – liegenden Namen bilden die erste Gruppe, die anderen die zweite Gruppe. Um die Abfrage zu vereinfachen, wird mit der Funktion `GROSS()` sichergestellt, dass der Anfangsbuchstabe des Namens immer als Großbuchstabe ausgewertet wird. Die Formel heißt dann:

```
=WENN(CODE(GROSS(A6))<77;1;2)
```

Sind die Gruppenzuordnungen berechnet, können Sie anschließend die Tabelle nach der Spalte mit den Gruppenziffern sortieren.

	A	B	C
1			
2	**Gruppieren nach dem ersten Buchstaben**		
3			
4	Gruppierung von Namen		
5	**Name**	**Gruppe**	
6	Hansen	1	**=WENN(CODE(GROSS(A6))<77;1;2)**
7	Bergmann	1	
8	Gange	1	
9	Erft	1	
10	Donan	1	
11	Karim	1	
12	Katar	1	
13	Schoner	2	
14	Zander	2	
15	Sonderbohm	2	

Abbildung 12.3 Die `CODE()`-Funktion lässt sich zum Gruppieren von Namen nach dem ersten Buchstaben verwenden.

DM()

DOLLAR()

Syntax: `DM(`**`Zahl`**`; Dezimalstellen)`

Beispiel: `=DM(58721,4567;2)`

Ergebnis: `58721,46 €`

Die Funktion `DM()` mit dem »alten« Namen wandelt den Wert für das Argument `Zahl` in eine Zeichenfolge im aktuellen Währungsformat um.

Durch das optionale Argument `Dezimalstellen` kann die Anzahl der Ziffern rechts vom Komma festgelegt werden, die dabei gerundet und nicht abgeschnitten werden. Wird das Argument nicht angegeben, dann wird es mit `2` angenommen. Wird ein negativer Wert eingegeben, wird auf entsprechende Stellen vor dem Komma gerundet.

Ist `Zahl` nicht numerisch, liefert die Funktion den Fehler `#WERT!`. Das gilt auch, wenn für `Dezimalstellen` ein Wert über `127` angegeben wird.

Wie die Spalte D in Abbildung 12.4 zeigt, wird bei einer Multiplikation des Funktionsergebnisses etwa mit dem Mehrwertsteuersatz nicht – wie es sonst der Fall ist – das Euroformat automatisch übernommen.

	A	B	C	D
1				
2	**Währungswert als Text**			
3				
4	**Zahl**	**Dezimalstellen**	**DM()**	**DM() * 1,19**
5	123	0	123 EUR	146,37
6	-123,5	1	-123,5 EUR	-146,965
7	123,45	2	123,45 EUR	146,9055
8	-123.456	3	-123.456,000 EUR	-146912,64

Abbildung 12.4 Die Funktion mit dem veralteten Namen liefert jetzt Eurowerte.

Während es sich bei dem Inhalt der Zellen C5:C8 um Zeichenfolgen und nicht um Zahlenwerte handelt, sind die Werte in D5:D8 wieder Zahlenwerte. Durch die Multiplikation wird also automatisch der Datentyp `Zeichenfolge` in den Datentyp `Zahl` umgewandelt.

ERSETZEN()

REPLACE()

Syntax: `ERSETZEN(`**`Alter_Text;Erstes_Zeichen;Anzahl_Zeichen;Neuer_Text`**`)`
Beispiel: `=ERSETZEN("Müller";2;5;"eier")`
Ergebnis: `Meier`

Mit der Funktion `ERSETZEN()` können in einer Zeichenfolge ein oder mehrere Zeichen durch neue Zeichen überschrieben werden.

	A	B	C	D	E
1					
2	**Ersetzen von Zeichen**				
3					
4	**Alter_Text**	**Erstes_Zeichen**	**Anzahl_Zeichen**	**Neuer_Text**	**ERSETZEN()**
5	Kapitel 1	1	7	Abschnitt	Abschnitt 1
6	Hansen	1		Dr.	Dr. Hansen
7	Gernot	7		@net.de	Gernot@net.de
8	Köln, den Dt	11	2	08.05.2013	Köln, den 08.05.2013

Abbildung 12.5 Beispiele für das Ersetzen von Textteilen

Die Funktion hat vier Argumente: `Alter_Text` ist die Zeichenfolge, in der Teile ersetzt werden sollen. `Erstes_Zeichen` bestimmt die Position in der Zeichenfolge, an der mit dem Ersetzen zu beginnen ist. Dabei wird das erste Zeichen der Zeichenfolge mit `1` gezählt. `Anzahl_Zeichen` gibt an, wie viele Zeichen aus der alten Zeichenfolge ab dem an-

gegebenen Startpunkt durch die mit Neuer_Text angegebene Zeichenfolge ersetzt werden sollen.

Die Werte für Erstes_Zeichen und Anzahl_Zeichen müssen > 0 sein, ansonsten erscheint als Ergebnis der Fehler #WERT!. Ist Erstes_Zeichen > als die Anzahl der Zeichen in der Zeichenfolge Alter_Text, wird die Zeichenfolge Neuer_Text einfach an die Zeichenfolge Alter_Text angehängt. Das gilt auch, wenn Alter_Text ein Bezug auf eine leere Zelle ist. Es macht auch nichts, wenn Anzahl_Zeichen größer ist als die Anzahl der Zeichen in Neuer_Text.

Wenn nicht bekannt ist, an welcher Stelle in einem gegebenen Text sich der Teil befindet, der ersetzt werden soll, und wie viele Zeichen er enthält, kann die Funktion mit der weiter unten beschriebenen Funktion SUCHEN() kombiniert werden.

```
=ERSETZEN(A8;SUCHEN("den Dt";A8);LÄNGE("den Dt");
"den "&TEXT(HEUTE();"TT.MM.JJJJ"))
```

FEST()

FIXED()

Syntax:	FEST(**Zahl**; Dezimalstellen; Keine_Punkte)
Beispiel:	=FEST(33654,236;2;WAHR)
Ergebnis:	"33654,24" (als Text)

Die Funktion FEST() wandelt einen numerischen Wert in eine Zeichenfolge mit einer fixierten Anzahl von Nachkommastellen um.

Mit Zahl wird der numerische Wert angegeben. Mit dem Argument Dezimalstellen kann die Anzahl der gewünschten Stellen bestimmt werden. Das Ergebnis der Funktion wird gerundet. Geben Sie das Argument Dezimalstellen nicht an, so wird es als 2 angenommen; wird es negativ eingegeben, wird auf die entsprechenden Stellen vor dem Komma gerundet. Obwohl Excel nur mit maximal 15 Stellen hinter dem Komma rechnet, kann bei dieser Funktion für Dezimalstellen ein Wert von bis zu 127 angegeben werden. Keine_Punkte ist ein Wahrheitswert, der darüber entscheidet, ob eine Tausenderabtrennung mittels Punkt vorgenommen werden soll. Wird kein Wahrheitswert oder FALSCH angegeben, dann wird die Tausenderabtrennung vorgenommen, andernfalls wird der Punkt nicht mit ausgegeben.

Ein Wert für Dezimalstellen über 127 führt zu dem Fehler #WERT!. Der erscheint auch, wenn Zahl nicht numerisch ist.

Ist `Zahl` ein Datumswert, liefert die Funktion die serielle Zahl als Text, wie in Abbildung 12.6 zu sehen. Ist `Zahl` ein Wahrheitswert, werden die Ziffern `1` und `0` als Text ausgegeben. Wird eine Zelle, deren Wert mit `FEST()` ermittelt wird, in einer Formel beispielsweise mit einer Zahl addiert, wandelt Excel für die Berechnung den Text der Zelle automatisch wieder in eine Zahl um.

	A	B	C	D	E
1					
2	**Zahl als Text mit Dezimalstellen ausgeben**				
3					
4	**Zahl**	**Dezimalstellen**	**Keine_Punkte**	**FEST()**	**ISTTEXT()**
5	123	2		123,00	WAHR
6	123	1		123,0	WAHR
7	123	0		123	WAHR
8	123	4		123,0000	WAHR
9	12.11.2007	0	WAHR	39398	WAHR
10	WAHR	0	WAHR	1	WAHR
11	FALSCH	0	WAHR	0	WAHR

Abbildung 12.6 Die `ISTTEXT()`-Funktion bestätigt, dass `FEST()` Zahlen als Text liefert.

Die Funktion `FEST()` kann genutzt werden, um Zeichenfolgen zusammenzustellen, die Text und Zahlen kombinieren.

```
="Der Mittelwert ist "&FEST(100/13;3)
```

gibt beispielsweise das Ergebnis der Division mit 3 Dezimalstellen aus.

```
="Der Mittelwert ist "&(100/13)
```

würde dagegen das Ergebnis

```
Der Mittelwert ist 7,692230769230769
```

ausgeben.

FINDEN()

FIND()

Syntax: `FINDEN(`**`Suchtext`**`;` **`Text`**`; Erstes_Zeichen)`
Beispiel: `=FINDEN("&";"Pasta & Vino")`
Ergebnis: `7`

Für die Suche nach Zeichenfolgen innerhalb einer Zeichenfolge stellt Excel zwei ähnliche Funktionen zur Verfügung, die Funktion FINDEN() und die Funktion SUCHEN(). Die Funktion FINDEN() unterscheidet bei der Suche zwischen Groß- und Kleinschreibung und lässt keine Platzhalterzeichen zu, die Funktion SUCHEN() ignoriert die Schreibung und lässt Platzhalter zu.

Die Funktion FINDEN() überprüft, ob die mit Suchtext angegebene Zeichenfolge in einer anderen Zeichenfolge vorkommt, die mit dem Argument Text angegeben wird. Suchtext darf keine Platzhalter enthalten. Die Suche beginnt bei der Position in der Zeichenfolge, die durch den Wert für Erstes_Zeichen bestimmt ist. Fehlt allerdings ein Wert für Erstes_Zeichen, wird 1 angenommen, also von Anfang an gesucht. Ergebnis der Funktion ist eine Zahl für die Position von Suchtext in Text. Dabei wird immer vom ersten Zeichen von Text aus gezählt unabhängig von dem Wert für Erstes_Zeichen.

	A	B	C	D
1				
2	**Zeichen finden**			
3				
4	**Suchtext**	**Text**	**Erstes_Zeichen**	**FINDEN()**
5	o	Anatomie	1	5
6	O	Anatomie	1	#WERT!
7	o	Anatomie	6	#WERT!
8	o	Anatomie	4	5
9	z	Anatomie	1	#WERT!
10	@	jk@net.de	1	3

Abbildung 12.7 Suchtext dreimal gefunden und dreimal nicht

Wird der Suchtext in der Zeichenfolge nicht gefunden, wird der Fehler #WERT! ausgegeben. Denselben Fehler liefert die Funktion, wenn Erstes_Zeichen nicht > 0 oder größer als die Anzahl von Zeichen in Text ist.

GLÄTTEN()

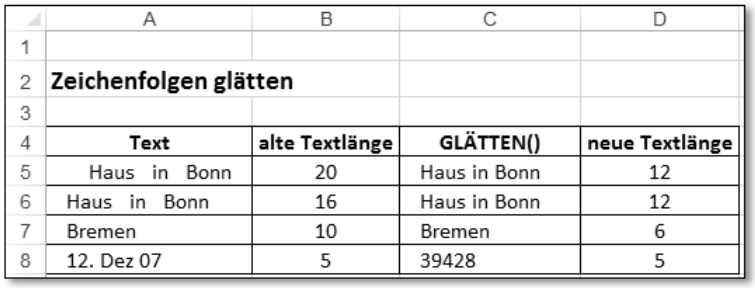
TRIM()

Syntax: GLÄTTEN(**Text**)
Beispiel: =GLÄTTEN("Egon Maier")
Ergebnis: Egon Maier

Die Funktion GLÄTTEN() entfernt überzählige Leerzeichen in Text, so dass jeweils nur Leerzeichen zur Worttrennung stehen bleiben. Ist Text ein Datumswert, liefert die

Funktion die serielle Zahl, d.h., das Ergebnis wird auf das Standardformat zurückgesetzt.

Die Funktion ist insbesondere bei der Übernahme von Daten aus anderen Anwendungen nützlich, um überflüssige Leerzeichen aufzuspüren und um mögliche Fehler beim Sortieren zu vermeiden. In dem Beispiel in Abbildung 12.8 werden die Textlängen vor und nach dem Glätten mit Hilfe der Funktion `LÄNGE()` geprüft.



	A	B	C	D
1				
2	**Zeichenfolgen glätten**			
3				
4	**Text**	**alte Textlänge**	**GLÄTTEN()**	**neue Textlänge**
5	Haus in Bonn	20	Haus in Bonn	12
6	Haus in Bonn	16	Haus in Bonn	12
7	Bremen	10	Bremen	6
8	12. Dez 07	5	39428	5

Abbildung 12.8 Entfernen überflüssiger Leerzeichen

Die Funktion entfernt allerdings nur Leerzeichen, die dem Codewert 32 im ASCII-Zeichensatz entsprechen. Geschützte Leerzeichen, wie sie in HTML mit der Entität ` ` vorkommen – im Unicode-Zeichensatz ist das der dezimale Wert 160 –, werden dagegen nicht entfernt.

GROSS()

UPPER()

Syntax: `GROSS(`**`Text`**`)`
Beispiel: `=GROSS("Kilo")`
Ergebnis: `KILO`

Die Funktion `GROSS()` wandelt alle Buchstaben aus der mit `Text` angegebenen Zeichenfolge in Großbuchstaben um. Sie erlaubt es beispielsweise, den Inhalt einer Textspalte mit uneinheitlicher Anwendung der Groß- und Kleinschreibung in eine einheitliche Schreibweise zu konvertieren, etwa um problemloser Daten sortieren zu können.

Verweist `Text` auf eine Zelle, die einen numerischen Wert oder ein Datum enthält, wird die Zahl als Text im Standardformat ausgegeben. Wenn Sie einen derartigen Zellinhalt mit einem anderen Wert multiplizieren oder einen Wert addieren, wird der Zell-

inhalt automatisch wieder als Zahl behandelt, die Berechnung ergibt also einen gültigen Zahlenwert.

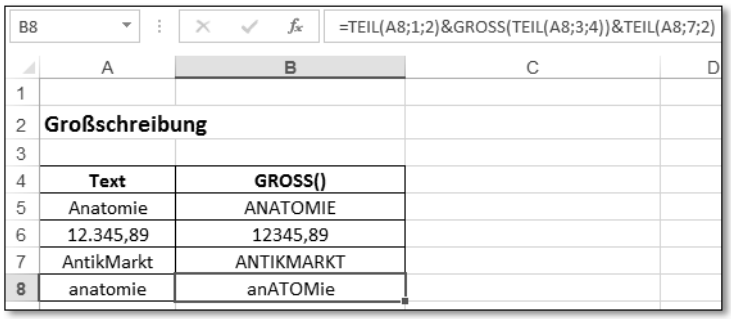


Abbildung 12.9 Generelle oder partielle Großschreibung

In der letzten Formel der Abbildung – B8 – wird als Spielerei eine Kombination der Funktion `GROSS()` mit der Funktion `TEIL()` verwendet, so dass nur ein Teil der ursprünglichen Zeichenfolge umgewandelt wird:

```
=TEIL(A8;1;2)&GROSS(TEIL(A8;3;4))&TEIL(A8;7;2)
```

Häufig wird die Funktion auch in Formeln benutzt, um Vergleiche zwischen Zeichenfolgen zu vereinfachen, wie es schon bei der Funktion `CODE()` gezeigt wurde.

GROSS2()

PROPER()

Syntax: `GROSS2(`**`Text`**`)`
Beispiel: `=GROSS2("karl-heinz")`
Ergebnis: `Karl-Heinz`

Die Funktion `GROSS2()` wandelt jeweils den ersten Buchstaben eines jeden Wortes in der mit `Text` angegebenen Zeichenfolge in Großbuchstaben um, die weiteren Zeichen in Kleinbuchstaben. Auch bei Worten, die durch Bindestrich getrennt sind, wird so verfahren. Bei Zahlen verhält sich die Funktion wie `GROSS()`.

Die Funktion kann zur Erhöhung der Lesbarkeit von Texteinträgen benutzt werden. Sie ist von Nutzen, wenn z.B. Daten von anderen Programmen oder Rechnern übernommen worden sind.

	A	B
1		
2	**Groß- und Kleinschreibung**	
3		
4	**Text**	**GROSS2()**
5	hans hagen	Hans Hagen
6	per saldo	Per Saldo
7	AntikMarkt	Antikmarkt
8	anatomie	Anatomie
9	ana-tomie	Ana-Tomie

Abbildung 12.10 Wortweise Groß- und Kleinschreibung

IDENTISCH()

EXACT()

Syntax: `IDENTISCH(Text1; Text2)`
Beispiel: `=IDENTISCH("ABBA";"Abba")`
Ergebnis: `FALSCH`

Die Funktion `IDENTISCH()` vergleicht die beiden mit `Text1` und `Text2` übergebenen Zeichenfolgen und prüft, ob sie exakt übereinstimmen. Nur dann wird als logischer Wert `WAHR` geliefert, im anderen Fall `FALSCH`. Leerzeichen und Groß- bzw. Kleinschreibung werden in den Vergleich einbezogen.

Beim Vergleich von numerischen Werten wird die Übereinstimmung unabhängig vom Zahlenformat bewertet, wie Abbildung 12.11 zeigt.

	A	B	C
1			
2	**Textvergleiche**		
3			
4	**Text1**	**Text2**	**IDENTISCH()**
5	Anatomie	Anatomie	WAHR
6	Anatomie	anatomie	FALSCH
7	1.200,23	1,200.23	FALSCH
8	1200,23	1.200,23	WAHR
9	12.10.2007	39367	WAHR

Abbildung 12.11 Vergleich von Texten auf Übereinstimmung

In Abbildung 12.12 wird noch ein Beispiel für die Verwendung der Funktion in einer Matrixformel gezeigt. In Zelle C5 wird geprüft, ob der in Zelle A5 stehende Name in der in Spalte B angezeigten Liste vorkommt. Dazu kann folgende Formel verwendet werden:

```
{ODER(IDENTISCH(A11;B11:B14))}
```

Die Formel wird mit Strg + ⇧ + ↵ abgeschlossen, also als Matrixformel für die Zelle C5.

	A	B	C
1			
2	**Kommt ein Element in einer Liste vor?**		
3			
4			**{=ODER(IDENTISCH(A5;B5:B8))}**
5	Lauber	Horten	WAHR
6		Hansen	
7		Lauber	
8		Karl	

Abbildung 12.12 Prüfen, ob ein Element in einer Liste vorkommt

KLEIN()

LOWER()

Syntax: `KLEIN(`**`Text`**`)`
Beispiel: `=KLEIN("MF2HD")`
Ergebnis: `mf2hd`

Die Funktion `KLEIN()` ist das Gegenstück zu `GROSS()`, sie wandelt alle Buchstaben von `Text` in Kleinbuchstaben um. Sie erlaubt es beispielsweise, den Inhalt einer Textspalte mit uneinheitlicher Anwendung der Groß- und Kleinschreibung in eine einheitliche Schreibweise zu konvertieren, etwa um problemloser Daten sortieren zu können.

Verweist `Text` auf eine Zelle, die einen numerischen Wert oder ein Datum enthält, wird die Zahl als Text im Standardformat ausgegeben.

Die Funktion kann auch in Formeln benutzt werden, um Vergleiche zwischen Zeichenfolgen zu vereinfachen, bei denen die Groß- und Kleinschreibung ignoriert werden soll.

	A	B
1		
2	**Kleinschreibung**	
3		
4	**Text**	**KLEIN()**
5	Anatomie	anatomie
6	ANATOMIE 2	anatomie 2
7	AntikMarkt	antikmarkt
8	12.10.2007	39367

Abbildung 12.13 Beispiele für den Effekt der Funktion `KLEIN()`

LÄNGE()

LEN()

Syntax: `LÄNGE(Text)`
Beispiel: `=LÄNGE("LÄNGE ")`
Ergebnis: `6`

Die Funktion `LÄNGE()` liefert die Zahl der Zeichen (einschließlich Leerzeichen) in der mit `Text` angegebenen Zeichenfolge. Ist das Argument der Funktion eine Zelladresse, wird als Ergebnis `0` ausgegeben, wenn die Zelle leer ist. Numerische Zeichen werden wie Text behandelt und auf die gleiche Weise gezählt. Dabei werden aber Zeichen ignoriert, die durch die Zahlenformatierung hinzugefügt sind, etwa Währungszeichen oder Tausenderabtrennung, während das Dezimalzeichen mitgezählt wird. Für Datumswerte wird die Länge der seriellen Zahl angezeigt.

Die Funktion wird häufig in Kombination mit anderen Funktionen wie `ERSETZEN()`, `LINKS()`, `RECHTS()` oder `TEIL()` eingesetzt. Zur Anwendung kommt die Funktion insbesondere auch bei der Prüfung von Einträgen, für die eine bestimmte Länge erforderlich ist, z. B. Postleitzahlen.

	A	B
1		
2	**Länge einer Zeichenfolge**	
3		
4	**Text**	**LÄNGE()**
5	Anatomie	8
6	A n a t o m i e	15
7	12.10.2007	5
8		1
9		0
10	1.234,56 €	7
11	A n a t o m i e - Kurs	22

Abbildung 12.14 Längen verschiedener Zellinhalte

LINKS()

LEFT()

Syntax: `LINKS(Text; Anzahl_Zeichen)`
Beispiel: `=LINKS("Version3";)`
Ergebnis: `Ver`

Die Funktion LINKS() liefert als Ergebnis die mit Anzahl_Zeichen festgelegte Zeichenzahl vom Beginn der Zeichenfolge an, die mit dem Argument Text angegeben ist, der Rest wird abgeschnitten. Wird für Anzahl_Zeichen kein Wert angegeben, so wird 1 vorgegeben.

Ist Anzahl_Zeichen größer als die Anzahl der vorhandenen Zeichen, wird der gesamte Text ausgegeben. Anzahl_Zeichen darf nicht negativ sein, sonst liefert die Funktion den Fehler #WERT!.

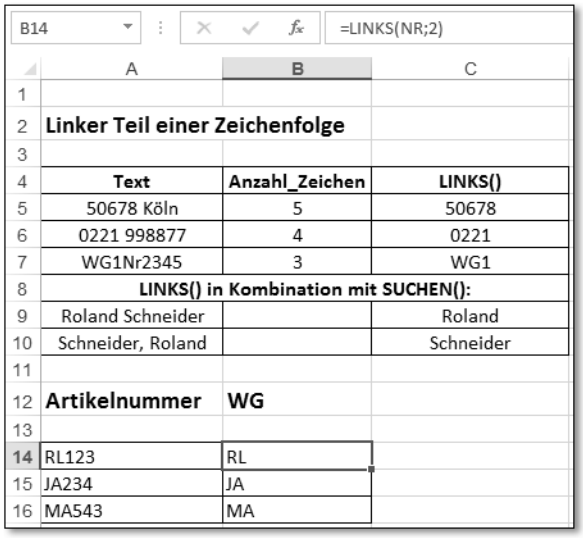


Abbildung 12.15 Abtrennen des linken Teils einer Zeichenfolge

Abbildung 12.16 zeigt eine Kombination der Funktion mit SUCHEN(), die es erlaubt, den Vor- oder den Nachnamen zu extrahieren. Die Anzahl der abzutrennenden Zeichen wird dabei errechnet, indem die Stelle gesucht wird, die das Leerzeichen zwischen den Namen enthält bzw. das Komma in der zweiten Formel.

```
=LINKS(A4;SUCHEN(" ";A4)-1)
=LINKS(A%;SUCHEN(",";A%)-1)
```

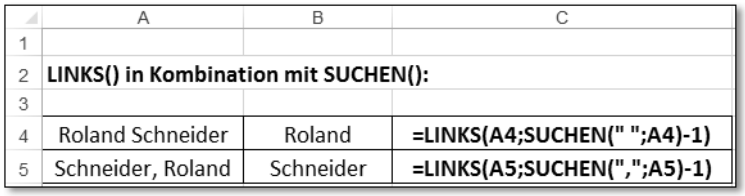


Abbildung 12.16 LINKS() in Kombination mit SUCHEN()

Die Funktion LINKS() bietet auch eine einfache Möglichkeit, Zeichenfolgen zu kürzen, um Kurzbezeichnungen zu produzieren. Enthalten beispielsweise die ersten beiden Zeichen einer Artikelnummer die Kennzeichnung der Warengruppe, kann diese mit

```
=LINKS(NR;2)
```

extrahiert werden, wenn der Bereich der Artikelnummern vorher mit dem Namen NR belegt wurde.

	A	B
1		
2	**Kürzel generieren**	
3		
4	**Artikelnummer**	**WG**
5	RL123	RL
6	JA234	JA
7	MA543	MA

Abbildung 12.17 Kürzel generieren

RECHTS()

RIGHT()

Syntax: RECHTS(**Text**; Anzahl_Zeichen)

Beispiel: =RECHTS("Artikelnummer 23456";5)

Ergebnis: 23456

Mit der Funktion RECHTS() können Sie aus einer Zeichenfolge den rechten Teil abgreifen. Die Funktion liefert als Ergebnis die mit Anzahl_Zeichen festgelegte Zeichenzahl vom Ende der Zeichenfolge gerechnet, die mit dem Argument Text angegeben ist, der vordere Rest wird also abgeschnitten. Wird für Anzahl_Zeichen kein Wert angegeben, so wird 1 vorgegeben.

	A	B	C
1			
2	**Rechter Teil einer Zeichenfolge**		
3			
4	**Text**	**Anzahl_Zeichen**	**RECHTS()**
5	50678 Köln	4	Köln
6	0221 998877	6	998877
7	WG1Nr2345	6	Nr2345

Abbildung 12.18 Abtrennen des rechten Teils einer Zeichenfolge

Ist der Wert größer als die Anzahl der vorhandenen Zeichen, wird der gesamte Text ausgegeben. `Anzahl_Zeichen` darf aber nicht negativ sein, sonst liefert die Funktion den Fehler `#WERT!`.

12.2.2 Abgreifen von Namen

Ist nicht vorgegeben, an welcher Stelle der Schnitt vorgenommen werden soll, können Sie die Funktion mit der Funktion `SUCHEN()` und mit der Funktion `LÄNGE()` kombinieren. Um aus einer Zelle mit dem Vor- und Nachnamen Letzteren zu extrahieren, suchen Sie das Leerzeichen zwischen den beiden Namen. Die Position dieses Zeichens können Sie dann von der Anzahl der Zeichen der gesamten Zeichenfolge abziehen, die Sie mit Hilfe der Funktion `LÄNGE()` ermittelt haben.

Die Formel ist:

```
=RECHTS(A9;LÄNGE(A9)-SUCHEN(" ";A9))
```

	A	B
1		
2	**Umgang mit Namen**	
3		
4	**RECHTS() in Kombination mit SUCHEN():**	
5	Roland Schneider	Schneider
6	Hans Werner	Werner
7	Ilse Dore	Dore
8	Karin Maier	Maier
9	**=RECHTS(A5;LÄNGE(A5)-SUCHEN(" ";A5))**	
10		
11	**RECHTS() in Kombination mit LINKS() und SUCHEN():**	
12	Schneider, Roland	Roland Schneider
13	Werner, Hans	Hans Werner
14	Dore, Ilse	Ilse Dore
15	Maier, Karin	Karin Maier
16	**=RECHTS(A12;LÄNGE(A12)-SUCHEN(", ";A12)-1)&" "&LINKS(A12;SUCHEN(",";A12)-1)**	

Abbildung 12.19 Freistellen des Nachnamens und Tausch der Namen

Soll die Reihenfolge der Namen vertauscht werden, brauchen Sie eine noch etwas komplexere Formel. Abbildung 12.19 zeigt in der angezeigten Formel für die Zelle B12 eine Kombination der Funktionen `RECHTS()` und `LINKS()` mit den Funktionen `SUCHEN()` und

LÄNGE(), die es erlaubt, die Reihenfolge von Namen und Vornamen in einer Zeichenfolge zu vertauschen:

```
=RECHTS(A12;LÄNGE(A12)-SUCHEN(",";A12)-1)
& " " & LINKS(A12;SUCHEN(",";A12)-1)
```

Die Formel setzt die neue Zeichenfolge aus drei Teilen zusammen, die mit dem &-Operator verknüpft werden.

12.2.3 Minuszeichen umstellen

Eine manchmal nötige Operation ist das Umstellen des Minuszeichens bei Daten, die von anderen Anwendungen übernommen werden. Es kommt vor, dass dabei das Minuszeichen rechts von der Zahl angezeigt wird. Excel interpretiert diese Werte als Text und richtet sie als Vorgabe linksbündig aus.

Zum Umstellen kann folgende Formel genutzt werden:

```
=WENN(RECHTS(A16;1)="-";LINKS(A16;LÄNGE(A16)-1)*-1;A16)
```

Die Formel prüft, ob das letzte Zeichen ein Minuszeichen ist. Wenn ja, wird mit der Funktion LINKS() die gesamte Zeichenfolge mit Ausnahme der letzten Stelle übernommen und mit -1 multipliziert, um einen negativen Wert zu erzeugen, im anderen Fall wird einfach der positive Wert ausgegeben. Die Formel kann dann für die ganze Spalte kopiert werden.

	A	B
1		
2	**Minuszeichen umstellen**	
3		
4	100,78-	-100,78
5	100	100
6	12,34-	-12,34
7	**=WENN(RECHTS(A4;1)="-";LINKS(A4;LÄNGE(A4)-1)*-1;A4)**	

Abbildung 12.20 Aus einem nachgestellten Minuszeichen ein vorgestelltes machen

Es sei noch einmal darauf hingewiesen, dass sich die hier beschriebenen Umstellungen per Funktion in Excel 2013 auch mit der neuen Funktion **Blitzvorschau** erledigen lassen, die bereits in Kapitel 1 beschrieben wurde.

SÄUBERN()

CLEAN()

Syntax: SÄUBERN(**Text**)
Beispiel: siehe Abbildung 12.21

Die Funktion SÄUBERN() entfernt aus der mit dem Argument Text gelieferten Zeichenfolge alle nicht druckbaren Zeichen. In der Regel sind dies Steuerzeichen, die mit Texten aus anderen Programmen übernommen wurden und nicht ausgedruckt werden können. Berücksichtigt werden nur die ersten 32 Zeichen des 7-Bit-ASCII-Codes.

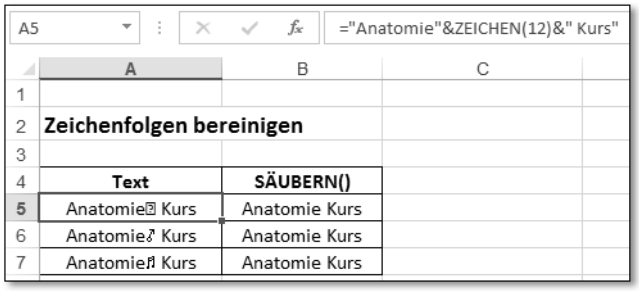


Abbildung 12.21 Entfernen von nicht druckbaren Zeichen

SUCHEN()

SEARCH()

Syntax: SUCHEN(**Suchtext**; **Text**; Erstes_Zeichen)
Beispiel: =SUCHEN(" ";"Artikel 3245")
Ergebnis: 8

Für die Suche nach Zeichenfolgen innerhalb einer Zeichenfolge stellt Excel zwei ähnliche Funktionen zur Verfügung, die Funktion SUCHEN() und die Funktion FINDEN(). Die Funktion SUCHEN() ignoriert die Groß- und Kleinschreibung und lässt Platzhalter zu. Die Funktion FINDEN() unterscheidet bei der Suche zwischen Groß- und Kleinschreibung und lässt keine Platzhalterzeichen zu.

Die Funktion SUCHEN() überprüft, ob die mit Suchtext angegebene Zeichenfolge in einer anderen Zeichenfolge vorkommt, die mit dem Argument Text angegeben wird. Suchtext darf die Platzhalter ? (für ein beliebiges Zeichen) und * (für ein oder meh-

rere beliebige Zeichen) enthalten. Die Suche beginnt bei der Position in der Zeichenfolge, die durch den Wert für `Erstes_Zeichen` bestimmt ist. Fehlt allerdings ein Wert für `Erstes_Zeichen`, wird `1` angenommen, also von Anfang an gesucht. Ergebnis der Funktion ist eine Zahl für die Position von `Suchtext` in `Text`. Dabei wird immer vom ersten Zeichen von Text aus gezählt, unabhängig von dem Wert für `Erstes_Zeichen`.

Wird der `Suchtext` in der Zeichenfolge nicht gefunden, wird der Fehler `#WERT!` ausgegeben. Der Wert von `Erstes_Zeichen` muss größer `0` sein, sonst liefert die Funktion ebenfalls den Fehler `#WERT!`.

	A	B	C	D	E
1					
2	**Zeichen suchen**				
3					
4	**Suchtext**	**Text**	**Erstes_Zeichen**	**SUCHEN()**	**FINDEN()**
5	a	Anatomie	1	1	3
6	O	Anatomie	1	5	#WERT!
7	o	Anatomie	6	#WERT!	#WERT!
8	z	Anatomie	1	#WERT!	#WERT!
9	o	Anatomie	4	5	5
10	@	jk@net.de	1	3	3
11	a?o	Anatomie	2	3	#WERT!
12	a*m	Anatomie	3	3	#WERT!

Abbildung 12.22 `SUCHEN()` und `FINDEN()` im Vergleich

T()

T()

Syntax:	`T(`**`Wert`**`)`
Beispiel:	`=T("Text")`
Ergibt:	`Text`

Die Funktion `T()` überprüft, ob das mit `Wert` angegebene Argument eine Zahl oder eine Zeichenfolge liefert. Wird eine Zahl oder ein Wahrheitswert gefunden, dann gibt die Funktion Leertext aus, wird Text gefunden, wird genau dieser ausgegeben. Fehlerwerte werden als Fehlerwerte ausgegeben.

Mit Hilfe von `T()` kann etwa von einer Spalte mit gemischten Datentypen eine andere Spalte erzeugt werden, die nur noch die Textdaten anzeigt.

	A	B	C
1			
2	**Umwandlung in Text**		
3			
4	**Wert**	**T()**	**ISTTEXT()**
5	8899		FALSCH
6	12.11.2007		FALSCH
7	Anatomie	Anatomie	WAHR
8	#NV	#NV	FALSCH
9	WAHR		FALSCH

Abbildung 12.23 Überprüfung der Ergebnisse von `T()` mit `ISTTEXT()`

TEIL()

MID()

Syntax: `TEIL(`**`Text; Erstes_Zeichen; Anzahl_Zeichen`**`)`

Beispiel: `=TEIL("ArtikelNr: 07560";12;5)`

Ergebnis: `07560`

Die wichtigste Funktion, um aus einer Zeichenfolge bestimmte Teile herauszuziehen, ist die Funktion `TEIL()`. Die Funktion schneidet aus der mit `Text` angegebenen Zeichenfolge eine Unterzeichenfolge ab der mit `Erstes_Zeichen` festgelegten Position in der mit `Anzahl_Zeichen` bestimmten Länge aus.

Ist `Erstes_Zeichen` größer als die Gesamtlänge von `Text`, gibt die Funktion eine leere Zeichenfolge zurück. Ist der Wert für `Erstes_Zeichen` plus `Anzahl_Zeichen` größer als die Länge von `Text`, gibt die Funktion die Zeichen ab `Erstes_Zeichen` zurück.

Ist `Erstes_Zeichen` kleiner `1` oder `Anzahl_Zeichen` negativ, liefert die Funktion den Fehler `#WERT!`.

	A	B	C	D
1				
2	**Teile von Zeichenfolgen extrahieren**			
3				
4	**Text**	**Erstes_Zeichen**	**Anzahl_Zeichen**	**TEIL()**
5	Anatomiekurs	1	8	Anatomie
6	Anatomiekurs	9	5	kurs
7	Anatomiekurs	12	2	s
8	Anatomiekurs	13	5	
9	Anatomiekurs	-5	5	#WERT!
10	Anatomiekurs	10	-5	#WERT!
11	12.345.678,00 €	3	1	3

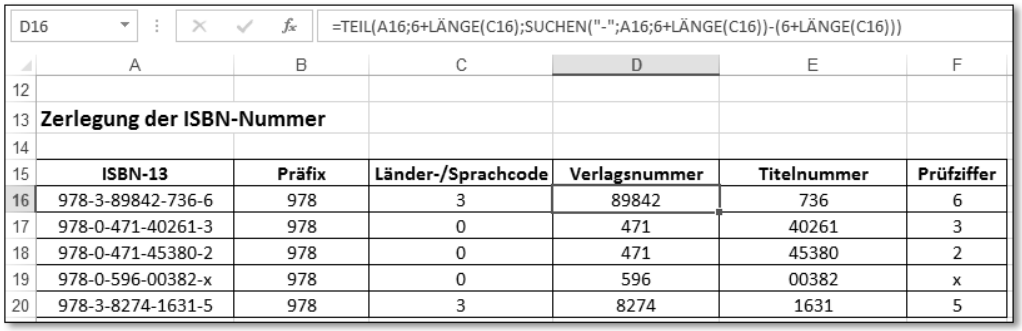
Abbildung 12.24 Beispiele für Teilzeichenfolgen

12.2.4 Zerlegung einer ISBN-Nummer

TEIL() ist insbesondere nützlich, wenn es darum geht, aus Zeichenfolgen, die in einer vorgeschriebenen Weise zusammengesetzt sind, bestimmte Teile zu extrahieren. Der andere Anwendungsbereich ist, eine Zeichenfolge aus Teilen anderer Zeichenfolgen neu zusammenzusetzen.

Die Zerlegung ist relativ einfach, wenn die einzelnen Teile immer eine festgelegte Länge haben. Bei einer ISBN-Nummer ist das aber beispielsweise nicht der Fall. Hier hilft dann eine Kombination der Funktion mit anderen Textfunktionen.

In der Tabelle in Abbildung 12.25 wird gezeigt, wie mit Trennstrichen gegliederte ISBN-Nummern in ihre Bestandteile zerlegt werden können:



	A	B	C	D	E	F
12						
13	**Zerlegung der ISBN-Nummer**					
14						
15	**ISBN-13**	**Präfix**	**Länder-/Sprachcode**	**Verlagsnummer**	**Titelnummer**	**Prüfziffer**
16	978-3-89842-736-6	978	3	89842	736	6
17	978-0-471-40261-3	978	0	471	40261	3
18	978-0-471-45380-2	978	0	471	45380	2
19	978-0-596-00382-x	978	0	596	00382	x
20	978-3-8274-1631-5	978	3	8274	1631	5

Abbildung 12.25 Zerlegung von ISBN-Nummern

Die Formel für die Spalte »Präfix« ist einfach:

```
=TEIL(A16;1;3)
```

Für die »Länder-/Sprachcode«-Spalte wird mit einer Kombination mit der SUCHEN()-Funktion gearbeitet, um die Stelle zu finden, wo der zweite Trennstrich auftaucht:

```
=TEIL(A16;5;SUCHEN("-";A16;5)-5)
```

Die Formel für die Verlagsnummer wertet nun die Länge des eben ermittelten Codes aus:

```
=TEIL(A16;6+LÄNGE(C16);SUCHEN("-";A16;6+LÄNGE(C16))-(6+LÄNGE(C16)))
```

Noch komplexer ist die Formel für die Titelnummer, die die Längen der beiden vorherigen Zellen auswertet:

```
=TEIL(A16;7+LÄNGE(C16)+LÄNGE(D16);LÄNGE(A16)-(LÄNGE(C16)+LÄNGE(D16)+8))
```

Die Formel für die Prüfziffer ist wieder einfach:

```
=TEIL(A16;17;1)
```

Nachdem die Daten zerlegt sind, lassen sich beispielsweise Buchdaten sehr einfach nach den Länder-/Sprachcodes oder nach der Verlagsnummer sortieren.

12.2.5 Datumswerte aufbereiten

Mit Hilfe der Funktionen `LINKS()`, `RECHTS()` und `TEIL()` ist es auch möglich, Datumswerte aufzubereiten, die Excel in der übermittelten Form zunächst nicht verarbeiten kann. Werden beispielsweise Daten importiert, die ein Datum in der Form »JJJJMMDD« – 20131221 – anzeigen, lassen sich die verschiedenen Bestandteile, die für die Funktion `DATUM()` benötigt werden, einzeln abgreifen. Steht die Zeichenfolge in A5, heißt die entsprechende Formel:

```
=DATUM(LINKS(A5;4);TEIL(A5;5;2);RECHTS(A5;2))
```

TEXT()

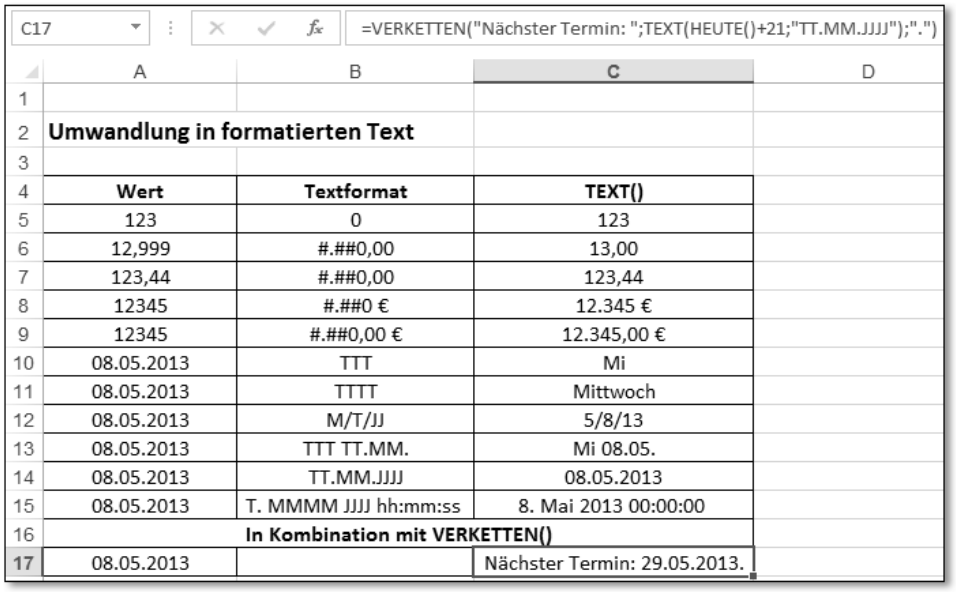
TEXT()

Syntax: `TEXT(`**`Wert; Textformat`**`)`
Beispiel: `=TEXT("34594";"TT.MM.JJJJ")`
Ergebnis: `17.09.1994`

Die Funktion `TEXT()` wandelt den mit dem Argument `Wert` übergebenen numerischen Wert in einen Text um und verwendet dabei das Format, das mit dem Argument `Textformat` angegeben wird. Gemeint ist damit eines der Zahlen-, Datums- oder Zeitformate, die Excel über den Dialog **Zellen formatieren** anbietet. Auch benutzerdefinierte Formate sind möglich. Dabei werden die Codes angegeben, mit denen das Format definiert wird, so wie es in Kapitel 1 im Abschnitt »Formatcodes« beschrieben ist. Nur die Roteinfärbung von negativen Werten wird nicht unterstützt.

Werden fehlerhafte Codes für das Format eingegeben, etwa ein zusätzliches 1000er-Trennzeichen, wird es ignoriert. Wird ein zweites Dezimalzeichen eingefügt, wird es dagegen mitausgegeben, so dass das Ergebnis numerisch nicht mehr korrekt ist.

Das Ergebnis der Funktion ist also eine Zeichenfolge, die aussieht wie eine gültig formatierte Zahl. Der Nutzen dieser Umwandlung liegt insbesondere darin, dass sich das Ergebnis der Funktion leicht mit einem anderen Text verketten lässt, wie das letzte Beispiel in Abbildung 12.26 zeigt:



	A	B	C	D
1				
2	**Umwandlung in formatierten Text**			
3				
4	**Wert**	**Textformat**	**TEXT()**	
5	123	0	123	
6	12,999	#.##0,00	13,00	
7	123,44	#.##0,00	123,44	
8	12345	#.##0 €	12.345 €	
9	12345	#.##0,00 €	12.345,00 €	
10	08.05.2013	TTT	Mi	
11	08.05.2013	TTTT	Mittwoch	
12	08.05.2013	M/T/JJ	5/8/13	
13	08.05.2013	TTT TT.MM.	Mi 08.05.	
14	08.05.2013	TT.MM.JJJJ	08.05.2013	
15	08.05.2013	T. MMMM JJJJ hh:mm:ss	8. Mai 2013 00:00:00	
16	**In Kombination mit VERKETTEN()**			
17	08.05.2013		Nächster Termin: 29.05.2013.	

Abbildung 12.26 Umwandlung numerischer Werte in formatierten Text

Falls Sie versuchen, den erläuternden Text einfach direkt mit dem Ergebnis der Funktion `HEUTE()` zu verketten, also mit:

```
=VERKETTEN("Nächster Termin: "; HEUTE())
```

würden Sie anstelle des formatierten Datums einfach nur die serielle Nummer des entsprechenden Tages finden.

UNICODE()

UNICODE()

Syntax: `UNICODE(`**`Text`**`)`
Beispiel: `=UNICODE("!")`
Ergebnis: `33`

Die Funktion UNICODE() liefert den dem Unicode-Standard entsprechenden Code für das erste Zeichen in Text. Der Code gibt den Codepoint an, dem ein bestimmtes Zeichen im Coderaum zugeordnet ist. Der Coderaum von Unicode umfasst Werte zwischen 0 und 1114111.

Ist Text eine leere Zeichenfolge oder der Verweis auf eine leere Zelle, liefert die Funktion den Fehlerwert #WERT!.

	A	B	C
1			
2	**UNICODE**		
3			
4	Text	UNICODE()	CODE()
5	A	65	65
6	B	66	66
7	C	67	67
8	a	97	97
9	b	98	98
10	c	99	99
11	1	49	49
12	2	50	50
13	3	51	51
14	α	945	63
15	β	946	63
16	γ	947	63
17	€	8364	128

Abbildung 12.27 Vergleich von UNICODE() und CODE()

UNIZEICHEN()

UNICHAR()

Syntax: UNIZEICHEN(**Zahl**)
Beispiel: =UNIZEICHEN(33)
Ergebnis: "!"

Die Funktion UNIZEICHEN() liefert das Zeichen, das dem mit Zahl angegebenen Code entspricht, siehe auch UNICODE().

Ist Zahl gleich 0 oder außerhalb des Bereichs vorhandener Codes oder nicht numerisch, liefert die Funktion den Fehler #WERT!.

	A	B	C
1			
2	**UNIZEICHEN**		
3			
4	UNICODE()	UNIZEICHEN()	
5	65	A	
6	66	B	
7	67	C	
8	97	a	
9	98	b	
10	99	c	
11	49	1	
12	50	2	
13	51	3	
14	945	α	
15	946	β	
16	947	γ	
17	8364	€	

Abbildung 12.28 Beispiele für `UNIZEICHEN()`

VERKETTEN()

CONCATENATE()

Syntax: `VERKETTEN(`**`Text1`**`; Text2; ...)`

Beispiel: `=VERKETTEN("Eigen";"anteil")`

Ergebnis: `Eigenanteil`

Die Funktion `VERKETTEN()` ist eine Alternative zu dem Verkettungsoperator `&`, mit dem Zeichenfolgen verknüpft werden können. Als Argumente sind seit Excel 2007 bis zu 255 Zeichenfolgen – `Text1`, `Text2` etc. – erlaubt, in den älteren Versionen bis zu 29.

Beachtet werden muss, dass die Verkettung ohne Einfügen von Leerzeichen geschieht. Wird ein Leerzeichen benötigt, muss es also gesondert angegeben werden. Die Funktion erfüllt somit die gleichen Aufgaben wie der &-Operator.

	A	B	C	D
1				
2	**Zeichenfolgen verketten**			
3				
4	**Text1**	**Text2**	**Text3**	**VERKETTEN()**
5	Da	Da	Da	DaDaDa
6	WG1	Art123	TypB	WG1Art123TypB
7	Termin:		22.Mai	Termin: 22.Mai

Abbildung 12.29 Textteile zusammenfügen

WECHSELN()

SUBSTITUTE()

Syntax: `WECHSELN(Text; Alter_Text; Neuer_Text; Ntes_Auftreten)`
Beispiel: `=WECHSELN("Aufschwung 1";"1";"2")`
Ergebnis: `Aufschwung 2`

Die Funktion `WECHSELN()` sucht in einem Text nach einer Zeichenfolge – `Alter_Text` –, um diese Zeichenfolge dann durch eine andere – `Neuer_Text` – zu ersetzen. Dabei wird die Groß- und Kleinschreibung berücksichtigt.

Das optionale Argument `Ntes_Auftreten` bestimmt, wie oft der Vorgang des Ersetzens wiederholt werden soll. Wird das Argument auf `1` gesetzt, dann wird nur die erste Fundstelle geändert; wird das Argument nicht belegt, werden dagegen alle Fundstellen geändert. Im Unterschied zu der ähnlichen Funktion `ERSETZEN()` ist hier also ein mehrfacher Austausch von Zeichenfolgen möglich.

Ist der Wert für `Ntes_Auftreten` höher als die Anzahl des tatsächlichen Auftretens, gibt die Funktion die Zeichenfolge `Alter_Text` unverändert wieder. Ist der Wert < `1`, liefert die Funktion den Fehlerwert `#WERT!`.

E13 =WECHSELN(WECHSELN(A13;"N";"L");"l";"n")

	A	B	C	D	E
1					
2	**Zeichenfolgen austauschen**				
3					
4	**Text**	**Alter_Text**	**Neuer_Text**	**ntes_Auftreten**	**WECHSELN()**
5	Anatomiekurs	kurs	seminar		Anatomieseminar
6	Jan Willem		-		Jan-Willem
7	Burg Erb	Erb	Erbenhausen		Burg Erbenhausen
8	Katoroko	o	u	2	Katoruko
9	123 €	€	EUR	1	123 EUR
10	**In Kombination mit LÄNGE() Vorkommen eines Zeichens zählen:**				
11	Katoroko	o			3
12	**Wechselweise tauschen**				
13	Nobel				Loben

Abbildung 12.30 Gezielter Austausch von Zeichenfolgen in Texten

In Kombination mit der Funktion `LÄNGE()` kann `WECHSELN()` auch verwendet werden, um das Vorkommen von Zeichen zu zählen. Die Formel zu der in Abbildung 12.30 gezeigten Zelle E11 sieht so aus:

```
=LÄNGE(A11)-LÄNGE(WECHSELN(A11;B11;""))
```

Der Trick dieser Formel besteht darin, dass von der Gesamtlänge der Zeichenfolge der Rest abgezogen wird, der übrig bleibt, nachdem alle gesuchten Zeichen durch ein Leerzeichen ersetzt wurden. In Zelle E13 finden Sie ein spielerisches Beispiel dafür, wie die Funktion verschachtelt genutzt werden kann, um gleich zweimal auszutauschen:

```
=WECHSELN(WECHSELN(A13;"N";"L");"l";"n")
```

12.2.6 Ersetzen von Trennzeichen

Werden Zahlenkolonnen aus anderen Systemen importiert, kann es nötig werden, die dort verwendeten Trennzeichen durch die in der eigenen Region gültigen Trennzeichen zu ersetzen. In den USA wird bekanntlich der Punkt als Dezimaltrennzeichen verwendet und das Komma zur Trennung der Tausendergruppen. Die Klippe in diesem Fall ist, dass zunächst das Komma durch ein beliebiges anderes Zeichen – etwa ein Leerzeichen – ersetzt werden muss, damit der Punkt anschließend durch Kommas ersetzt werden kann. Zuletzt wird dann das Leerzeichen durch den Punkt ersetzt. Dazu ist eine dreifache Schachtelung notwendig:

```
=WECHSELN(WECHSELN(WECHSELN(A5;",";" ");".";",");" ";".")
```

Das Ergebnis ist eine Zeichenfolge, die wie eine ordentlich formatierte Zahl aussieht. Um daraus eine Zahl zu machen, fügen Sie die Funktionsfolge als Argument in die Funktion `WERT()` ein.

```
=WERT(WECHSELN(WECHSELN(WECHSELN(A5;",";" ");".";",");" ";"."))
```

Dabei geht das 1000er-Trennzeichen zwar wieder verloren, weil die Funktion ihr Ergebnis im Standardformat ausgibt, aber dagegen hilft ja die Zuordnung eines anderen Zahlenformats. Abbildung 12.31 zeigt einige Beispiele.

	A	B
1		
2	**Trennzeichen austauschen**	
3		
4	**Wert**	**WECHSELN()**
5	1,000,000.55	1.000.000,55
6	1,500,000.56	1.500.000,56
7	400,000.57	400.000,57
8	3,000,000.58	3.000.000,58
9	1,000,700.59	1.000.700,59

Abbildung 12.31 Austausch von Trennzeichen

12.2.7 Ersetzen von nicht druckbaren Zeichen

Anstatt nicht druckbare Zeichen mit der Funktion SÄUBERN() zu entfernen, wie oben beschrieben, können Sie auch mit einer gezielten Ersetzung arbeiten. Wenn beispielsweise größere Textmengen in eine Zelle A1 importiert werden, in denen sich Zeilenumbruchzeichen befinden, können Sie diese mit der Formel

```
=WECHSELN(A1;ZEICHEN(10);" ")
```

durch Leerzeichen ersetzen.

WERT()

VALUE()

Syntax: WERT(**Text**)
Beispiel: =WERT("19.09.1994")
Ergebnis: 34596

Die Funktion WERT() wandelt die mit dem Argument Text gelieferte Zeichenfolge in einen Wert um und gibt als Ergebnis dann die entsprechende Zahl aus. Text darf dabei die entsprechenden Zeichen aus einem Excel-Zahlenformat enthalten. Zeichenfolgen, die als Datums- oder Zeitformate interpretiert werden können, werden als die jeweilige serielle Zahl ausgegeben.

Lässt sich Text nicht als Zahl interpretieren, gibt die Funktion #WERT! aus.

	A	B	C	D
1				
2	**Texte in numerische Werte umwandeln**			
3				
4	**Text**	**WERT()**	**ISTZAHL()**	**N()**
5	666,99	666,99	WAHR	666,99
6	144	144	WAHR	144,00
7	-122	-122	WAHR	-122,00
8	-12,56	-12,56	FALSCH	0
9	3,14	3,14	FALSCH	0
10	28.03.2007	39169	WAHR	39169
11	28.07.2007 18:42	39291,77917	WAHR	39291,78
12	1. Platz	#WERT!	FALSCH	0
13	WAHR	#WERT!	FALSCH	1

Abbildung 12.32 Test des Ergebnisses von WERT() mit ISTZAHL() und N()

WIEDERHOLEN()

REPT()

Syntax: `WIEDERHOLEN(`**`Text`**`;` **`Multiplikator`**`)`
Beispiel: `=WIEDERHOLEN("=";13)`
Ergebnis: `=============`

Die Funktion `WIEDERHOLEN()` trägt die mit dem Argument `Text` angegebene Zeichenfolge so oft hintereinander in die Zelle ein, wie durch das Argument `Multiplikator` angegeben wird. Die Zeichenfolge kann aus mehr als einem Zeichen bestehen.

	A	B	C
1			
2	**Zeichen wiederholen**		
3			
4	**Text**	**Multiplikator**	**WIEDERHOLEN()**
5	Da	3	DaDaDa
6	#	4	####
7	-	15	---------------
8	=	15	================

Abbildung 12.33 Beispiele für `WIEDERHOLEN()`

Hat `Multiplikator` den Wert `0`, wird eine leere Zeichenfolge ausgegeben. Ist der Wert `> 32767`, übersteigt also die maximale Zeichenanzahl für eine Zelle, erscheint der Fehlerwert `#WERT!`. Nachkommastellen werden dagegen abgeschnitten.

Eine weitere Möglichkeit, die die Funktion `WIEDERHOLEN()` eröffnet, sind Pseudodiagramme, die bestimmte Werte durch eine entsprechende Wiederholung eines Zeichens darstellen. Abbildung 12.34 zeigt einen Urlaubsplan, in dem die Anzahl der Tage mit einem Zeichen aus dem Font Webdings angezeigt wird.

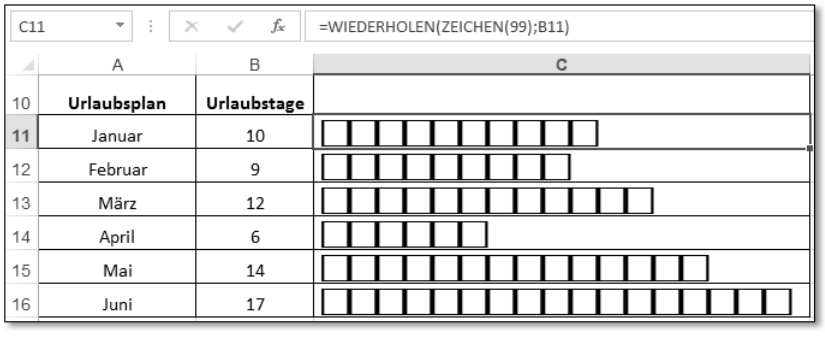


Abbildung 12.34 Pseudodiagramme mit der Funktion `WIEDERHOLEN()`

Die Funktion kann auch verwendet werden, um Zellen mit führenden oder nachfolgenden Zeichen aufzufüllen, um beispielsweise eine Spalte mit einer einheitlichen Länge zu erreichen. Das funktioniert ganz gut, wenn Sie dabei eine Schriftart wie Courier New verwenden, bei der jeder Buchstabe die gleiche Breite hat.

Abbildung 12.35 zeigt ein Beispiel für den Entwurf eines Inhaltsverzeichnisses. Die in Spalte C verwendete Formel nimmt Bezug auf eine ausgeblendete Spalte D, die jeweils den Text des Kapiteltitels enthält:

```
=D4 & WIEDERHOLEN(".";30 - LÄNGE(D4))
```

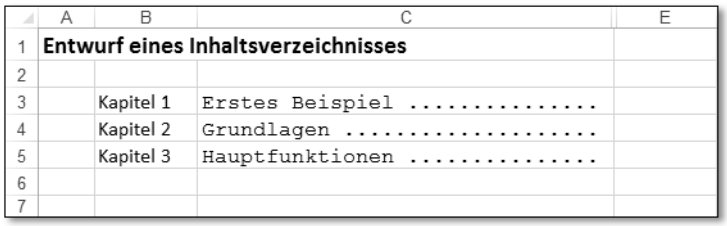


Abbildung 12.35 Zellen variabel auffüllen mit der Funktion `WIEDERHOLEN()`

ZAHLENWERT()

NUMBERVALUE()

Syntax: `ZAHLENWERT(`**`Text`**`; Dezimaltrennzeichen; Gruppentrennzeichen)`

Beispiel: `=ZAHLENWERT("100,500.50";".";",")`

Ergebnis: `100500,5`

Die in Excel 2013 neue Funktion `ZAHLENWERT()` konvertiert unabhängig vom aktuellen Gebietsschema Text in Zahlen. Die Funktion kann beispielsweise helfen, Datenbestände, die aus anderen Ländern übernommen worden sind, in denen mit anderen Trennzeichen gearbeitet wird, in ein für die hiesige Region passendes Format zu bringen.

Mit den optionalen Argumenten `Dezimaltrennzeichen` und `Gruppentrennzeichen` geben Sie an, welche Zeichen in dem Argument `Text` für die Trennung der Dezimalstellen und für die Bildung von 1000er-Gruppen aktuell verwendet werden. Excel gibt das Ergebnis der Funktion im Standardformat aus, also ohne Gruppentrennzeichen und unter Abschneidung nachfolgender Nullen.

	A	B	C
1			
2	**Texte in Zahlen umwandeln**		
3			
4	**Text**	**ZAHLENWERT()**	
5	12,000.50	12000,5	**=ZAHLENWERT(A5;".";",")**
6	1.44	1,44	**=ZAHLENWERT(A6;".";",")**
7	-122,000.250	#WERT!	**=ZAHLENWERT(A7;".";",")**
8	1,123,123.55	1123123,55	**=ZAHLENWERT(A8;".";",")**
9	40 000	40000	**=ZAHLENWERT(A9;".";",")**
10		0	**=ZAHLENWERT(A10;".";",")**
11	12.000.50	#WERT!	**=ZAHLENWERT(A11;".";",")**
12	12.000,51	#WERT!	**=ZAHLENWERT(A12;".";",")**

Abbildung 12.36 Umwandlung von Zahlenwerten

Werden die Argumente `Dezimaltrennzeichen` und `Gruppentrennzeichen` nicht angegeben, interpretiert die Funktion die mit dem Argument `Text` angegebene Zeichenfolge nach dem aktuellen Gebietsschema. Werden für die Argumente mehrere Zeichen angegeben, wird nur das erste verwendet. Ist `Text` eine leere Zeichenfolge, ist das Ergebnis `0`. Leerzeichen innerhalb der Zeichenfolge werden ignoriert. "40 000" wird also als `40000` gewertet.

Wird ein Dezimaltrennzeichen mehrfach verwendet, gibt die Funktion den Fehlerwert `#WERT!` zurück. Das gilt auch, wenn in der Zeichenfolge das Dezimalzeichen vor dem Gruppentrennzeichen erscheint. Lässt sich `Text` nicht als Zahl verstehen, liefert die Funktion ebenfalls den Fehlerwert `#WERT!`.

ZEICHEN()

CHAR()

Syntax: `ZEICHEN(`**`Zahl`**`)`
Beispiel: `=ZEICHEN(33)`
Ergebnis: `!`

Die Funktion `ZEICHEN()` ist die Umkehrung der `CODE()`-Funktion. Sie liefert für eine bestimmte Codezahl das entsprechende Zeichen. Welches Zeichen angezeigt wird, hängt allerdings auch von dem gewählten Zeichensatz ab. Spezialzeichensätze wie Symbol oder Wingdings zeigen andere Zeichen an als "normale" Zeichensätze wie Calibri oder Arial.

`Zahl` darf ein Wert von `1` bis `255` sein, ansonsten liefert die Funktion den Fehler `#WERT!`.

Die Funktion kann unter anderem dazu verwendet werden, um bestimmte Sonderzeichen anzuzeigen oder in Zeichenfolgen einzubinden:

```
=ZEICHEN(169) & " " & "Helmut Vonhoegen"
```

ergibt beispielsweise

```
© Helmut Vonhoegen
```

	B	C	E	F	H	I	K	L	N	O	Q	R	T	U
4	ZEICHEN()	CODE()	ZEICHEN()	CODE()	ZEICHEN()	CODE()	ZEICHEN()	CODE()	ZEICHEN()	CODE()	ZEICHEN()	CODE()	ZEICHEN()	CODE()
5		1	&	38	K	75	p	112	•	149	º	186	ß	223
6	┐	2	'	39	L	76	q	113	–	150	»	187	à	224
7	└	3	(	40	M	77	r	114	—	151	¼	188	á	225
8	┘	4	)	41	N	78	s	115	˜	152	½	189	â	226
9	├	5	*	42	O	79	t	116	™	153	¾	190	ã	227
10	─	6	+	43	P	80	u	117	š	154	¿	191	ä	228
11	•	7	,	44	Q	81	v	118	›	155	À	192	å	229
12	■	8	-	45	R	82	w	119	œ	156	Á	193	æ	230
13		9	.	46	S	83	x	120		157	Â	194	ç	231
14		10	/	47	T	84	y	121	ž	158	Ã	195	è	232
15	♂	11	0	48	U	85	z	122	Ÿ	159	Ä	196	é	233
16	◙	12	1	49	V	86	{	123		160	Å	197	ê	234
17		13	2	50	W	87	|	124	¡	161	Æ	198	ë	235
18	♫	14	3	51	X	88	}	125	¢	162	Ç	199	ì	236
19	‼	15	4	52	Y	89	~	126	£	163	È	200	í	237
20	┼	16	5	53	Z	90	⌂	127	¤	164	É	201	î	238
21	◄	17	6	54	[	91	€	128	¥	165	Ê	202	ï	239
22	↕	18	7	55	\	92		129	¦	166	Ë	203	ð	240
23	‼	19	8	56	]	93	‚	130	§	167	Ì	204	ñ	241
24	¶	20	9	57	^	94	ƒ	131	¨	168	Í	205	ò	242
25	┴	21	:	58	_	95	„	132	©	169	Î	206	ó	243
26	┬	22	;	59	`	96	…	133	ª	170	Ï	207	ô	244
27	┤	23	<	60	a	97	†	134	«	171	Ð	208	õ	245
28	↑	24	=	61	b	98	‡	135	¬	172	Ñ	209	ö	246
29	├	25	>	62	c	99	ˆ	136	-	173	Ò	210	÷	247
30	→	26	?	63	d	100	‰	137	®	174	Ó	211	ø	248
31	←	27	@	64	e	101	Š	138	¯	175	Ô	212	ù	249
32		28	A	65	f	102	‹	139	°	176	Õ	213	ú	250
33		29	B	66	g	103	Œ	140	±	177	Ö	214	û	251
34		30	C	67	h	104		141	²	178	×	215	ü	252
35		31	D	68	i	105	Ž	142	³	179	Ø	216	ý	253
36		32	E	69	j	106		143	´	180	Ù	217	þ	254
37	!	33	F	70	k	107		144	µ	181	Ú	218	ÿ	255
38	"	34	G	71	l	108	‘	145	¶	182	Û	219		
39	#	35	H	72	m	109	’	146	·	183	Ü	220		
40	$	36	I	73	n	110	“	147	¸	184	Ý	221		
41	%	37	J	74	o	111	”	148	¹	185	Þ	222		

Abbildung 12.37 ANSI-Zeichentabelle mit den entsprechenden `CODE()`-Werten

12.2.8 Fortlaufende Zeichenreihen

Die Funktion `ZEICHEN()` bietet auch eine Lösung für die Bildung von Reihen, die alphabetisch geordnet sind. Die im ersten Kapitel beschriebenen Reihenbildungen betreffen ja nur numerische Reihen oder Datums- oder Zeitreihen. Wenn Sie Zeilen oder Spalten mit alphabetischen Reihen beschriften wollen, können Sie aber weder mit dem Dialog **Reihe** noch mit dem Ausfüllkästchen oder dem Symbol **AutoAusfüllen** etwas ausrichten.

Eine der möglichen Lösungen ist eine Formel, die mit der `ZEICHEN()`-Funktion in Kombination mit der Funktion `ZEILE()` oder mit der Funktion `SPALTE()` arbeitet. Dabei wird

die Nummer des Zeichens von der jeweiligen Zeilen- oder Spaltennummer abhängig gemacht. Wenn die Länge der Spalte vorgegeben ist, können Sie hier gut mit einer Matrixformel arbeiten. In dem Beispiel in Abbildung 12.38 wird zunächst der Bereich A5:A18 markiert. Sollen die Zeilen mit "Versuch a", "Versuch b" etc. beschriftet werden, können Sie folgende Formel verwenden:

```
{="Versuch " & ZEICHEN(97+ZEILE(A5:A18)-ZEILE(A5))}
```

Die Formel wird mit Strg + ⇧ + ↵ abgeschlossen, also als Matrixformel für den markierten Bereich.

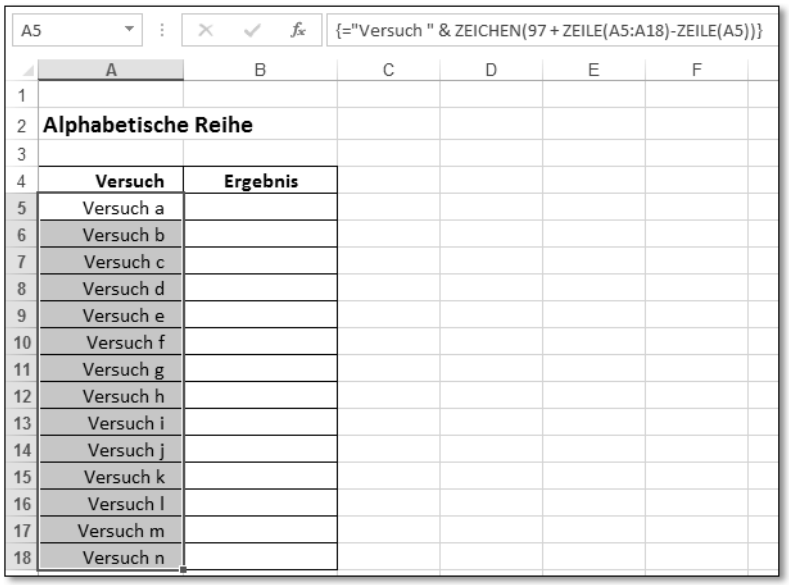


Abbildung 12.38 Spalte mit einer alphabetischen Reihe

Sollen die Spalten einer Tabelle mit Buchstaben bezeichnet werden, verwenden Sie in der Formel statt der Funktion `ZEILE()` die Funktion `SPALTE()`. Wenn Sie statt der Kleinbuchstaben mit Großbuchstaben arbeiten wollen, beginnen Sie mit `ZEICHEN(65)`.

Ist die Beschriftungsspalte in der Länge nicht fixiert, können Sie in der ersten Zelle eine etwas andere Formel ablegen:

```
="Versuch " & ZEICHEN(97+ZEILE(A5)-ZEILE($A5$))
```

Diese Formel, die einen relativen Bezug mit einem absoluten Bezug kombiniert, kann beispielsweise mit dem Ausfüllkästchen nach unten kopiert werden, bis der Buchstabe z erreicht wird.

13 Logische Funktionen

Funktion	Seite	Funktion	Seite
FALSCH()	790	WENN()	795
NICHT()	791	WENNFEHLER()	796
ODER()	791	WENNNV()	797
UND()	793	XODER()	797
WAHR()	794		

13.1 Einsatzbereiche der logischen Funktionen

Entscheidungen, sofern sie nicht willkürlich sind, werden in der Regel auf der Basis von Informationen getroffen, die eine bestimmte Situation ausreichend genug kennzeichnen, um zu einer begründeten Beurteilung zu kommen. Die Lage eines Unternehmens wird als positiv eingeschätzt, wenn bestimmte Indikatoren bestimmte Werte erreichen, und umgekehrt lässt sich eine kritische Entwicklung feststellen, wenn diese Indikatoren sich in die entgegengesetzte Richtung bewegen.

In all diesen Fällen können Sie Ihr Urteil an die Prüfung von Bedingungen und Nebenbedingungen knüpfen, die erfüllt sein müssen, um zu einem bestimmten Ergebnis zu kommen. Genau dies ist der Bereich, in dem die logischen Funktionen, die Excel zur Verfügung stellt, von Nutzen sind.

Für die Formulierung der zu prüfenden Bedingungen werden hauptsächlich die Vergleichsoperatoren `=`, `<`, `>`, `<=`, `>=`, `<>` herangezogen. Mit Ausnahme der in dieser Kategorie mit aufgeführten Funktion `WENN()`, `WENNFEHLER()` und `WENNNV()`, die Verzweigungen ermöglichen, liefern logische Funktionen als Ergebnis die Wahrheitswerte `WAHR` oder `FALSCH`.

Die Wahrheitswerte werden in bestimmtem Umfang von Excel zusätzlich mit `1` für `WAHR` und `0` für `FALSCH` ausgewertet, so dass sie in Berechnungen einbezogen werden können. Es macht aber einen Unterschied, ob Bezüge auf Zellen, die Wahrheitswerte enthalten, mit einfachen Operatoren wie + oder – verknüpft werden oder ob sie in Bereichsargumenten von Funktionen erscheinen. Abbildung 13.1 zeigt, dass die Addition dreier Wahrheitswerte in der Form

```
=B5+B9+B13
```

den Wert 2 liefert, eine entsprechende Summenfunktion aber den Wert 0, weil die Funktion die Wahrheitswerte nicht numerisch auswerten kann. Auch die Funktion WERT() liefert, wenn das Argument ein Wahrheitswert ist, keinen numerischen Wert, sondern den Fehlerwert #WERT!.

Die Abbildung zeigt außerdem in dem Bereich E9 bis H11, dass Excel einen Zelleintrag mit WAHR, =WAHR und =WAHR() als gleichwertig betrachtet.

	A	B	C	D	E	F	G	H
1								
2	**Umgang mit logischen Werten**							
3								
4	**FALSCH()**			=B5+B9+B13				
5	=FALSCH()	FALSCH		2				
6	=FALSCH()+1	1		=SUMME(B5;B9;B13)				
7	=FALSCH()*1	0		0				
8	**WAHR()**					WERT()	=E9+1	gleichwertig?
9	=WAHR()	WAHR		WAHR	WAHR	#WERT!	2	WAHR
10	=WAHR()+1	2		=WAHR	WAHR	#WERT!	2	WAHR
11	=WAHR()*1	1		=WAHR()	WAHR	#WERT!	2	WAHR
12	**NICHT()**							
13	=NICHT(B5)	WAHR						
14	=NICHT(B9)	FALSCH						
15	=NICHT("Berlin">"Bonn")	WAHR						
16	**UND()**							
17	=UND(B5;B9)	FALSCH						
18	=UND(5<10;5>4)	WAHR						
19	=UND(WAHR;NICHT(WAHR))	FALSCH						
20	**ODER()**							
21	=ODER(B5;B9)	WAHR						
22	=ODER(B10<2;B10>4)	FALSCH						
23	=ODER(WAHR;NICHT(WAHR))	WAHR						
24	**XODER()**							
25	=XODER(B5;B9)	FALSCH						
26	=XODER(B10<2;B10>4)	WAHR						
27	=XODER(WAHR;NICHT(WAHR)	WAHR						

Abbildung 13.1 Ergebnisse logischer Funktionen

Eine logische Formel kann nicht nur eine Bedingung enthalten, sondern auch mehrere gleichzeitig. Diese Bedingungen können entweder alternativ oder additiv formuliert werden. Alternative Bedingungen werden entweder mit der ODER()-Funktion oder der neuen XODER()-Funktion verknüpft, additive Bedingungen mit der UND()-Funktion. Hier einige Beispiele:

```
=UND(B6>B7; B9>B10)
=UND(A10>50;A10<100)
```

In diesen Fällen ist die Bedingung jeweils nur erfüllt, also wahr, wenn beide Teilbedingungen erfüllt sind.

```
=ODER(B6>B7; B9>B10)
```

Bei dieser Formel ist das Ergebnis WAHR, wenn die erste oder die zweite oder beide Bedingungen erfüllt sind.

```
=XODER(B6>B7; B9>B10)
```

Beim ausschließenden Oder ist dagegen das Ergebnis nur dann WAHR, wenn entweder die erste oder die zweite Bedingung erfüllt ist. Sind beide Bedingungen erfüllt, ist das Ergebnis also FALSCH.

Die Funktion NICHT() verneint einen Vergleich. Ein Beispiel:

```
=NICHT(A1=100)
```

Diese Bedingung ist in allen Fällen erfüllt, wo A1 nicht gleich 100. Die Formeln

```
=A1<>100
=NICHT(A1=100)
```

sind also austauschbar.

13.2 Prüfen mit der WENN()-Funktion

Logische Vergleiche werden insbesondere auch mit Hilfe der Funktion WENN() durchgeführt. Manchmal ist es dabei bequemer, eine komplexe Bedingung zunächst in einer Zelle zu prüfen und dann in einer WENN()-Funktion nur den Wahrheitswert dieser Zelle abzufragen, als die komplexe Bedingung in die WENN()-Funktion selbst mit aufzunehmen. Das ist insbesondere zu empfehlen, wenn diese Bedingung an mehreren Stellen abgefragt werden soll.

Abbildung 13.2 zeigt ein einfaches Beispiel. In Zelle D6 wird zunächst mit der logischen Formel

```
=B7>B8
```

geprüft, ob der Deckungsbeitrag die fixen Kosten übersteigt. Das Ergebnis ist ein Wahrheitswert. Dieser wird dann in der WENN()-Funktion in Zelle D7 abgefragt.

	A	B	C	D	E
1					
2	**Logische Formeln**				
3					
4	Absatz:	6.000,00		Ergebnis:	Formel:
5	Verkaufserlöse:	1.494.000,00		FALSCH	=B9<(B5*0,05)
6	variable Kosten:	300.000,00		WAHR	=B7>B8
7	Deckungsbeitrag:	1.194.000,00		Gewinn	=WENN(D6=WAHR;"Gewinn";"Verlust")
8	fixe Kosten:	1.100.000,00			
9	Betriebsgewinn:	94.000,00			

Abbildung 13.2 Beispiel für logische Formeln

Die WENN()-Funktion prüft jedes Mal zunächst, ob eine bestimmte Bedingung erfüllt, also wahr ist. Trifft dies zu, wird die WAHR-Anweisung ausgeführt und bestimmt das Ergebnis der Formel. Trifft dies nicht zu, wird zur FALSCH-Anweisung verzweigt und diese ausgeführt. Die Funktion hat deshalb drei Argumente, die in der vorgeschriebenen Reihenfolge eingetragen werden müssen, weil sonst unerwünschte Ergebnisse produziert werden.

1. Prüfung ist eine Bedingung in Form eines logischen Ausdrucks.
2. Eine Anweisung für den Fall, dass die Bedingung erfüllt ist, die als Dann_Wert bezeichnet wird.
3. Eine Anweisung für den Fall, dass die Bedingung nicht erfüllt ist, der Sonst_Wert.

Sowohl bei der Formulierung der Bedingungen als auch bei der Formulierung der WAHR- und FALSCH-Anweisungen können natürlich alle sinnvollen Kombinationen von Operatoren und Funktionen benutzt werden.

Die WENN()-Funktion kann generell für zwei ganz unterschiedliche Zwecke eingesetzt werden:

1. Die Funktion erlaubt die Durchführung von Prüfungen. Zum Beispiel kann mit dieser Funktion die Frage beantwortet werden, ob bestimmte Grenzwerte überschritten sind, ob Abweichungen von einer Norm vorliegen oder ob bestimmte Zielwerte erreicht worden sind.
2. Mit der WENN()-Funktion kann Excel veranlasst werden, unterschiedliche Operationen durchzuführen, je nachdem, ob eine Bedingung erfüllt ist. Das entspricht etwa einer Verzweigung, die in allen Computersprachen zu den Grundoperationen gehört.

Allerdings gilt die Verzweigung immer nur in Bezug auf das Ergebnis in der Zelle selbst. Es ist nicht möglich, mit der WENN()-Funktion direkt einen unterschiedlichen Wert in eine andere Zelle einzutragen. Es ist zwar möglich, etwa in die Zelle B3 zu schreiben:

```
=WENN(B1>1000;B2=100;B2=200)
```

Ist B1 tatsächlich größer 1000, bedeutet diese Formel aber lediglich, dass Excel nun prüft, ob die Zelle B2 tatsächlich den Wert 100 enthält. Ist das der Fall, ist das Ergebnis in Zelle B3 der Wert WAHR.

Eine direkte Wertzuweisung an eine Zelle, die dann eventuell auch Werte, die bisher in dieser Zelle eingetragen waren, überschreibt, kann nur über Makrobefehle geleistet werden. Das schließt natürlich nicht aus, dass eine andere Zelle durch einen Adressbezug den Wert aus der Zelle, in der die WENN()-Funktion steht, übernimmt. Wenn Sie in B3:

```
=WENN(B1>1000;100;200)
```

schreiben und in Zelle B2:

```
=B3
```

erhalten Sie das Ergebnis, das Sie mit der ersten Formel erreichen wollten. In den folgenden Abschnitten soll die Nützlichkeit der WENN()-Funktion noch an einigen typischen Beispielen verdeutlicht werden.

13.3 Bewertungen erzeugen

In der in Abbildung 13.3 gezeigten Tabelle sind Kundenumsätze einer Firma für zwei Jahre eingetragen. Die Kunden, deren realer Umsatz – es wird eine Inflationsrate von 3% unterstellt – gegenüber dem Vorjahr abgesunken ist, sollen in der Spalte D durch einen entsprechenden Hinweis gekennzeichnet werden. Die Formel kann in die Zelle D6 eingetragen und dann die Spalte hinunterkopiert werden. Sie könnte lauten:

```
=WENN(C6<(B6*1,03); "Umsatzrückgang";"")
```

Bei den Kunden, deren Umsatz inflationsbereinigt gesunken ist, erscheint in der Spalte D der Hinweis »Umsatzrückgang«. Bei allen anderen Kunden bleibt die Zelle in der Spalte D als Ergebnis der Formel leer.

	A	B	C	D
1				
2	**Prüfung von Bedingungen**			
3				
4		Umsatzentwicklung		Umsatzentwicklung inflationsbereinigt
5		2011	2012	
6	Kunde A	100000	770088	
7	Kunde B	120000	50000	Umsatzrückgang
8	Kunde C	87000	120000	
9	Kunde D	250000	250100	Umsatzrückgang
10		**=WENN(C6<(B6*1,03);"Umsatzrückgang";"")**		

Abbildung 13.3 Prüfung von Bedingungen

13.4 Bedingte Berechnungen

Während es im letzten Fall in erster Linie um die Prüfung der Entwicklung der Kundenumsätze ging, soll die nächste Formel zeigen, wie die `WENN()`-Funktion benutzt werden kann, um auf verschiedene Zustände unterschiedlich zu reagieren. Die Tabelle in Abbildung 13.4 berechnet für eine Firma einen Planwert für die Umsatzentwicklung in verschiedenen Regionen. Er soll entweder den Wert des aktuellen Jahres oder den des Vorjahres um 20% übersteigen, abhängig davon, welcher Wert höher ist. Die verwendete Formel ist:

```
=WENN(C5>B5;C5*1,2;B5*1,2)
```

	A	B	C	D	E
1					
2	**Bedingte Berechnungen**				
3					
4	Region	Umsatz 2011	Umsatz 2012	Planwert	
5	Süd	1.200.000,00 €	1.300.000,00 €	1.560.000,00 €	**=WENN(C5>B5;C5*1,2;B5*1,2)**
6	West	900.000,00 €	870.000,00 €	1.080.000,00 €	
7	Nord	600.000,00 €	440.000,00 €	720.000,00 €	
8	Ost	300.000,00 €	440.000,00 €	528.000,00 €	

Abbildung 13.4 Rechnen in Abhängigkeit von einer Prüfung

13.5 Bedingte Textanzeige

Besteht die Möglichkeit, in einem Arbeitsblatt Texte nur unter bestimmten Voraussetzungen zu drucken? Die `WENN()`-Funktion bietet sich auch in diesem Fall an. Angenommen, das Arbeitsblatt wird zur Aufbereitung und zum Druck von Rechnungen benutzt.

Auf den Dezember-Rechnungen soll im Rechnungsfuß ein Hinweis auf ein spezielles Weihnachtsangebot erscheinen. Die Lösung könnte folgende Formel sein:

```
=WENN(MONAT(JETZT())=12;"Sonderrabatt von 20% für alle Orientteppiche";"")
```

Die WENN()-Funktion kann selbstverständlich nicht nur zur Prüfung numerischer Zellen benutzt werden. Zellen, die Texte oder Zeichenfolgen enthalten, können ebenfalls befragt werden. Der Text kann direkt eingetragen werden oder indirekt über einen Zellbezug oder als Ergebnis einer Formel, die eine Zeichenfolge bereitstellt. Hier ein einfaches Beispiel:

```
=WENN(B10="Berlin";"Hauptstadt";"")
```

Allerdings kann im Fall von Zeichenfolgenvergleichen eine unangenehme Überraschung auftreten. Die Bedingung, dass in der Zelle B10 der Name »Berlin« steht, ist nur dann erfüllt, wenn dort tatsächlich auch nur »Berlin« steht. Ein Fehler, der bei der Arbeit mit Excel gelegentlich plagt, weil er so schwer erkennbar ist, besteht darin, überflüssige Leerzeichen zu benutzen.

Wenn also hinter dem Wort »Berlin« versehentlich noch einmal die Leertaste gedrückt worden ist, wird Excel sich so verhalten, als sei der Inhalt der Zelle B10 nicht »Berlin«. Im strengen Sinne stimmt das ja auch, aber zu sehen ist der Unterschied nur, wenn im Bearbeitungsfeld Ende gedrückt wird. Der Fehler kann allerdings durch die Funktion =GLÄTTEN() abgefangen werden. Die Formel

```
=WENN(GLÄTTEN(B10)="Berlin" ...
```

ist gegen die angesprochene Fehlermöglichkeit gefeit. Die Funktion GLÄTTEN() ist eine Textfunktion, die überflüssige Leerzeichen entfernt.

13.6 Prüfungen mit komplexen Bedingungen

Das Argument Prüfung kann auch aus mehreren Einzelbedingungen zusammengesetzt sein. In vielen Fällen ist eine Operation gleich von mehreren Bedingungen abhängig. Eine Komponente eines Produkts soll nur dann bei einem bestimmten Lieferanten bestellt werden, wenn der Preis akzeptabel ist und gleichzeitig die Lieferfrist maximal einen Monat beträgt. Die Formel könnte heißen:

```
=WENN(UND(C12<12500;D12<4);"bestellen";"nicht bestellen")
```

Die Zelle C12 enthält den Preis, die Zelle D12 die Lieferzeit in Wochen. Die Bedingung ist nur dann erfüllt, wenn beide Teile der Bedingung gleichzeitig erfüllt sind.

In anderen Fällen hängt eine Entscheidung davon ab, ob eine bestimmte Bedingung erfüllt ist oder eine andere. Im Beispiel wäre denkbar, dass auch ein höherer Preis akzeptiert wird, wenn die Lieferfrist kurz ist. Eine kleine Änderung der Formel trägt der Situation Rechnung. Statt der Funktion UND() wird ODER() eingesetzt:

```
=WENN(ODER(C12<1250O;D12<4);"bestellen";"nicht bestellen")
```

Die neue Bedingung ist in drei Fällen wahr:

1. Der erste Teil der Bedingung ist erfüllt.

2. Der zweite Teil der Bedingung ist erfüllt.

3. Beide Bedingungen sind erfüllt.

Die Funktion ODER() ist also kein ausschließendes Oder. Das ausschließende Oder würde den dritten Fall nicht zulassen. Ein solches Entweder-oder, aber nicht beides gleichzeitig, konnte in einer älteren Excel-Version erreicht werden durch eine Kombination von UND() und ODER():

```
=WENN(UND(ODER(B1O=5;C1O=7;NICHT(UND(B1O=5;C1O=7) ) );"ok";"prüfen")
```

Wenn die Zelle B10 den Wert 5 hat und die Zelle C10 den Wert 7, ist das Ergebnis prüfen, die Bedingung ist also nicht erfüllt.

In Excel 2013 wird die Sache einfacher durch die neue Funktion XODER(). Die obige Formel lautet dann:

```
=WENN(XODER(B1O=5;C1O=7);"ok";"prüfen")
```

13.7 Mehrfachverzweigungen

Sowohl die WAHR-Anweisung als auch die FALSCH-Anweisung können selbst wieder eine WENN()-Funktion enthalten, so dass eine Mehrfachverzweigung erreicht werden kann.

Die Weihnachtsgeschenke einer Firma sind nach Wert in zwei Gruppen eingeteilt. Geschenke der Gruppe A sollen Kunden erhalten, deren Umsatz über 500.000 EUR liegt, Gruppe B verlangt mindestens 10.000 EUR, um Minikunden ganz herauszunehmen. Zuerst werden in der Formel die Minikunden abgefangen, dann wird zwischen Gruppe A und Gruppe B unterschieden.

1. Schritt

```
WENN(B5>=10000;"Geschenk";"kein Präsent")
```

Damit sind die Minikunden herausgenommen.

2. Schritt

```
=WENN(B5>10000;WENN(B5>=50000;"Geschenk A";"Geschenk B");"kein Präsent")
```

Die WAHR-Anweisung aus dem ersten Schritt ist jetzt ersetzt durch eine komplette WENN-Funktion, die selbst wiederum eine Bedingung, eine WAHR-Anweisung und eine FALSCH-Anweisung enthält.

	A	B	C
1			
2	**Verschachtelte Bedingungen**		
3			
4	Kunden	Umsatz	Geschenkkategorie
5	Berger	600.000,00 €	Geschenk A
6	Wehner	430.000,00 €	Geschenk B
7	Brech	30.000,00 €	Geschenk B
8	Schub	3.290,00 €	kein Präsent
9	Scheimer	24.000,00 €	Geschenk B
10	Vosken	1.200,00 €	kein Präsent
11	Erber	530.000,00 €	Geschenk A
12	Kosinsky	23.000,00 €	Geschenk B
13	Marchat	12.000,00 €	Geschenk B
14			
15	Formel:	=WENN(B5>=10000;WENN(B5>=500000; "Geschenk A";"Geschenk B");"kein Präsent")	

Abbildung 13.5 Verschachtelte Bedingungen

13.8 Bedingte Formate

Ein weiterer Anwendungsbereich für logische Formeln und Funktionen sind die bedingten Formate. Hier ein praktisches Beispiel: Wenn Sie in einer Arbeitsmappe einen Kalender oder einen Zeitplan anlegen, lässt sich mit einem bedingten Format das Wochenende leicht mit einem anderen Hintergrund hervorheben.

```
=ODER(WOCHENTAG(B4)=7;WOCHENTAG(B4)=1)
```

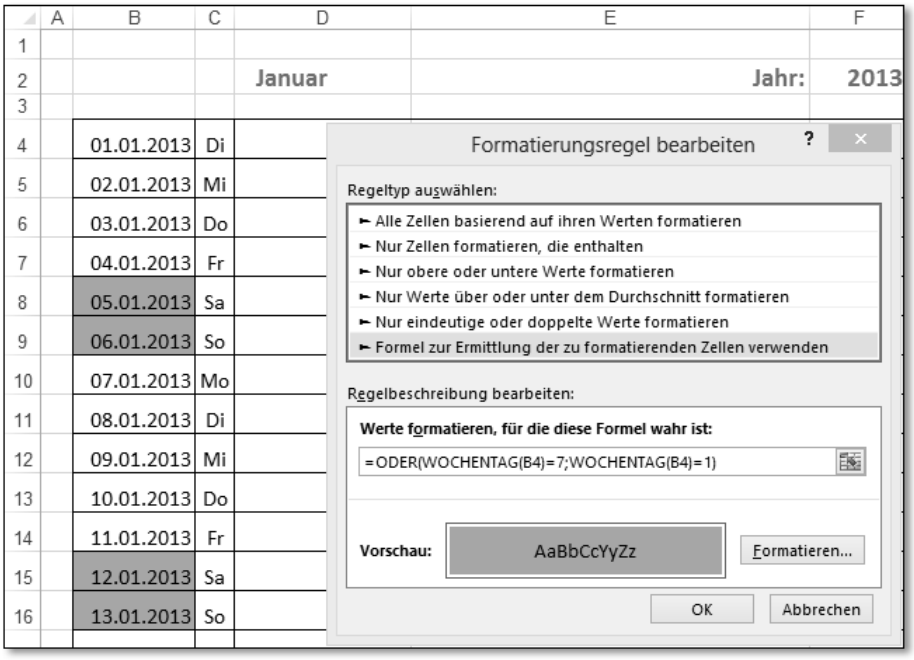


Abbildung 13.6 Kalender mit bedingtem Format für das Wochenende

13.9 Referenz der logischen Funktionen

FALSCH()

FALSE()

Syntax: `FALSCH()`
Beispiel: `=FALSCH()`
Ergebnis: `FALSCH`

Die Funktion `FALSCH()` legt den Wahrheitswert `FALSCH` in der Zelle ab oder übergibt ihn an eine andere Funktion. Stattdessen kann auch direkt der Wert `FALSCH` in der Zelle abgelegt werden. Excel vergibt bei der Eingabe von »falsch« oder »FALSCH« für die Zelle automatisch den Datentyp `boolean` und zentriert die in Großbuchstaben ausgegebene Anzeige. (Um das Wort »Falsch« nur als Text in einer Zelle abzulegen, müssen Sie die Zelle als Text formatieren oder einen Apostroph davorsetzen.)

Die Funktion FALSCH() kann in einer WENN()-Funktion benutzt werden, um anzuzeigen, ob die Bedingung, die diese Funktion prüft, erfüllt ist. Das kann sinnvoll sein, wenn bei einer komplexen Bedingung nicht gleich ersichtlich ist, ob die Bedingung erfüllt ist oder nicht. Hier ein Beispiel:

```
=WENN(Gewinne01>500000; WAHR(); FALSCH())
```

Das Ergebnis ist FALSCH, wenn Gewinne01 beispielsweise den Wert 300000 hat. Die Formel könnte auch lauten:

```
=WENN(Gewinne01>500000;WAHR;FALSCH)
```

Die beiden Wahrheitswerte müssen dabei nicht in Anführungszeichen gesetzt werden.

NICHT()

NOT()

Syntax: NICHT(**Wahrheitswert**)
Beispiel: =NICHT(13>14)
Ergebnis: WAHR

Durch die Funktion NICHT() wird der Wert von Wahrheitswert umgekehrt. Ergibt der Ausdruck oder der Bezug des Arguments den Wahrheitswert WAHR, ist also das Ergebnis der Funktion der Wahrheitswert FALSCH.

Wahrheitswert muss ein Ausdruck oder ein Bezug auf einen Ausdruck sein, der nur den Wert WAHR oder FALSCH annehmen kann. Ist das nicht der Fall, etwa weil der Inhalt einer mit einem Bezug angegebenen Zelle eine Zeichenfolge ist, liefert die Funktion den Fehler #WERT!. Bei einem Bezug auf eine leere Zelle liefert die Funktion dagegen den Wert WAHR, beim Bezug auf eine Zelle mit einem numerischen Inhalt den Wert FALSCH.

ODER()

OR()

Syntax: ODER(**Wahrheitswert1**; Wahrheitswert2; ...)
Beispiel: =ODER(13>14;13<14)
Ergebnis: WAHR

Die Funktion ODER() vergleicht seit Excel 2007 bis zu 255 Argumente miteinander, in den älteren Versionen bis zu 30. Sind alle Argumente FALSCH, so liefert die Funktion den Wahrheitswert FALSCH, andernfalls WAHR. Das gilt auch, wenn alle Wahrheitswerte WAHR sind. Die Funktion stellt also kein ausschließendes Oder im Sinne eines Entweder-oder dar. Dafür können Sie in Excel 2013 die neue Funktion XODER() verwenden.

Mindestens eines der angegebenen Argumente muss ein Ausdruck oder ein Bezug auf einen Ausdruck sein, der nur den Wert WAHR oder FALSCH annehmen kann. Ist das nicht der Fall, etwa bei Bezügen auf Zellen mit Zeichenfolgen oder bei leeren Zellen, liefert die Funktion den Fehler #WERT!. Bei einem numerischen Wert oder einem Bezug auf eine Zelle mit einem numerischen Wert liefert die Funktion dagegen den Wert WAHR.

```
=ODER(5;6)
```

ergibt also WAHR.

13.9.1 Vorkommen prüfen

Da es bei der ODER()-Funktion genügt, wenn ein Argument wahr ist, um das Gesamtergebnis WAHR zu erzeugen, kann diese Funktion auch verwendet werden, um zu prüfen, ob ein bestimmtes Element in einer Liste vorkommt. Abbildung 13.7 zeigt ein Beispiel, in dem geprüft werden soll, ob ein Artikel mit einer bestimmten Artikelnummer vorkommt. Sie können eine solche Frage natürlich auch mit Hilfe der Filterfunktionen prüfen, die in Excel für Tabellen zur Verfügung stehen, aber es geht auch mit der Funktion ODER().

Dazu ist es sinnvoll, den Bereich, in dem das Vorkommen geprüft werden soll, vorher zu benennen. Dann kann mit einer einfachen Matrixformel ein Vergleich zwischen dem Inhalt der Zelle F2 und allen Einträgen in dem benannten Bereich vorgenommen werden. Die Formel ist:

```
{=ODER(F2=artikelliste)}
```

Wichtig ist, sie mit der Tastenkombination Strg + ⇧ + ↵ zu beenden, damit sie als Matrixformel arbeiten kann.

Um die Tabelle noch etwas verständlicher zu machen, können Sie die Funktion noch in eine WENN()-Funktion einfügen:

```
{=WENN(ODER(F2=artikelliste);"Artikel vorhanden.";"Artikel fehlt!")}
```

	A	B	C	D	E	F	G
1							
2	**Vorkommen prüfen**				Artnr	8443	**Artikel vorhanden**
3							
4	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Farbe**	**Absatz**	**Umsatz**
5	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1232	26181
6	7778	Jalousie Ccs	Jalousie	Metall	rot	1463	63360
7	7578	Jalousie Ccw	Jalousie	Metall	rot	1353	62810
8	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1243	25631
9	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	770	12650
10	8443	Jalousie V	Jalousie	Metall	silbergrau	660	44000
11	8243	Jalousie VV	Jalousie	Metall	silbergrau	253	43450

Abbildung 13.7 Prüfen, ob ein Artikel in der Liste vorkommt

UND()

AND()

Syntax: `UND(`**`Wahrheitswert1`**`; Wahrheitswert2; ...)`

Beispiel: `=UND(13>14;14<13)`

Ergebnis: `FALSCH`

Die Funktion `UND()` vergleicht seit Excel 2007 bis zu 255 Argumente miteinander, in den älteren Versionen bis zu 30. Nur wenn alle Argumente `WAHR` ergeben, wird als Ergebnis `WAHR` zurückgegeben, andernfalls ist das Ergebnis `FALSCH`.

Mindestens eines der angegebenen Argumente muss ein Ausdruck oder ein Bezug auf einen Ausdruck sein, der nur den Wert `WAHR` oder `FALSCH` annehmen kann. Ist das nicht der Fall, etwa bei Bezügen auf Zellen mit Zeichenfolgen oder bei leeren Zellen, liefert die Funktion den Fehler `#WERT!`. Bei einem numerischen Wert oder einem Bezug auf eine Zelle mit einem numerischen Wert liefert die Funktion dagegen den Wert `WAHR`.

```
=UND(5;6)
```

ergibt also `WAHR`.

13.9.2 Mehrfachprüfung über einen Bereich

Wenn Sie in einer bestimmten Situation zahlreiche Bedingungen gleichzeitig prüfen wollen, kann es umständlich werden, die einzelnen Ausdrücke alle nacheinander in eine `UND()`-Funktion einzubauen. Hier hilft eine Formel, die die Liste der Bedingungen

aus einem angegebenen Bereich übernimmt. In dem folgenden Beispiel sind neben den Datenspalten einer Tabelle drei Spalten mit Kriterien angelegt, die mit strukturierten Verweisen arbeiten. Die Kriterien in den Zellen H5 bis J5 heißen:

```
=[@Warengruppe]=$G$2
=[@Material]=$I$2
=[@Farbe]=$K$2
```

Der aktuelle Wert in der Zeile wird also jedes Mal mit den in der zweiten Zeile eingetragenen Vergleichswerten verglichen. In der Spalte K kann dann mit der Formel

```
=UND(Artikeltabelle[@[Kriterium1]:[Kriterium3]])
```

gearbeitet werden, um die Artikel zu kennzeichnen, die alle drei Kriterien erfüllen. Sie brauchen dann nur noch die Filterschaltfläche für die Spalte »Ergebnis« zu benutzen, um alle Artikel, die die Kriterien nicht alle erfüllen, auszublenden.

	A	B	C	D	E	F	G	H	I	J	K
1											
2	**Mehrfach prüfen**					Warengruppe:	Jalousie	Material:	Kunststoff	Farbe:	grau
3											
4	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Farbe**	**Absatz**	**Umsatz**	**Kriterium1**	**Kriterium2**	**Kriterium3**	**Ergebnis**
5	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1232	26181	WAHR	WAHR	WAHR	WAHR
6	7778	Jalousie Ccs	Jalousie	Metall	rot	1463	63360	WAHR	FALSCH	FALSCH	FALSCH
7	7578	Jalousie Ccw	Jalousie	Metall	rot	1353	62810	WAHR	FALSCH	FALSCH	FALSCH
8	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1243	25631	WAHR	WAHR	WAHR	WAHR
9	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	770	12650	WAHR	FALSCH	FALSCH	FALSCH
10	8443	Jalousie V	Jalousie	Metall	silbergrau	660	44000	WAHR	FALSCH	FALSCH	FALSCH
11	8243	Jalousie VV	Jalousie	Metall	silbergrau	253	43450	WAHR	FALSCH	FALSCH	FALSCH

Abbildung 13.8 Markierung mit mehrfachen Kriterien

WAHR()

TRUE()

Syntax: `WAHR()`
Beispiel: `=WAHR()`
Ergebnis: `WAHR`

Die Funktion `WAHR()` legt den Wahrheitswert `WAHR` in der Zelle ab oder übergibt ihn an eine andere Funktion. Stattdessen kann auch direkt der Wert `WAHR` in der Zelle abgelegt werden. Excel vergibt bei der Eingabe von »wahr« oder »WAHR« für die Zelle automatisch den Datentyp `boolean` und zentriert die in Großbuchstaben ausgegebene Anzeige.

(Um das Wort »Wahr« nur als Text in einer Zelle abzulegen, müssen Sie die Zelle als Text formatieren oder einen Apostroph davorsetzen.)

Die Funktion WAHR() kann in einer WENN()-Funktion benutzt werden, um anzuzeigen, ob die Bedingung, die diese Funktion prüft, erfüllt ist. Das kann sinnvoll sein, wenn bei einer komplexen Bedingung nicht gleich ersichtlich ist, ob die Bedingung erfüllt ist oder nicht. Hier ein Beispiel:

```
=WENN(Gewinne01>500000; WAHR(); FALSCH())
```

Das Ergebnis ist WAHR, wenn Gewinne01 beispielsweise den Wert 600000 hat.

Die Formel könnte auch lauten:

```
=WENN(Gewinne01>500000;WAHR;FALSCH)
```

Die beiden Wahrheitswerte müssen dabei nicht in Anführungszeichen gesetzt werden.

WENN()

IF()

Syntax:	WENN(**Prüfung; Dann_Wert**; Sonst_Wert)
Beispiel:	=WENN(A2>B2;A2-B2;B2-A2)
Ergebnis:	Bewirkt, dass jeweils der kleinere Wert vom größeren abgezogen wird.

Die Funktion WENN() liefert in Abhängigkeit von einer Bedingung je nach Wahrheitsprüfung unterschiedliche Resultate. Die WENN()-Funktion prüft zunächst, ob die mit dem Argument Prüfung formulierte Bedingung erfüllt, also WAHR ist. Trifft dies zu, gibt die Funktion den Dann_Wert aus, trifft es nicht zu, wird der Sonst_Wert ausgegeben. Ist dieser nicht angegeben, wird der logische Wert ausgegeben, den die Prüfung liefert.

Die Funktion erwartet für das Argument Prüfung einen Ausdruck, der einen logischen Wert liefert. Ist Prüfung eine Zahl oder der Verweis auf eine Zahl, wird dies als WAHR gewertet. Ist Prüfung eine Zeichenfolge, wird sie als FALSCH gewertet. Ist Prüfung ein ungültiger Name, erscheint der Fehler #NAME?.

Die Funktion erlaubt die Durchführung von Prüfungen, z. B. kann mit dieser Funktion die Frage beantwortet werden, ob bestimmte Grenzwerte überschritten oder bestimmte Zielwerte erreicht worden sind.

Es kann auch veranlasst werden, unterschiedliche Werte in eine Zelle einzutragen oder unterschiedliche Berechnungen durchzuführen, je nachdem, ob eine Bedingung erfüllt ist oder nicht.

In dem folgenden Beispiel wird mit Hilfe der WENN()-Funktion eine Bewertung des Vergleichs von zwei Jahresergebnissen erzeugt:

	A	B	C	D	E
1					
2	**Bedingte Texte**				
3					
4	Region	Umsatz 2006	Umsatz 2007	Bewertung	
5	Süd	1.200.000,00 €	1.300.000,00 €	Wachstum	**=WENN(C13>B13;"Wachstum";"Rückschritt")**
6	West	900.000,00 €	870.000,00 €	Rückschritt	
7	Nord	600.000,00 €	440.000,00 €	Rückschritt	
8	Ost	300.000,00 €	440.000,00 €	Wachstum	

Abbildung 13.9 Bewertung mit Hilfe bedingter Texte

Als Ergebnis der Wahrheitsprüfung kann eine Verschachtelung durch eine erneute Verwendung der WENN()-Funktion erfolgen. Beispiele finden Sie in der Einführung des Kapitels. Seit Excel 2007 sind bis zu 64 Schachtelungen erlaubt, in den älteren Versionen nur 7.

WENNFEHLER()

IFERROR()

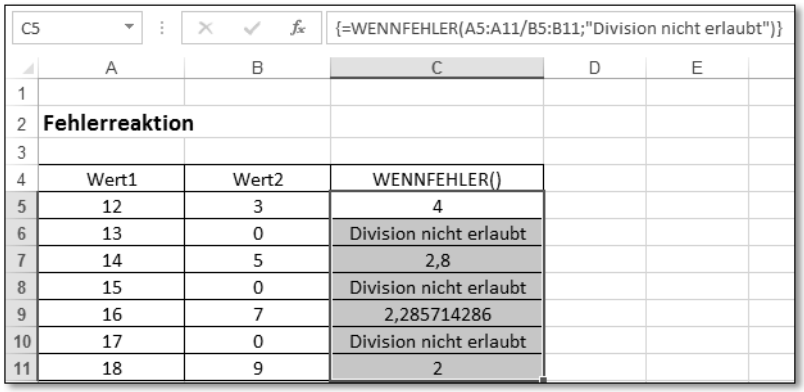
Syntax: WENNFEHLER(**Wert; Wert_falls_Fehler**)

Beispiel: =WENNFEHLER(A2/B2;"unerlaubte Division")

Ergebnis: Unerlaubte Division, wenn B2 den Wert 0 hat.

Die Funktion WENNFEHLER() wird seit Excel 2007 angeboten. Sie erlaubt eine Reaktion des Anwenders auf den Fall, dass ein mit dem Argument Wert angegebener Ausdruck oder Bezug statt eines ordentlichen Ergebnisses einen Fehlerwert liefert. Liefert Wert keinen Fehler, wird das Ergebnis von Wert in der Zelle angezeigt, anderenfalls wird der Wert ausgewertet, der mit dem Argument Wert_falls_Fehler angegeben wird. Das kann ein Fehlerhinweis oder auch ein Ausdruck mit einer alternativen Berechnung sein oder auch eine leere Zeichenfolge.

Die Funktion kann auch für eine Matrixformel verwendet werden, um, wie in dem folgenden Beispiel etwa, eine Spalte auf Fehlerwerte zu prüfen.



C5 {=WENNFEHLER(A5:A11/B5:B11;"Division nicht erlaubt")}

	A	B	C	D	E
1					
2	**Fehlerreaktion**				
3					
4	Wert1	Wert2	WENNFEHLER()		
5	12	3	4		
6	13	0	Division nicht erlaubt		
7	14	5	2,8		
8	15	0	Division nicht erlaubt		
9	16	7	2,285714286		
10	17	0	Division nicht erlaubt		
11	18	9	2		

Abbildung 13.10 Benutzerdefinierte Fehleranzeige

WENNNV()

IFNA()

Syntax: `WENNNV(Wert; Wert_bei_NV)`

Beispiel: `=WENNNV(SVERWEIS("Atlaa";A6:B46;2;"Wert fehlt!")`

Ergebnis: `"Wert fehlt!"`, wenn die Zeichenfolge `"Atlaa"` nicht mit der `SVERWEIS()`-Funktion gefunden wird.

Die Funktion `WENNNV()` gibt den ersten Wert zurück, wenn der Ausdruck nicht den Fehlerwert `#NV` liefert, anderenfalls den zweiten Wert.

XODER()

XOR()

Syntax: `XODER(Wahrheitswert1; Wahrheitswert2; ...)`

Beispiel: `=XODER(5>0;2>9)`

Ergebnis: `WAHR`, weil nur der erste Wahrheitswert `WAHR` ergibt.

Diese in Excel lang vermisste Funktion gibt ein ausschließliches Oder aller Argumente zurück. Die Funktion `XODER()` ergibt den Wert `WAHR`, wenn nur einer der Wahrheitswerte `WAHR` ergibt, in allen anderen Fällen ist das Ergebnis `FALSCH`.

Mindestens eines der angegebenen Argumente muss ein Ausdruck oder ein Bezug auf einen Ausdruck sein, der nur den Wert `WAHR` oder `FALSCH` annehmen kann. Ist das nicht

der Fall, etwa bei Bezügen auf Zellen mit Zeichenfolgen oder bei leeren Zellen, liefert die Funktion den Fehler #WERT!. Bei einem numerischen Wert oder einem Bezug auf eine Zelle mit einem numerischen Wert liefert die Funktion den Wert FALSCH.

```
=XODER(5;6)
```

ergibt also FALSCH.

14 Informationsfunktionen

Funktion	Seite	Funktion	Seite
BLATT()	801	ISTLEER()	808
BLÄTTER()	802	ISTLOG()	809
FEHLER.TYP()	803	ISTNV()	810
INFO()	804	ISTTEXT()	810
ISTBEZUG()	806	ISTUNGERADE()	811
ISTFEHL()	806	ISTZAHL()	812
ISTFEHLER()	806	N()	813
ISTFORMEL()	807	NV()	814
ISTGERADE()	807	TYP()	815
ISTKTEXT()	808	ZELLE()	817

Mit Hilfe der Informationsfunktionen lassen sich bestimmte Daten über Elemente eines Tabellenblatts abfragen, etwa der Typ eines Fehlers, der Zustand einer Zelle oder der Datentyp eines Zellinhalts. Außerdem lassen sich bestimmte Daten über die Umgebung der Anwendung erfahren, beispielsweise die Nummer eines Blatts, die Anzahl der Arbeitsblätter oder der aktuell eingestellte Rechenmodus.

14.1 Einsatzbereiche für Informationsfunktionen

Die meisten der Informationsfunktionen werden nur gelegentlich innerhalb von Tabellen, sonst eher in Makros benötigt. Ein Hauptanwendungsgebiet der Funktionen ist, in Verbindung mit der Funktion `WENN()` Ergebnisse von Berechnungen von den Inhalten bestimmter Zellen abhängig zu machen.

14.1.1 Beispiel für bedingte Berechnungen

Nehmen Sie das folgende einfache Beispiel: Sie haben in einer Tabelle in den Spalten A und B ab Zeile 3 sowohl Zahlen als auch Texteinträge. Sie wollen in C das Produkt von A und B ausgeben, falls in der Zeile entsprechende Werte vorhanden sind. Wenn Sie nun

in C3 die Formel `=A3*B3` eintragen und nach unten kopieren, erhalten Sie immer dann, wenn in Spalte A und B keine Zahlen stehen, die Fehlermeldung `#WERT!`, weil ja mit einem Text nicht multipliziert werden kann.

Um dies zu vermeiden, können Sie in C3 eintragen:

```
=WENN(UND(ISTZAHL(A3);ISTZAHL(B3));A3*B3;"")
```

Dann wird das Produkt nur gebildet, wenn beide Zellen eine Zahl enthalten, sonst wird eine leere Zeichenfolge ausgegeben. Wenn Sie diese Formel nach unten kopieren, dann bleiben Ihnen die unangenehmen Fehlermeldungen erspart. (Die vorgegebenen Fehlermeldungen sind in Abschnitt 1.9, »Hinweise zu den Grundrechenarten«, beschrieben.)

	A	B	C	D
1				
2	**Bedingte Berechnung**			
3				
4	**Wert**	**Multiplikator**	**Produkt**	
5	1000	3,5	3500	**=A5*B5**
6	unbekannt	100	#WERT!	
7	2000	2,8	5600	
8				
9	**Wert**	**Multiplikator**	**Produkt**	
10	1000	3,5	3500	**=WENN(UND(ISTZAHL(A10);ISTZAHL(B10));A10*B10;"")**
11	unbekannt	100		
12	2000	2,8	5600	
13				

Abbildung 14.1 Beispiel für bedingte Berechnungen

14.1.2 Funktionen zur Prüfung des Datentyps

Ein Dutzend der folgenden Funktionen beginnen mit dem Präfix `IST` und dienen dem Zweck, zu prüfen, ob es sich bei dem angegebenen Argument um einen bestimmten Datentyp handelt. Das erlaubt insbesondere in der Kombination mit der `WENN()`-Funktion, bedingte Verfahren einzurichten, also etwa je nach dem vorliegenden Datentyp unterschiedliche Berechnungsschritte vorzunehmen. Sehr nützlich sind diese Funktionen aber auch im Zusammenhang mit der bedingten Formatierung oder als Teil von Gültigkeitsregeln. Neu hinzugekommen in Excel 2013 ist dabei die Funktion `ISTFORMEL()`, mit der Sie prüfen, ob eine Zelle tatsächlich eine Formel enthält oder nicht.

Während die Funktion `TYP()` ganz allgemein nach dem Datentyp des angegebenen Arguments fragt, werden die IST-Funktionen eingesetzt, um gezielt zu prüfen, ob ein bestimmter Datentyp vorliegt. Das Argument `Wert` kann dabei ein Bereichsname, ein Zell-

bezug, eine Zahl, ein Text, ein logischer Wert oder ein Fehlerwert sein. Das Ergebnis der Funktionen ist jeweils ein logischer Wert, also `WAHR` oder `FALSCH`. Anders als bei den anderen Funktionen interpretieren die IST-Funktionen Zahlen, die in Anführungszeichen gesetzt sind, nicht als Zahlen, sondern als Zeichenfolgen.

Abbildung 14.2 gibt eine Übersicht über die IST-Funktionen:

	A	B	C
1			
2	**IST-Funktionen**		
3			
4	**Wert**	**Funktionsergebnis**	**Funktion**
5	Daten1	WAHR	=ISTBEZUG(Daten1)
6	#DIV/0!	WAHR	=ISTFEHL(A6)
7	#NV	WAHR	=ISTFEHLER(A7)
8	12	WAHR	=ISTFORMEL(A8)
9	3	FALSCH	=ISTGERADE(A9)
10	Text	FALSCH	=ISTKTEXT(A10)
11		WAHR	=ISTLEER(A11)
12	FALSCH	WAHR	=ISTLOG(A12)
13	#NV	WAHR	=ISTNV(A13)
14	123	FALSCH	=ISTTEXT(A14)
15	5	WAHR	=ISTUNGERADE(A15)
16	333	FALSCH	=ISTZAHL(A16)

Abbildung 14.2 Übersicht über die IST-Funktionen

14.2 Referenz der Informationsfunktionen

BLATT()

SHEET()

Syntax: `BLATT(Wert)`
Beispiel: `=BLATT("Tabelle5")`
Ergebnis: `5`

Die Funktion `BLATT()` liefert die Nummer des mit `Wert` angegebenen Blatts oder, wenn `Wert` nicht angegeben ist, die Nummer des aktuellen Blatts. `Wert` kann auch ein Zellbezug in dem betreffenden Blatt sein.

Ist `Wert` kein gültiger Wert, liefert die Funktion den Fehler `#BEZUG!`. Ist `Wert` kein gültiger Blattname, liefert die Funktion den Fehler `#NV`.

Zu beachten ist, dass Excel dabei alle Blatttypen fortlaufend zählt, also auch Diagrammblätter etc. Werden Blätter während der Arbeitssitzung verschoben, werden die Blattnummern sofort angepasst, solange die automatische Neuberechnung aktiviert ist.

	A	B	C
1			
2	**Blatt und Blätter**		
3			
4		BLATT()	
5	Aktuelles Blatt	3	**=BLATT()**
6	Blatt in der aktuellen Arbeitsmappe	9	**=BLATT("INFO")**
7	Blatt eines Zellbezugs in der aktuellen Arbeitsmappe	2	**=BLATT(BedingteBerechnung!A1)**
8			
9		BLÄTTER()	
10	Anzahl der Blätter der Arbeitsmappe	14	**=BLÄTTER()**
11	Anzahl der Blätter eines 3D-Bereichs	2	**=BLÄTTER(BedingteBerechnung:BLATT!A1)**
12	Anzahl der Blätter eines benannten 3D-Bereichs	11	**=BLÄTTER(Kopfzelle)**
13			

Abbildung 14.3 Beispiele für die Funktionen `BLATT()` und `BLÄTTER()`

Wenn Sie die Blätter einer großen Arbeitsmappe durchnummerieren wollen, können Sie im Gruppenmodus, also nach Auswahl aller Blätter, beispielsweise in der ersten Zelle die Formel ablegen:

```
="Blatt " & BLATT()
```

BLÄTTER()

SHEETS()

Syntax: `BLÄTTER(Bezug)`
Beispiel: `=BLÄTTER(Tabelle1:Tabelle5!A1)`
Ergebnis: `5`

Die Funktion `BLÄTTER()` liefert die Anzahl der Blätter eines Bezugs. Wird `Bezug` nicht angegeben, gibt die Funktion die Anzahl der Blätter in der Arbeitsmappe aus. Als `Bezug` kann ein 3D-Bezug angegeben werden oder der Name eines benannten 3D-Bezugs.

Ist `Bezug` kein gültiger Bezug, liefert die Funktion den Fehler `#BEZUG!`.

Zu beachten ist, dass Excel immer alle Blatttypen fortlaufend zählt, also auch Diagrammblätter etc. Siehe auch Abbildung 14.3.

Wenn Sie die Blätter einer großen Arbeitsmappe durchnummerieren wollen, können Sie im Gruppenmodus, also nach Auswahl aller Blätter, beispielsweise in der ersten Zelle die Formel ablegen:

```
="Blatt " & BLATT() & " von " & BLÄTTER()
```

FEHLER.TYP()

ERROR.TYPE()

Syntax: `FEHLER.TYP(`**`Fehlerwert`**`)`
Beispiel: `=FEHLER.TYP(falscherName)`
Ergebnis: `5`, wenn der Name nicht definiert ist

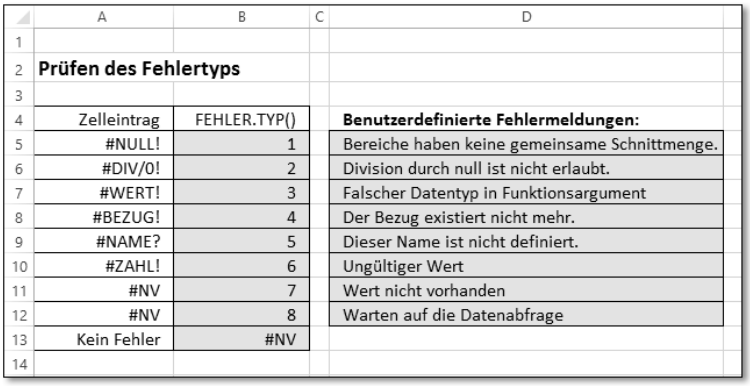
Die Funktion `FEHLER.TYP()` liefert entweder eine Zahl, die einen bestimmten Fehlerwert kennzeichnet, oder den Fehlerwert `#NV`, falls keiner der in Tabelle 14.1 aufgeführten Fehlerwerte vorliegt. Das Argument `Fehlerwert` ist ein Bezug auf eine Zelle, ein Ausdruck, der einen Fehlerwert liefert, oder ein direkt eingegebener Fehlerwert.

Fehlerwert	Rückgabewert
#NULL!	1
#DIV/0!	2
#WERT!	3
#BEZUG!	4
#NAME?	5
#ZAHL!	6
#NV	7
#GETTING_DATA	8
Sonstiges	#NV

Tabelle 14.1 Rückgabewerte der Fehler in Excel

Hauptsächlich wird diese Funktion verwendet, um anstelle einer Standardfehlermeldung eine benutzerdefinierte Erläuterung eines Fehlers anzuzeigen:

```
=WENN(FEHLER.TYP(A6)=2;"Division durch Null ist nicht erlaubt";)
```



	A	B	C	D
1				
2	**Prüfen des Fehlertyps**			
3				
4	Zelleintrag	FEHLER.TYP()		**Benutzerdefinierte Fehlermeldungen:**
5	#NULL!	1		Bereiche haben keine gemeinsame Schnittmenge.
6	#DIV/0!	2		Division durch null ist nicht erlaubt.
7	#WERT!	3		Falscher Datentyp in Funktionsargument
8	#BEZUG!	4		Der Bezug existiert nicht mehr.
9	#NAME?	5		Dieser Name ist nicht definiert.
10	#ZAHL!	6		Ungültiger Wert
11	#NV	7		Wert nicht vorhanden
12	#NV	8		Warten auf die Datenabfrage
13	Kein Fehler	#NV		
14				

Abbildung 14.4 Die Ermittlung der Fehlerkennung mit `FEHLER.TYP()` erlaubt benutzerdefinierte Fehlermeldungen.

Wenn Sie die Sache perfektionieren wollen, können Sie auch mit einer `WENN()`-Funktion arbeiten, die für jede Fehlernummer die entsprechende Meldung ausgibt. Dabei müssen Sie nur noch die Funktion `WAHL()` heranziehen:

```
=WENN(FEHLER.TYP(A6)<=8; WAHL(FEHLER.TYP(A6);
    "Bereiche haben keine gemeinsame Schnittmenge.";
    "Division durch null ist nicht erlaubt." ... ))
```

INFO()

INFO()

Syntax: `INFO(`**Typ**`)`

Beispiel: `=INFO("System")`

Ergebnis: `pcdos`

Die Funktion `INFO()` liefert einige Informationen über die aktuelle Betriebssystemumgebung, wobei mit dem Argument `Typ` angegeben wird, welche Daten nachgefragt werden. Tabelle 14.2 listet die möglichen Werte für das Argument `Typ` auf:

Typ	Rückgabewert
Verzeichnis	Pfad des aktuellen Verzeichnisses oder Ordners
Dateienzahl	Anzahl aktiver Arbeitsblätter in den geöffneten Arbeitsmappen. Ausgeblendete Mappen werden mitgezählt.

Tabelle 14.2 Werte für das Argument `Typ` und ihre Bedeutung

Typ	Rückgabewert
Ursprung	Gibt den absoluten Zellbezug der sichtbaren obersten linken Zelle im aktuellen Fensterbereich zurück. Dabei wird dem Text `"$A:"` vorangestellt. Dieser Wert dient der Kompatibilität mit Lotus 1-2-3, Version 3.x. Der tatsächlich zurückgegebene Wert hängt von der aktuellen Einstellung für die Bezugsart ab. Für D6 lautet der zurückgegebene Wert wie folgt: `A1-Bezugsart "$A:$D$6"`. `Z1S1-Bezugsart "$A:Z6S4"`.
Berechne	Liefert den aktuellen Berechnungsmodus (automatisch oder manuell).
Sysversion	Name der Version des aktuellen Betriebssystems
Version	Name der Version von Microsoft Excel
System	Name des Betriebssystems: Macintosh = `"mac"`, für Windows wird seltsamerweise immer noch = `"pcdos"` ausgegeben.

Tabelle 14.2 Werte für das Argument `Typ` und ihre Bedeutung (Forts.)

Einige Typwerte, die in früheren Versionen von Excel noch unterstützt wurden, werden seit Excel 2007 nicht mehr unterstützt. Das gilt für `BenutztSpeich`, `VerfSpeich`, `Gesamtspeich` und `Rechenmodus`. Die erlaubten Werte werden bei der direkten Eingabe der Funktion zur Auswahl angeboten, sobald die öffnende Klammer gesetzt ist.

Ein ungültiger Wert für `Typ` führt zu dem Fehler `#WERT!`, nicht mehr unterstützte Werte für `Typ` zu dem Fehler `#NV`.

	A	B
1		
2	**Infos zur Systemumgebung**	
3		
4	Zelleintrag	INFO()
5	=INFO("Verzeichnis")	C:\Users\Helmut Vonhoegen\Documents\
6	=INFO("Dateienzahl")	38
7	=INFO("Ursprung")	$A:$A$1
8	=INFO("Sysversion")	Windows (64-bit) NT 6.02
9	=INFO("Berechne")	Automatisch
10	=INFO("Version")	15.0
11	=INFO("System")	pcdos

Abbildung 14.5 Die Funktion `INFO()` liefert Informationen zur Umgebung.

Es ist unter Umständen notwendig, eine Neuberechnung zu starten, um den aktuellen Wert der Funktion zu erhalten.

ISTBEZUG()

ISREF()

Syntax:	ISTBEZUG(**Wert**)
Beispiel:	=ISTBEZUG(Daten1)
Ergebnis:	FALSCH, wenn der Bereichsname nicht existiert

Die Funktion ISTBEZUG() testet, ob das mit Wert angegebene Argument ein gültiger Bereichsname oder eine Bereichsadresse ist. Ist das der Fall, ergibt die Funktion den Wahrheitswert WAHR, sonst den Wert FALSCH. Zellbezüge werden ohne Anführungszeichen angegeben. Wird ein Blattname angegeben, der nicht existiert, versucht Excel, eine solche Tabelle zu finden, und bietet den **Öffnen**-Dialog an. Wird dieser abgebrochen, liefert die Funktion trotzdem den Wert WAHR. Die Prüfung ist also nur formal.

ISTFEHL()

ISERR()

Syntax:	ISTFEHL(**Wert**)
Beispiel:	=ISTFEHL(A10)
Ergebnis:	WAHR, wenn A10 eine Fehlermeldung (außer #NV) enthält

Die Funktion ISTFEHL() testet, ob das mit Wert angegebene Argument einen Fehlerwert enthält oder liefert (Ausnahme: #NV). Ist ein Fehlerwert vorhanden, wird WAHR ausgegeben, in allen anderen Fällen FALSCH.

Ist Wert keine gültige Zelladresse und kein gültiger Name, liefert die Funktion ebenfalls das Ergebnis WAHR. Wird als Wert ein Zellbereich angegeben, wird der gesamte Bereich geprüft. Die Funktion liefert WAHR, wenn alle Zellen des Bereichs Fehlerwerte liefern, ansonsten ist das Ergebnis FALSCH.

ISTFEHLER()

ISERROR()

Syntax:	ISTFEHLER(**Wert**)
Beispiel:	=ISTFEHLER(A10)
Ergebnis:	WAHR, wenn A10 = #NV

Die Funktion ISTFEHLER() testet, ob das mit Wert angegebene Argument einen Fehlerwert enthält oder liefert, und zwar diesmal einschließlich des Fehlerwerts #NV. Ist ein Fehlerwert vorhanden, wird WAHR ausgegeben, in allen anderen Fällen FALSCH.

Ist Wert keine gültige Zelladresse und kein gültiger Name, liefert die Funktion ebenfalls das Ergebnis WAHR. Wird als Wert ein Zellbereich angegeben, wird der gesamte Bereich geprüft. Die Funktion liefert WAHR, wenn alle Zellen des Bereichs Fehlerwerte liefern, ansonsten ist das Ergebnis FALSCH.

ISTFORMEL()

ISFORMULA()

Syntax: ISTFORMEL(**Bezug**)
Beispiel: =ISTFORMEL(A10)
Ergebnis: WAHR, wenn A10 eine Formel enthält

Die in Excel 2013 neu eingeführte Funktion ISTFORMEL() testet, ob die mit Bezug angegebene Zelle eine Formel enthält oder nicht. Ist eine Formel vorhanden, wird WAHR ausgegeben, in allen anderen Fällen FALSCH.

Ist Bezug keine gültige Zelladresse und kein gültiger Name, liefert die Funktion den Fehler #NAME?. Wird als Bezug ein Zellbereich angegeben, wird nur die erste Zelle des Bereichs geprüft, anders als bei den Funktionen ISTFEHL() und ISTFEHLER().

ISTGERADE()

ISEVEN()

Syntax: ISTGERADE(**Zahl**)
Beispiel: =ISTGERADE(3)
Ergebnis: FALSCH

Die Funktion ISTGERADE() testet, ob das mit Zahl angegebene Argument durch 2 ganzzahlig teilbar ist, und liefert WAHR, wenn das der Fall ist, vergleiche ISTUNGERADE(). Dabei wird erwartet, dass sich das Argument Zahl numerisch auswerten lässt.

Ist Zahl nicht numerisch, liefert die Funktion unterschiedliche Ergebnisse: Bei einem Text wird die Fehlermeldung #WERT! ausgegeben, ebenso bei einem logischen Wert, bei einem Fehlerwert wird dieselbe ausgegeben. Wird als Bezug ein Zellbereich angegeben, zeigt die Funktion ebenfalls den Fehler #WERT! an. Ist Zahl keine gültige Zelladresse und kein gültiger Name, liefert die Funktion das Ergebnis #NAME?.

Entspricht Wert einer gebrochenen Zahl, werden ohne Rundung nur die Vorkommastellen ausgewertet.

```
=ISTGERADE(2,3)
```

liefert WAHR,

```
=ISTGERADE(1,6)
```

dagegen FALSCH.

Ist Wert keine gültige Zelladresse und kein gültiger Name, liefert die Funktion ebenfalls das Ergebnis #NAME?.

ISTKTEXT()

ISNONTEXT()

Syntax:	ISTKTEXT(**Wert**)
Beispiel:	=ISTKTEXT(A1)
Ergebnis:	FALSCH, wenn A1 Text enthält

Die Funktion ISTKTEXT() testet, ob in einer Zelle kein Text vorhanden ist. Ist Text vorhanden, so wird als Ergebnis FALSCH ausgegeben, andernfalls und bei einer leeren Zelle WAHR. Auch wenn Wert der Bezug auf eine leere Zelle ist, ist das Ergebnis WAHR. Enthält die Zelle dagegen eine leere Zeichenfolge, ist das Ergebnis FALSCH!

ISTLEER()

ISBLANK()

Syntax:	ISTLEER(**Wert**)
Beispiel:	=ISTLEER(A10)
Ergebnis:	WAHR, wenn A10 leer ist

Die Funktion ISTLEER() testet, ob das Argument Wert eine leere Zelle liefert. Wenn das der Fall ist, wird WAHR ausgegeben, andernfalls FALSCH. Das Ergebnis FALSCH wird also auch ausgegeben, wenn Wert einen Fehlerwert liefert, eine Bereichsadresse oder einen nicht existierenden Bereichsnamen enthält.

ISTLOG()

ISLOGICAL()

Syntax: ISTLOG(**Wert**)
Beispiel: =ISTLOG(FALSCH)
Ergebnis: WAHR

Die Funktion ISTLOG() überprüft, ob das Argument Wert einen Wahrheitswert liefert. Ist Wert ein Zellbezug, kann so festgestellt werden, ob die Zelle einen Wahrheitswert enthält oder ergibt. Ist ein Wahrheitswert, eine Formel oder Funktion, die zu einem Wahrheitswert führt, eingetragen, wird WAHR ausgegeben, andernfalls FALSCH.

Die Funktion kann genutzt werden, um über ein bedingtes Format die Zellen zu kennzeichnen, die den Wert WAHR oder den Wert FALSCH liefern. In dem Beispiel in Abbildung 14.6 werden dazu für den markierten Bereich zwei Regeln definiert, die mit unterschiedlichen Hintergrundfarben verknüpft werden. Den Zellen, die die Regel

```
=UND(ISTLOG(F5);F5)
```

erfüllen, wird die Farbe Grün zugeordnet. Die Bedingung ist nur erfüllt, wenn die Zelle sowohl einen logischen Wert liefert als auch den Wert WAHR.

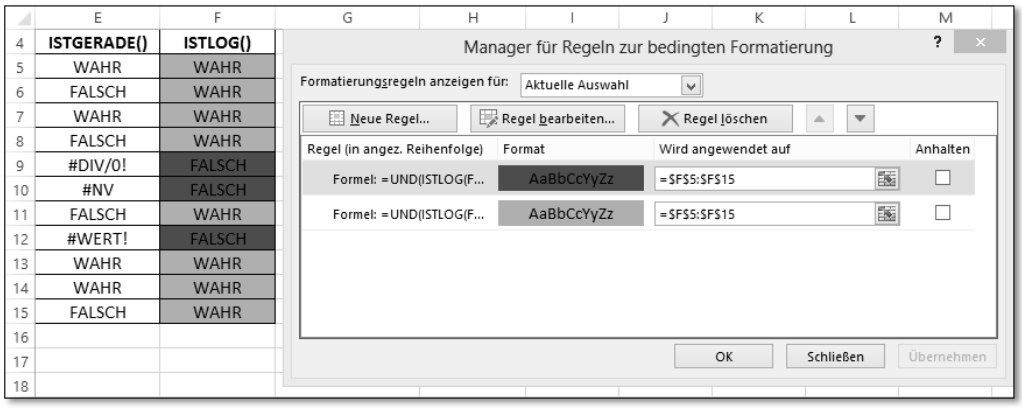


Abbildung 14.6 Bedingte Formate für logische Werte

Die Zellen, die die Regel

```
=UND(ISTLOG(F5);NICHT(F5))
```

erfüllen, also den logischen Wert FALSCH ergeben, werden mit Rot verknüpft.

ISTNV()

ISNA()

Syntax:	ISTNV(**Wert**)
Beispiel:	=ISTNV(A10)
Ergebnis:	FALSCH, wenn A10 =#DIV/0!

Die Funktion ISTNV() überprüft, ob eine Zelle den Fehlerwert #NV enthält. Das Ergebnis wird als Wahrheitswert ausgegeben. Sollen in einem Zellbereich mit #NV gekennzeichnete Zellen farbig hervorgehoben werden, kann ein bedingtes Format mit einer Formel wie

```
=ISTNV(A1)
```

verwendet werden.

ISTTEXT()

ISTEXT()

Syntax:	ISTTEXT(**Wert**)
Beispiel:	=ISTTEXT("666")
Ergebnis:	WAHR

Die Funktion ISTTEXT() prüft, ob das Argument Wert eine Zeichenfolge liefert. Ist Wert ein Bezug auf eine Zelle, die Text enthält, wird WAHR ausgegeben. Eine leere Zelle ergibt den Wert FALSCH.

Enthält eine Spalte Werte, bei denen nicht sicher ist, ob es sich um Zahlen oder um Texte aus Ziffern handelt, kann mit dieser Funktion festgestellt werden, was der Fall ist. Um die Daten zu bereinigen und einheitlich als Zahlen auszugeben, kann beispielsweise eine Formel wie

```
=WENN(ISTTEXT(A5);WERT(A5);A5)
```

verwendet werden. Abbildung 14.7 zeigt eine kleine Tabelle mit einer solchen Umwandlung.

	A	B	C	D
1				
2	**Typumwandlung**			
3				
4	**Werte**	**ISTTEXT()**	**Umwandlung in Zahlen**	
5	123	FALSCH	123	**=WENN(ISTTEXT(A5);WERT(A5);A5)**
6	456	FALSCH	456	
7	789	WAHR	789	
8	101	WAHR	101	
9				

Abbildung 14.7 Umwandlung »unechter« Zahlen

Die Formel

```
=ISTTEXT(A5&A7)
```

liefert ebenfalls den Wert `WAHR`, weil der Verkettungsoperator in diesem Beispiel dafür sorgt, dass der numerische Wert aus der Zelle A5 in eine Zeichenfolge umgewandelt wird.

ISTUNGERADE()

ISODD()

Syntax:	`ISTUNGERADE(`**`Zahl`**`)`
Beispiel:	`=ISTUNGERADE(4)`
Ergebnis:	`FALSCH`

Die Funktion `ISTUNGERADE()` testet, ob das mit `Zahl` angegebene Argument durch 2 ganzzahlig teilbar ist, und liefert `FALSCH`, wenn das der Fall ist. Diese Funktion ist also das Gegenstück zu der Funktion `ISTGERADE()`. Auch hier gilt, dass die Nachkommastellen von `Zahl` ignoriert werden. Dabei wird erwartet, dass sich das Argument `Zahl` numerisch auswerten lässt.

Ist `Zahl` nicht numerisch, liefert die Funktion unterschiedliche Ergebnisse: Bei einem Text wird die Fehlermeldung `#WERT!` ausgegeben, ebenso bei einem logischen Wert, bei einem Fehlerwert wird dieselbe ausgegeben. Wird als Bezug ein Zellbereich angegeben, zeigt die Funktion ebenfalls den Fehler `#WERT!` an. Ist `Zahl` keine gültige Zelladresse und kein gültiger Name, liefert die Funktion das Ergebnis `#NAME?`.

ISTZAHL()

ISNUMBER()

Syntax: `ISTZAHL(`**`Wert`**`)`
Beispiel: `=ISTZAHL(0)`
Ergebnis: `WAHR`

Die Funktion `ISTZAHL()` prüft, ob das Argument `Wert` eine Zahl liefert. Ist `Wert` ein Zellbezug, wird getestet, ob die Zelle eine Zahl enthält. Ist eine Zahl enthalten, dann wird als Ergebnis `WAHR` ausgegeben, enthält die Zelle keine Zahl, wird `FALSCH` ausgegeben. Das gilt auch für eine leere Zelle, obwohl diese ansonsten mit dem Wert `0` ausgewertet wird. Datums- und Zeitwerte werden ebenfalls als gültige Zahlenwerte behandelt, da sie ja intern über serielle Zahlen und Bruchzahlen dargestellt werden. Ist `Wert` ein ungültiger Bezug, ist das Ergebnis ebenfalls `FALSCH`.

Die bei der Funktion `ISTTEXT()` verwendete Formel zur Umwandlung »unechter« Zahlen kann in diesem Fall umgedreht werden, wie Abbildung 14.8 zeigt:

	A	B	C	D
1				
2	**Typumwandlung für Zeichenfolgen**			
3				
4	**Werte**	**ISTZAHL()**	**Umwandlung in Zahlen**	
5	123	WAHR	123	**=WENN(ISTZAHL(A5);A5;WERT(A5))**
6	456	WAHR	456	
7	789	FALSCH	789	
8	101	FALSCH	101	
9				
10			**Test der Datentypanpassung**	
11		123789	WAHR	**=ISTTEXT(A5&A7)**
12		912	WAHR	**=ISTZAHL(A5+A7)**
13				

Abbildung 14.8 Umwandlung »unechter« Zahlen, die Zweite

Die Formel

```
=ISTZAHL(A5+A7)
```

liefert ebenfalls den Wert `WAHR`, weil der Operator in diesem Beispiel dafür sorgt, dass der nicht numerische Wert aus der Zelle A7 in eine Zahl umgewandelt wird.

Anders als bei den anderen Funktionen interpretieren die IST-Funktionen Zahlen, die in Anführungszeichen gesetzt sind, nicht als Zahlen, sondern als Zeichenfolgen.

```
=ISTZAHL("123")
```

ergibt also FALSCH.

14.2.1 Nicht numerische Werte zählen

Das in Abbildung 14.9 gezeigte Beispiel ist eine der Möglichkeiten, die Funktion nutzbringend anzuwenden. Häufig ist es ja so, dass die numerischen Daten für einen Tabellenbereich nicht alle gleichzeitig eingetragen werden können. Wenn Sie für einen solchen Tabellenbereich feststellen wollen, wie viele Einträge noch fehlen, können Sie mit einer Matrixformel arbeiten:

```
{=SUMME(WENN(ISTZAHL(Wertebereich); 0; 1))}
```

Die Formel setzt voraus, dass der Zellbereich, für den numerische Werte erwartet werden – hier C5:D11 –, mit dem Namen "Wertebereich" belegt worden ist. Über die Funktion WENN() wird für jede Zelle geprüft, ob sie numerisch ist. Ist das der Fall, wird für die Zelle der Wert 0 angegeben, andernfalls 1. Die umgebende Funktion SUMME() zählt dann die vergebenen Werte und gibt die Zahl der nicht numerischen Zellen als Ergebnis aus.

	A	B	C	D
1				
2	**Nicht numerische Werte zählen**		Anzahl	4
3				
4	**Artnr**	**Bezeichnung**	**Absatz**	**Umsatz**
5	7777	Jalousie Cc	1232	26181
6	7778	Jalousie Ccs	1463	63360
7	7578	Jalousie Ccw	Wert fehlt	Wert fehlt
8	7577	Jalousie Ccx	1243	25631
9	6378	Jalousie Louise	770	12650
10	8443	Jalousie V	Wert fehlt	Wert fehlt
11	8243	Jalousie VV	253	43450

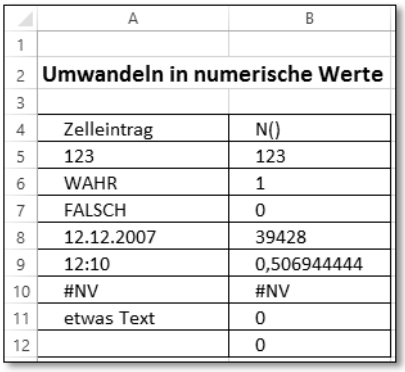
Abbildung 14.9 Zählen nicht numerischer Werte

N()

N()

Syntax: N(**Wert**)
Beispiel: =N("text")
Ergebnis: 0

Die Funktion N() gibt das mit Wert angegebene Argument als einen in eine Zahl gewandelten Wert zurück. Findet die Funktion eine Zahl vor, so wird diese unverändert als Zahl ausgegeben. Der Wahrheitswert WAHR wird als 1 wiedergegeben, FALSCH als 0, Datums- und Zeitangaben in einem der integrierten Excel-Formate werden als entsprechende serielle Zahlen ausgegeben. Bei Datumswerten steht dabei 1 für den 1.1.1900, Zeitangaben werden mit einer entsprechenden Bruchzahl ausgegeben. Fehlerwerte werden auch als Fehlerwerte ausgegeben. Texte oder leere Zellen werden als 0 ausgegeben. Das gilt auch, wenn eine Zahl mit dem Format **Text** belegt wurde.



	A	B
1		
2	**Umwandeln in numerische Werte**	
3		
4	Zelleintrag	N()
5	123	123
6	WAHR	1
7	FALSCH	0
8	12.12.2007	39428
9	12:10	0,506944444
10	#NV	#NV
11	etwas Text	0
12		0

Abbildung 14.10 Beispiele für Umwandlungen mit der Funktion N()

NV()

NA()

Syntax: NV()
Beispiel: =NV()
Ergebnis: #NV

Die Funktion NV() trägt den Fehlerwert #NV (**n**icht **v**orhanden) in die Zelle ein, die die Funktion enthält. Obwohl die Funktion keine Argumente verlangt, ist es notwendig, die beiden Klammern mit anzugeben, damit Excel den Eintrag als Funktion erkennt.

Hauptsächlich lässt sich die Funktion zur auffälligen Kennzeichnung von Zellen einsetzen, in die (noch) keine Werte eingetragen sind. Das hat zudem den Vorteil, dass sich in einem Tabellenblatt die Zellen mit fehlenden Werten über den Dialog **Inhalte auswählen** und die Option **Formeln • Fehler** in einem Zug markieren lassen, so dass beim nachträg-

lichen Eintragen mit Hilfe der ⇥-Taste die entsprechenden Zellen angesteuert werden können, wie Abbildung 14.11 zeigt:

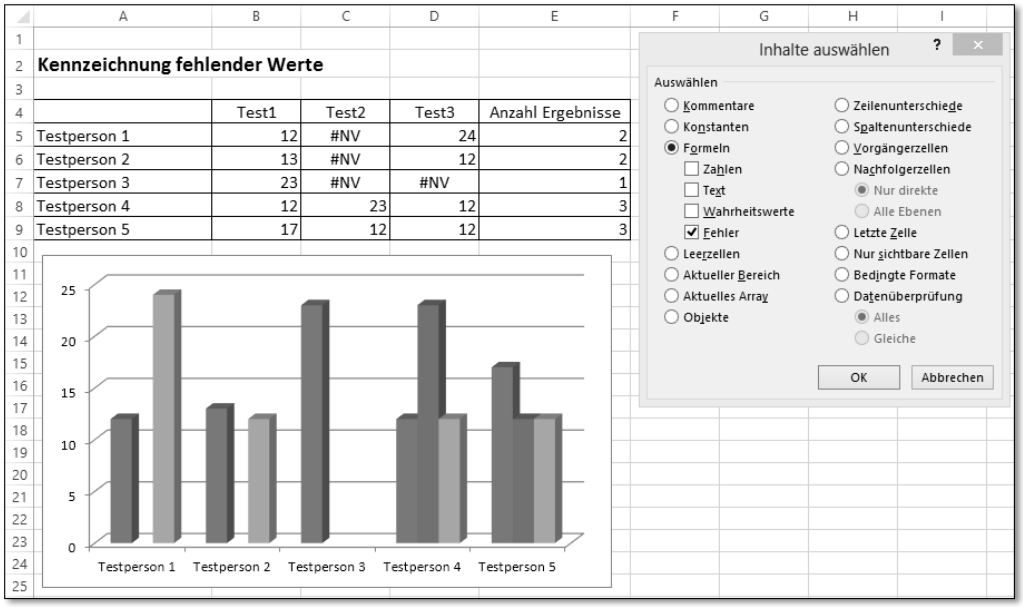


Abbildung 14.11 Zellen mit dem Fehlerwert `#NV` lassen sich in einem Zug auswählen.

Dies gilt nicht, wenn der Fehlerwert `#NV` direkt in die Zelle eingetragen wird, da dies keine Funktion darstellt.

Für einen Bereich kann die Funktion auch als Matrixformel eingegeben werden. Formeln, die sich auf eine Zelle beziehen, in der `#NV` eingetragen ist, liefern auch `#NV`. Das gilt auch, wenn eine solche Zelle im Bereich einer `SUMME()`-Funktion vorkommt. Die Funktion `ANZAHL()` dagegen behandelt eine Zelle mit `=NV()` wie eine leere Zelle.

TYP()

TYPE()

Syntax: `TYP(`**`Wert`**`)`

Beispiel: `=TYP(15)`

Ergebnis: `1`

Die Funktion TYP() wird verwendet, um den Datentyp des angegebenen Arguments Wert abzufragen. Als Ergebnis werden numerische Typkennzeichen geliefert, die in Tabelle 14.3 zusammengestellt sind:

Argument	Rückgabewert
Zahl	1
Text	2
Wahrheitswert	4
Fehlerwert	16
Matrix	64

Tabelle 14.3 Codes für die Datentypen in Excel

Die Funktion ist nützlich, um festzustellen, ob das Ergebnis einer anderen Funktion oder einer Formel einem bestimmten Datentyp entspricht. Dies ist unter Umständen notwendig, wenn ein Wert als Argument für eine Funktion oder innerhalb einer Formel verwendet werden soll, die einen bestimmten Datentyp erwartet. Die Formel

```
=WENN(TYP(A11)=1;A11*1,19;"Wert in A11 ist keine Zahl")
```

prüft zunächst, ob die Zelle A11 eine Zahl enthält, bevor die Multiplikation mit dem angegebenen Faktor 1,19 ausgeführt wird.

	A	B	C
1			
2	**Prüfen des Datentyps von Argumenten**		
3			
4	Skye	2	=TYP(A4)
5		2	=TYP("Frau "&A4)
6		16	=TYP(2+A4)
7	20	1	=TYP(A7)
8	WAHR	4	=TYP(A8)
9		64	=TYP({1,2;3,4})
10			
11	100	119	=WENN(TYP(A11)=1;A11*1,19;"Wert in A11 ist keine Zahl")

Abbildung 14.12 Verschiedene Beispiele für die Verwendung der TYP-Funktion

Dagegen ist es nicht möglich, mit TYP() festzustellen, ob eine Zelle eine Formel enthält.

Anstelle der Abfrage mit der Funktion TYP() kann mit den Funktionen ISTZAHL() und ISTTEXT() etc. auch gezielt geprüft werden, ob ein bestimmter Datentyp vorliegt.

ZELLE()

CELL()

Syntax: `ZELLE(`**`Infotyp`**`; Bezug)`
Beispiel: `=ZELLE("Spalte";A1)`
Ergebnis: `1`

Die Funktion `ZELLE()` gibt vielfältige Informationen über eine Zelle bzw. die Zelle in der linken oberen Ecke eines Bereichsbezugs. Unter `Infotyp` wird ein Textstring eingetragen, der Informationen abfragt, die durch diesen Text definiert werden; `Bezug` gibt die Zelle an, über die Informationen eingeholt werden sollen. Wenn `Bezug` nicht angegeben ist, werden immer die Informationen für die zuletzt geänderte Zelle zurückgegeben.

Infotyp	Rückgabewert
Adresse	der absolute Bezug der ersten Zelle in `Bezug` als Text
Breite	Spaltenbreite der Zelle, auf eine ganze Zahl gerundet. Jede Einheit der Spaltenbreite ist gleich der Breite eines Zeichens im Standardschriftgrad.
Dateiname	Liefert den Dateinamen (und den vollständigen Pfad) der Datei, die `Bezug` enthält, als Text. Ergibt eine leere Textzeichenfolge (`""`), wenn das Tabellenblatt, das den Bezug enthält, noch nicht gespeichert wurde.
Farbe	Ergibt `1`, wenn die Zelle für negative Werte farbig formatiert ist, sonst `0`.
Format	Liefert einen Textwert, der das Zahlenformat der Zelle anzeigt. Die möglichen Werte sind in der folgenden Tabelle aufgeführt. Gibt – am Ende zurück, wenn die Zelle für negative Werte farbig formatiert ist. Gibt `()` am Ende zurück, wenn die Zelle für positive oder auch für alle Werte mit Klammern formatiert ist.
Inhalt	Liefert den Wert der linken obersten Zelle, die zu `Bezug` gehört, also keine Formel.
Klammern	Liefert `1`, wenn die Zelle für positive oder alle Werte mit Klammern formatiert ist, sonst `0`.

Tabelle 14.4 Werte für das Argument `Infotyp` und ihre Bedeutung

Infotyp	Rückgabewert
Präfix	Liefert einen Textwert, der dem »Beschriftungspräfix« der Zelle entspricht. Ein einfaches Anführungszeichen (') zeigt an, dass die Zelle linksbündigen Text enthält, ein doppeltes Anführungszeichen (") zeigt rechtsbündigen Text an, ein Zirkumflexzeichen (^) zeigt zentrierten Text an. Ein umgekehrter Schrägstrich (\) zeigt an, dass die Zelle mit Textzeichen aufgefüllt ist, und eine leere Textzeichenfolge ("") zeigt an, dass die Zelle keinen Text enthält.
Schutz	Liefert `0`, wenn die Zelle nicht gesperrt ist, sonst `1`.
Spalte	Liefert die Spaltennummer der Zelle in `Bezug`.
Typ	Informiert über den Datentyp des Zellinhalts. Liefert `b`, wenn die Zelle leer (blank) ist, `l` für Beschriftung (`label`), also wenn die Zelle eine Textkonstante enthält, und `w` für Wert, wenn die Zelle etwas anderes enthält.
Zeile	Liefert die Zeilennummer der Zelle in Bezug.

Tabelle 14.4 Werte für das Argument `Infotyp` und ihre Bedeutung (Forts.)

Die erlaubten Werte für `Infotyp` werden bei der direkten Eingabe der Funktion zur Auswahl angeboten, sobald die öffnende Klammer gesetzt ist.

	A	B	C
1			
2	**Informationen zu Zellen**		
3			
4	**Zelle**	**ZELLE()**	
5	0	$A$5	=ZELLE("adresse";A5)
6	1.000,11	1	=ZELLE("spalte";A6)
7	-100,00 €	1	=ZELLE("farbe";A7)
8	02/14/07	39127	=ZELLE("inhalt";A8)
9	11.12	$A:$A$9	=ZELLE("koord";A9)
10		C:\ExcelFunktionen\Beispiele Funktionen Excel 2013\[Informationsfunktionen.xlsx]ZELLE	=ZELLE("dateiname")
11	-100,00	0	=ZELLE("klammern";A11)
12	100	"	=ZELLE("präfix";A12)
13	10,00%	1	=ZELLE("schutz";A13)
14	Standard	14	=ZELLE("zeile";A14)
15	1 1/3	w	=ZELLE("typ";A15)
16	12. Feb. 07	19	=ZELLE("breite";A16)
17	10:55:20	11	=ZELLE("breite")
18	12:12 PM	S	=ZELLE("format")

Abbildung 14.13 Beispiele für die Funktion `ZELLE()`

Tabelle 14.5 listet die Textwerte auf, die ZELLE() zurückgibt, wenn Infotyp vom Typ Format und Bezug eine Zelle ist, die mit einem integrierten Zahlenformat formatiert ist.

Format	Rückgabewert
Standard	S
0	F0
#.##0	0,0
0,00	F2
#.##0,00	,2
#.##0_€;-#.##0_€	W0
#.##0_€;[Rot]-#.##0_€	W0-
#.##0,00_€;-#.##0,00_€	W2
#.##0,00_€;[Rot]-#.##0,00_€	W2-
0%	P0
0,00%	P2
0,00E+00	E2
# ?/? oder # ??/??	S
m/t/jj oder m/t/jj h:mm oder mm/tt/jj	D4
t-mmm-jj oder tt-mmm-jj	D1
t-mmm oder tt-mmm	D2
mmm-jj	D3
mm/tt/jjjj	D4
mm/tt	D5
h:mm:ss AM/PM	U1
h:mm AM/PM	U2
h:mm:ss	U3
h:mm	U4

Tabelle 14.5 Codes für Formate

Wenn die Zelle später mit einem benutzerdefinierten Format formatiert wird, muss das Tabellenblatt eventuell neu berechnet werden, um die ZELLE()-Formel zu aktualisieren.

14.2.2 Einsatz mit bedingten Formaten

Praktische Einsatzmöglichkeiten der Funktion ergeben sich insbesondere im Rahmen von bedingten Formatierungen von Zellbereichen. Sollen beispielsweise in einer Tabelle alle Einträge mit einer Hintergrundfarbe gekennzeichnet werden, die ein beliebiges Währungsformat verwenden, kann im Dialog für die bedingte Formatierung eine Formel eingesetzt werden, die die erste Stelle der Formatkennzeichnung abfragt, die die Funktion mit dem Infotyp `Format` liefert, beispielsweise:

```
=LINKS(ZELLE("Format";A5);1)="W"
```

Abbildung 14.14 zeigt die Anwendung einer solchen Formatierungsregel auf den Zellbereich in Spalte A.

	A	B	C
1			
2	**Bedingte Formate, abhängig vom Zahlenformat**		
3			
4	**Zelle**	**ZELLE("format")**	
5	0	.0	
6	1.000,11	.2	
7	100.001 €	W0	
15	-100.000,50 €	W2-	
16	100.000,50 €	W2	
17	13.02.2007	D1	
18	02/14/07	D4	
19	11.12	D5	
20	1,11E+16	E2	
21	1,11	F2	
22	10%	P0	

Abbildung 14.14 Bedingte Formatierung auf der Basis der Formatzeichen

Ungültige Werte für `Infotyp` führen zu dem Fehler `#WERT!`. Ungültige Werte für `Bezug` führen zu dem Fehler `#NAME?`.

15 Webfunktionen

Funktion	Seite	Funktion	Seite
URLCODIEREN()	822	XMLFILTERN()	825
WEBDIENST()	825		

15.1 Einsatzbereich der Webfunktionen

Unter der ganz neuen Kategorie **Web** tauchen in Excel 2013 drei Funktionen auf, die den Zugriff auf Daten im Web vereinfachen. Die Funktion `URLCODIEREN()` verwandelt Zeichenfolgen in formal gültige Webadressen oder Teile davon um. Die `WEBDIENST()`-Funktion fragt Daten von einem Webdienst ab, dessen URL angegeben ist. Mit der Funktion `XMLFILTERN()` können Auszüge aus XML-Inhalten mit Hilfe eines entsprechenden XPath-Ausdrucks eingelesen werden.

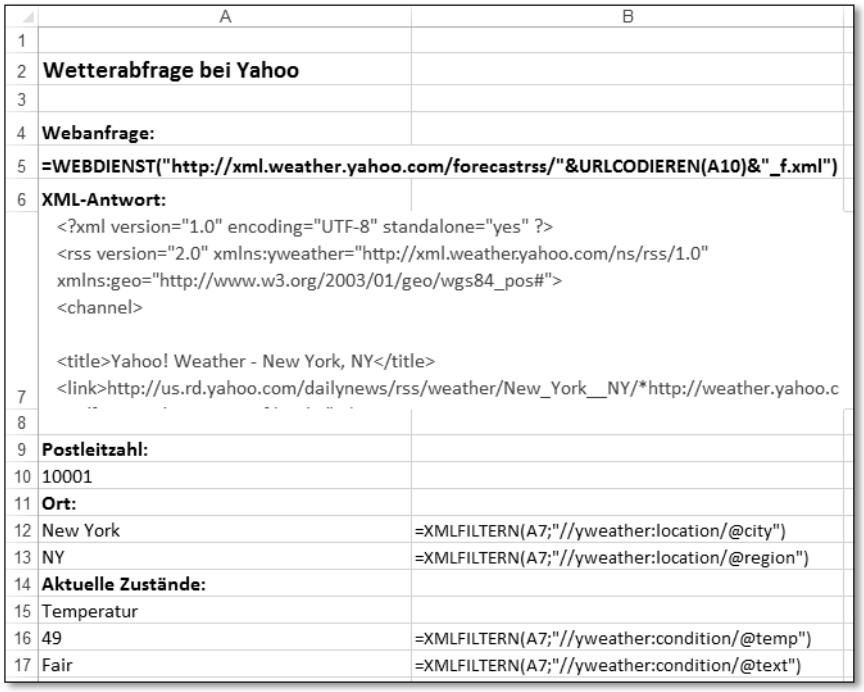
Diese drei Funktionen sind hauptsächlich im Zusammenspiel von praktischem Nutzen. Wenn Sie Daten von einem Webdienst auswerten wollen, die im XML-Format bereitgestellt werden, können Sie dafür die Funktion `WEBDIENST()` verwenden. Die dafür benötigte Webanfrage lässt sich mit Hilfe der Funktion `URLCODIEREN()`, wo es nötig ist, aufbereiten, damit der Browser damit arbeiten kann. Sind die Daten in einen Zellbereich eingelesen, lassen sich die für Sie interessanten Informationen daraus mit Hilfe der Funktion `XMLFILTERN()` herausholen.

Da die Zugangsdaten bekannt sind, verwenden wir als Beispiel eine einfache Wetterabfrage an den von Yahoo! dafür zur Verfügung gestellten Webdienst.

Die Formel in Zelle A37 fragt den Webdienst von Yahoo nach Wetterinformationen für New York ab. Mit Hilfe der Funktion `URLCODIEREN()` wird der Inhalt der Zelle A10 in die Webadresse eingebaut. Sie übergibt als Parameter für die Abfrage die in A10 abgelegte Postleitzahl. Die Zellen A12, A13, A16 und A17 enthalten verschiedene `XMLFILTERN()`-Funktionen, um aus den in A7 eingelesenen Webdienst-Daten einzelne Informationen herauszufiltern:

```
=XMLFILTERN(A7;"//yweather:condition/@temp")
```

liefert beispielsweise die aktuelle Temperatur in Fahrenheit.



	A	B
1		
2	**Wetterabfrage bei Yahoo**	
3		
4	**Webanfrage:**	
5	**=WEBDIENST("http://xml.weather.yahoo.com/forecastrss/"&URLCODIEREN(A10)&"_f.xml")**	
6	**XML-Antwort:**	
7	<?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <rss version="2.0" xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"> <channel> <title>Yahoo! Weather - New York, NY</title> <link>http://us.rd.yahoo.com/dailynews/rss/weather/New_York__NY/*http://weather.yahoo.c	
8		
9	**Postleitzahl:**	
10	10001	
11	**Ort:**	
12	New York	=XMLFILTERN(A7;"//yweather:location/@city")
13	NY	=XMLFILTERN(A7;"//yweather:location/@region")
14	**Aktuelle Zustände:**	
15	Temperatur	
16	49	=XMLFILTERN(A7;"//yweather:condition/@temp")
17	Fair	=XMLFILTERN(A7;"//yweather:condition/@text")

Abbildung 15.1 Beispiel für die Abfrage eines Webdienstes, der Daten im Format *XML* bereitstellt

Das Beispiel profitiert davon, dass in einer einzelnen Zelle notfalls auch ein ganzes XML-Dokument abgelegt werden kann. Der Webdienst liefert das Ergebnis der Webabfrage im XML-Format.

URLCODIEREN()

ENCODEURL()

Syntax: `URLCODIEREN(`**`Text`**`)`

Beispiel: `=URLCODIEREN("meine abfrage")`

Ergebnis: `"meine%20abfrage"`

Die Funktion `URLCODIEREN()` erstellt aus der mit `Text` angegebenen Zeichenfolge eine URL-kodierte Zeichenfolge, die dann als Webadresse oder Teil einer Webadresse verwendet werden kann. Dabei werden alle in einer URL nicht erlaubten Zeichen entsprechend maskiert.

15.1.1 Exkurs zur URL-Kodierung

Die URL-Kodierung ist notwendig, wenn in einer Webadresse Zeichen enthalten sind, die aufgrund der für URLs vorgeschriebenen Syntax nicht zugelassen sind. Diese Syntax lässt nur bestimmte Zeichen des ASCII-Zeichensatzes zu. Werden andere Zeichen benötigt, müssen sie durch Zeichenfolgen ersetzt werden, die mit einem Prozentzeichen beginnen, weshalb auch von Prozent-Kodierung gesprochen wird. Hinter dem Prozentzeichen, das für das Escape-Zeichen steht, folgen zwei Ziffern, die als Hexadezimalzahl den ASCII-Code des maskierten Zeichens angeben.

Der einfachste Fall ist das Leerzeichen. Der Browser würde ein unmaskiertes Leerzeichen als Ende der Webadresse interpretieren, deshalb wird statt des Leerzeichens die Zeichenfolge `%20` verwendet.

Einige Zeichen haben in einer URL eine festgelegte Bedeutung:

- Das Fragezeichen (`?`) leitet den Datenteil (Query String) einer URL ein.
- Das Gleichheitszeichen (`=`) steht zwischen dem Namen eines Parameters und seinem Wert.
- Das Et-Zeichen (`&`) wird im Datenteil als Trennzeichen zwischen »Parameter=Wert«-Paaren eingesetzt.
- Dem Doppelkreuz (`#`) folgt der Name eines Dokumentenankers.

Insgesamt gelten folgende Zeichen als reserviert:

```
! # $ % & ' ( ) * + , / : ; = ? @ [ ]
```

Die Tabelle in Abbildung 15.2 zeigt in Spalte E die hexadezimalen Codes der reservierten Zeichen.

Nicht reserviert sind die Buchstaben `[A-Z, a-z]`, die Ziffern `[0-9]` und die Sonderzeichen _, ., ~.

Soll eines der reservierten Zeichen nicht zu dem vorgegebenen Zweck verwendet werden, muss es maskiert werden. Wenn also beispielsweise das Doppelkreuz Teil eines Parameterwerts sein soll, können Sie dafür `%23` verwenden. Die folgende Adresse

```
www.meine-site.de?seite=ms8#kopf
```

adressiert den Anker `kopf`.

Soll dagegen der Wert des Parameters `seite = ms8#kopf` sein, müssen Sie schreiben:

```
www.meine-site.de?seite=ms8%23kopf
```

	C	D	E
11	Reservierte Zeichen	CODE()	DEZINHEX()
12	!	33	21
13	#	35	23
14	$	36	24
15	%	37	25
16	&	38	26
17	(	40	28
18	)	41	29
19	*	42	2A
20	+	43	2B
21	,	44	2C
22	/	47	2F
23	:	58	3A
24	;	59	3B
25	=	61	3D
26	?	63	3F
27	@	64	40
28	[	91	5B
29	]	93	5D

Abbildung 15.2 Die Codes der reservierten Zeichen, dezimal und hexadezimal

Wird das Prozentzeichen als Teil einer Adresse benötigt, muss es ebenfalls maskiert werden: `%25`.

Auch Zeichen, die nicht im ASCII-Zeichensatz enthalten sind, lassen sich mit dem vorangestellten Prozentzeichen kodieren. Wie die Kodierung erfolgt, hängt allerdings von der Kodierung ab, die die Webseite selbst verwendet. Wird mit UTF-8 gearbeitet, lässt sich beispielsweise ein Umlaut wie ö mit dem Unicode-Zeichenwert 246 als `%C3%B6` darstellen. Abbildung 15.3 zeigt einige Beispiele, wie die Funktion arbeitet.

	A	B	C
1			
2	**Webabfragen codieren**		
3			
4	Adresse	URL-codierte Adresse	
5	www.meine-site.de?seite=ms 8#kopf	www.meine-site.de%3Fseite%3Dms%208%23kopf	=URLCODIEREN(A5)
6	www.meine-site.de?seite=öffentlich	www.meine-site.de?seite=ms%208%23kopf	="www.meine-site.de?seite="&URLCODIEREN("ms 8#kopf")
7	www.meine-site.de?seite=ms 8#kopf	www.meine-site.de?seite=%C3%B6ffentlich	="www.meine-site.de?seite="&URLCODIEREN("öffentlich")

Abbildung 15.3 Beispiele für die URL-Kodierung

Das Ergebnis der Formel in B5 zeigt allerdings, dass die Funktion `URLCODIERUNG()` alle reservierten Zeichen kodiert, wenn Sie die komplette Webadresse als Zeichenfolge angeben, also auch die Zeichen ? und =, die in diesem Fall gerade nicht maskiert werden sollen. Die Funktion kann also in der Regel nur verwendet werden, um genau die Teile einer Webadresse zu kodieren, bei denen die reservierten Zeichen gerade nicht ihre übliche Funktion innerhalb der URL-Syntax ausüben sollen.

WEBDIENST()

WEBSERVICE()

Syntax:	`WEBDIENST(`**`Url`**`)`
Beispiel:	`=WEBDIENST("http://xml.weather.yahoo.com/forecastrss/10001_f.xml")`
Ergebnis:	Liefert eine XML-Datei mit den Wetterdaten für New York.

Die Funktion `WEBDIENST()` gibt Daten eines Webdienstes zurück, der das XML-Format unterstützt. Das Argument `Url` muss die Webadresse des Dienstes und eine gültige Form der Abfrage gegenüber diesem Dienst enthalten. Da die Funktion mit einer GET-Anforderung an den Webdienst arbeitet, darf die maximale Größe einer solchen Anforderung – 2048 Zeichen – nicht überschritten werden. Die Funktion unterstützt nur das *http*-Protokoll, das *ftp*-Protokoll also nicht.

Kann die Funktion keine Daten liefern, ist das Ergebnis `#WERT!`.

XMLFILTERN()

FILTERXML()

Syntax:	`XMLFILTERN(`**`xml; xpath`**`)`
Beispiel:	`=XMLFILTERN (A7;"//yweather:condition/@temp)`
Ergebnis:	Filtert aus in A7 abgelegten XML-Daten die Informationen heraus, die über den XPath-Ausdruck ausgewählt werden.

Die Funktion `XMLFILTERN()` filtert mit Hilfe eines XPath-Ausdrucks Informationen aus Daten, die im XML-Format vorliegen. Benötigt werden dazu zwei Argumente: `xml` ist eine Zeichenfolge im XML-Format bzw. der Verweis auf eine Zelle, die solche Daten enthält.

Sind die Daten nicht wohlgeformt, liefert die Funktion den Fehlerwert `#WERT!`.

Da eine Zelle maximal 32.767 Zeichen aufnehmen kann, liegt hier auch die Grenze für die Menge der Zeichen, die das XML-Dokument enthalten darf. Die Möglichkeit, das XML-Dokument auf einen Zellbereich auszubreiten, wird nicht unterstützt.

Das zweite Argument, das benötigt wird, ist ein XPath-Ausdruck, der in der Lage ist, Daten aus dem XML-Dokument herauszuziehen. Über die Syntax solcher Ausdrücke finden Sie erste Informationen in dem nachfolgenden Exkurs.

Abbildung 15.4 zeigt in der Zelle A5 ein XML-Dokument mit Daten über Autoren. In Zelle A7 ist ein XPath-Ausdruck abgelegt, der die Autorennamen aus dem Dokument herausziehen soll. Um mehrere Namen herauszuziehen, wird zunächst ein entsprechend großer Bereich markiert und die Formel `=XMLFILTERN(A5;A7)` als Matrixformel abgelegt.

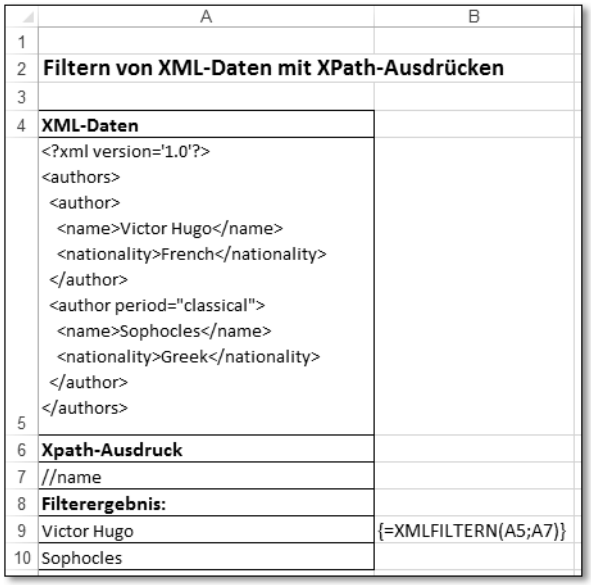


Abbildung 15.4 Filtern von XML-Daten

15.1.2 Exkurs zu XPath

Die logische Struktur eines XML-Dokuments lässt sich in einer Baumstruktur darstellen, in der die Informationseinheiten als Knoten erscheinen. XPath arbeitet auf der Basis eines solchen Baummodells. Ein XPath-Ausdruck wird ausgewertet, um ein Objekt zu gewinnen. Dabei sind vier Datentypen möglich. Neben den einfachen Typen `string`, `number` und `boolean` liefert `node-set` eine Knotenmenge.

Der Knotenbaum geht von einem Wurzelknoten aus. Die Zuordnung der Knoten des Baums zu den Komponenten des XML-Dokuments ist durch die Dokumentreihenfolge geregelt, die der Reihenfolge der Start-Tags im Dokument entspricht. Diese Reihenfolge bestimmt, wie ein Prozessor die Knoten ablaufen – traversieren – wird, wenn er den Auftrag erhält, sämtliche Knoten nacheinander zu besuchen. Es werden dabei nicht zunächst alle Knoten derselben Ebene nacheinander abgelaufen, sondern immer erst die Kinder und Kindeskinder des aktuellen Knotens abgearbeitet.

15.1.3 Lokalisierungspfade

Der wichtigste Ausdruckstyp in XPath ist der Lokalisierungspfad. Er gibt an, welcher Teil der Knotenmenge eines Dokuments ausgewählt und bearbeitet werden soll. Dabei werden zwei Schreibweisen verwendet, eine ausführliche und eine bequemere Kurzform. Die Funktion `XMLFILTERN()` unterstützt allerdings nur diese Kurzform.

Um in dem Beispiel in Abbildung 14.4 den Knoten `<name>` zu erreichen, genügt in der Kurzform folgender Ausdruck:

```
/authors/author/name
```

oder noch kürzer:

```
//name
```

Dieser Ausdruck liefert die entsprechende Knotenmenge, den `node-set`. Die einzelnen Stufen in der Hierarchie der Knoten werden also immer mit einem Schrägstrich getrennt, wobei der erste Schrägstrich für den Wurzelknoten steht. Ähnlich wie bei Dateisystemen dürfen absolute und relative Pfade verwendet werden. Ein absoluter Pfad geht immer vom Wurzelknoten aus, ein relativer dagegen von dem Knoten, der gerade der Kontextknoten ist. Das ist jeweils der Knoten, der durch die vorausgehenden Schritte erreicht wurde. Statt immer die vollständige Hierarchie anzugeben, lässt sich auch mit dem Doppelschrägstrich arbeiten wie im letzten Beispiel.

Anstelle konkreter Namen dürfen auch Wildcards verwendet werden:

```
/authors/author/*
```

liefert zum Beispiel alle Kinder des Elements `<authors>`.

Obwohl XPath die Elementknoten als Eltern ihrer Attributknoten behandelt, werden die Attributknoten nicht wie Kindknoten des Elementknotens angesprochen. Um den Attributwert zu erhalten, wird in der Kurzform eine Kombination aus dem @-Zeichen und Attributnamen verwendet:

```
//author/@period
```

liefert das Periodenattribut der Elementknoten `<author>`. Auch Attribute lassen sich über Wildcards suchen:

```
//author/@*
```

liefert alle Attribute des Elements `<author>`.

16 Analyse mit Pivot-Tabellen und -Diagrammen

Je größer die Datenmengen sind, die für die Abwicklung der Aufgaben in einem Bereich notwendig sind, umso stärker entwickelt sich in der Regel das Bedürfnis, die darin enthaltenen Informationen unter den verschiedensten Gesichtspunkten zusammenzufassen und auszuwerten. Die stärkste Methode, die Excel für diesen Zweck zur Verfügung stellt, ist zweifellos das Pivotieren, sei es in Form von Tabellen oder in Form von Diagrammen.

Excel 2013 macht auf diesem Feld einen enormen Sprung nach vorn, indem nun für solche Analysen nicht mehr nur einzelne Tabellen, sondern auch Datenmodelle aus mehreren über Fremdschlüssel verknüpfte Tabellen zugrunde gelegt werden können. Dabei wird im Arbeitsspeicher jeweils ein entsprechendes Datenmodell aufgebaut.

Da die verwerteten Daten stark komprimiert werden, lassen sich auch große Datenmengen in Excel-Dateien speichern und mit anderen Personen ohne Umstände austauschen. Dabei gelten folgende Regeln:

- Pro Arbeitsmappe ist nur ein Datenmodell erlaubt.
- Zwischen den Tabellen eines Datenmodells lassen sich mehrfache Verbindungen herstellen. Das Limit liegt aber bei 5.

Dieses Kapitel bietet Ihnen wenigstens einen Einstieg in diese komplexe Materie. Um Ihnen den Zugang zu erleichtern, verwenden wir ein leicht nachvollziehbares Beispiel mit zwei Tabellen. Die erste Tabelle enthält nur ein paar Daten zu den Regionen.

	A	B	C
1	**Region**	**Bezeichnung**	**Niederlassung**
2	1	Ost	Berlin
3	2	Nord	Bremen
4	3	Süd	Augsburg
5	4	West	Bonn

Abbildung 16.1 Tabelle zu den Regionen

Die zweite Tabelle enthält Umsatz- und Absatzdaten von Artikeln für den Hausbedarf, die nach Regionen aufgegliedert sind. Die Beziehung der beiden Tabellen soll über einen einfachen Schlüssel für die jeweilige Region hergestellt werden.

	A	B	C	D	E	F	G	H	I
1									
2	**Verkäufe in den Regionen**								
3									
4	**Region**	**Artnr**	**Bezeichnung**	**Warengruppe**	**Material**	**Farbe**	**Absatz**	**Umsatz_lfd_Jahr**	**Umsatz_Vorjahr**
5	1	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1232	26181	19787
6	2	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1120	23801	17988
7	3	7777	Jalousie Cc	Jalousie	Kunststoff	grau	1064	22611	17089
8	4	7777	Jalousie Cc	Jalousie	Kunststoff	grau	896	19041	14390
9	1	7778	Jalousie Ccs	Jalousie	Metall	rot	1463	63360	51480
10	2	7778	Jalousie Ccs	Jalousie	Metall	rot	1330	57600	46800
11	3	7778	Jalousie Ccs	Jalousie	Metall	rot	1264	54720	44460
12	4	7778	Jalousie Ccs	Jalousie	Metall	rot	1064	46080	37440
13	1	7578	Jalousie Ccw	Jalousie	Metall	rot	1353	62810	39930
14	2	7578	Jalousie Ccw	Jalousie	Metall	rot	1230	57100	36300
15	3	7578	Jalousie Ccw	Jalousie	Metall	rot	1169	54245	34485
16	4	7578	Jalousie Ccw	Jalousie	Metall	rot	984	45680	29040
17	1	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1243	25631	19237
18	2	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1130	23301	17488
19	3	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	1074	22136	16614
20	4	7577	Jalousie Ccx	Jalousie	Kunststoff	grau	904	18641	13990
21	1	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	770	12650	21450
22	2	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	700	11500	19500
23	3	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	665	10925	18525
24	4	6378	Jalousie Louise	Jalousie	Naturfaser	silbergrau	560	9200	15600

Abbildung 16.2 Tabelle über Verkäufe in verschiedenen Regionen

In der Tabelle mit den Verkaufsdaten sind ganz unterschiedliche Dimensionen miteinander verknüpft. Zum einen wird die zeitliche Entwicklung der Umsätze durch den Vergleich mit dem Vorjahr dargestellt, zum anderen die räumliche Verteilung nach Regionen. Die dritte Dimension ist die Verteilung der Umsätze nach Warengruppen, also eine sachliche Ordnung.

In der abgebildeten Fassung ist die Liste nicht besonders aussagekräftig. Sie können zwar schnell nachsehen, wie viel Umsatz ein Artikel in einer Region gebracht hat, aber welchen Anteil die entsprechende Warengruppe am Gesamtumsatz in einer bestimmten Region hat, ist nicht auf einen Blick zu sehen. Ob sich die Region Ost gegenüber dem Vorjahr besser entwickelt hat als die Region West, ist ebenfalls nicht gleich zu erkennen. Dennoch ist in der Liste alles enthalten, um diese Fragen zu beantworten. Es kommt nur darauf an, die Rohdaten zu brauchbaren Informationen aufzubereiten.

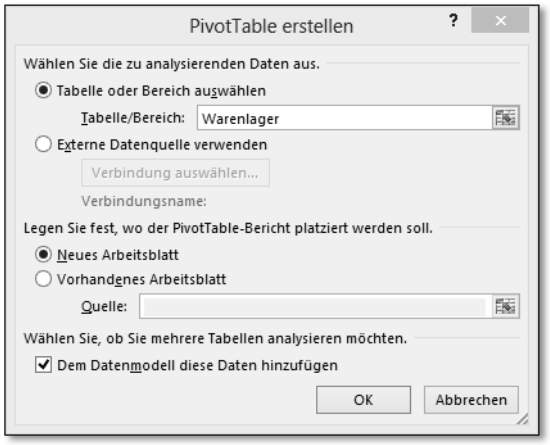
Anders als bei der Arbeit mit explizit eingegebenen Formeln übernimmt Excel die für die Beantwortung solcher Fragen notwendige Rechenarbeit weitgehend automatisch. Sie haben aber auch die Möglichkeit, detailliert festzulegen, wie Excel Daten zusammenfassen soll.

16.1 Eine Pivot-Tabelle anlegen

Im Unterschied zu den bisher beschriebenen Tabellen handelt es sich bei der Pivot-Tabelle um eine »interaktive« Tabelle. *Pivot* ist das englische Wort für einen Drehpunkt, wie die Angel in der Tür. Damit wird angesprochen, dass in der Pivot-Tabelle die Daten nach wechselnden Gesichtspunkten an- und umgeordnet, zusammengefasst und ausgewertet werden können.

Bedienungstechnisch wird das dadurch erreicht, dass für die Spaltennamen aus der Tabelle Felder erzeugt werden, die zwischen den verschiedenen Bereichen der Pivot-Tabelle hin- und hergeschoben werden können, wobei die zugehörigen Daten jeweils mitwandern. Die Felder und ihre zugehörigen Daten sind also intern miteinander verknüpft. Je nach Anordnung werden automatisch zusammenfassende Berechnungen erzeugt, die jedes Mal der Gruppierung der Daten angepasst werden. Die folgenden Schritte genügen, um eine Pivot-Tabelle einzurichten:

1 Der Dialog für die Erstellung einer Pivot-Tabelle wird über das Register **Einfügen** mit dem Symbol **PivotTable** in der Gruppe **Tabellen** aufgerufen.

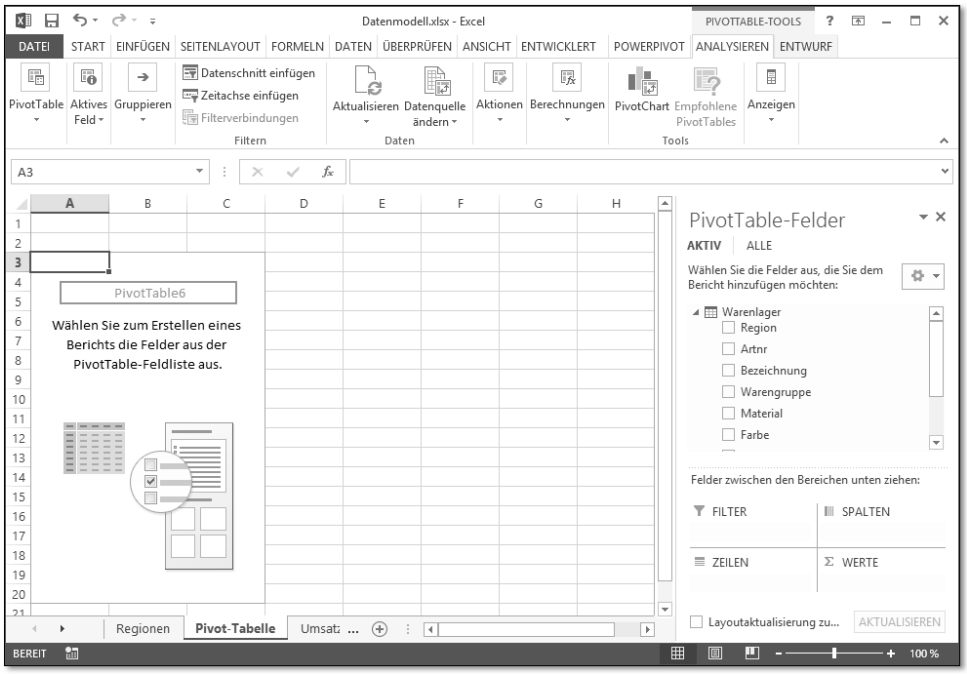


2 Zunächst wird dort die Herkunft der Daten angegeben, die auszuwerten sind. Steht der Zellzeiger in dem Datenblock, der ausgewertet werden soll, zeigt Excel unter **Tabelle/Bereich** den gefundenen Bereich an, der bei Bedarf korrigiert werden kann. Bei normalen Zellbereichen müssen Sie darauf achten, dass die Zeile mit den Spaltenüberschriften mit in den Bereich eingeschlossen wird. Handelt es sich um eine Excel-Tabelle, wird der für die Tabelle vergebene Name angeboten.

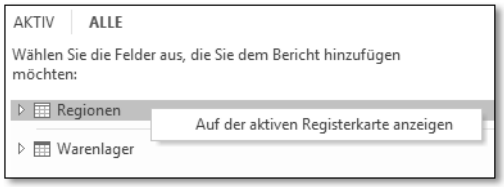
3 Im unteren Teil des Dialogs legen Sie fest, wo die Pivot-Tabelle eingefügt werden soll. Sie kann auf einem neuen Blatt angelegt werden oder an einer bestimmten Stelle des aktiven Blatts beginnen. Im zweiten Fall geben Sie unter **Quelle** die Adresse der oberen Eckzelle für die Pivot-Tabelle auf diesem Blatt an oder markieren sie. Beachten Sie, dass die erzeugte Tabelle alle bereits vorhandenen Daten in dem angegebenen Bereich überschreibt. Es ist deshalb in der Regel günstiger, ein eigenes Blatt in der Arbeitsmappe zu verwenden.

4 Wenn die Pivot-Tabelle die Daten von mehr als einer Tabelle auswerten soll, müssen Sie noch die letzte Option aktivieren: **Dem Datenmodell diese Daten hinzufügen**.

5 Wird der Dialog bestätigt, erstellt Excel einen zunächst leeren Pivot-Tabellenbereich und blendet den Aufgabenbereich **PivotTable-Felder** ein. Außerdem werden die **PivotTable-Tools** aktiviert, und das Register **Analysieren** wird eingeblendet, das gleich neun Gruppen von Befehlen und Schaltflächen zur Arbeit mit Pivot-Tabellen anbietet. Außerdem stehen nun auf dem Register **Entwurf** drei Gruppen zum Layout und Format der Pivot-Tabelle zur Verfügung. Es ist sinnvoll, zunächst den vorgegebenen Namen für die Pivot-Tabelle durch einen aussagekräftigeren zu ersetzen. Dies geschieht in der Gruppe **PivotTable** unter **PivotTable-Name**.



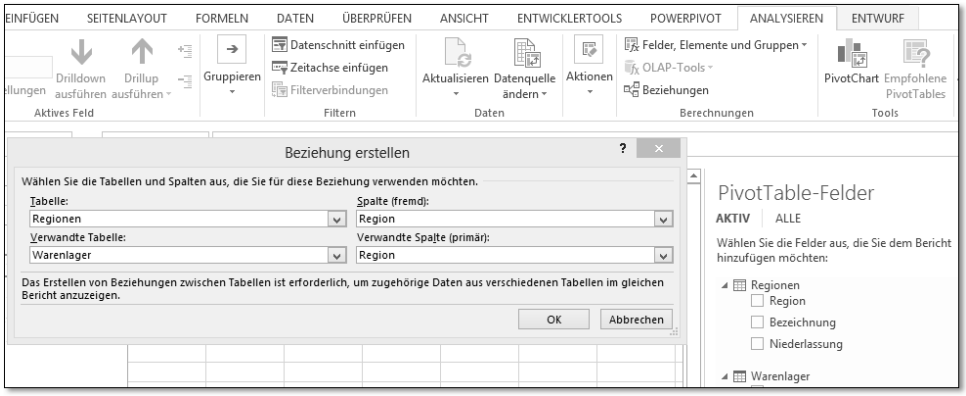
6 Anders als bei einer Pivot-Tabelle, die nur die Daten einer Tabelle auswertet, erscheinen in diesem Fall unter der Überschrift des Aufgabenbereichs die beiden Links **Aktiv** und **Alle**. Auf der Seite zu **Aktiv** erscheinen zunächst aber nur die Felder der Tabelle mit den Verkaufsdaten. Um die Felder der zweiten Tabelle zu den Regionen mit in das Modell einzubeziehen, benutzen Sie zunächst den Link **Alle** und finden dort auch die zweite Tabelle zu den Regionen.



7 Benutzen Sie im Kontextmenü zu dem Symbol **Regionen** die Option **Auf der aktiven Registerkarte anzeigen**, werden die Felder mit den Daten über die Regionen auf der Seite **Aktiv** für die Verteilung auf die verschiedenen Bereiche der Pivot-Tabelle angeboten.

8 Nun muss noch die Beziehung zwischen den beiden Tabellen geklärt werden. Benutzen Sie dazu auf dem Register **Analysieren** in der Gruppe **Berechnungen** die Schaltfläche **Beziehungen**.

9 Sie öffnet einen Dialog, in dem Sie zunächst die Schaltfläche **Neu** benutzen, da ja eine neue Beziehung eingerichtet werden soll.



10 Im Dialog **Beziehung erstellen** wählen Sie in dem ersten Listenfeld zu **Tabelle** die Tabelle zu den Regionen aus. Unter **Spalte (fremd)** geben Sie das Feld **Region** als Schlüsselfeld an.

11 Unter **Verwandte Tabelle** wählen Sie die Tabelle mit den Verkaufsdaten aus und unter **Verwandte Spalte (primär)** ebenfalls das Feld **Region**.

12 Die neue Beziehung wird anschließend in dem vorherigen Dialog angezeigt, den Sie nun schließen können.

Nun stehen alle Felder des Datenmodells zur Verfügung, um verschiedene Auswertungen auf der Basis der Daten beider Tabellen einzurichten.

16.2 Layout der Pivot-Tabelle

Zunächst sollte geklärt werden, unter welchem Gesichtspunkt die Daten hauptsächlich zu betrachten sind. In der neuen Tabelle sollen die Daten so angeordnet und zusammengefasst werden, dass erkennbar wird, wie sich der Anteil der verschiedenen Warengruppen im Lauf der beiden Jahre entwickelt hat, und zwar sowohl insgesamt als auch in den vier Regionen.

Excel bietet im Aufgabenbereich **PivotTable-Felder** unter den Tabellennamen für jede Spalte ein auswählbares Element an. Diese Elemente können einem der vier Bereiche der Pivot-Tabelle zugewiesen werden:

1. Die Felder, die die Ordnung der Seiten des Berichts bestimmen, werden dem Bereich **Filter** zugewiesen.
2. Die Felder, die die Ordnung der Spalten bestimmen, gehören in den Bereich **Spalten**, und zwar in der Reihenfolge von links nach rechts.
3. Die Felder, die die Ordnung der Zeilen bestimmen, gehören untereinander in den Bereich **Zeilen**.
4. Unbedingt muss wenigstens eines der Felder in den Bereich **Werte** eingefügt werden, damit das Programm erkennt, welche Werte in der Tabelle auszugeben sind.

Wird ein Feld in der Feldliste per Abhaken ausgewählt, wird es zunächst einem vorgegebenen Bereich zugeordnet. Diese Zuordnung lässt sich aber leicht durch Ziehen der Schaltflächen in einen anderen Bereich oder über die Kontextmenüs der Felder ändern.

Für eine erste Auswertung sollen hier aus der Tabelle **Regionen** lediglich das Feld **Bezeichnung** in den Bereich **Spalten** übernommen werden, aus der Tabelle **Warenlager** das Feld **Warengruppe** in den Bereich **Zeilen** sowie die Felder **Umsatz_Vorjahr** und **Umsatz_lfd_Jahr** in den Bereich **Werte**. Die Reihenfolge ist hier wichtig, wenn das Vorjahr links von dem laufenden Jahr erscheinen soll.

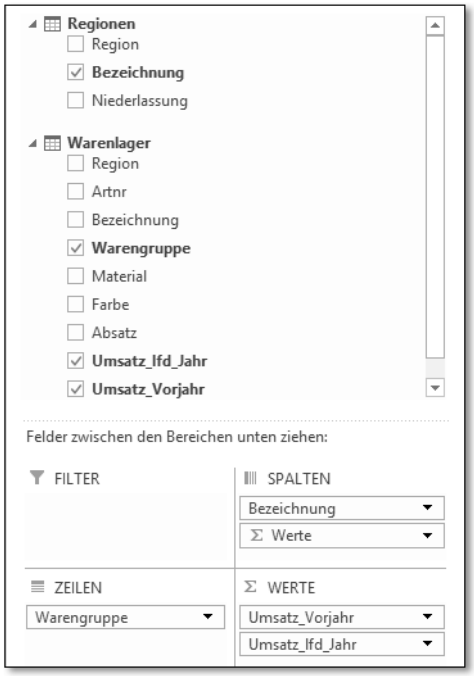


Abbildung 16.3 Feldaufteilung für eine erste Auswertung

Wählen Sie auf dem Register **Entwurf** unter **Berichtslayout** noch die Option **In Tabellenformat anzeigen**, damit die Feldnamen für die Beschriftung der ersten Zeile und der ersten Spalte übernommen werden. Außerdem können Sie die etwas umständlich klingenden vorgegebenen Spaltenüberschriften – »Summe von ...« – für die **Werte**-Felder über den Befehl **Feldeinstellungen** auf dem Register **Analysieren** ändern.

Über das Register **Entwurf** können Sie noch ein passendes **PivotTable-Format** zuweisen und die Breite der Spalten und die Ausrichtung der Beschriftungen bei Bedarf anpassen. Abbildung 16.4 zeigt den leicht überarbeiteten Bericht.

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3		Region	Werte								
4		Nord		Ost		Süd		West		Gesamt: Umsatz Vorjahr	Gesamt: Umsatz lfd. Jahr
5	Warengruppe	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr		
6	Jalousie	334976	397182	368474	436900	318227	377323	267981	317746	1289658	1529151
7	Markise	23500	45500	25850	50050	22325	43225	18800	36400	90475	175175
8	Rollo	304390	314500	334829	345950	289171	298775	243512	251600	1171902	1210825
9	**Gesamtergebnis**	**662866**	**757182**	**729153**	**832900**	**629723**	**719323**	**530293**	**605746**	**2552034**	**2915151**

Abbildung 16.4 Erste Auswertung mit Daten aus beiden Tabellen

Die vergebenen Bezeichnungen für die **Werte**-Felder wie »Summe von Umsatz Vorjahr« lassen erkennen, welche Art der Zusammenfassung der Daten die Pivot-Funktion vorgenommen hat. Nehmen Sie die erste Zelle im **Werte**-Bereich. Sie ist der Schnittpunkt zwischen Region **Nord** und der Warengruppe **Jalousie**, was auch bestätigt wird, wenn Sie mit der Maus die Zelle berühren. In dieser Zelle erscheint also der summierte Umsatz der Region **Nord** in der betreffenden Warengruppe. Excel berechnet als Vorgabe zusätzlich auch das Gesamtergebnis pro Region und pro Warengruppe.

Excel nimmt hier also Zusammenfassungen von Daten vor, die Sie normalerweise mit Hilfe der Funktion `SUMME()` vornehmen würden. Solange Excel Felder zusammenfassen soll, die Zahlen enthalten, wird das Programm als Vorgabe eine Summe bilden. Enthalten die Felder dagegen Texte, wird eine Zusammenfassung im Bereich **Werte** als Vorgabe mit der Funktion `ANZAHL()` operieren, also zählen, wie oft die betreffenden Elemente vorkommen.

16.3 Elementfilter und Datenschnitte

Zu den beiden Feldnamen, die als Ordnungskriterien dienen, **Region** und **Warengruppe**, erscheinen Filterschaltflächen, wie Sie sie verwenden können, um vorübergehend bestimmte Elemente auszublenden.

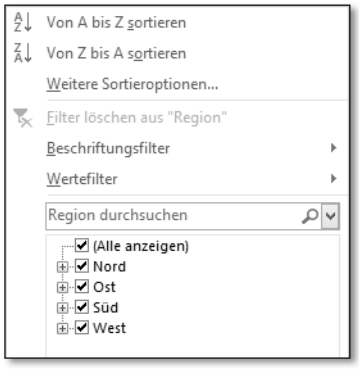
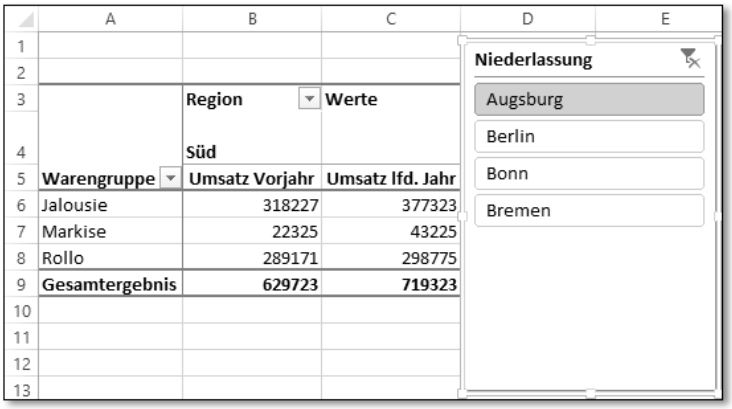


Abbildung 16.5 Filtermenü zu dem Feld **Region**

Jede Schaltfläche bietet ein umfangreiches Filtermenü mit der Liste der jeweiligen Elemente. Löschen Sie beispielsweise das Häkchen bei einer Region, werden diese Daten aus der Tabelle ausgeblendet.

Noch praktischer ist das Filtern, wenn Sie auf dem Register **Analysieren** die Schaltfläche **Datenschnitt einfügen** in der Gruppe **Filtern** verwenden und beispielsweise das Feld **Nie-**

derlassung auswählen. Excel blendet ein grafisches Element mit auswählbaren Schaltflächen für jede Niederlassung ein. Per Klick oder Tipp filtern Sie so die Daten für die einzelnen Regionen heraus.



	A	B	C
1			
2			
3		Region	Werte
4		Süd	
5	Warengruppe	Umsatz Vorjahr	Umsatz lfd. Jahr
6	Jalousie	318227	377323
7	Markise	22325	43225
8	Rollo	289171	298775
9	Gesamtergebnis	629723	719323
10			
11			
12			
13			

Abbildung 16.6 Datenschnitt-Element zur Auswahl einzelner Regionen über die Niederlassung

16.4 Ändern der Berechnungsart

Sie haben die Möglichkeit, auch andere Berechnungsmethoden zur Zusammenfassung der Daten zu verwenden. Eine solche Änderung gilt jeweils für ein bestimmtes Wertfeld. Wählen Sie dazu ein Wertfeld aus, und verwenden Sie **Analysieren • Aktives Feld • Feldeinstellungen**. Excel öffnet den Dialog **Wertfeldeinstellungen**.

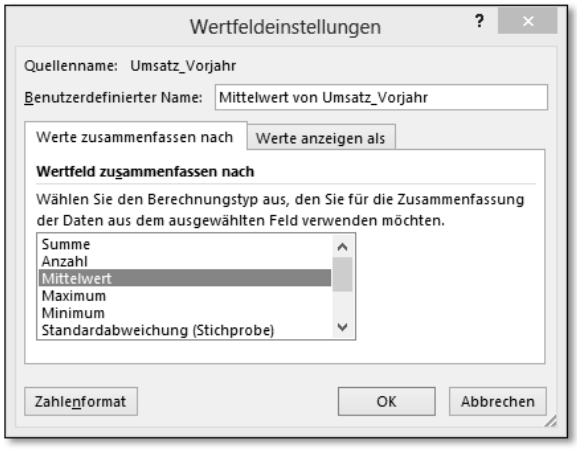


Abbildung 16.7 Auswahl einer anderen Funktion für die Zusammenfassung der Daten

Auf dem Register **Werte zusammenfassen nach** wählen Sie die Funktion aus, die zur Berechnung verwendet werden soll. Sie finden hier die statistischen Funktionen, die auch der Befehl **Teilergebnisse** anbietet. Statt der Summe können Sie beispielsweise den Mittelwert der Verkäufe in einer Warengruppe anzeigen lassen.

Über die Schaltfläche **Zahlenformat** haben Sie außerdem die Möglichkeit, die Werte im Datenbereich zu formatieren. Excel übernimmt zunächst immer das Format der Quelldaten, es kann aber gut sein, dass dieses Format für die zusammenfassenden Ergebnisse nicht günstig ist, beispielsweise wenn die Nachkommastellen nicht benötigt werden.

Interessante Möglichkeiten bietet hier auch das zweite Register **Werte anzeigen als**. Wenn Sie z. B. anstelle der Summenwerte jeweils die Prozentanteile der zusammengefassten Werte sehen wollen, können Sie unter **Werte anzeigen als** den Punkt **% des Gesamtergebnisses** wählen. Auch Differenzbildungen zu anderen Spalten sind möglich. Dafür muss dann unter **Basisfeld** bzw. **Basiselement** der passende Eintrag ausgewählt werden.

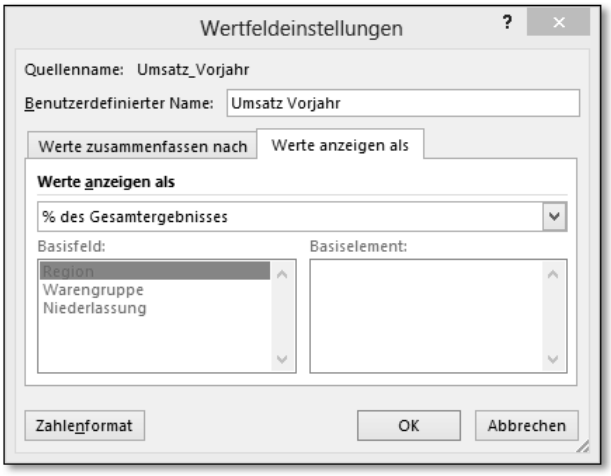


Abbildung 16.8 Wahl der Prozentanzeige in Bezug auf das Gesamtergebnis

Abbildung 16.9 zeigt eine Tabelle, die die Einzelwerte als Prozentanteile des Gesamtergebnisses pro Jahr ausgibt.

Auch Differenzbildungen zu anderen Spalten sind möglich. Um beispielsweise die Differenz der anderen Regionen zur Region **West** zu berechnen, wählen Sie aus der Liste die Option **% Differenz von**, dann unter **Basisfeld** das Feld **Region** und unter **Basiselement** eben **West** aus.

	A	B	C	D	E	F	G	H	I	J	K
3		Region	Werte								
4		Nord		Ost		Süd		West		Gesamt: Umsatz Vorjahr	Gesamt: Umsatz lfd. Jahr
5	Warengruppe	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr	Umsatz Vorjahr	Umsatz lfd. Jahr		
6	Jalousie	13,13%	13,62%	14,44%	14,99%	12,47%	12,94%	10,50%	10,90%	50,53%	52,46%
7	Markise	0,92%	1,56%	1,01%	1,72%	0,87%	1,48%	0,74%	1,25%	3,55%	6,01%
8	Rollo	11,93%	10,79%	13,12%	11,87%	11,33%	10,25%	9,54%	8,63%	45,92%	41,54%
9	**Gesamtergebnis**	**25,97%**	**25,97%**	**28,57%**	**28,57%**	**24,68%**	**24,68%**	**20,78%**	**20,78%**	**100,00%**	**100,00%**

Abbildung 16.9 Tabelle mit Prozentanteilen

16.5 Auswertungen mit Pivot-Diagrammen

Schon seit Excel 2002 haben Sie auch bei den Pivot-Diagrammen die interaktiven Möglichkeiten, Daten neu zusammenzufassen oder zu gruppieren. Excel 2013 bietet dafür auf dem Register **Einfügen** nun eine eigene Schaltfläche **PivotChart**.

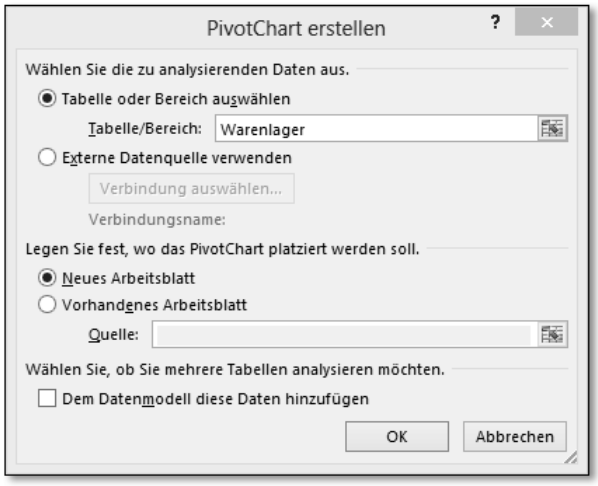


Abbildung 16.10 Dialog für PivotChart

Der aufgerufene Dialog entspricht – abgesehen von der Titelzeile – dem, der auch für die Erstellung einer Pivot-Tabelle angeboten wird. Wenn Sie den Dialog quittieren, wird ein leerer Diagrammbereich angelegt, und im Aufgabenbereich erscheint nun die Liste der **PivotChart-Felder**.

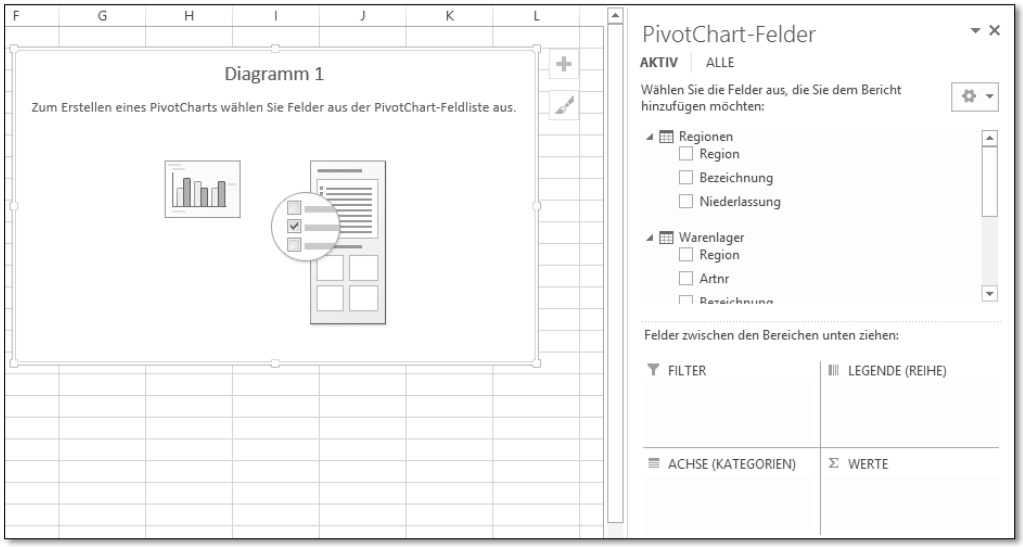


Abbildung 16.11 Der noch leere Bereich für ein Pivot-Diagramm und die Liste der Felder

Die Namen für die Bereiche und die Zuordnungsoptionen im Kontextmenü eines Felds sind nun etwas anders benannt. Anstelle des Bereichs **Zeilen** wird der Bereich **Achse (Kategorien)** angeboten, anstelle des Bereichs **Spalten** der Bereich **Legende (Reihen)**.

Besteht bereits eine Pivot-Tabelle, können Sie über die Schaltfläche **PivotChart** auch nachträglich ein Pivot-Diagramm hinzufügen. In diesem Fall öffnet der Befehl sofort den Dialog zur Auswahl des Diagrammtyps und übernimmt dabei die Feldauswahl, die für die Pivot-Tabelle vorgenommen wurde.

Um die Verkaufsdaten als Diagramm auszuwerten, kann nun beispielsweise das Feld **Warengruppe** als Legendenfeld verwendet werden, das Feld **Bezeichnung** aus der Tabelle **Regionen** als Achsenfeld. Das Feld **Umsatz_lfd_Jahr** wird dem Wertebereich zugeordnet. Um die Handhabung zu vereinfachen, werden im Diagramm noch Filterschaltflächen eingeblendet, über deren Listenfelder Elemente aus dem Diagramm herausgenommen werden können, etwa eine bestimmte Warengruppe.

Falls die Filterschaltflächen nicht sichtbar sind, benutzen Sie auf dem Register **Analysieren** in der Gruppe **Einblenden/Ausblenden** das Menü der Schaltfläche **Feldschaltflächen**. Die Feldschaltflächen lassen sich einzeln aus- und einblenden.

Das Ergebnis ist als Vorgabe ein Säulendiagramm, in dem die Gesamtumsätze pro Region nach Warengruppen unterschieden werden. Durch Ziehen der Felder zwischen den Bereichen im Aufgabenbereich lässt sich das Diagramm beliebig umbauen, wobei eine mit dem Diagramm verknüpfte Tabelle automatisch angepasst wird.

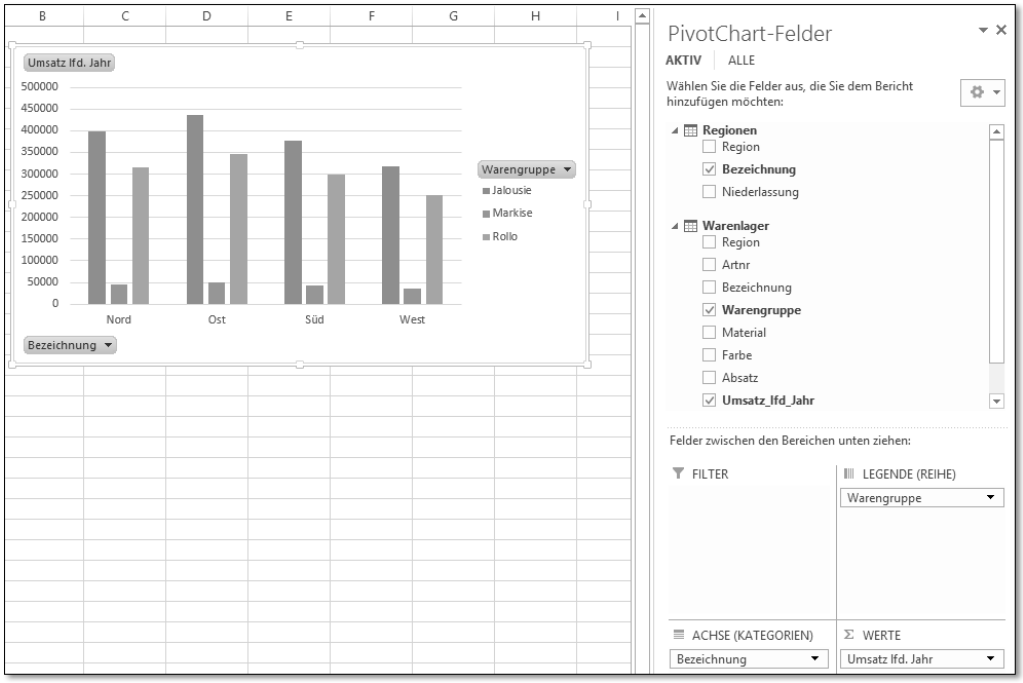


Abbildung 16.12 Das Diagramm mit den Filterschaltflächen

Ist das Diagramm ausgewählt, können Sie den Diagrammtyp jederzeit ändern. Für das Pivot-Diagramm gelten weitgehend die gleichen Bearbeitungsmöglichkeiten wie für die »normalen« Diagramme, nur das Symbol **Diagrammfilter** am rechten Rand fehlt, weil ja bereits Filterschaltflächen innerhalb des Diagramms angeboten werden.

16.6 Anwenden der Cube-Funktionen bei Datenmodellen

In Kapitel 11, »Cube-Funktionen«, ist eingangs angesprochen worden, dass sich in Excel 2013 Cube-Funktionen auch für interne Datenmodelle anwenden lassen. Das Argument `Verbindung` heißt in diesem Fall einfach `ThisWorkbookDataModel`. In Bezug auf das hier verwendete Beispiel legt die Formel

```
=CUBEMENGE("ThisWorkbookDataModel";"[Warenlager].[Warengruppe].
[All].children";"Warengruppen")
```

beispielsweise in Zelle A30 die Anzahl der Warengruppen ab, wobei in der Zelle aber nur die Beschriftung zu sehen ist, die als letztes Argument angegeben ist. Eine zweite

Funktion, die sich auf die Zelle mit der ersten Funktion bezieht, gibt dann die Menge der Gruppen aus:

```
=CUBEMENGENANZAHL(A30)
```

Soll eine der in der Pivot-Tabelle berechneten Gesamtsummen angezeigt werden, verwenden Sie beispielsweise:

```
=CUBEWERT("ThisWorkbookDataModel";"[Measures].[Summe von Umsatz_Vorjahr]")
```

Das liefert die Gesamtsumme aller Verkäufe im Vorjahr.

17 Zusätzliche Tools für die Datenanalyse

Für die statistische Analyse von Daten in einer Tabelle stehen nicht nur die zahlreichen statistischen Tabellenfunktionen zur Verfügung, die in Kapitel 7, »Statistische Funktionen«, beschrieben sind. Zusätzlich kann ein umfangreiches Analysewerkzeug eingesetzt werden, das die statistische Auswertung vorgegebener Daten vereinfacht.

17.1 Aktivieren der Analyse-Funktionen

Voraussetzung für die Nutzung der zusätzlichen Funktionen ist, dass das Add-In **Analyse-Funktionen** installiert und in die Liste der Zusatzprogramme, die Excel bereitstellt, eingefügt ist, wie es bereits in Kapitel 1, »Einstieg in Berechnungen mit Excel«, beschrieben wurde. Neben den zusätzlichen Funktionen, die durch dieses Add-In direkt im Tabellenblatt zur Verfügung stehen, erscheint seit Excel 2007 auf dem Register **Daten** eine neue Gruppe **Analyse** mit der Schaltfläche **Datenanalyse**.

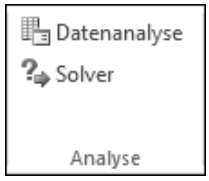


Abbildung 17.1 Die Schaltfläche für die Analyse-Funktionen in Excel 2013

Die Schaltfläche öffnet den Dialog **Analyse-Funktionen**, der die verschiedenen Tools über ein Listenfeld anbietet.

In den älteren Excel-Versionen wird das Add-In über den Befehl **Extras • Add-Ins** aktiviert, der den Dialog **Add-Ins** öffnet. Sind die Analyse-Funktionen aktiviert, wird der Dialog **Analyse-Funktionen** über den gleichnamigen Befehl im Menü **Extras** eingeblendet, der dann zur Verfügung steht. Bis auf die Tatsache, dass seit Excel 2007 die ersten drei Funktionen jetzt als *Anova-Funktionen* kategorisiert sind, ist die Liste der im Dialog angebotenen Tools gegenüber den älteren Versionen unverändert geblieben. Anova ist ein Akronym für **An**alysis **o**f **Va**riance.

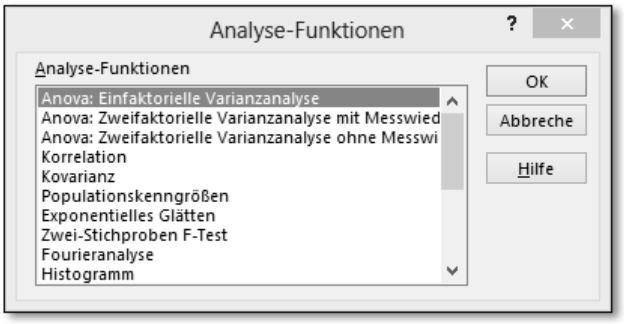


Abbildung 17.2 Der Dialog bietet die Analyse-Funktionen als Liste an.

Bevor Sie eine der hier angebotenen Funktionen verwenden können, müssen die Daten, die analysiert werden sollen, in einem Tabellenblatt so angeordnet werden, dass die gewählte Analyse-Funktion damit arbeiten kann. Wie die Daten für eine bestimmte Funktion vorliegen müssen, hängt von der jeweiligen Funktion ab. In dem entsprechenden Dialog werden jedes Mal mindestens ein Eingabebereich und ein Ort für die Ausgabe abgefragt, wobei die Ausgabe auch auf ein neues Tabellenblatt oder gar in eine eigene Arbeitsmappe gelegt werden kann. Für andere Einstellungen werden Vorgaben gemacht, die geändert werden können.

Wird der Dialog bestätigt, erscheinen im für die Ausgabe bestimmten Bereich jeweils die errechneten Werte mit entsprechenden Beschriftungen. Es werden also keine Formeln im Arbeitsblatt angelegt. Werden Werte im Eingabebereich nachträglich geändert, muss die Funktion neu aufgerufen werden, um die entsprechenden Ergebnisse neu zu erzeugen. Um die Neuberechnung zu vereinfachen, bleibt der Eingabebereich aber voreingestellt. Wird der Ausgabebereich für die erneute Berechnung nicht geändert, muss bestätigt werden, dass der bisherige Bereich überschrieben werden darf. Dabei gehen allerdings eventuelle Formatierungen des Ausgabebereichs, die zwischenzeitlich vorgenommen wurden, wieder verloren. Es wird deshalb häufig sinnvoll sein, vor dem Überschreiben Kopien anzulegen, um den älteren Zustand zu behalten. Bei einigen Funktionen werden zusätzlich zu den ausgegebenen Werten gleich auch grafische Darstellungen generiert.

In den folgenden Abschnitten werden die Tools an kleinen Beispielen vorgestellt. Die erste Gruppe der Tools, die wie gesagt seit Excel 2007 jetzt alle als Anova-Analysetools bezeichnet werden, erlaubt verschiedene Varianten der Varianzanalyse. Welches Tool brauchbar ist, hängt dabei von der Anzahl der beobachteten Faktoren und der Anzahl der Stichproben aus entsprechenden Grundgesamtheiten ab, die in dem jeweiligen statistischen Test ins Spiel gebracht werden.

17.2 Anova-Varianzanalyse mit einem Faktor

Solange immer nur zwei voneinander unabhängige Stichproben in Bezug auf ihre Streuungsmerkmale untersucht werden, ist die `F.TEST()`-Funktion ein praktikabler Weg. Schwieriger wird es, wenn die Zahl der zu vergleichenden Stichproben ansteigt. Normalerweise müssten dann jeweils paarweise Vergleiche zwischen den Stichproben ausgeführt werden, was mit steigender Zahl der Stichproben aufwendig wird.

Hier hat sich nun ein Verfahren durchgesetzt, das als einfache Varianzanalyse bezeichnet wird. Das Verfahren soll an einem hypothetischen Beispiel beschrieben werden. In der Tabelle in Abbildung 17.3 sind die Ergebnisse dreier Zufallsstichproben zusammengestellt. In drei Städten wird eine unterschiedliche Zahl von Haushalten nach der Zimmeranzahl ihrer Wohnungen befragt. Zur ersten Auswertung sind die arithmetischen Mittelwerte eingetragen.

	A	B	C	D
1				
2	**Anova: Einfaktorielle Varianzanalyse**			
3				
4	Anzahl Zimmer pro Haushalt	Berlin	Hamburg	Essen
5		3	4	3
6		2	3	2
7		4	5	5
8		1	2	2
9		2	3	1
10		2	2	4
11		3	1	3
12		5	2	5
13		2	2	2
14		3	3	3
15			4	2
16			3	1
17			1	
18			2	
19	Mittelwert:	2,7	2,6	2,8

Abbildung 17.3 Der Eingabebereich für die Varianzanalyse

Die Varianzanalyse soll nun prüfen, ob behauptet werden kann, dass alle diese Stichproben aus einer einzigen Grundgesamtheit stammen können und deshalb auch in der zugrunde liegenden Wahrscheinlichkeitsverteilung übereinstimmen. Oder belegen die Daten die entgegengesetzte Vermutung, dass die Stichproben für ganz unterschiedliche Wahrscheinlichkeitsverteilungen sprechen? Die zu prüfende Nullhypothese kann also so formuliert werden: Alle Stichproben stammen aus der gleichen Grundgesamtheit, und die beobachteten Mittelwertunterschiede sind rein zufällig und statistisch nicht signifikant.

Das Verfahren, um die Nullhypothese zu prüfen, verwendet eine Quadratsummenzerlegung. Dabei wird die Summe der quadrierten Abweichungen der Merkmalswerte vom Gesamtmittelwert in zwei Teile zerlegt. Das eine – `q1` – ist die Summe der quadrierten und jeweils mit dem Stichprobenumfang gewichteten Abweichungen der Stichprobenmittelwerte vom Gesamtmittelwert, das andere – `q2` – ist die Summe der quadrierten Abweichungen der einzelnen Merkmalswerte vom jeweiligen Stichprobenmittelwert. Die erste Summe ist ein Maß für die Streuung zwischen den Stichproben, das zweite ein Maß für die Streuung innerhalb der Stichproben. Mit der Formel

```
q = q1 + q2
```

wird dann ein Wert ermittelt, der als Maß der Varianz aller Werte interpretiert werden kann.

Sind die Daten wie in Abbildung 17.3 vorbereitet, kann das Analysetool eingesetzt werden. Im Dialog **Analyse-Funktionen** wird dazu die Option **Anova: Einfaktorielle Varianzanalyse** gewählt. Anschließend werden die Angaben zum **Eingabebereich** eingetragen. In diesem Fall ist es der Bereich B4:D18, die Spaltenbeschriftungen werden mit eingeschlossen, wenn gleichzeitig die Option **Beschriftungen in erster Zeile** aktiviert ist. Unter **Geordnet nach:** wird festgelegt, ob die Stichprobenwerte jeweils in einer Spalte oder Zeile angeordnet sind. In dem Textfeld **Alpha:** wird das Signifikanzniveau angegeben, also die maximal zulässige geschätzte Irrtumswahrscheinlichkeit. Die Vorgabe 0,05 für 5% kann hier übernommen werden. Unter **Ausgabe** kann die Option **Neues Tabellenblatt** genutzt werden.

Der Ausgabebereich kann mit ein paar Formatierungsmaßnahmen etwas übersichtlicher gemacht werden, insbesondere durch Anpassung der Spaltenbreite, durch Zahlenformate mit weniger Dezimalstellen und durch einen Umbruch der Beschriftungen.

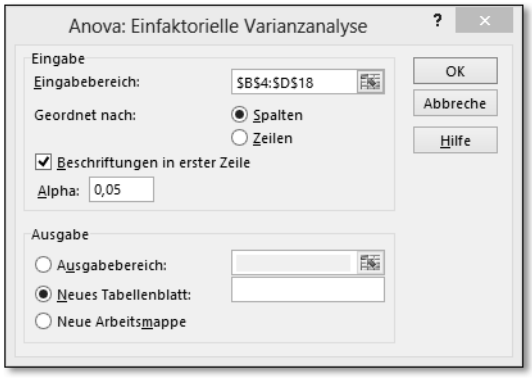


Abbildung 17.4 Der Dialog der Anova-Funktion

	A	B	C	D	E	F	G
1	**Anova: Einfaktorielle Varianzanalyse**						
2							
3	ZUSAMMENFASSUNG						
4	*Gruppen*	*Anzahl*	*Summe*	*Mittelwert*	*Varianz*		
5	Berlin	10	27	2,7000	1,3444		
6	Hamburg	14	37	2,6429	1,3242		
7	Essen	12	33	2,7500	1,8409		
8							
9							
10	ANOVA						
11	*Streuungsursache*	*Quadrat-summen (SS)*	*Freiheits-grade (df)*	*Mittlere Quadratsumme (MS)*	*Prüfgröße (F)*	*P-Wert*	*kritischer F-Wert*
12	Unterschiede zwischen den Gruppen	0,0746	2	0,0373	0,0248	0,9755	3,2849
13	Innerhalb der Gruppen	49,5643	33	1,5019			
14							
15	Gesamt	49,6389	35				

Abbildung 17.5 Der Bereich mit den berechneten Werten, zur besseren Lesbarkeit leicht formatiert

Die erste Gruppe der Auswertung gibt für jede Stichprobe einfach den Stichprobenumfang (`Anzahl`), die Summe der Werte, den arithmetischen Mittelwert und die Varianz aus, die auch die Funktion `VAR.S()` oder `VARIANZ()` liefern könnte. Wie schon erwähnt, werden in der Auswertung aber keine Formeln, sondern nur die berechneten Werte ausgegeben.

Interessanter sind die Zahlen in der unteren Tabelle. Hier finden Sie die drei genannten Quadratsummen, die die Unterschiede zwischen und innerhalb der Gruppen und den Gesamtunterschied beschreiben. Der erste Wert unter `Freiheitsgrade` ist `2`, weil von `r`, der Gesamtzahl der Stichproben, 1 abgezogen wird. Der zweite Wert wird berechnet, indem von der Gesamtzahl der Werte die Anzahl der Stichproben abgezogen wird, also: `36 – 3 = 33`. Der dritte Wert addiert die beiden Werte. In der Spalte D wird die mittlere Quadratsumme ausgegeben. Dazu wird die Quadratsumme aus Spalte B durch die Freiheitsgrade dividiert.

Das Ergebnis zeigt in diesem Fall, dass die Streuung innerhalb der Gruppen größer ist als die Streuung zwischen den Gruppen. Es spielt also nur eine geringe Rolle, in welcher Stadt die Daten über die Wohnungsgröße erhoben werden.

Die Prüfgröße (`F`) wird nun ermittelt, indem die mittlere Quadratsumme für die Gruppenunterschiede durch die mittlere Quadratsumme der Unterschiede innerhalb der Gruppen dividiert wird, also hier: `=D12/D13`.

Der ermittelte Wert in E12 kann mit der Funktion `FVERT()` geprüft werden. Der P-Wert in F12 wird deshalb mit der Formel

`=FVERT(E12;C12;C13)` oder `=F.VERT(E12;C12;C13;FALSCH)`

berechnet. Er liefert die Überschreitungswahrscheinlichkeit. (Die neuere, verbesserte Funktion liefert allerdings ein etwas abweichendes Ergebnis, die Analyse-Funktion arbeitet also noch mit der älteren Funktion.) Diese Überschreitungswahrscheinlichkeit ist größer als das Signifikanzniveau, das ja mit dem Wert `0,05` für `Alpha` angegeben wurde. Damit wird bestätigt, die Null-Hypothese, dass die Stichproben alle aus der gleichen Grundgesamtheit stammen, kann nicht verworfen werden.

Fehlt noch der kritische F-Wert, der Prüfwert eines F-Tests. Der hier angezeigte Wert wird mit der Funktion `FINV()` oder `F.INV.RE()` berechnet. Die Formel würde hier lauten:

`=FINV(0,05;C12;C13)` oder `=F.INV.RE(0,05;C12;C13)`

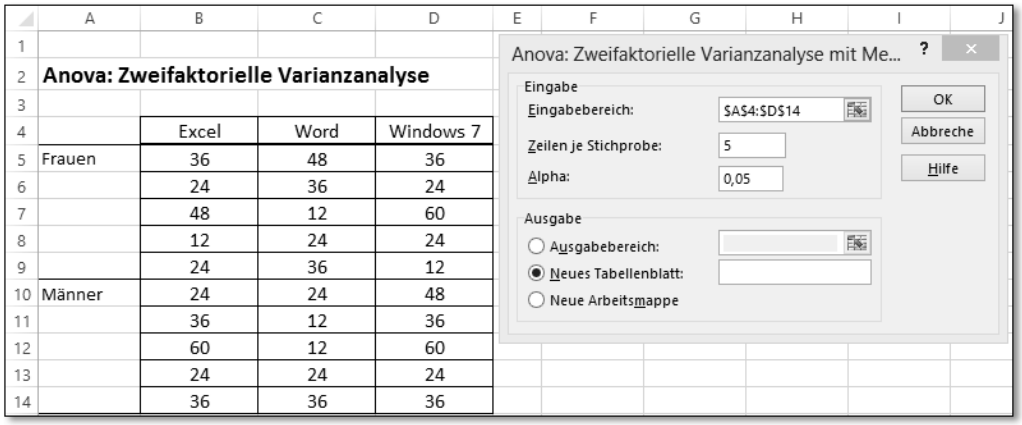
wobei `0,05` der im Dialog angegebene **Alpha**-Wert für die Irrtumswahrscheinlichkeit ist, die beiden anderen Werte sind wieder die Freiheitsgrade. Die Funktion liefert den kritischen Wert der F-Verteilung. Würde der in E12 berechnete Quotient der beiden Varianzwerte den hier errechneten kritischen Wert erreichen oder sogar überschreiten, müsste die Nullhypothese, dass die geprüften Stichproben aus der gleichen Grundgesamtheit stammen können, verworfen werden. Das ist hier offensichtlich nicht der Fall. (Zu den hier angesprochenen Funktionen vergleiche die Beschreibungen in Kapitel 7, »Statistische Funktionen«.)

17.3 Anova: Zweifaktorielle Varianzanalyse mit Messwiederholung

Im Unterschied zu der im letzten Abschnitt beschriebenen Form der Varianzanalyse versucht dieses Analysewerkzeug, gleich zwei Faktoren zu berücksichtigen, die Einfluss auf die Daten von mehreren Stichproben haben, deren Varianz verglichen werden soll. Es gibt diese Auswertung mit Messwiederholung und ohne Messwiederholung. Im ersten Fall müssen die Daten so angeordnet werden, dass immer die gleiche Anzahl von Datenzeilen zu einem Messwerteblock zusammengefasst wird.

Im Folgenden wird ein hypothetisches Beispiel gezeigt, das die Häufigkeit von Supportanfragen für Software pro Monat auflistet und dabei gleichzeitig unterscheidet, ob die Anfrage von Frauen oder Männern gestellt wird.

Abbildung 17.6 zeigt, wie bei diesem Tool im Dialog der Eingabebereich angegeben wird. Die Spalten- und Zeilenbeschriftungen werden mit übernommen. Die Zahl der Datenzeilen pro Stichprobe wird unter **Zeilen je Stichprobe** angegeben.



Anova: Zweifaktorielle Varianzanalyse

	Excel	Word	Windows 7
Frauen	36	48	36
	24	36	24
	48	12	60
	12	24	24
	24	36	12
Männer	24	24	48
	36	12	36
	60	12	60
	24	24	24
	36	36	36

Abbildung 17.6 Beispiel für zweifaktorielle Varianzanalyse

Das Ergebnis ist ähnlich aufgebaut wie bei der einfaktoriellen Analyse. In der unteren Gruppe werden aber gleich drei Prüfgrößen, P-Werte und kritische F-Werte ermittelt.

Anova: Zweifaktorielle Varianzanalyse mit Messwiederholung

ZUSAMMENFASSUNG	Excel	Word	Vista	Gesamt
Frauen				
Anzahl	5	5	5	15
Summe	144	156	156	456
Mittelwert	28,8	31,2	31,2	30,4
Varianz	187,2	187,2	331,2	202,971429
Männer				
Anzahl	5	5	5	15
Summe	180	108	204	492
Mittelwert	36	21,6	40,8	32,8
Varianz	216	100,8	187,2	215,314286
Gesamt				
Anzahl	10	10	10	
Summe	324	264	360	
Mittelwert	32,4	26,4	36	
Varianz	193,6	153,6	256	

ANOVA

Streuungsursache	*Quadrat-summen (SS)*	*Freiheits-grade (df)*	*Mittlere Quadrat-summe (MS)*	*Prüfgröße (F)*	*P-Wert*	*kritischer F-Wert*
Stichprobe	43,2	1	43,2	0,21428571	0,64759887	4,25967721
Spalten	470,4	2	235,2	1,16666667	0,32844667	3,40282611
Wechselwirkung	547,2	2	273,6	1,35714286	0,27644714	3,40282611
Fehler	4838,4	24	201,6			
Gesamt	5899,2	29				

Abbildung 17.7 Ergebnis der zweifaktoriellen Varianzanalyse

Die erste Zeile ist mit `Stichprobe` beschriftet und vergleicht die Mittelwerte der beiden Zeilengruppen. Die zweite Zeile vergleicht die Mittelwerte der Spalten. In beiden Auswertungen wird jeweils der zweite Faktor zunächst vernachlässigt. Unter `Wechselwirkung` wird geprüft, ob die Auswirkung des ersten Faktors (hier das Geschlecht) auch davon abhängt, welcher Wert für den zweiten Faktor (hier die Software) gegeben ist. Es wird also einmal geprüft, ob es signifikante Unterschiede zwischen Männern und Frauen gibt, dann, ob es ungeachtet des Geschlechts Unterschiede gibt, die mit der betreffenden Software zu tun haben, und schließlich, ob es signifikante Unterschiede zwischen Faktorenpaaren (Geschlecht + Software) gibt.

17.4 Anova: Zweifaktorielle Varianzanalyse ohne Messwiederholung

Die dritte Option für eine Anova-Analyse im Dialog **Analyse-Funktionen** ist eine vereinfachte Variante der zuletzt beschriebenen Varianzanalyse mit zwei Faktoren. Dabei wird vorausgesetzt, dass für jedes mögliche Faktorenpaar nur Daten aus einer Beobachtung vorliegen. Daraus folgt, dass über eine Wechselwirkung zwischen den Faktoren nichts ausgesagt werden kann. Das oben beschriebene Beispiel sieht in dieser Vereinfachung so aus:

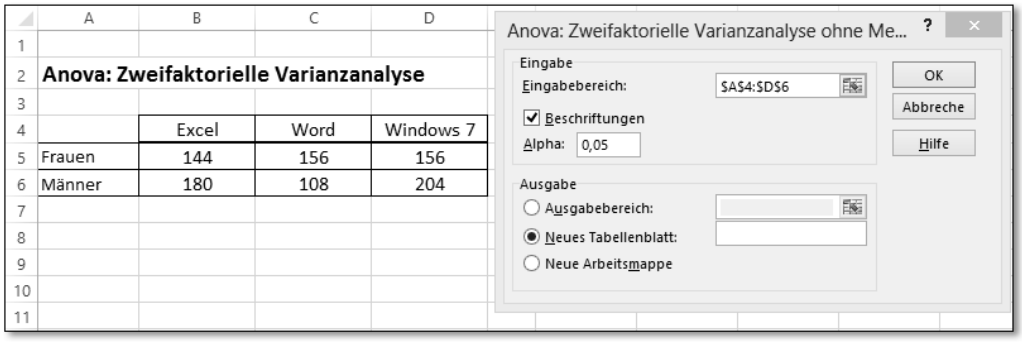


Abbildung 17.8 Die Variante ohne Messwertwiederholung

Das Ergebnis zeigt in diesem Fall nur zwei Prüfgrößen, P-Werte und kritische F-Werte.

	A	B	C	D	E	F	G
1	**Anova: Zweifaktorielle Varianzanalyse ohne Messwiederholung**						
2							
3	*ZUSAMMENFASSUNG*	*Anzahl*	*Summe*	*Mittelwert*	*Varianz*		
4	Frauen	3	456	152	48		
5	Männer	3	492	164	2496		
6							
7	Excel	2	324	162	648		
8	Word	2	264	132	1152		
9	Vista	2	360	180	1152		
10							
12	ANOVA						
13	*Streuungs-ursache*	*Quadrat-summen (SS)*	*Freiheits-grade (df)*	*Mittlere Quadrat-summe (MS)*	*Prüfgröße (F)*	*P-Wert*	*kritischer F-Wert*
14	Zeilen	216	1	216	0,15789474	0,72949911	18,5128205
15	Spalten	2352	2	1176	0,85964912	0,53773585	19
16	Zufallsfehler	2736	2	1368			
18	Gesamt	5304	5				

Abbildung 17.9 Die Auswertung bei fehlenden Messwertwiederholungen

17.5 Korrelation

Um den Korrelationskoeffizienten zwischen zwei Wertereihen zu ermitteln, kann die Funktion `KORREL()` verwendet werden, wie in Kapitel 7, »Statistische Funktionen«, beschrieben. Sollen gleichzeitig Korrelationen zwischen mehreren Datenreihen geprüft werden, bietet sich das Analysetool **Korrelation** an.

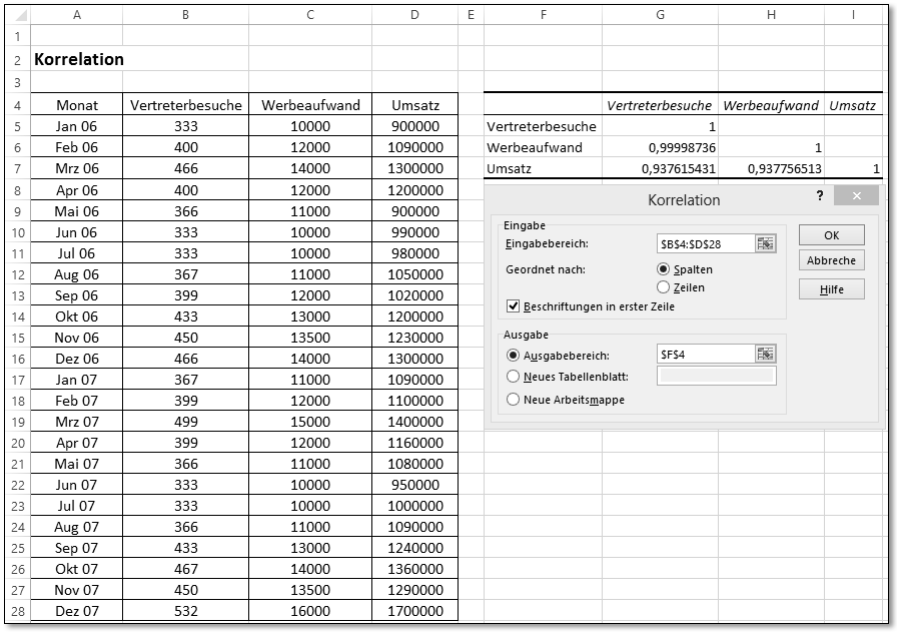


Abbildung 17.10 Ermittlung der Korrelation mehrerer Datenreihen

In dem Beispiel in Abbildung 17.10 sind zeilenweise drei Werte aufgeführt. Es ist zu erwarten, dass die Entwicklung der Umsatzwerte sowohl mit der Zahl der Vertreterbesuche pro Monat korreliert als auch mit dem zeitgleichen Werbeaufwand.

Im Dialog **Korrelation** wird der **Eingabebereich** unter Einschluss der drei Spaltenbeschriftungen angegeben. Die Auswertung liefert eine kleine Tabelle, in der die Werte, die die Funktion `KORREL()` für jedes mögliche Paar von Variablen liefert, zusammengestellt sind. Der errechnete Korrelationskoeffizient ist unabhängig von den jeweils verwendeten Maßeinheiten. Er muss zwischen `-1` und `+1` (einschließlich) liegen. In diesem Fall liegen die drei maßgeblichen Koeffizienten alle über `0,9`, was belegt, dass ein enger Zusammenhang zwischen den Werten besteht.

17.6 Kovarianz

Das Tool zur Berechnung der Kovarianz wird ähnlich eingesetzt wie das Tool zur Berechnung der Korrelation. Es liefert die Ergebnisse der Funktion `KOVAR()` bzw. der Funktion `KOVARIANZ.P()` für mehrere Datenreihen gleichzeitig. Die Ausgabetabelle zeigt für jedes Variablenpaar die Kovarianz. Da hier alle Werte positiv sind, heißt das, den hohen Werten bei Vertreterbesuchen entsprechen steigende Umsätze etc.

Kovarianz

Monat	Vertreterbesuche	Werbeaufwand	Umsatz
Jan 06	333	10000	900000
Feb 06	400	12000	1090000
Mrz 06	467	14000	1300000
Apr 06	400	12000	1200000
Mai 06	366	11000	900000
Jun 06	333	10000	990000
Jul 06	333	10000	980000
Aug 06	366	11000	1050000
Sep 06	400	12000	1020000
Okt 06	432	13000	1200000
Nov 06	450	13500	1230000
Dez 06	467	14000	1300000
Jan 07	366	11000	1090000
Feb 07	400	12000	1100000
Mrz 07	499	15000	1400000
Apr 07	400	12000	1160000
Mai 07	366	11000	1080000
Jun 07	332	10000	950000
Jul 07	333	10000	1000000
Aug 07	366	11000	1090000
Sep 07	433	13000	1240000
Okt 07	466	14000	1360000
Nov 07	449	13500	1290000
Dez 07	532	16000	1700000

	Vertreterbesuche	*Werbeaufwand*	*Umsatz*
Vertreterbesuche	3147,934469		
Werbeaufwand	94477,72121	2838541,667	
Umsatz	9493412,697	284895833,3	32515972222

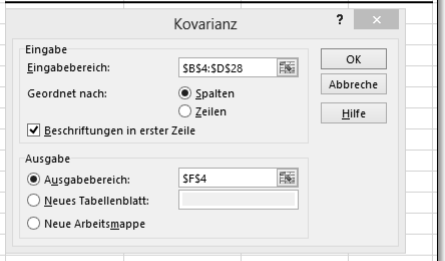


Abbildung 17.11 Kovarianzberechnung für mehrere Datenreihen

Die Werte, die auf der Diagonalen der Ausgabetabelle angezeigt werden, liefern jeweils die Kovarianz der n-ten Messvariable mit sich selbst. Das entspricht der Varianz der Grundgesamtheit für diese Variable, wie sie mit der Funktion `VAR.P()` oder `VARIANZEN()` berechnet wird.

17.7 Populationskenngrößen

Das Tool **Populationskenngrößen** liefert gewissermaßen einen Rundumschlag der univariaten statistischen Auswertungen, die sich auf eine Datenreihe anwenden lassen. Im Dialog kann ausgewählt werden, welche Werte gewünscht sind.

Populationskenngrößen

Monat	Umsatz
Jan 06	900000
Feb 06	1090000
Mrz 06	1300000
Apr 06	1200000
Mai 06	900000
Jun 06	990000
Jul 06	980000
Aug 06	1050000
Sep 06	1020000
Okt 06	1200000
Nov 06	1230000
Dez 06	1300000
Jan 07	1090000
Feb 07	1100000
Mrz 07	1400000
Apr 07	1160000
Mai 07	1080000
Jun 07	950000
Jul 07	1000000
Aug 07	1090000
Sep 07	1240000
Okt 07	1360000
Nov 07	1290000
Dez 07	1700000

Umsatz	
Mittelwert	1150833,333
Standardfehler	37599,70642
Median	1095000
Modus	1090000
Standardabweichung	184200,1904
Stichprobenvarianz	33929710145
Kurtosis	1,972607703
Schiefe	1,095479774
Wertebereich	800000
Minimum	900000
Maximum	1700000
Summe	27620000
Anzahl	24
k-größter Wert(3)	1360000
k-kleinster Wert(3)	950000
Konfidenzniveau(95,0%)	77780,91883

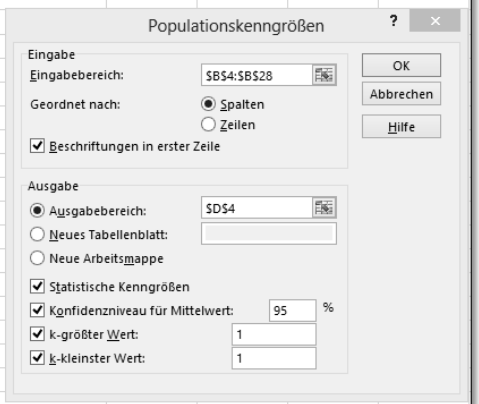


Abbildung 17.12 Kenngrößen zu einer Datenreihe

17.8 Exponentielles Glätten

Mit dem Tool **Exponentielles Glätten** lassen sich für die Werte einer Datenreihe alternative Prognosewerte errechnen, die einen glatteren Verlauf eines exponentiellen Wachs-

tums darstellen. Das Ausmaß der Glättung kann im Dialog des Tools über den **Glättungsparameter** angegeben werden, wobei sinnvolle Werte zwischen 0,2 und 0,3 liegen. Bei dieser Vorgehensweise wird davon ausgegangen, dass der vorhergehende Prognosewert in bestimmtem Umfang fehlerhaft sein kann und dass deshalb für den nächsten Prognosewert eine Anpassung von 20 bis 30% vorzunehmen ist. Wahlweise kann eine **Diagrammdarstellung** und die Ausgabe der **Standardfehler** gewählt werden.

Abbildung 17.13 zeigt in der Spalte D die geglättete Folge von Werten. Dabei wird bei einem Glättungsparameter von `0,3` jeweils ab dem zweiten Wert mit folgender Formel gearbeitet:

```
=0,7*B6+0,3*D5
```

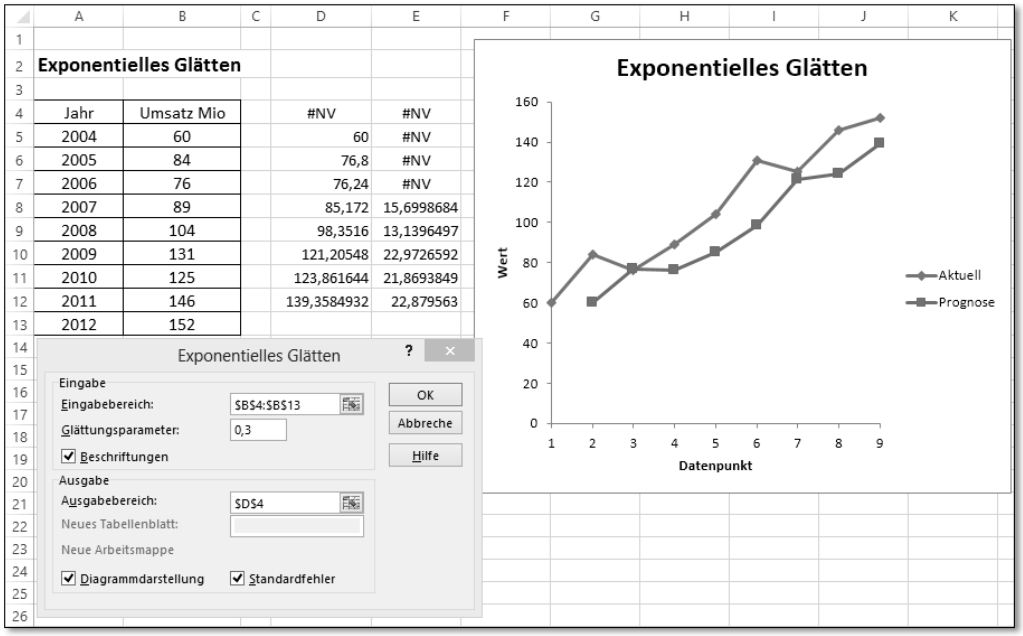


Abbildung 17.13 Glätten eines exponentiellen Anstiegs von Werten

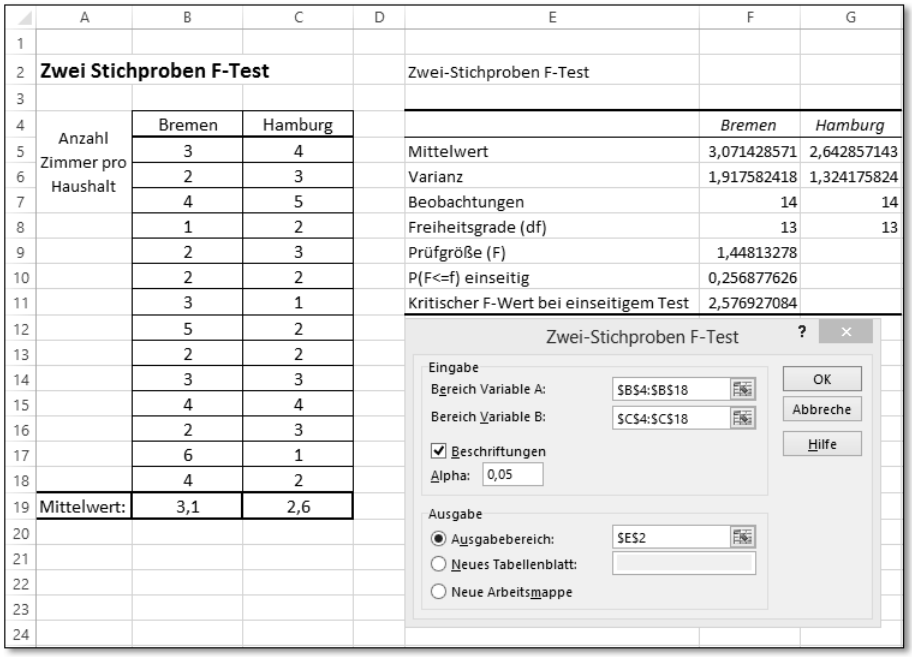
Die Werte für **Standardfehler** werden dabei mit folgender Formel berechnet:

```
=WURZEL(SUMMEXMY2(B6:B8;D5:D7)/3)
```

Es wird also die Wurzel aus der gedrittelten Summe der quadrierten Differenzen zwischen drei alten und drei neuen Werten gezogen.

17.9 Zwei-Stichproben F-Test

Das Tool **Zwei-Stichproben F-Test** erlaubt die Prüfung der Nullhypothese, nach der zwei Stichproben aus Grundgesamtheiten mit gleicher Varianz stammen. Im Dialog werden unter **Bereich Variable A** und **Bereich Variable B** die Daten der beiden Stichproben angegeben. Daraus wird als Prüfgröße der Stichproben-F-Wert berechnet, als Quotient der beiden Varianzen, im Beispiel also `=F3/G3`. Liegt dieser Wert F unterhalb des kritischen F-Werts, der für die angegebenen Freiheitsgrade und die im Dialog mit dem **Alpha**-Wert angegebene Irrtumswahrscheinlichkeit berechnet wird, kann die Nullhypothese nicht verworfen werden. Die Varianzen zwischen den beiden Stichproben unterscheiden sich dann nicht signifikant, sondern nur zufällig. Siehe auch die Beschreibungen der Funktionen `F.VERT()` und `FVERT()`, `F.INV()` und `FINV()` sowie `F.TEST()` und `FTEST()` in Kapitel 7, »Statistische Funktionen«.



Zwei Stichproben F-Test

	Bremen	Hamburg
Anzahl Zimmer pro Haushalt	3	4
	2	3
	4	5
	1	2
	2	3
	2	2
	3	1
	5	2
	2	2
	3	3
	4	4
	2	3
	6	1
	4	2
Mittelwert:	3,1	2,6

Zwei-Stichproben F-Test

	Bremen	*Hamburg*
Mittelwert	3,071428571	2,642857143
Varianz	1,917582418	1,324175824
Beobachtungen	14	14
Freiheitsgrade (df)	13	13
Prüfgröße (F)	1,44813278	
P(F<=f) einseitig	0,256877626	
Kritischer F-Wert bei einseitigem Test	2,576927084	

Abbildung 17.14 Vergleich der Varianz zweier Stichproben

17.10 Fourieranalyse

Mit dem Tool **Fourieranalyse** ist es möglich, periodische Daten mit Hilfe der Schnellen Fouriertransformation (*Fast Fourier Transform* oder kurz *FFT*) zu analysieren. FFT ist

nur anwendbar, wenn die Anzahl der Messwerte unter **Eingabebereich** eine Potenz von 2 ist. Der Eingabebereich kann reelle oder komplexe Zahlen enthalten. Komplexe Zahlen müssen als Zeichenfolge in dem Format `x+yi` oder `x+yj` eingegeben werden, siehe dazu Kapitel 6, »Technische Funktionen«.

	A	B	C
1			
2	**Fourieranalyse**		
3			
4	n	1	
5	a_n	1	
6	ωt	a_n*sin(nωt)	FFT-Werte
7	0	0	1,7271157934689
8	0,0628319	0,06279052	-0,25422547109261+0,585472555334632i
9	0,1256637	0,125333234	-0,244767899084127+0,241353092246537i
10	0,1884956	0,187381315	-0,2431543032370 99+9,98899170938845E-002i
11	0,2513274	0,248689887	-0,242820446641221
12	0,3141593	0,309016994	-0,2431543032371-9,98899170938843E-002i
13	0,3769911	0,368124553	-0,244767899084128-0,241353092246537i
14	0,439823	0,425779292	-0,254225471092611-0,585472555334632i

Abbildung 17.15 Schnelle Fouriertransformation bei Schwingungen

Das Tool unterstützt auch die umgekehrte Transformation, die aus transformierten Daten wieder die ursprünglichen Daten zurückgibt. Die Fouriertransformation ist beispielsweise ein fundamentales Verfahren in der Signalverarbeitung. Dabei wird die Darstellung von Signalen mit den Parametern Zeitpunkt und Abtastwert in die Darstellung mit Frequenzanteil, Amplitude, Phase überführt.

17.11 Histogramme für die Darstellung von Häufigkeiten

Mit dem Analysetool **Histogramm** können individuelle und kumulierte Häufigkeiten für einen Zellbereich von Daten berechnet werden. Dabei können die Daten innerhalb von frei wählbaren Grenzen klassifiziert werden. Gleichzeitig kann aus den errechneten Häufigkeiten ein Histogramm erzeugt werden. Diese spezielle Diagrammform wird gerne eingesetzt, um Häufigkeitsverteilungen darzustellen.

Eine typische Anwendung für ein Histogramm ist die Darstellung der Häufigkeitsverteilung der Abweichungen von einem bestimmten Standardwert. Denken Sie an eine Tabelle, in der für eine Stichprobe von 500 Personen der Prozentsatz der Abweichung vom Normalgewicht eingetragen ist.

Die Funktion erwartet in der Tabelle einen Eingabebereich, der nur aus numerischen Daten bestehen darf, abgesehen von der Beschriftung der Spalten bzw. Zeilen. Die Daten können in Spalten oder in Zeilen angeordnet werden.

Zusätzlich zum Eingabebereich kann noch ein Klassenbereich angelegt werden. Hier werden Grenzwerte eingetragen, die den Umfang der einzelnen Klassen definieren. Die Grenzwerte müssen in aufsteigender Reihenfolge angegeben werden. Sind ein oder mehrere Grenzwerte vorhanden, ermittelt Excel jeweils die Anzahl der Fälle, die unter die verschiedenen Klassen fallen, wobei immer die Werte gezählt werden, die kleiner oder gleich dem Grenzwert sind. Dabei werden die Werte bis zur untersten Klassengrenze und die Werte über der obersten Klassengrenze jeweils zusammengefasst. Wird kein Klassenbereich verwendet, teilt Excel die gesamte Spannweite der Daten selbstständig in gleich breite Abschnitte auf.

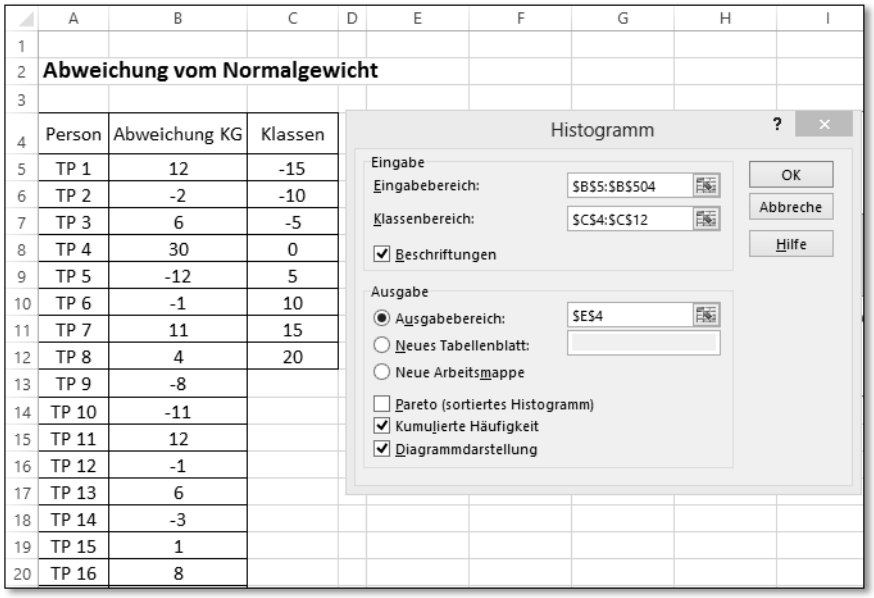


Abbildung 17.16 Bestimmung der Ausgangsdaten für das Histogramm

Sind diese Vorbereitungen getroffen, geben Sie im Dialog **Histogramm** unter **Eingabebereich** den Bezug auf den Bereich ein, der ausgewertet werden soll. Unter **Klassenbereich** bestimmen Sie den Bereich mit den Grenzwerten. Das Kontrollkästchen **Beschriftungen** sollte aktiviert werden, wenn die erste Zeile des Eingabebereichs und des Klassenbereichs Beschriftungen enthält. Ist das nicht der Fall, werden Vorgabebeschriftungen erstellt. Dann können Sie unter **Ausgabe** angeben, wo die Ergebnisse der Analyse dargestellt werden sollen.

Mit einem Häkchen bei **Pareto** kann erreicht werden, dass die Ergebnisse mit absteigender Häufigkeitsfolge sortiert dargestellt werden. Zusätzlich kann noch **Kumulierte Häufigkeit** gewählt werden. Dann liefert die Funktion eine zusätzliche Spalte mit den kumulierten Häufigkeiten und fügt dafür im Histogramm selbst eine entsprechende Kurve ein. Ob direkt ein Diagramm zu den Ergebnissen geplottet werden soll, legen Sie durch ein Häkchen bei **Diagrammdarstellung** fest.

Das Diagramm sieht noch etwas stärker wie ein übliches Histogramm aus, wenn Sie die Abstände zwischen den Säulen auf null setzen. Klicken Sie dazu auf das Diagramm und dann mit der rechten Maustaste auf eine der Säulen. Wählen Sie **Datenreihen formatieren** und dann **Reihenoptionen**.

Wenn Sie die Angaben bestätigen, wird zunächst eine Häufigkeitstabelle erstellt. Daneben erscheint das Diagramm.

Klassen	*Häufigkeit*	*Kumuliert %*
-15	0	0,00%
-10	72	14,43%
-5	80	30,46%
0	82	46,89%
5	75	61,92%
10	88	79,56%
15	86	96,79%
20	15	99,80%
und größer	1	100,00%

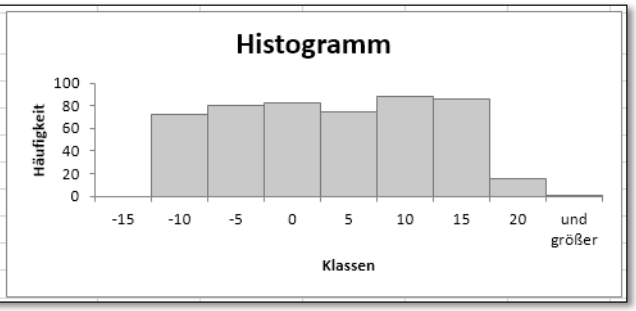


Abbildung 17.17 Auswertungstabelle und das Histogramm in überarbeiteter Form

17.12 Gleitender Durchschnitt

Insbesondere bei Zeitreihen kann es sinnvoll sein, erkennbare kurzfristige oder saisonale Schwankungen aus Trendberechnungen herauszurechnen, um ein besseres Bild von der zu erwartenden Gesamtentwicklung zu erhalten. Das Analysetool **Gleitender Durchschnitt** bietet dazu eine Möglichkeit. Dabei werden Prognosewerte generiert, die auf dem Mittelwert der Variable für eine bestimmte Anzahl vorhergehender Zeiträume basieren. Wenn die bisher bekannten Daten beispielsweise erkennen lassen, dass jede Woche freitags immer ein überdurchschnittlicher Spitzenwert erreicht wird, kann im Dialog ein **Intervall** von 7 angegeben werden.

Die Funktion berechnet dann jeweils für 7 Tage den Mittelwert und ordnet diesen dem mittleren Wert im Intervall zu, also dem 4. Tag. Im nächsten Rechenschritt wird diese Rechnung für den 2. bis 8. Tag vorgenommen und der Mittelwert wieder dem Tag in der

Mitte zugeordnet. So gleitet die Berechnung über den gesamten Zeitraum. Neben den berechneten Prognosewerten können noch die Ausgabe des Standardfehlers und auch die Darstellung der neuen Werte im Diagramm angefordert werden.

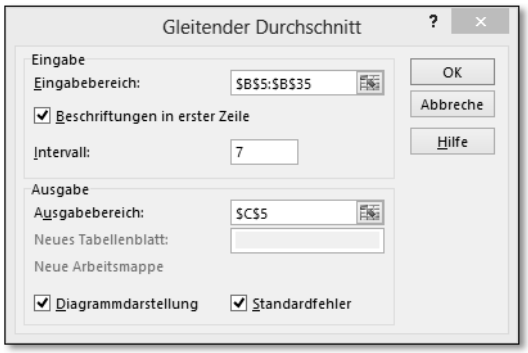
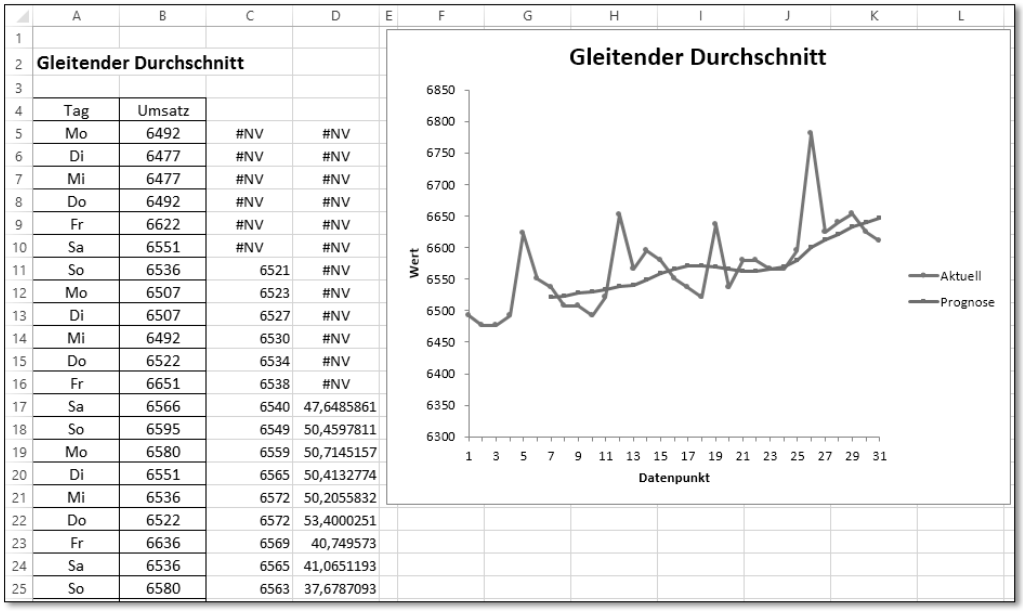


Abbildung 17.18 Dialog für das Tool »Gleitender Durchschnitt«

Für den Eingabebereich müssen mindestens vier Zellen angegeben werden, damit die Funktion arbeiten kann. Wie das Ergebnis im Folgenden zeigt, kann in Spalte C für die ersten Werte aus Spalte B kein geglätteter Wert zugeordnet werden. In das Diagramm in Abbildung 17.19 wurden auch noch die Ausgangsdaten eingefügt, so dass die Glättung im Vergleich gut zu sehen ist.



	A	B	C	D
1				
2	**Gleitender Durchschnitt**			
3				
4	Tag	Umsatz		
5	Mo	6492	#NV	#NV
6	Di	6477	#NV	#NV
7	Mi	6477	#NV	#NV
8	Do	6492	#NV	#NV
9	Fr	6622	#NV	#NV
10	Sa	6551	#NV	#NV
11	So	6536	6521	#NV
12	Mo	6507	6523	#NV
13	Di	6507	6527	#NV
14	Mi	6492	6530	#NV
15	Do	6522	6534	#NV
16	Fr	6651	6538	#NV
17	Sa	6566	6540	47,6485861
18	So	6595	6549	50,4597811
19	Mo	6580	6559	50,7145157
20	Di	6551	6565	50,4132774
21	Mi	6536	6572	50,2055832
22	Do	6522	6572	53,4000251
23	Fr	6636	6569	40,749573
24	Sa	6536	6565	41,0651193
25	So	6580	6563	37,6787093

Abbildung 17.19 Geglättete Prognosewerte

17.13 Zufallszahlengenerierung

Gerade für den Test von statistischen Lösungen in Excel ist das Tool **Zufallszahlengenerierung** nützlich. Es erlaubt, einen Bereich mit unabhängigen Zufallszahlen zu füllen, wobei die Art der Daten durch die Wahl der entsprechenden Verteilung bestimmt wird.

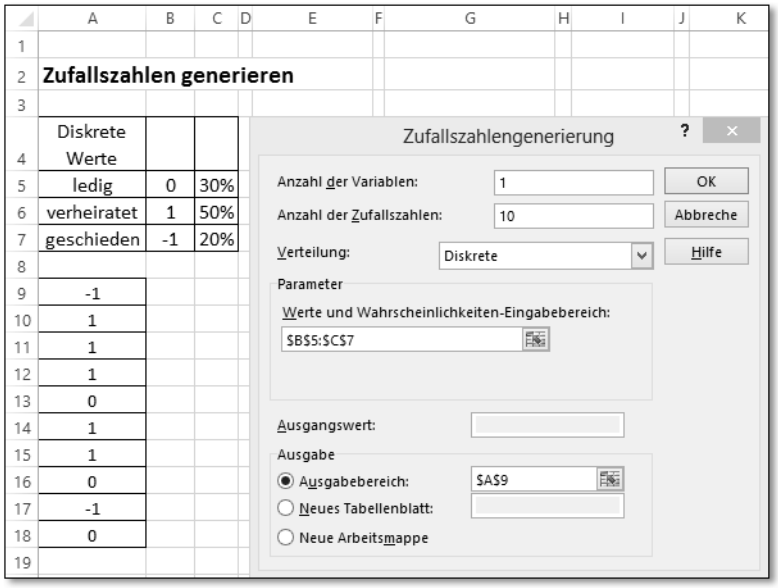


Abbildung 17.20 Zufallszahlen für eine Standardnormalverteilung generieren

Im Dialog wird zunächst die **Anzahl der Variablen** angegeben. Für jede Variable wird eine Spalte ausgegeben. Die **Anzahl der Zufallszahlen** bestimmt dann die Anzahl der Zeilen. Unter **Verteilung** stehen die meisten der von den Excel-Funktionen genutzten Verteilungen zur Auswahl. Je nach Auswahl des Verteilungstyps sind jeweils die verteilungsspezifischen **Parameter** einzutragen. Meist kann auch ein **Ausgangswert** eingetragen werden.

- **Diskrete**: Angegeben werden ein numerischer Wertebereich und ein zugeordneter Wahrscheinlichkeitenbereich.
- **Gleichverteilt**: Angegeben wird ein oberer und unterer Grenzwert, die Werte werden mit derselben Wahrscheinlichkeit aus diesem Bereich gezogen.
- **Standard**: Für eine Standardnormalverteilung werden **Mittelwert** = 0 und **Standardabweichung** = 1 angegeben.

- **Bernoulli**: Werte sind `0` oder `1`, angegeben wird unter **p-Wert** die Erfolgswahrscheinlichkeit.
- **Binomial**: Angegeben werden unter **p-Wert** die Erfolgswahrscheinlichkeit und die Zahl der Versuche.
- **Poisson**: Angegeben wird der **Lambda**-Wert, der gleich `1/Mittelwert` ist.
- **Schematisch**: Angegeben werden mit **Von** ein unterer und mit **bis** ein oberer Grenzwert, eine **Schrittlänge**, eine Wiederholungsrate für Werte und eine Wiederholungsrate für Sequenzen.

Abbildung 17.21 zeigt einige Beispiele:

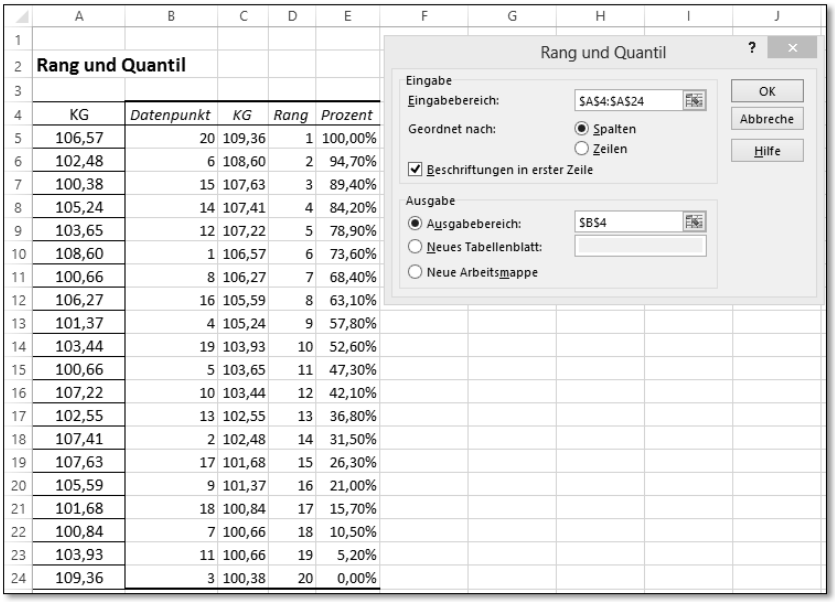
Zufallszahlen generieren

Diskrete Werte			Gleichverteilt	Standardnormal-verteilung	Bernoulli	Binomial	Poisson	Schematisch
ledig	0	30%	zwischen					
verheiratet	1	50%	0					
geschieden	-1	20%	100					
-1			82,99813837	-0,314105364	1	1	7	1
1			88,65016633	0,979482593	0	1	3	1
1			25,51957762	0,931895556	1	1	5	21
1			65,90166936	0,328604983	0	1	3	21
0			4,416028321	0,857137366	0	0	2	41
1			21,94586016	-0,443465069	0	4	2	41
1			83,92895291	0,381899099	0	2	4	61
0			62,82540361	-0,332240688	0	1	1	61
-1			55,98620563	0,203388026	0	1	2	81
0			63,85082553	-1,450073341	0	1	5	81

Abbildung 17.21 Beispiele mit generierten Werten für verschiedene Verteilungen

17.14 Rang und Quantil

Mit dem Tool **Rang und Quantil** lässt sich für den angegebenen **Eingabebereich** eine Tabelle erstellen, die den Rang und den Quantilsrang jedes Werts anzeigt. Dabei werden die Funktionen `RANG()` und `QUANTILSRANG()` benutzt. Allerdings werden Rangwerte nur absteigend ermittelt, wenn die aufsteigende Reihenfolge benötigt wird, sollte direkt mit `RANG()` gearbeitet werden.



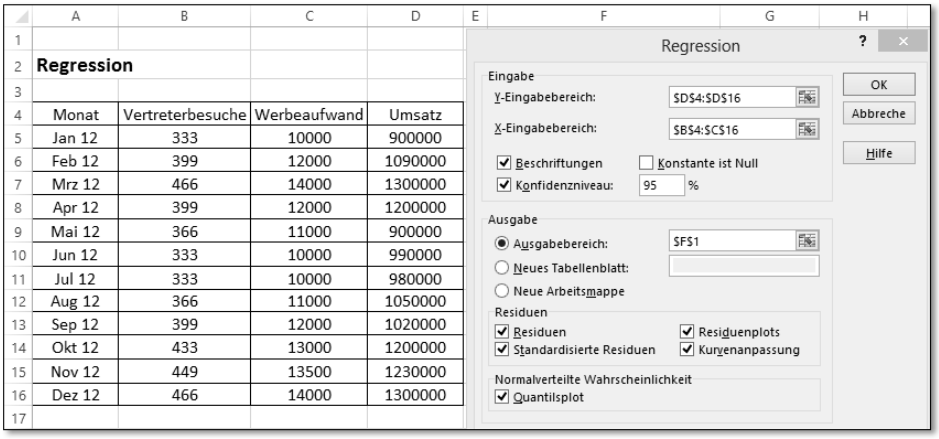
Rang und Quantil

KG	Datenpunkt	KG	Rang	Prozent
106,57	20	109,36	1	100,00%
102,48	6	108,60	2	94,70%
100,38	15	107,63	3	89,40%
105,24	14	107,41	4	84,20%
103,65	12	107,22	5	78,90%
108,60	1	106,57	6	73,60%
100,66	8	106,27	7	68,40%
106,27	16	105,59	8	63,10%
101,37	4	105,24	9	57,80%
103,44	19	103,93	10	52,60%
100,66	5	103,65	11	47,30%
107,22	10	103,44	12	42,10%
102,55	13	102,55	13	36,80%
107,41	2	102,48	14	31,50%
107,63	17	101,68	15	26,30%
105,59	9	101,37	16	21,00%
101,68	18	100,84	17	15,70%
100,84	7	100,66	18	10,50%
103,93	11	100,66	19	5,20%
109,36	3	100,38	20	0,00%

Abbildung 17.22 Ermittlung von Rang und Quantilsrang in einem Zug

17.15 Regression

Das Analysetool **Regression** liefert mit Hilfe der RGP()-Tabellenfunktion eine lineare Regressionsanalyse und bereitet die errechneten Daten in einer übersichtlichen Form auf. Siehe dazu die Beschreibung von RGP() in Kapitel 7, »Statistische Funktionen«.



Regression

Monat	Vertreterbesuche	Werbeaufwand	Umsatz
Jan 12	333	10000	900000
Feb 12	399	12000	1090000
Mrz 12	466	14000	1300000
Apr 12	399	12000	1200000
Mai 12	366	11000	900000
Jun 12	333	10000	990000
Jul 12	333	10000	980000
Aug 12	366	11000	1050000
Sep 12	399	12000	1020000
Okt 12	433	13000	1200000
Nov 12	449	13500	1230000
Dez 12	466	14000	1300000

Abbildung 17.23 Ausgangsdaten und Dialogparameter für das Tool »Regression«

Im Dialog werden unter **Y-Eingabebereich** die Werte der abhängigen Variablen angegeben, in diesem Beispiel also die Umsatzwerte, unter **X-Eingabebereich** die Daten einer oder wie hier mehrerer unabhängiger Variablen.

	F	G	H	I	J	K	L	M	N
3	*Regressions-Statistik*								
4	Multipler Korrelationskoeffizient	0,91483084							
5	Bestimmtheitsmaß	0,836915466							
6	Adjustiertes Bestimmtheitsmaß	0,800674459							
7	Standardfehler	64875,18019							
8	Beobachtungen	12							
9									
10	ANOVA								
11		*grade (df)*	*summen*	*Quadratsumme*	*(F)*	*F krit*			
12	Regression	2	1,9439E+11	97193782813	23,0930519	0,00028567			
13	Residue	9	3,7879E+10	4208789004					
14	Gesamt	11	2,3227E+11						
15									
16		*Koeffizienten*	*Standard-*	*t-Statistik*	*P-Wert*	*Untere 95%*	*Obere 95%*	*Untere*	*Obere*
17	Schnittpunkt	77793,15892	160824,652	0,483714143	0,6401403	-286017,48	441603,798	-286017,48	441603,798
18	Vertreterbesuche	28267,06766	51451,2775	0,549394866	0,596101	-88123,8082	144657,943	-88123,8082	144657,943
19	Werbeaufwand	-855,0844873	1716,23722	-0,498232109	0,63026823	-4737,4828	3027,31383	-4737,4828	3027,31383

Abbildung 17.24 Die aufbereitete Auswertung der Regressionsanalyse

Die Auswertung liefert eine Regressions-Statistik und eine ANOVA-Analyse. Im dritten Abschnitt werden für die beiden unabhängigen Variablen und für den Schnittpunkt die Koeffizienten, der Standardfehler, die t-Statistik und der P-Wert angegeben.

Zusätzlich können über die Dialogoptionen weitere Auswertungen angefordert werden:

Residuen liefert eine Ausgabetabelle, die eine Spalte mit der Schätzung für die abhängigen Werte und eine Spalte für die Residuen enthält, also die Abweichungen der geschätzten Werte von den beobachteten Werten.

	F	G	H	I
23	AUSGABE: RESIDUENPLOT			
24				
25	*Beobachtung*	*Schätzung für Umsatz*	*Residuen*	*Standardisierte Residuen*
26	1	926461,0972	-26461,0972	-0,419338904
27	2	1110360,311	-20360,3112	-0,322657466
28	3	1292623,314	7376,68649	0,11690111
29	4	1101276,42	98723,5804	1,5645095
30	5	1024700,831	-124700,831	-1,976180703
31	6	935101,1329	54898,8671	0,870002878
32	7	924978,8256	55021,1744	0,871941128
33	8	1032281,158	17718,8416	0,280797109
34	9	1123247,23	-103247,23	-1,636197471
35	10	1186803,122	13196,8775	0,209135853
36	11	1238901,934	-8901,93428	-0,141072282
37	12	1291873,236	8126,76432	0,128787873

Abbildung 17.25 Schätzwerte und Residuen

Residuenplot erzeugt für jede unabhängige Variable und die jeweiligen Abweichungen ein eigenes Diagramm.

Standardisierte Residuen bewirkt, dass die Ausgabetabelle zusätzlich noch eine Spalte für standardisierte Residuen enthält. Sie sind ein Maß für den Vergleich der Differenz zwischen erwartetem und beobachtetem Wert.

Kurvenanpassung sorgt dafür, dass ein Diagramm mit den prognostizierten und den tatsächlich beobachteten Werten erzeugt wird. **Quantilsplot** liefert ein Diagramm für die Quantile.

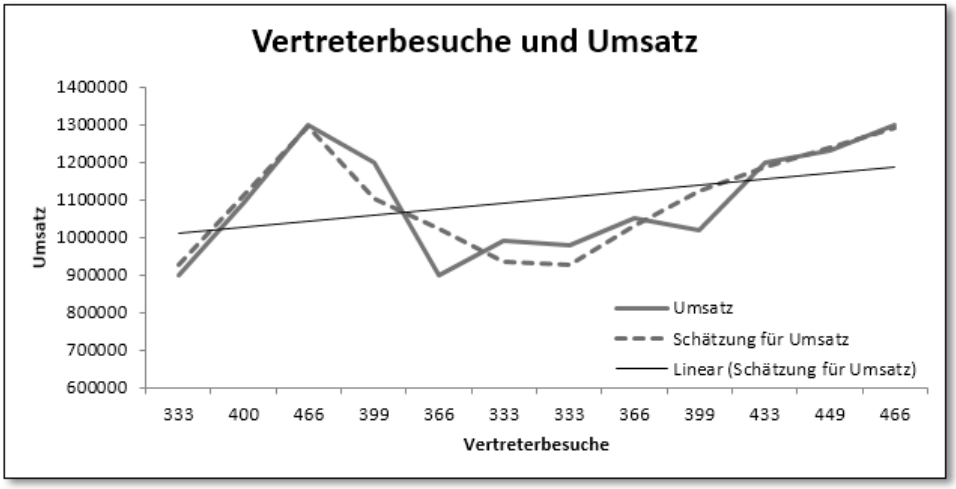


Abbildung 17.26 Leicht angepasstes Diagramm

17.16 Stichprobenziehung

Die Funktion **Stichprobenziehung** ist eine Hilfsfunktion, die es erlaubt, aus einem **Eingabebereich**, der statistisch als Grundgesamtheit zu interpretieren ist, eine reduzierte Datenmenge als Stichprobe zu ziehen. Dabei sind zwei Verfahren möglich. Soll wie in Abbildung 17.27 nur jeder 3. Monat berücksichtigt werden, geben Sie unter **Periode** den Wert »3« an.

Soll die Auswahl zufällig erfolgen, geben Sie unter **Stichprobenanzahl** die Größe der Stichprobe an. Die Funktion kann nur die Werte herausziehen, die Beschriftungen werden nicht mitgezogen.

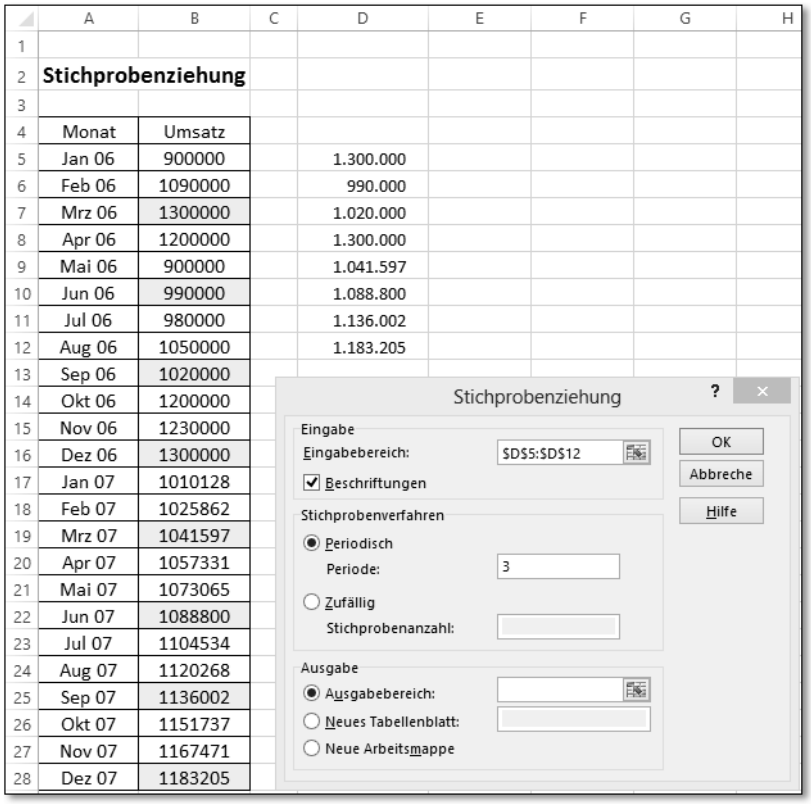


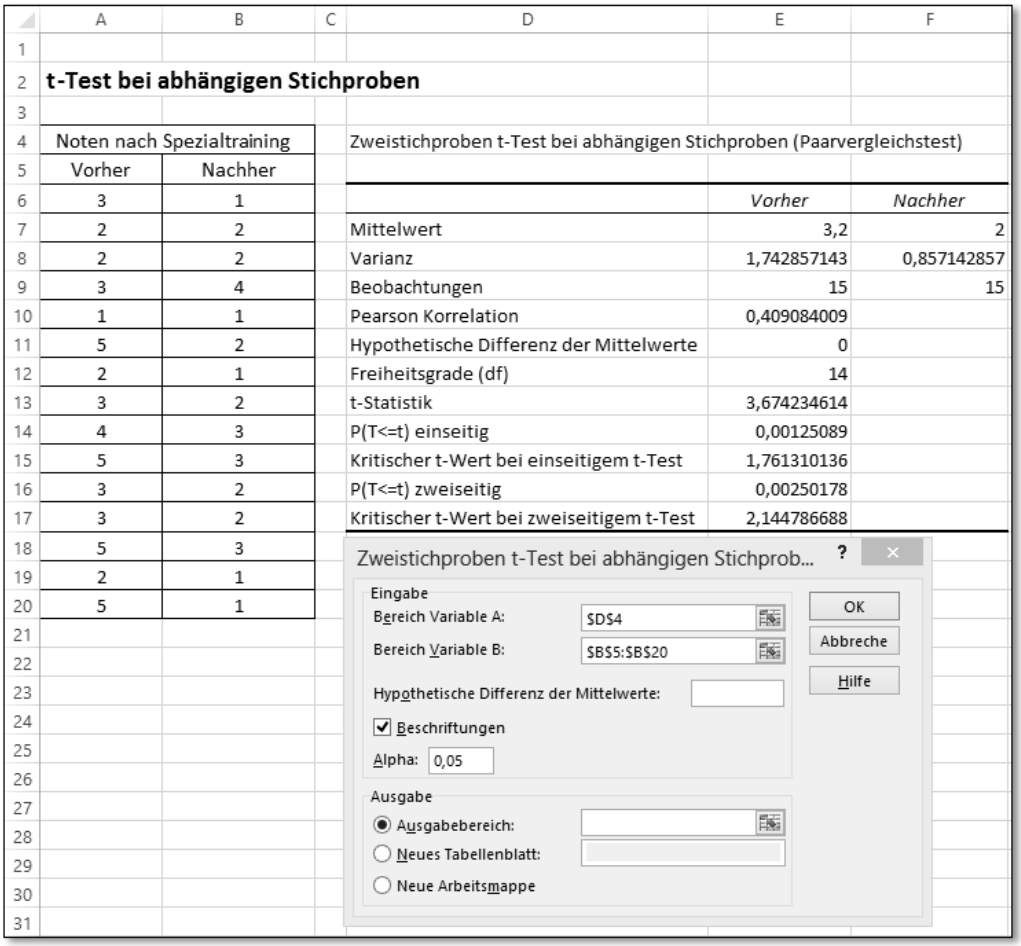
Abbildung 17.27 Aus den Monatswerten wird jeder dritte Wert herausgezogen.

17.17 t-Test-Varianten

Für den Vergleich von zwei Stichproben in Bezug auf die Übereinstimmung ihrer Mittelwerte gibt es gleich drei Tools, die sich dadurch unterscheiden, dass sie von unterschiedlichen Annahmen in Bezug auf die Varianz der Grundgesamtheiten ausgehen. In den Dialogen der Tools werden die beiden Stichproben jeweils unter **Bereich Variable A:** und **Bereich Variable B:** angegeben. Dabei kann entweder jeweils eine Spalte oder eine Zeile gewählt werden. Ist die Option **Beschriftungen** aktiviert, muss die Zelle mit den Beschriftungen in den Bereichen mit einbezogen werden.

Unter **Hypothetische Differenz der Mittelwerte** kann ein Wert für die Differenz der Stichprobenmittelwerte eingegeben werden. Wird `0` eingegeben, wird angenommen, dass die Mittelwerte gleich sind. Unter **Alpha** wird das Konfidenzniveau für den Test angegeben. Der Wert muss im Bereich `0` bis `1` liegen. Vorgabe ist `0,05`.

In allen Fällen wird ein t-Statistik-Wert `t` in der Ausgabetabelle ausgegeben. Je nach angegebenen Stichprobenwerten kann dieser positiv oder negativ sein.



t-Test bei abhängigen Stichproben

Noten nach Spezialtraining	
Vorher	Nachher
3	1
2	2
2	2
3	4
1	1
5	2
2	1
3	2
4	3
5	3
3	2
3	2
5	3
2	1
5	1

Zweistichproben t-Test bei abhängigen Stichproben (Paarvergleichstest)

	Vorher	*Nachher*
Mittelwert	3,2	2
Varianz	1,742857143	0,857142857
Beobachtungen	15	15
Pearson Korrelation	0,409084009	
Hypothetische Differenz der Mittelwerte	0	
Freiheitsgrade (df)	14	
t-Statistik	3,674234614	
P(T<=t) einseitig	0,00125089	
Kritischer t-Wert bei einseitigem t-Test	1,761310136	
P(T<=t) zweiseitig	0,00250178	
Kritischer t-Wert bei zweiseitigem t-Test	2,144786688	

Abbildung 17.28 Paarvergleichstest

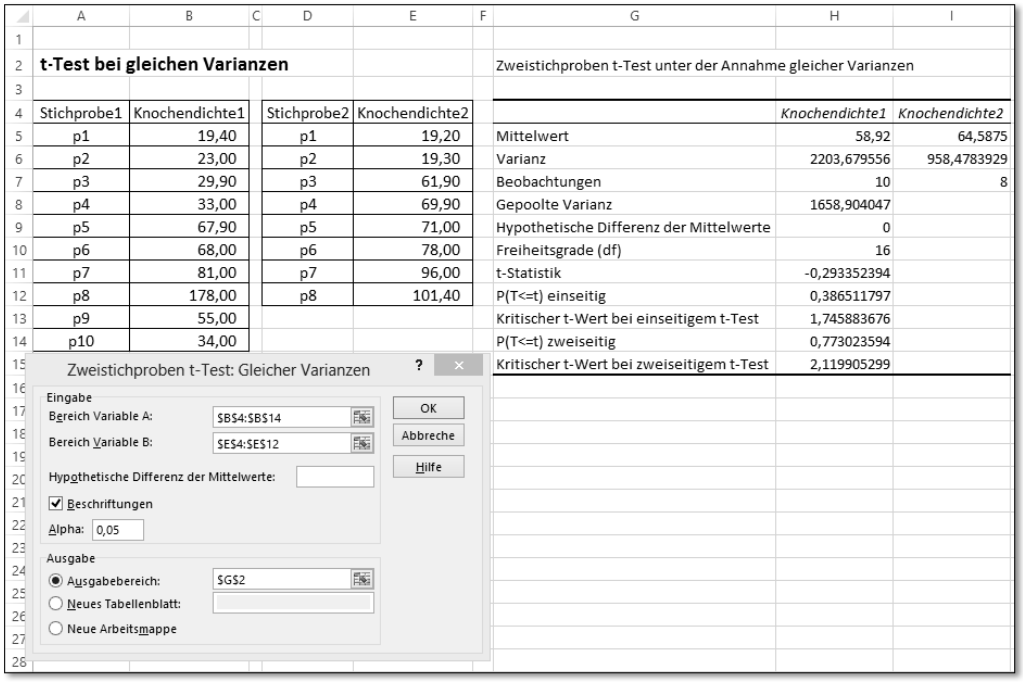
Bei der Option **Zweistichproben t-Test bei abhängigen Stichproben** wird davon ausgegangen, dass es sich bei den beiden Stichproben um beobachtete Werte für dieselben Objekte handelt, etwa vor und nach einer bestimmten Behandlung. Bei dieser Variante, die auch als Paarvergleichstest bezeichnet wird, muss die Zahl der Werte in beiden Stichproben gleich sein, was bei den anderen Varianten nicht erforderlich ist. Es wird dabei aber nicht angenommen, dass die Varianzen der Grundgesamtheiten gleich sind. Geprüft wird beispielsweise, ob die vor und nach einer Behandlung oder einem Experiment beobachteten Werte signifikant unterschiedlich sind. In Abbildung 17.28 werden

beispielsweise Notenwerte vor und nach einem Spezialtraining verglichen. Die Auswertung zeigt, dass das Tool mit einem Wert für Freiheitsgrade von 14 arbeitet, also der Anzahl der Datenpaare –1.

Der errechnete t-Wert ist mit 3,67 größer als die beiden kritischen t-Werte für einen ein- oder zweiseitigen t-Test bei einem Alpha-Wert von 0,05. Damit ist eine signifikante Veränderung zwischen der ersten und der zweiten Stichprobe festzustellen.

Die Option **Zweistichproben t-Test: Gleicher Varianzen** geht von der Annahme aus, dass die Varianzen der Grundgesamtheiten, denen die Stichproben entnommen sind, gleich sind. Der Test wird auch als homoskedastischer t-Test bezeichnet. Er prüft, ob die beiden Stichproben mit genügend großer Wahrscheinlichkeit aus Verteilungen mit gleichen Erwartungswerten der Zufallsvariablen stammen.

Der Wert für Freiheitsgrade ist in diesem Fall gleich der Gesamtzahl der Testwerte –2. Im Ausgabebereich wird auch die gepoolte Varianz ausgegeben, ein akkumuliertes Maß für die Abweichung vom Mittelwert.



Stichprobe1	Knochendichte1	Stichprobe2	Knochendichte2
p1	19,40	p1	19,20
p2	23,00	p2	19,30
p3	29,90	p3	61,90
p4	33,00	p4	69,90
p5	67,90	p5	71,00
p6	68,00	p6	78,00
p7	81,00	p7	96,00
p8	178,00	p8	101,40
p9	55,00		
p10	34,00		

Zweistichproben t-Test unter der Annahme gleicher Varianzen

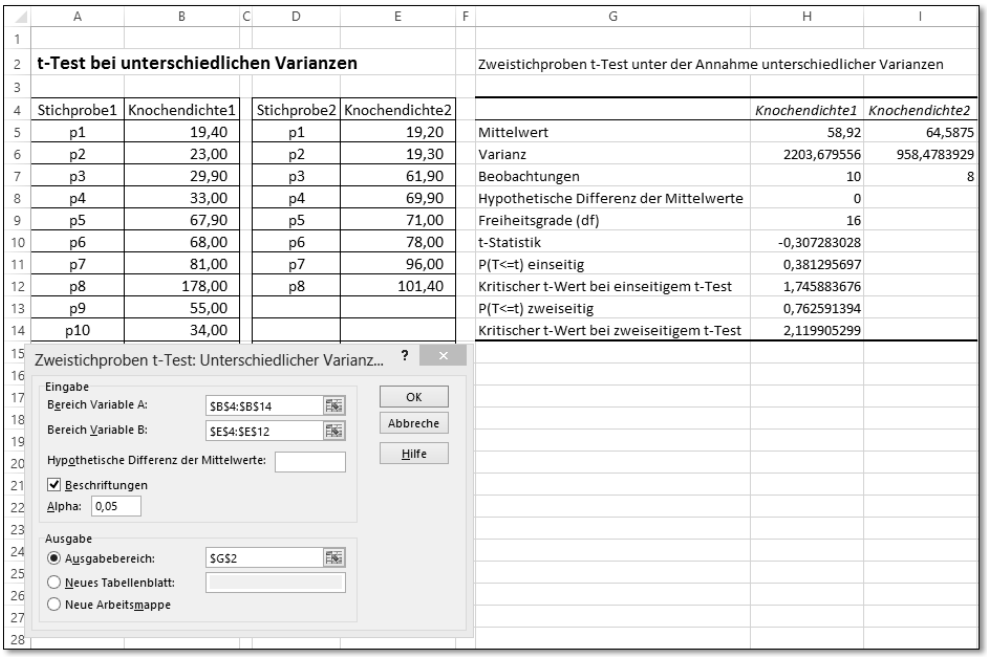
	Knochendichte1	*Knochendichte2*
Mittelwert	58,92	64,5875
Varianz	2203,679556	958,4783929
Beobachtungen	10	8
Gepoolte Varianz	1658,904047	
Hypothetische Differenz der Mittelwerte	0	
Freiheitsgrade (df)	16	
t-Statistik	-0,293352394	
P(T<=t) einseitig	0,386511797	
Kritischer t-Wert bei einseitigem t-Test	1,745883676	
P(T<=t) zweiseitig	0,773023594	
Kritischer t-Wert bei zweiseitigem t-Test	2,119905299	

Abbildung 17.29 t-Test bei der Annahme gleicher Varianzen

Die dritte Option **Zweistichproben t-Test: Unterschiedlicher Varianzen** geht davon aus, dass die Varianzen der entsprechenden Grundgesamtheiten nicht gleich sind.

Dieser Test wird auch als heteroskedastischer t-Test bezeichnet. Wie bei dem letzten Test wird auch hier geprüft, ob die beiden Stichproben mit genügend großer Wahrscheinlichkeit aus Verteilungen mit gleichen Erwartungswerten der Zufallsvariablen stammen.

Beachtet werden muss, dass das Analysetool bei der Berechnung der Freiheitsgrade den erhaltenen Wert immer auf die nächste ganze Zahl rundet. Die Tabellenfunktion `TTEST()` rundet diesen Wert aber nicht, weshalb es zu Abweichungen zwischen dem Ergebnis von `TTEST()` und dem vom Analysetool errechneten t-Wert kommen kann.



t-Test bei unterschiedlichen Varianzen

Stichprobe1	Knochendichte1	Stichprobe2	Knochendichte2
p1	19,40	p1	19,20
p2	23,00	p2	19,30
p3	29,90	p3	61,90
p4	33,00	p4	69,90
p5	67,90	p5	71,00
p6	68,00	p6	78,00
p7	81,00	p7	96,00
p8	178,00	p8	101,40
p9	55,00		
p10	34,00		

Zweistichproben t-Test unter der Annahme unterschiedlicher Varianzen

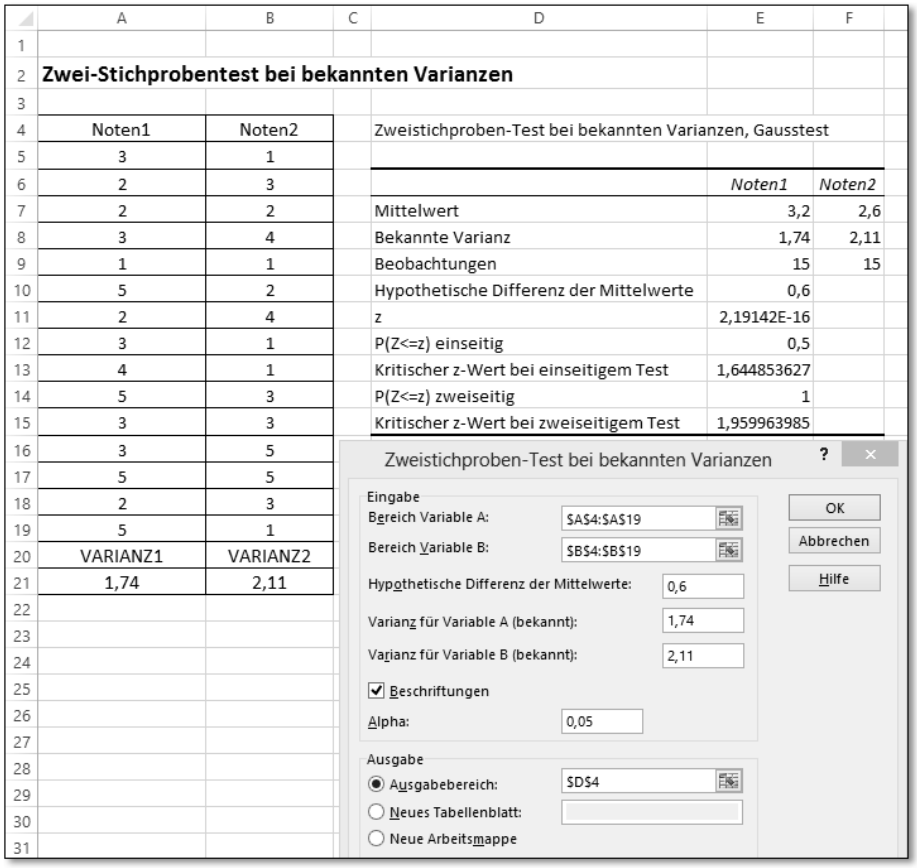
	Knochendichte1	*Knochendichte2*
Mittelwert	58,92	64,5875
Varianz	2203,679556	958,4783929
Beobachtungen	10	8
Hypothetische Differenz der Mittelwerte	0	
Freiheitsgrade (df)	16	
t-Statistik	-0,307283028	
P(T<=t) einseitig	0,381295697	
Kritischer t-Wert bei einseitigem t-Test	1,745883676	
P(T<=t) zweiseitig	0,762591394	
Kritischer t-Wert bei zweiseitigem t-Test	2,119905299	

Abbildung 17.30 t-Test bei der Annahme unterschiedlicher Varianzen

17.18 Gauß-Test

Die Option **Zwei-Stichprobentest bei bekannten Varianzen** wird verwendet, um die Nullhypothese, dass es keine Differenz zwischen den Erwartungswerten zweier Zufallsvariablen gibt, gegenüber einseitigen oder zweiseitigen alternativen Hypothesen zu testen. Sind die Varianzen nicht bekannt, sollte lieber die `G.TEST()`-Tabellenfunktion verwendet werden.

Im Dialog, der ansonsten dem der zuvor beschriebenen t-Tests ähnelt, werden die Varianzen der beiden Variablen direkt angegeben.



Zwei-Stichprobentest bei bekannten Varianzen

Noten1	Noten2
3	1
2	3
2	2
3	4
1	1
5	2
2	4
3	1
4	1
5	3
3	3
3	5
5	5
2	3
5	1
VARIANZ1	VARIANZ2
1,74	2,11

Zweistichproben-Test bei bekannten Varianzen, Gausstest

	Noten1	*Noten2*
Mittelwert	3,2	2,6
Bekannte Varianz	1,74	2,11
Beobachtungen	15	15
Hypothetische Differenz der Mittelwerte	0,6	
z	2,19142E-16	
P(Z<=z) einseitig	0,5	
Kritischer z-Wert bei einseitigem Test	1,644853627	
P(Z<=z) zweiseitig	1	
Kritischer z-Wert bei zweiseitigem Test	1,959963985	

Abbildung 17.31 Stichprobentest mit bekannten Varianzen

Bei der Ausgabe ist Folgendes zu beachten:

`P(Z <= z) einseitig` entspricht `P(Z >= ABS(z))`, ist also die Wahrscheinlichkeit, dass ein `z`-Wert in gleicher Richtung weiter von 0 entfernt ist als der beobachtete `z`-Wert, wenn es keine Differenz zwischen den Erwartungswerten der Zufallsvariablen gibt.

`P(Z <= z) zweiseitig` entspricht `P(Z >= ABS(z)` oder `Z <= -ABS(z))`, ist also die Wahrscheinlichkeit, dass ein `z`-Wert in beliebiger Richtung weiter von 0 entfernt ist als der beobachtete `z`-Wert, wenn es keine Differenz zwischen den Erwartungswerten der Zufallsvariablen gibt. Das zweiseitige Ergebnis ist somit einfach das mit 2 multiplizierte einseitige Ergebnis.

18 Entwicklung eigener Funktionen

18.1 Funktionen mit VBA erstellen

Mit den über 450 Tabellenfunktionen, die Excel mitbringt, müssen Sie sich keineswegs zufriedengeben. Wenn sich bei Ihnen spezielle Berechnungen häufig wiederholen, können Sie sich mit Hilfe der in Excel integrierten Programmiersprache *Visual Basic for Applications* – kurz *VBA* – auch eigene Funktionen zusammenbauen, die Ihnen aus bestimmten Ausgangsdaten einen berechneten Wert zurückgeben.

18.1.1 Eigene Zinseszinsfunktion

Um die Vorgehensweise zu demonstrieren, soll zunächst eine einfache Funktion gebildet werden, die eine Zinseszinsberechnung durchführt. Eine solche Funktion ist ja in Excel nicht direkt vorhanden, auch wenn die `ZW()`-Funktion ein entsprechendes Ergebnis liefern kann, wenn anstelle der regelmäßigen Zahlungen ein Barwert als Argument angegeben wird, wobei die Funktion das Ergebnis als negativen Betrag auswirft.

1. Damit Sie die Funktion in allen Arbeitsmappen anwenden können, ist es sinnvoll, sie in der normalerweise verborgenen persönlichen Arbeitsmappe *Personal.xlsb* einzutragen. (In den älteren Versionen ist dies *Personal.xls*.)

2. Wechseln Sie mit `Alt` + `F11` in die VBA-Umgebung, und wählen Sie in dem Fenster **Projekt** den Eintrag **VBAProjekt(PERSONAL.XSLB)** aus. Wenn Sie die neuen Funktionen in einem eigenen Modul sammeln wollen, wählen Sie **Einfügen • Modul**, ansonsten können Sie auch ein bereits bestehendes Modul per Doppelklick oder -tipp im Projekt-Explorer öffnen.

3. Wählen Sie dann **Einfügen • Prozedur**, und tragen Sie den Namen der ersten Funktion ein. Nehmen Sie unter **Typ** die Option **Function**. Übernehmen Sie die Option **Public** für den Gültigkeitsbereich, damit die Funktion allseitig aufgerufen werden kann. Im Modulfenster erscheint ein Skelett für die Funktion:

```
Public Function ZZWert()
End Function
```

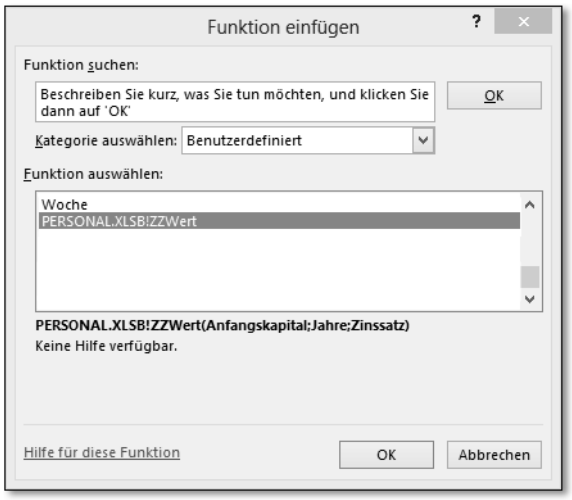
4 Fügen Sie in die Klammer hinter dem Funktionsnamen die Namen der Argumente ein, die an die Funktion übergeben werden sollen. In diesem Fall werden drei Argumente benötigt: `Anfangskapital`, `Jahre`, `Zinssatz`. Anders als bei den Tabellenfunktionen werden die Argumente innerhalb von VBA – der amerikanischen Schreibweise entsprechend – durch Kommata getrennt. Wenn Sie den Datentyp der Argumente genau festlegen wollen, können Sie mit `As <Datentyp>` jedem Argument eine entsprechende Festlegung zuordnen. Hinter der Klammer kann etwa mit `As Double` noch der Datentyp festgelegt werden, den die Funktion für ihr Ergebnis verwenden soll, das sie zurückgibt. Dies entspricht dem Standardzahlenformat von Excel. Ist ein Argument optional, wird vor den Argumentnamen das Schlüsselwort `Optional` gesetzt.

5 Nun schreiben Sie zwischen die beiden Zeilen die Anweisung, wie aus den gegebenen Argumenten der verzinste Endbetrag errechnet werden soll. Das Ergebnis wird an eine Variable übergeben, die den Namen der Funktion trägt. Das ist zugleich das Ergebnis, das die Funktion liefert, wenn sie im Tabellenblatt benutzt wird. Die Funktion sieht dann insgesamt so aus:

```
Public Function ZZWert(Anfangskapital, Jahre, Zinssatz) _
    As Double
    ZZWert = Anfangskapital * (1 + Zinssatz) ^ Jahre
End Function
```

6 Sichern Sie das VBA-Projekt mit der Schaltfläche **Speichern**. Wechseln Sie zurück in das Fenster der Arbeitsmappe per Klick oder Tipp auf das Symbol **Ansicht Microsoft Excel**.

7 Um die Funktion im Tabellenblatt zu verwenden, kann wie gewohnt der Dialog **Funktion einfügen** verwendet werden. Wählen Sie unter **Kategorie auswählen** den Eintrag **Benutzerdefiniert** und dann die gewünschte Funktion.



8 Im Dialogfeld **Funktionsargumente** werden automatisch die für die Funktion festgelegten Argumente angeboten. Nur eine Hilfe ist nicht vorhanden. Das Einbinden entsprechender Hilfedateien ist aber ebenfalls möglich, würde aber an dieser Stelle zu weit führen.

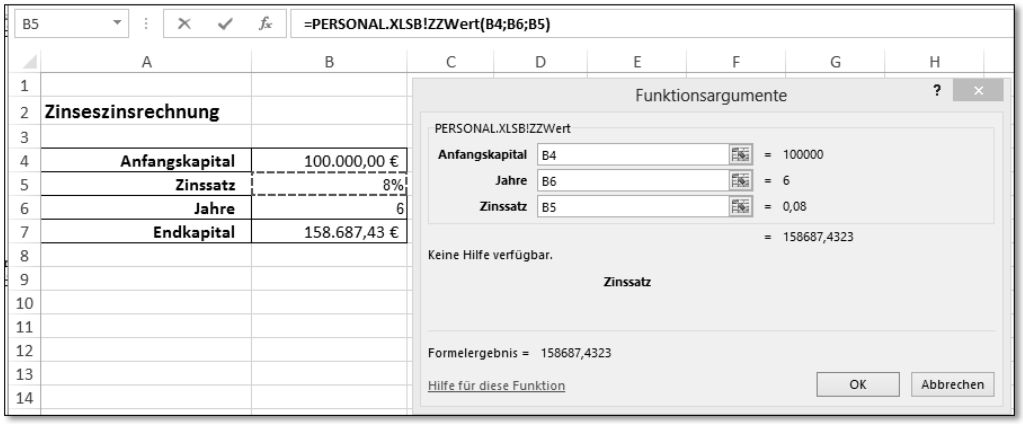


Abbildung 18.1 Eingabe der Argumente

Beachten Sie, dass Excel bei benutzerdefinierten Funktionen, die nicht in der aktiven Arbeitsmappe gespeichert sind, erwartet, dass der Name der Mappe, die die Funktion

enthält, vor den Namen der Funktion gesetzt wird, also hier `=Personal.xlsb!ZZWert()` statt `=ZZWert()`. Wenn Sie die Funktionen in die aktive Arbeitsmappe kopieren, müssen Sie dagegen nur den Funktionsnamen angeben.

Werden Arbeitsmappen an andere Arbeitsplätze exportiert, die benutzerdefinierte Funktionen enthalten, muss natürlich darauf geachtet werden, dass die entsprechenden Module auch mitgeliefert werden.

18.1.2 Eigene Funktionen in einem Add-In zur Verfügung stellen

Eine weitere Möglichkeit ist es, eine Arbeitsmappe, die eigene Funktionen enthält, als Excel-Add-In zu speichern. Vor dem Speichern sollte innerhalb des VBA-Editors mit **Debuggen • Kompilieren von VBAProjekt** geprüft werden, ob alle syntaktischen Fehler ausgeschlossen sind.

Sobald die Datei mit dem Dateityp *.xlam* gespeichert ist, kann sie in die Liste der aktiven Add-Ins übernommen werden, wie es bereits im ersten Kapitel, »Einstieg in Berechnungen mit Excel«, beschrieben wurde.

18.2 Hinweise zu VBA

Um komplexere Funktionen in VBA zu entwerfen, bedarf es entsprechender Kenntnisse in Visual Basic for Applications. Es würde den Rahmen dieses Buches sprengen, VBA an dieser Stelle im Detail vorzustellen. Wenigstens einige der Sprachkonstrukte, die in Funktionen häufiger benötigt werden, sollen jedoch kurz vorgestellt werden.

18.2.1 Objekte, Eigenschaften und Methoden

VBA ist eine objektorientierte Programmiersprache. Dahinter steht die Idee, dass Anwendungen mit Objekten zu tun haben. Für jedes Objekt gibt es einen bestimmten Satz von Eigenschaften und eine Reihe von Verfahren, die ein Objekt beherrscht. Sie werden Methoden genannt. Welche Objekte eine Anwendung bereitstellt und wie diese miteinander zusammenhängen, wird durch das Objektmodell – eine vorgegebene Hierarchie von Objekten – festgelegt, in dem die Anwendung selbst – `Application` – das oberste Objekt darstellt.

Soll beispielsweise in einer benutzerdefinierten Funktion eine der Standardfunktionen von Excel benutzt werden, wird diese Funktion als Methode des `Application`-Objekts

aufgerufen. Eine Funktion zur Berechnung der Kreisfläche kann beispielsweise so aussehen:

```
Public Function Kreisfläche(Radius) As Double
  Kreisfläche = Radius ^2 * Application.WorksheetFunction.Pi
End Function
```

Bei der Eingabe der Funktion werden Sie unterstützt, indem jeweils die zu einem Objekt möglichen Methoden oder Eigenschaften über Listenfelder angeboten werden, wie Abbildung 18.2 zeigt.

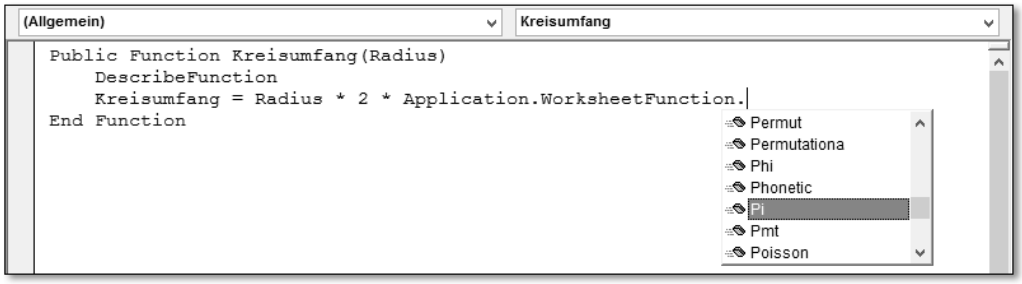
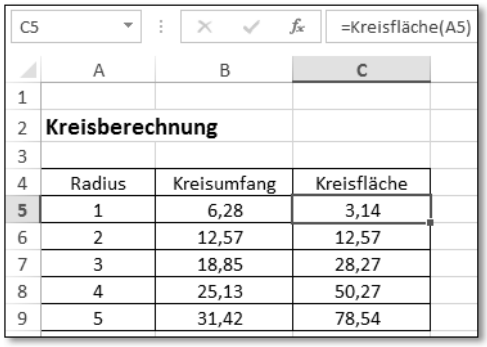


Abbildung 18.2 Unterstützung bei der Eingabe der Funktion

Eine Funktion für den Kreisumfang sieht ähnlich aus:

```
Public Function Kreisumfang(Radius)
  Kreisumfang = Radius * 2 * Application.WorksheetFunction.Pi
End Function
```

In der Arbeitsmappe können die Funktionen wie gewohnt eingesetzt werden, wobei es keine Rolle spielt, ob der Funktionsname in Großbuchstaben oder in gemischter Schreibweise eingegeben wird.



	A	B	C
1			
2	**Kreisberechnung**		
3			
4	Radius	Kreisumfang	Kreisfläche
5	1	6,28	3,14
6	2	12,57	12,57
7	3	18,85	28,27
8	4	25,13	50,27
9	5	31,42	78,54

Abbildung 18.3 Einsatz der Funktion zur Berechnung der Kreisfläche

Zu beachten ist dabei, was in diesem Fall nicht gleich erkennbar ist: Innerhalb einer VBA-Funktion muss der englische Funktionsname verwendet werden. Aus diesem Grund wurde in der Funktionsreferenz in den Kapiteln 3 bis 15 auch jeweils der englische Funktionsname mit angegeben. In der Klammer müssen die Argumente dann auch mit einem Komma statt mit dem sonst üblichen Semikolon getrennt werden. Enthält der Funktionsname einen Punkt, wird er durch einen Unterstrich ersetzt, etwa `VAR_P()` statt `VAR.P()`.

Die Zuordnung der Objekte kann über den Objektkatalog im VBA-Editor eingesehen werden. Die Objektmodelle haben Ähnlichkeit mit den bekannten russischen Puppen. Das oberste Objekt ist wie erwähnt die Anwendung selbst. Die anderen Objekte sind in diesem Gesamtobjekt wie in einem Container eingeschlossen. Die nächsttiefere Ebene bei Excel etwa ist die Arbeitsmappe. Innerhalb der Arbeitsmappen werden verschiedene Auflistungen – `collections` – von gleichrangigen Objekten unterschieden: Arbeitsblätter, Diagramme, Module. Die einzelnen Elemente einer solchen Auflistung können über Indizes oder über zugeordnete Namen angesprochen werden.

```
Worksheets(3)
Worksheets("Prognose")
```

sind Beispiele, wie auf ein Element einer Sammlung von Objekten zugegriffen werden kann.

Abbildung 18.4 Der Objektkatalog für Excel 2013 mit der Liste der Funktionen

Wollen Sie beispielsweise die Namen der Blätter in einer Arbeitsmappe in einer Tabelle zusammenstellen, hilft eine ganz einfache Funktion, die die Namenseigenschaft der Blätter abfragt:

```
Public Function BlattName(id As Integer) As Variant
    BlattName = Sheets(id).Name
End Function
```

Manchmal wird, was oft etwas verwirrend ist, für ein Objekt und für eine Eigenschaft dieselbe Bezeichnung verwendet. `Legend` ist z. B. eine Eigenschaft, die ein `Chart`-Objekt hat. Diese Eigenschaft liefert aber als Ergebnis wieder ein Objekt, nämlich das `Legend`-Objekt, das ein Unterobjekt eines `Chart`-Objekts ist.

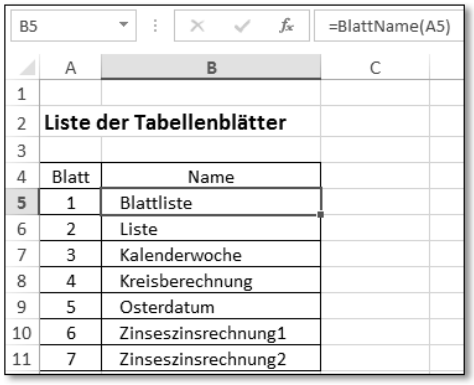


Abbildung 18.5 Liste der Blätter mit einer eigenen Funktion erstellen

18.2.2 Variablen und Konstanten in VBA

Variablen werden in VBA wie in jeder anderen Programmiersprache benutzt, um bestimmte Werte, die während des Programmablaufs entstehen, festzuhalten, so dass sie bei Bedarf an einer anderen Stelle weiterverwendet werden können. Um die verschiedenen Werte unterscheiden zu können, werden Namen für die Variablen vergeben. Zusätzlich wird festgelegt, welche Art von Informationen mit Hilfe einer bestimmten Variablen gespeichert werden soll, etwa Zahlenwerte, Texte etc. Diese Festlegung erfolgt mit Hilfe des Datentyps, der zudem auch Bedeutung für die maximale Größe oder Länge der von der Variablen festgehaltenen Information hat.

Variablen können entweder implizit oder explizit deklariert werden. Im ersten Fall wird der Datentyp `Variant` benutzt. Dieser Datentyp ist besonders flexibel, weil er Daten jeder Art akzeptiert. Allerdings muss darauf geachtet werden, welche Daten einer Variab-

len dieses Typs tatsächlich zugeordnet werden, bevor damit weitergearbeitet werden kann.

Implizit wird eine Variable direkt innerhalb einer Funktion oder Prozedur durch Wertzuweisung definiert. So werden beispielsweise durch die beiden Zeilen

```
wert1 = 3
wert2 = "xyz"
```

zwei Variablen mit den Namen `wert1` und `wert2` definiert und zugleich mit den Werten `3` bzw. `xyz`, also einer Zahl und einer Zeichenkette, belegt. Diese einfache Art der Variablendefinition ist sehr bequem. Sie definieren eine Variable genau dann, wann Sie sie benötigen, können also intuitiv arbeiten. Es ist nicht festgelegt, welcher Datentyp (Zahl, Text etc.) für die Variable gilt, d. h., die Definition ist sehr flexibel. Ein und dieselbe Variable könnte innerhalb einer Prozedur einmal für eine Zahl und einmal für eine Zeichenkette usw. verwendet werden.

18.2.3 Explizite Deklaration

Die Alternative zu dem impliziten Verfahren ist die ausdrückliche Deklaration einer Variablen unter Verwendung etwa der `Dim`-Anweisung. In diesem Fall wird der Datentyp ausdrücklich festgelegt, es sei denn, es wird der Typ `Variant` verwendet.

```
Dim text1 As String
Dim zahl1 As Double
text1 = ""
zahl1 = 0
```

Zunächst wird also eine Variable mit der `Dim`-Anweisung definiert. Anschließend wird ihr ein Wert zugeordnet.

Als Variablentypen stellt VBA folgende Möglichkeiten zur Verfügung:

Datentyp	Beschreibung
`Byte`	0–255
`Boolean`	Wahrheitswerte (`True` oder `False`). Statt der Wahrheitswerte werden auch Zahlen akzeptiert, wobei 0 `False` und alle anderen Zahlen `True` ergeben.

Tabelle 18.1 mögliche Variablentypen

Datentyp	Beschreibung
`Integer`	ganze Zahlen von –32768 bis 32767
`Long`	ganze Zahlen von ca. –2 Mrd. bis +2 Mrd.
`Single`	Gleitkommazahlen von 1,4E-45 bis 3,4E-38, positive und negative Werte
`Double`	Gleitkommazahlen von 4,9E-324 bis 1,79E-308, positive und negative Werte
`Decimal`	nur als Untertyp von `Variant` möglich: max. 28 Stellen vor und nach dem Komma
`Currency`	Festkommazahlen von –9,22E-15 bis 9,22E-15 auf vier Stellen hinter dem Komma gerundet
`String`	Zeichenfolge mit variabler Länge (bis ca. 2 Milliarden Zeichen)
`String * Length`	Zeichenfolge mit festgelegter Länge; längere Folgen werden auf die vorgegebene Länge gestaucht, und zwar 1 bis 65536 Zeichen.
`Date`	Serielle Zahlen (wie in den Tabellenfunktionen für Datum und Zeit) oder Datums- und Zeitangaben, die zwischen #-Zeichen geschrieben werden, wie `#1 Jan 99#`.
`Object`	Verweise auf ein Objekt
`Variant`	Variabler Datentyp, der anstelle der spezifischen Datentypen verwendet werden kann.

Tabelle 18.1 mögliche Variablentypen (Forts.)

Variablen können auch als Arrays definiert werden, die über Indizes angesprochen werden. Das gestattet es, mit einer Variablendefinition gleich eine große Anzahl gleichartiger Variablen festzulegen. Dies ist natürlich nur sinnvoll, wenn es sich tatsächlich um Arrays gleichartiger Variablen handelt, vergleichbar etwa einer Matrix in Excel. Die einzelnen Elemente eines Arrays werden über die Indizes angesprochen.


```
Dim gebiete() As String
gebiete = Array("Nord","Ost","Süd","West")
region = gebiet(0)
```

Ob die Indizes mit `0` oder `1` beginnen, kann generell mit dem Befehl `Option Base` festgelegt werden. Vorgabe ist `0`. Die Anweisung `Option Base 1` muss am Anfang eines Moduls noch vor der Deklaration von Datenfeldern eingetragen werden.

Alle Variablen haben einen Geltungsbereich, mit dem festgelegt wird, für welche Teile des Programms sie gelten und wann sie wieder aus dem Speicher gelöscht werden. Eine Variable, die innerhalb einer Funktion deklariert wird, wird normalerweise auch nur innerhalb dieser Funktion verwendet.

18.2.4 Zum Einsatz von Operatoren

In vielen Funktionen werden Operatoren verwendet, die weitgehend identisch mit denen sind, die bei der Bildung von Formeln in den Excel-Tabellen eingesetzt werden. Da sie aber nicht völlig identisch sind und da bei der Aufstellung von Bedingungen für den Programmablauf häufig Operatoren benötigt werden, folgt hier ein kurzer Überblick. VBA unterscheidet zwischen arithmetischen, logischen, Vergleichs- und Verkettungsoperatoren.

Arithmetische Operatoren: +, –, *, / und ^ sind identisch mit den in Tabellen verwendeten Operatoren. \ ist der Operator für die ganzzahlige Division (5\3 liefert 1), `Mod` liefert den Rest einer derartigen Division (5 Mod 3 liefert 2).

Logische Operatoren: Die logischen Operatoren dienen (mit der Ausnahme von `Not`) der Verknüpfung von Ausdrücken, die Wahrheitswerte liefern, und führen selbst wieder zu Wahrheitswerten. In einer Tabelle sollen `a` und `b` Ausdrücke sein, die Wahrheitswerte liefern. Für die Operatoren gilt dann:

- `Not a` ist wahr, wenn `a` falsch, und falsch, wenn `a` wahr ist.
- `a Eqv b` (`a` ist äquivalent `b`) ist wahr, wenn `a` und `b` den gleichen Wahrheitswert haben.
- `a Imp b` (`a` impliziert `b`) ist wahr, wenn `b` wahr ist oder wenn `a` falsch ist.
- `a Or b` ist wahr, wenn mindestens einer der beiden Ausdrücke wahr ist.
- `a And b` ist wahr, wenn beide Ausdrücke wahr sind.
- `a Xor b` (`a` exklusiv-oder `b`) ist wahr, wenn genau einer der beiden Ausdrücke wahr ist.

Wenn die logischen Operatoren in Kombinationen verwendet werden, müssen die zusammengehörenden Elemente eingeklammert werden (ähnlich wie bei arithmetischen Operatoren).

Vergleichsoperatoren: Sie dienen dem Vergleich von Werten. Sie sind identisch mit den in Tabellen verwendeten: `<`, `>`, `<=`, `>=`, `=` und `<>`. Sie stehen für kleiner, größer, klei-

ner/gleich, größer/gleich, gleich und ungleich. Bei Zahlen ist die Bedeutung klar, bei Zeichenfolgen heißt > später in der alphabetischen Reihenfolge (`a > b` ist demnach falsch). Zusätzlich können noch die Operatoren `Like` und `Is` verwendet werden. `Like` wird für einen Mustervergleich verwendet, etwa:

`Text = "Hanna" Like "Ha*a"` ergibt `True`.

`Is` wird verwendet, um festzustellen, ob zwei Objektverweise auf dasselbe Objekt verweisen:

```
Test = Objekt1 Is Objekt2
```

Verkettungsoperatoren: & und +. Mit & lassen sich (wie in Tabellen) Zeichenketten verknüpfen. + sollte als Verkettungsoperator möglichst nicht verwendet werden, um Verwechslungen mit dem Additionsoperator zu vermeiden.

18.2.5 Anweisungen und VBA-Funktionen

Anweisungen steuern, ganz allgemein gesagt, den Programmablauf. Die Deklarierung und Definition von Variablen, die Festlegung von Anfang und Ende einer Prozedur oder Funktion, die Konstruktion von Verzweigungen und Schleifen – dies alles wird über Anweisungen erledigt.

Neben den schon angesprochenen Funktionen, die Sie selbst programmieren können, bietet VBA noch eine große Zahl von integrierten Funktionen. Auch diese Funktionen dienen dazu, Werte zu liefern oder umzuwandeln. Ein Teil dieser Funktionen leistet Ähnliches wie die entsprechenden Tabellenfunktionen, etwa die mathematischen Funktionen: `Abs()`, `Cos()`, `Exp()`, `Log()` etc.

Im Allgemeinen werden an eine Funktion Werte übergeben, und die Funktion liefert jeweils einen Wert oder ein Array von Werten zurück. Hierher gehören neben den mathematischen Funktionen Funktionen, die Datums- und Zeitwerte verarbeiten, Funktionen, die Zeichenketten manipulieren oder auswerten usw.

18.2.6 Mit Verzweigungen und Schleifen Abläufe steuern

Wenn eine Funktion auf unterschiedliche Argumente unterschiedlich reagieren soll, sind Verzweigungen notwendig. Soll dagegen ein Arbeitsschritt in einer bestimmten Häufigkeit durchgeführt werden, sind Schleifen angebracht. VBA stellt hier verschiedene Kontrollstrukturen für solche Abläufe zur Verfügung:

If – Then

Eine häufige Aufgabe besteht darin, auf das Eintreten einer Situation zu reagieren. Diese Aufgabe wird in VBA im einfachsten Fall mit einer Wenn-dann-Struktur gelöst. Soll beispielsweise eine Funktion ein optionales Argument enthalten, kann mit einer solchen Verzweigung geregelt werden, was zu geschehen hat, wenn das Argument nicht angegeben wird, etwa die Zuordnung eines vorgegebenen Werts:

```
Function Woche(Optional d As Date)
If d = 0 Then
  d = Date
End If
```

Derartige Strukturen können mit einem `Else`-Zweig erweitert werden. Hierfür wieder ein Beispiel aus einer Funktion, die weiter unten noch vorgestellt wird:

```
If d7 <= 31 Then
  Datum = CDate(CStr(d7) & ". 3. " & CStr(Jahr))
Else
  Datum = DateValue(CStr(d7 - 31) & ". 4. " & CStr(Jahr))
End If
```

Bei Bedarf können `If-Then`-Strukturen auch verschachtelt werden, allerdings kann dann auch schnell die Übersicht verloren gehen. Bei komplexen Verzweigungen sollten Sie deshalb prüfen, ob sich das Problem nicht mit der folgenden Struktur lösen lässt.

Select Case

Wenn die Verzweigung davon abhängt, welchen Wert ein Ausdruck annimmt, können Sie mit einer Struktur arbeiten, die für verschiedene Werte eines Ausdrucks unterschiedliche Reaktionen anbietet.

Beim folgenden Beispiel, das weiter unten in einer finanzmathematischen Funktion verwendet wird, kann der Benutzer unterschiedliche Perioden für die Berechnung auswählen. Die Funktion reagiert darauf mit einer `Select Case`-Verzweigung, die sehr übersichtlich ist:

```
Select Case periode
      Case "Jahre"
         fn = 1
      Case "Monate"
         fn = 1 / 12
```

```
      Case "Tage"
         fn = 1 / 360
End Select
```

While

Eine oft beim Programmieren auftretende Aufgabe besteht darin, dass ein Programmteil so lange ablaufen soll, bis eine bestimmte Bedingung eintritt. Das ist z. B. der Fall, wenn Benutzereingaben in eine Liste oder eine Feldvariable eingelesen werden sollen, bis der Benutzer den Vorgang beendet, oder wenn eine Operation so lange durchgeführt werden soll, bis ein bestimmter Wert erreicht ist.

```
While diff < 0
    versuch = nomrzins(m, erg, bab)
    diff = versuch - nz
    erg = erg + 0.1
Wend
```

Diese Schleife wird so lange durchlaufen, solange die Variable `diff` einen Wert < `0` hat. Innerhalb der Schleife stehen alle Anweisungen, die für diesen Fall auszuführen sind.

Do – Loop

Besonders elegant lassen sich derartige Aufgaben mit einer ähnlichen, aber flexibleren Struktur lösen. Sie gestattet es, die Bedingung für die Schleife an den Anfang oder an das Ende der Schleife zu setzen (im vorhergehenden Beispiel muss die Bedingung am Anfang stehen). Außerdem können innerhalb der Schleife noch Abbruchbedingungen formuliert werden. Das vorhergehende Beispiel könnte so formuliert werden:

```
Do While diff < 0
    versuch = nomrzins(m, erg, bab)
    diff = versuch - nz
    erg = erg + 0.1
Loop
```

For – Next

Wenn der Fall auftritt, dass eine Anweisung oder eine Folge von Anweisungen in einer definierten Häufigkeit wiederholt werden soll, wird eine `For-Next`-Schleife verwendet. Ein Beispiel hierfür ist das Einlesen der Blattnamen in ein Array:

```
anzahl = Sheets.Count
Dim bliste() As String
ReDim bliste(anzahl)
For i = 1 To anzahl
    bliste(i) = Sheets(i).Name
Next i
```

Zunächst wird die Anzahl der Blätter an die Variable `anzahl` übergeben. Damit kann das zunächst variabel definierte Array `bliste` redimensioniert werden. Die Schleife kann dann bis zur festgestellten Anzahl durchlaufen werden, um die Blattnamen in das Array zu schreiben.

18.2.7 Einschränkungen bei Funktionen

Beachtet werden muss noch, dass in Funktionen nicht alle Visual-Basic-Schlüsselwörter auftauchen können. Es ist z.B. nicht möglich, einer Zelle über eine Funktion eine bestimmte Schrift zuzuweisen. Selbstdefinierte Funktionen können ansonsten nicht nur direkt im Tabellenblatt, sondern auch innerhalb von VBA-Programmen benutzt werden.

Erwähnt sei noch, wie verfahren wird, wenn eine Programmzeile sehr lang wird. Damit sie als eine Zeile behandelt wird, kann die Zeile auf mehrere Zeilen verteilt werden, indem jedes Mal nach einer Leertaste ein Unterstrich eingegeben wird. Danach kann die Zeile durch eine Zeilenschaltung getrennt werden.

18.3 Eine komfortablere Funktion für die Zinseszinsberechnung

Die im letzten Abschnitt vorgestellte Funktion für die Zinseszinsberechnung lässt sich noch etwas komfortabler gestalten. Dazu ist eine etwas aufwendigere Funktion erforderlich, die im Folgenden wiedergegeben wird. Genauer gesagt werden gleich zwei Funktionen angeboten: eine, die das Endkapital bei anfallenden Zinseszinsen berechnet, und eine, die den effektiven Jahreszins ausgibt.

```
Function endkapital(anfangskapital, periode, _
anzahl_perioden, zinssatz, zinszeitraum, zinsabrechnung)
   Select Case periode
      Case "Jahre"
         fn = 1
      Case "Monate"
         fn = 1 / 12
```

```
        Case "Tage"
            fn = 1 / 360
    End Select
    Select Case zinsabrechnung
        Case "jährlich"
            m = 1
        Case "halbjährlich"
            m = 2
        Case "vierteljährlich"
            m = 4
        Case "monatlich"
            m = 12
        Case "täglich"
            m = 360
    End Select
    Select Case zinszeitraum
        Case "Jahr"
            fi = 1
        Case "Monat"
            fi = 1 / 12
    End Select
    i = zinssatz
    j = (1 + i / (fi * m)) ^ m - 1
    kn = anfangskapital * (1 + j) ^ (anzahl_perioden * fn)
    endkapital = kn
End Function
```

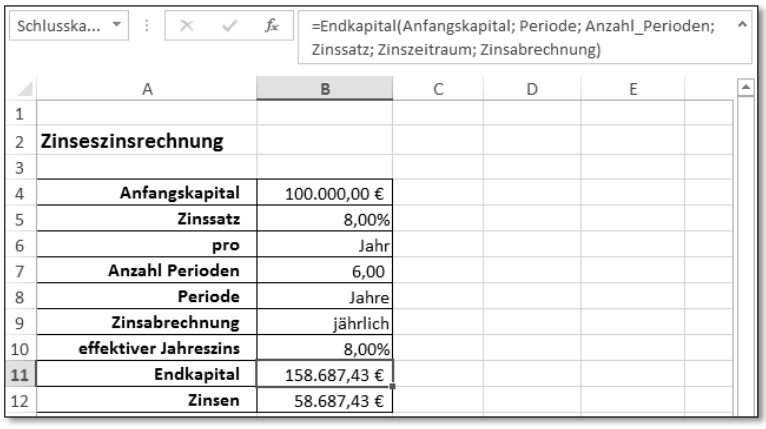


Abbildung 18.6 Zinseszinsberechnung mit variablen Abrechnungsperioden

Die zweite Funktion zur Berechnung des Effektivzinses sieht so aus:

```
Function effzins(anfangskapital, periode, anzahl_perioden, _
zinssatz, zinszeitraum, zinsabrechnung)
   Select Case periode
     Case "Jahre"
        fn = 1
     Case "Monate"
        fn = 1 / 12
     Case "Tage"
        fn = 1 / 360
   End Select
   Select Case zinsabrechnung
      Case "jährlich"
         m = 1
      Case "halbjährlich"
         m = 2
      Case "vierteljährlich"
         m = 4
      Case "monatlich"
         m = 12
      Case "täglich"
         m = 360
   End Select
   Select Case zinszeitraum
      Case "Jahr"
         fi = 1
      Case "Monat"
         fi = 1 / 12
   End Select
   i = zinssatz
   j = (1 + i / (fi * m)) ^ m - 1
   effzins = j
End Function
```

18.4 Effektivzins und Nominalzins bei Ratenkrediten

Excel hat zwar einige Funktionen, mit denen sich Annuitätenkredite berechnen lassen, z.B. die `RMZ()`-Funktion zur Berechnung der Rate. Auf die üblichen Ratenkredite für Verbraucher lassen sich diese Funktionen aber nicht direkt anwenden.

Bei den klassischen Ratenkrediten, wie sie z. B. zur Finanzierung von Möbeln oder Autos benutzt werden, wird in der Regel ein monatlicher Zinssatz angegeben, und die Zinsen werden für jeden Monat vom vollen Kreditbetrag gerechnet. Der Monatszinssatz sieht deshalb meist ziemlich klein aus, beispielsweise aktuell häufig 0,49%. Die effektive Jahresverzinsung ist aber in diesem Fall keineswegs – wie man vielleicht annehmen könnte – `0,49% * 12 = 5,88%`, sondern in Wirklichkeit 12,82%, also mehr als das Doppelte.

Die Kreditgeber sind verpflichtet, immer beide Werte – den Monatszins und den effektiven Jahreszins – anzugeben. In der Werbung geschieht das aber beispielsweise nicht immer. Deshalb hier zwei zusätzliche Funktionen, mit denen jeweils vom angegebenen Monatszins auf den effektiven Jahreszins und vom Jahreszins auf den Monatszins zurückgerechnet werden kann.

Die Funktion `NOMRZINS()` benötigt drei Argumente: die Anzahl der Monatsraten, den Effektivzins und den %-Satz der Bearbeitungsgebühren.

Die Funktion `EFFRZINS()` verwendet entsprechend die Argumente Monatsraten, Monatszins und %-Satz der Bearbeitungsgebühren. Sie ruft iterativ die `NOMRZINS()`-Funktion mehrmals auf, bis ein brauchbares Ergebnis erreicht ist.

Um die Funktionen bequem handhaben zu können, sollten die Zellen, die die Argumente liefern, entsprechend benannt werden.

Die folgende Listung zeigt die `NOMRZINS()`-Funktion:

```
Function nomrzins(m, efz, bab)
    'liefert den nominalen Monatszins aus dem effektiven Jahreszins
    'angegeben werden Laufzeit in Monaten
    'und der effektive Jahreszins als Zahl (z.B. 11,11)
    'und die Bearbeitungsgebühr in % (z.B. 2)
    Dim ergebnis As Currency
    mr = m Mod 12
    n = (m - mr) / 12
    q = 1 + efz / 100
    ba = bab / 100
    nz = q ^ n / ((q ^ n - 1) * (12 / (q - 1) + 5.5) + _
        ((1 + (mr - 1) / 24 * (q - 1)) * 12 * mr / _
        (12 + mr * (q - 1)))) - (1 + ba) / m
    ergebnis = nz * 100
    nomrzins = ergebnis
End Function
```

Die nächste Listung zeigt die EFFRZINS()-Funktion:

```
Function effrzins(m, nz, bab)
    'rechnet iterativ aus dem Nominalzins per Monat
    'den Effektivzins per Jahr mit Hilfe der Funktion nomrzins
    Dim erg As Currency
    Dim diff As Currency
    Dim versuch As Currency
    erg = nz * 12
    diff = -1
    While diff < 0
       versuch = nomrzins(m, erg, bab)
       diff = versuch - nz
       erg = erg + 1
    Wend
    erg = erg - 2
    diff = -1
    While diff < 0
       versuch = nomrzins(m, erg, bab)
       diff = versuch - nz
       erg = erg + 0.1
    Wend
    erg = erg - 0.2
    diff = -1
    While diff < 0
       versuch = nomrzins(m, erg, bab)
       diff = versuch - nz
       erg = erg + 0.01
    Wend
    erg = erg - 0.02
    diff = -1
    While diff < 0
       versuch = nomrzins(m, erg, bab)
       diff = versuch - nz
       erg = erg + 0.001
    Wend
```

```
    erg = erg - 0.002
    diff = -1
    While diff < 0
       versuch = nomrzins(m, erg, bab)
       diff = versuch - nz
       erg = erg + 0.0001
    Wend
    effrzins = erg - 0.0001
End Function
```

Die anderen Berechnungen im Tabellenblatt sind sehr einfach. In Zelle B11 wird die Gesamtsumme der Zinsen berechnet. Die Formel ergibt sich aus dem, was oben gesagt ist:

```
= Kreditbetrag * Monate * Monatszins
```

Der Betrag der Bearbeitungsgebühr ist:

```
= Kreditbetrag * %-Satz der Bearbeitungsgebühr
```

Die Gesamtbelastung ist die Summe aus Kreditbetrag, Zinsen und Bearbeitungsgebühr. Die Monatsrate kann dann einfach durch Division der Gesamtbelastung durch die Zahl der Monate errechnet werden. In der Praxis werden die Raten allerdings meist auf einen etwas glatteren Betrag gebracht, wobei die erste oder die letzte Rate dann etwas höher oder niedriger ist.

D9 =Effrzins(m; nz; bab)

	A	B	C	D
1				
2	**Nominalzins und Effektivzins von Ratenkrediten**			
3				
4	Kreditbetrag	20000		
5	Laufzeit in Monaten	36	Bearbeitungsgebühr %-Satz	2
6	Wenn der Effektive Jahreszins angegeben ist:		Wenn der Nominalzins pro Monat angegeben ist:	
7	Effektiver Jahreszins	12,82	Nominalzins pro Monat	0,49
8				
9	Nominalzins pro Monat	0,49	Effektiver Jahreszins	12,82
10				
11	Zinsen	3528,00		
12	Bearbeitungsgebühr	400,00		
13	Gesamtbelastung	23928,00		
14	Rate	664,67		

Abbildung 18.7 Berechnung eines Ratenkredits

18.5 Funktion für eine normgerechte Berechnung der Kalenderwoche

In Kapitel 4, »Datums- und Zeitfunktionen«, ist bereits angesprochen worden, dass die in Excel integrierte Funktion KALENDERWOCHE() nicht den Normen entspricht, die in Deutschland für die Berechnung der Kalenderwoche gelten. Seit Excel 2013 steht für die normgerechte Berechnung die neue Funktion ISOKALENDERWOCHE() zur Verfügung. Wenn Sie mit einer älteren Excel-Version arbeiten müssen, können Sie sich aber notfalls eine eigene Funktion zusammenbauen, deren Code im Folgenden aufgelistet wird:

```
Function Woche(Optional d As Date)
  If d = 0 Then
    d = Date
  End If
  Dim t As Long
  t = DateSerial(Year(d + (8 - Weekday(d)) Mod 7 - 3), 1, 1)
  Woche = ((d - t - 3 + (Weekday(t) + 1) Mod 7)) \ 7 + 1
End Function
```

18.6 Anlegen einer Blattliste

Bei umfangreichen Arbeitsmappen ist es häufig sinnvoll, das erste Blatt als eine Art Inhaltsverzeichnis mit Links zu den einzelnen Blättern einzurichten. Dabei wird eine Liste der aktuellen Blattnamen der Arbeitsmappe erzeugt. Der folgende Code gibt dafür eine einfache Lösung:

```
Public Sub ArbeitsblattListe()
    Dim i, s, r, anzahl As Integer
    i = 2
    r = 2
    s = 1

    If Sheets(1).Name = "Blattliste" Then
      Sheets(1).Delete
    End If

    Sheets.Add Before:=Sheets(1)
    Sheets(1).Name = "Blattliste"
```

```
    anzahl = Sheets.Count
    Cells(2, s).Value = "Blattliste der Arbeitsmappe"

    Do While i <= anzahl
        Cells(r + 2, s).Hyperlinks.Add _
        Anchor:=Cells(r + 2, s), _
        Address:="", _
        SubAddress:=Sheets(i).Name & "!A1", _
        TextToDisplay:=Sheets(i).Name
        i = i + 1
        r = r + 1
        If r > 25 Then
            r = 2
            s = s + 1
        End If
    Loop
End Sub
```

Abbildung 18.8 zeigt, wie ein solches Blatt für die in diesem Kapitel verwendeten Beispielblätter aussehen kann.

	A	B
1		
2	Blattliste der Arbeitsmappe	
3		
4	Liste	
5	Kalenderwoche	
6	Kreisberechnung	
7	Osterdatum	
8	Zinseszinsrechnung1	
9	Zinseszinsrechnung2	
10	Ratenkredit	

Abbildung 18.8 Automatisch erzeugte Blattliste

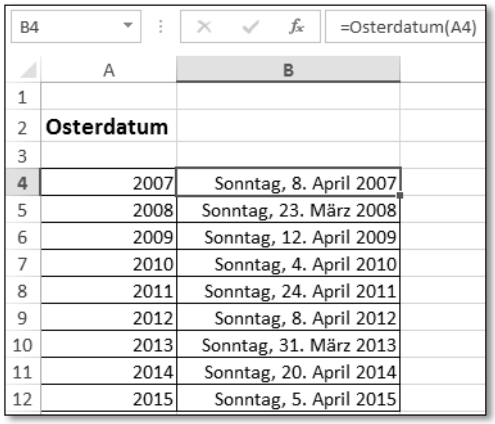
18.7 Funktion zur Berechnung des Osterdatums

Als letztes Beispiel sei hier noch eine Funktion wiedergegeben, die in der Lage ist, für eine bestimmte Jahreszahl das Datum vom Ostersonntag zu berechnen. Da sich ein Teil der jährlichen Feiertage nach dem Osterdatum richtet, lässt sich mit Hilfe dieser Funk-

tion ein größerer Teil der entsprechenden Termine berechnen, etwa um sie für Anwendungen mit den Funktionen ARBEITSTAG() oder NETTOARBEITSTAGE() zu nutzen.

Die Funktion verwendet ein Berechnungsverfahren, das bereits um das Jahr 1800 von Carl Friedrich Gauß entwickelt worden ist.

```
Function OsterDatum(Jahr) As Date
  Dim d1&, d2&, d3&, d4&, d5&, d6&, d7&
  d1 = Jahr Mod 19 + 1
  d2 = Fix(Jahr / 100) + 1
  d3 = Fix(3 * d2 / 4) - 12
  d4 = Fix((8 * d2 + 5) / 25) - 5
  d5 = Fix(5 * Jahr / 4) - d3 - 10
  d6 = (11 * d1 + 20 + d4 - d3) Mod 30
  If (d6 = 25 And d1 > 11) Or d6 = 24 Then d6 = d6 + 1
  d7 = 44 - d6
  If d7 < 21 Then d7 = d7 + 30
  d7 = d7 + 7
  d7 = d7 - (d5 + d7) Mod 7
  If d7 <= 31 Then
    OsterDatum = CDate(CStr(d7) & ". 3. " & CStr(Jahr))
  Else
    OsterDatum = DateValue(CStr(d7 - 31) & ". 4. " & CStr(Jahr))
  End If
End Function
```



	A	B
1		
2	**Osterdatum**	
3		
4	2007	Sonntag, 8. April 2007
5	2008	Sonntag, 23. März 2008
6	2009	Sonntag, 12. April 2009
7	2010	Sonntag, 4. April 2010
8	2011	Sonntag, 24. April 2011
9	2012	Sonntag, 8. April 2012
10	2013	Sonntag, 31. März 2013
11	2014	Sonntag, 20. April 2014
12	2015	Sonntag, 5. April 2015

Abbildung 18.9 Berechnen des Osterdatums

A Alphabetische Liste der Tabellenfunktionen

Funktion	Beschreibung
ABRUNDEN(**Zahl**; **Anzahl_Stellen**)	Rundet eine Zahl in Richtung 0 auf dem Zahlenstrahl.
ABS(**Zahl**)	Eliminiert die Vorzeichen von Zahlenwerten.
ACHSENABSCHNITT(**Y_Werte**; **X_Werte**)	Liefert den Ordinatenabschnitt für den Schnittpunkt der Regressionsgeraden mit der y-Achse.
ADRESSE(**Zeile**; **Spalte**; Abs; A1; Tabellenname)	Liefert die Adresse der mit Zeile und Spalte angegebenen Zelle.
AGGREGAT(**Funktion**; **Optionen**; **Array**; k)	Liefert Aggregate von Daten in Zellbereichen oder Matrizen, wobei die Art der Auswertung und die Behandlung bestimmter Werte gewählt werden können.
AGGREGAT(**Funktion**; **Optionen**; **Bezug1**; Bezug2...)	
AMORDEGRK(**Ansch_Wert**; **Kaufdatum**; **Erster_Zinstermin**; **Restwert**; **Termin**; **Satz**; Basis)	Liefert den degressiven Abschreibungsbetrag nach dem französischen Buchhaltungssystem.
AMORLINEARK(**Ansch_Wert**; **Kaufdatum**; **Erster_Zinstermin**; **Restwert**; **Termin**; Satz; Basis)	Liefert den linearen Abschreibungsbetrag nach dem französischen Buchhaltungssystem.
ANZAHL(**Wert1**; Wert2; ...)	Berechnet die Anzahl der numerischen Werte.
ANZAHL2(**Wert1**; Wert2; ...)	Ermittelt die Anzahl der Werte, wobei Texte und logische Werte mitgezählt werden.
ANZAHLLEEREZELLEN(**Bereich**)	Ermittelt die Anzahl der leeren Zellen im Bereich.
ARABISCH(**Text**)	Konvertiert eine römische Zahl in eine arabische Zahl.
ARBEITSTAG(**Ausgangsdatum**; **Tage**; Freie_Tage)	Berechnet ein neues Datum aufgrund der angegebenen Arbeitstage und freien Tage.

Funktion	Beschreibung
ARBEITSTAG.INTL(**Ausgangsdatum**; **Tage**; Wochenende; Freie_Tage)	Berechnet ein neues Datum aufgrund der angegebenen Arbeitstage und freien Tage, wobei die Wochenendtage frei gewählt werden können.
ARCCOS(**Zahl**)	Liefert zu einem Kosinuswert den Winkel im Bogenmaß.
ARCCOSHYP(**Zahl**)	Liefert den umgekehrten hyperbolischen Kosinus.
ARCCOT(**Zahl**)	Liefert den Arkuskotangens einer Zahl im Bogenmaß.
ARCCOTHYP(**Zahl**)	Liefert den umgekehrten hyperbolischen Kotangens einer Zahl.
ARCSIN(**Zahl**)	Liefert zu einem Sinuswert den Winkel im Bogenmaß.
ARCSINHYP(**Zahl**)	Umkehrfunktion zu der Funktion SINHYP()
ARCTAN(**Zahl**)	Errechnet zu einem Tangenswert den Winkel im Bogenmaß.
ARCTAN2(**x_Koordinate**; **y_Koordinate**)	Ermittelt den Arkustangens direkt aus den xy-Koordinaten.
ARCTANHYP(**Zahl**)	Umkehrfunktion zu TANHYP()
AUFGELZINS(**Emission**; **Erster_Zinstermin**; **Abrechnung**; **Satz**; **Nennwert**; **Häufigkeit**; **Basis**; Berechnungsmethode)	Liefert für Wertpapiere die aufgelaufenen Zinsen.
AUFGELZINSF(**Emission**; **Abrechnung**; **Nominalzins**; Nennwert; Basis)	Liefert für Wertpapiere die bei Fälligkeit aufgelaufenen Zinsen.
AUFRUNDEN(**Zahl**; **Anzahl_Stellen**)	Rundet eine Zahl von der Null weg.
AUSZAHLUNG(**Abrechnung**; **Fälligkeit**; **Anlage**; **Disagio**; Basis)	Liefert den Rückzahlungsbetrag eines festverzinslichen Wertpapiers.
BAHTEXT(**Zahl**)	Wandelt eine Zahl in Text in dem thailändischen Währungsformat um.
BASIS(**Zahl**; **Basis**; Mindestlänge)	Konvertiert eine Zahl in eine Textdarstellung mit der angegebenen Basis.

Funktion	Beschreibung
BEREICH.VERSCHIEBEN(**Bezug**; **Zeilen**; **Spalten**; Höhe; Breite)	Liefert einen Bereichsbezug, der um eine Anzahl von Zeilen und Spalten verschoben ist.
BEREICHE(**Bezug**)	Ermittelt die Anzahl der Bereiche, die in Bezug enthalten sind.
BESSELI(**x**; **n**)	Liefert die modifizierte Besselfunktion In(x).
BESSELJ(**x**; **n**)	Liefert die Besselfunktion Jn(x).
BESSELK(**x**; **n**)	Liefert die modifizierte Besselfunktion Kn(x).
BESSELY(**x**; **n**)	Liefert die Besselfunktion Yn(x) zurück.
BESTIMMTHEITSMASS(**Y_Werte**; **X_Werte**)	Ermittelt das Quadrat des Pearsonschen Korrelationskoeffizienten.
BETA.INV(**Wahrsch**; **Alpha**; **Beta**; A; B)	Liefert Quantile einer Betaverteilung.
BETA.VERT(**x**; **Alpha**; **Beta**; **Kumuliert**; A; B)	Liefert die Wahrscheinlichkeitsverteilung für eine betaverteilte Zufallsvariable.
BETAINV(**Wahrsch**; **Alpha**; **Beta**; A; B)	Liefert Quantile einer Betaverteilung.
BETAVERT(**x**; **Alpha**; **Beta**; A; B)	Liefert die Wahrscheinlichkeitsverteilung für eine betaverteilte Zufallsvariable.
BININDEZ(**Zahl**)	Wandelt einen binären Wert in eine Dezimalzahl um.
BININHEX(**Zahl**; Stellen)	Wandelt einen binären Wert in eine Hexadezimalzahl um.
BININOKT(**Zahl**; Stellen)	Wandelt einen binären Wert in eine Oktalzahl um.
BINOM.INV(**Versuche**; **Erfolgswahrsch**; **Alpha**)	Liefert die kleinste Anzahl erfolgreicher Versuche, für die die kumulierten Wahrscheinlichkeiten größer oder gleich einer Grenzwahrscheinlichkeit sind.
BINOM.VERT(**Zahl_Erfolge**; **Versuche**; **Erfolgswahrsch**; **Kumuliert**)	Liefert die Wahrscheinlichkeit von Zufallsvariablen bei einer Binomialverteilung.

Funktion	Beschreibung
BINOM.VERT.BEREICH(**Versuche**; **Erfolgswahrsch**; **Zahl_Erfolge**; Zahl2_Erfolge)	Gibt die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung zurück.
BINOMVERT(**Zahl_Erfolge**; **Versuche**; **Erfolgswahrsch**; **Kumuliert**)	Liefert die Wahrscheinlichkeit von Zufallsvariablen bei einer Binomialverteilung.
BITLVERSCHIEB(**Zahl**; **Verschiebebetrag**)	Liefert die Zahl, die sich ergibt, wenn Bits um den Verschiebebetrag nach links verschoben werden.
BITODER(**Zahl1**; **Zahl2**)	Gibt ein bitweises Oder zweier Zahlen zurück.
BITRVERSCHIEB(**Zahl**; **Verschiebebetrag**)	Liefert die Zahl, die sich ergibt, wenn Bits um den Verschiebebetrag nach rechts verschoben werden.
BITUND(**Zahl1**; **Zahl2**)	Gibt ein bitweises Und zweier Zahlen zurück.
BITXODER(**Zahl1**; **Zahl2**)	Gibt ein bitweises ausschließliches Oder zweier Zahlen zurück.
BLATT(Wert)	Gibt die Nummer des Blatts in einem Bezug zurück.
BLÄTTER(Bezug)	Gibt die Anzahl der Blätter in einem Bezug zurück.
BOGENMASS(**Winkel**)	Wandelt den Gradwert des angegebenen Winkels in Bogenmaß um.
BRTEILJAHRE(**Ausgangsdatum**; **Enddatum**; Basis)	Berechnet den Jahresbruchteil für die Zeitspanne zwischen Ausgangsdatum und Enddatum.
BW(**Zins**; **Zzr**; **Rmz**; Zw; F)	Berechnet den Barwert einer Investition.
CHIINV(**Wahrsch**; **Freiheitsgrade**)	Liefert Quantile der Chi-Quadrat-Verteilung.
CHIQU.INV(**Wahrsch**; **Freiheitsgrade**)	Liefert Quantile der linksseitigen Chi-Quadrat-Verteilung.
CHIQU.INV.RE(**Wahrsch**; **Freiheitsgrade**)	Liefert Quantile der rechtsseitigen Chi-Quadrat-Verteilung.

Funktion	Beschreibung
CHIQU.TEST(**Beob_Messwerte**; **Erwart_Werte**)	Liefert den Wahrscheinlichkeitswert für den Chi-Quadrat-Test beim Vergleich zwischen beobachteten und erwarteten Größen.
CHIQU.VERT(**x**; **Freiheitsgrade**; **Kumuliert**)	Liefert Werte der linksseitigen Verteilungsfunktion für eine chi-quadrat-verteilte Zufallsvariable.
CHIQU.VERT.RE(**x**; **Freiheitsgrade**)	Liefert Werte der rechtsseitigen Verteilungsfunktion für eine chi-quadrat-verteilte Zufallsvariable.
CHITEST(**Beob_Messwerte**; **Erwart_Werte**)	Liefert den Wahrscheinlichkeitswert für den Chi-Quadrat-Test.
CHIVERT(**x**; **Freiheitsgrade**)	Berechnet aus dem Wert für x und für Freiheitsgrade die Überschreitungswahrscheinlichkeit für die Übereinstimmung von beobachteten und erwarteten Werten.
CODE(**Text**)	Liefert den Code des ersten Zeichens in der Zeichenfolge.
COS(**Zahl**)	Berechnet den Kosinus des angegebenen Winkels.
COSEC(**Zahl**)	Berechnet den Kosekans eines Winkels.
COSECHYP(**Zahl**)	Berechnet den hyperbolischen Kosekans eines Winkels.
COSHYP(**Zahl**)	Liefert den hyperbolischen Kosinus.
COT(**Zahl**)	Liefert den Kotangens eines Winkels.
COTHYP(**Zahl**)	Liefert den hyperbolischen Kotangens eines Winkels.
CUBEELEMENT(**Verbindung**; **Element_Ausdruck**; Beschriftung)	Liefert ein Element oder Tupel aus einem Cube.
CUBEELEMENTEIGENSCHAFT(**Verbindung**; **Element_Ausdruck**; **Eigenschaft**)	Liefert den Wert einer Eigenschaft eines Elements im Cube.
CUBEKPIELEMENT(**Verbindung**; **KPI_Name**; **KPI_Eigenschaft**; Beschriftung)	Gibt die Eigenschaft eines Key Performance Indicator (KPI) zurück.

Funktion	Beschreibung
CUBEMENGE(**Verbindung**; **Menge_Ausdruck**; Beschriftung; Sortierreihenfolge; Sortieren_nach)	Sendet einen Mengenausdruck für einen bestimmten Satz von Elementen oder Tupeln an den Cube, der die entsprechenden Daten zusammenstellt und an Excel liefert.
CUBEMENGENANZAHL(**Menge**)	Liefert die Anzahl der Elemente in einer definierten Menge.
CUBERANGELEMENT(**Verbindung**; **Menge_Ausdruck**; **Rang**; Beschriftung)	Gibt das mit Rang angegebene Element in einer Menge zurück.
CUBEWERT(**Verbindung**; **Element_Ausdruck1**; Element_Ausdruck2 ...)	Liefert einen aggregierten Wert aus einem Cube.
DATEDIF(**Ausgangsdatum**; **Enddatum**; **Einheit**)	Berechnet die Differenz von Ausgangs- und Enddatum.
DATUM(**Jahr**; **Monat**; **Tag**)	Berechnet eine serielle Zahl für das eingegebene Datum.
DATWERT(**Datumstext**)	Wandelt die Zeichenfolge in einen Datumswert um.
DBANZAHL(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Ermittelt die Anzahl der Datensätze, die mit den Suchkriterien übereinstimmen.
DBANZAHL2(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Ermittelt die Anzahl der Datensätze, die mit den Suchkriterien übereinstimmen, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
DBAUSZUG(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Liefert den Inhalt des angegebenen Felds des Datensatzes, der mit den Suchkriterien übereinstimmt.
DBMAX(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Liefert den größten Wert in einem Datenbankfeld aus den Datensätzen, die den angegebenen Kriterien entsprechen.
DBMIN(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Liefert den kleinsten Wert in einem Datenbankfeld aus den Datensätzen, die den angegebenen Kriterien entsprechen.
DBMITTELWERT(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Liefert den Mittelwert aller Werte in einem Datenbankfeld aus den Datensätzen, die den angegebenen Kriterien entsprechen.

Funktion	Beschreibung
DBPRODUKT(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Multipliziert alle Werte eines Datenbankfelds der Datensätze, die die Suchkriterien erfüllen.
DBSTDABW(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Berechnet die Standardabweichung des Datenbankfelds für die Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Stichprobe behandelt.
DBSTDABWN(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Berechnet die Standardabweichung des Datenbankfelds für die Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Grundgesamtheit behandelt.
DBSUMME(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Summiert alle Werte eines Datenbankfelds für die Datensätze, die die Suchkriterien erfüllen.
DBVARIANZ(**Datenbank**; **Datenbankfeld**; **Suchkriterien**)	Berechnet die Varianz der Werte eines Datenbankfelds für die Datensätze, die die Suchkriterien erfüllen. Die Datensätze werden als Stichprobe behandelt.
DELTA(**Zahl1**; Zahl2)	Prüft, ob Werte gleich sind.
DEZIMAL(**Text**; **Basis**)	Konvertiert die Textdarstellung einer Zahl mit der angegebenen Basis in eine Dezimalzahl.
DEZINBIN(**Zahl**; Stellen)	Liefert den Binärwert einer Dezimalzahl.
DEZINHEX(**Zahl**; Stellen)	Liefert den Hexadezimalwert einer Dezimalzahl.
DEZINOKT(**Zahl**; Stellen)	Liefert den Oktalwert einer Dezimalzahl.
DIA(**Ansch_Wert**; **Restwert**; **Nutzungsdauer**; **Zr**)	Berechnet die Abschreibungsrate nach der digitalen Methode.
DISAGIO(**Abrechnung**; **Fälligkeit**; **Kurs**; **Rückzahlung**; Basis)	Berechnet den Disagio beim Handel mit einem Wertpapier.
DM(**Zahl**; Dezimalstellen)	Wandelt den Wert in eine Zeichenfolge im aktuellen Währungsformat um.
DURATION(**Abrechnung**; **Fälligkeit**; **Nominalzins**; **Rendite**; **Häufigkeit**; Basis)	Liefert die hypothetische durchschnittliche Kapitalbindungsdauer eines festverzinslichen Wertpapiers.

Funktion	Beschreibung
EDATUM(**Ausgangsdatum**; **Monate**)	Berechnet ein um Monate verschobenes Datum.
EFFEKTIV(**Nominalzins**; **Perioden**)	Berechnet den effektiven Jahreszins für eine Anlage oder ein Darlehen.
ERSETZEN(**Alter_Text**; **Erstes_Zeichen**; **Anzahl_Zeichen**; **Neuer_Text**)	Ersetzt ab einer bestimmten Stelle eine Anzahl Zeichen in einer Zeichenfolge durch neue Zeichen.
EXP(**Zahl**)	Bewirkt, dass die Eulersche Zahl e (2,71828...) mit dem angegebenen Wert potenziert wird.
EXPON.VERT(**x**; **Lambda**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für eine exponentialverteilte Zufallsvariable.
EXPONVERT(**x**; **Lambda**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für eine exponentialverteilte Zufallsvariable.
F.INV(**Wahrsch**; **Freiheitsgrade1**; **Freiheitsgrade2**)	Liefert Quantile der linksseitigen F-Verteilung.
F.INV.RE(**Wahrsch**; **Freiheitsgrade1**; **Freiheitsgrade2**)	Liefert Quantile der rechtsseitigen F-Verteilung.
F.TEST(**Matrix1**; **Matrix2**)	Liefert die Wahrscheinlichkeit der Übereinstimmung zweier Stichproben hinsichtlich ihrer Varianzen.
F.VERT(**x**; **Freiheitsgrade1**; **Freiheitsgrade2**; **Kumuliert**)	Liefert Werte der Verteilungsfunktion (1-Alpha) einer linksseitigen F-verteilten Zufallsvariablen.
F.VERT.RE(**x**; **Freiheitsgrade1**; **Freiheitsgrade2**)	Liefert Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitigen F-verteilten Zufallsvariablen.
FAKULTÄT(**Zahl**)	Liefert die Fakultät der angegebenen Zahl.
FALSCH()	Legt den Wahrheitswert FALSCH in der Zelle ab.
FEHLER.TYP(**Fehlerwert**)	Liefert die Fehlernummer für den vorliegenden Fehler.
FEST(**Zahl**; Dezimalstellen; Keine_Punkte)	Wandelt einen numerischen Wert in eine Zeichenfolge um.

Funktion	Beschreibung
FINDEN(**Suchtext**; **Text**; Erstes_Zeichen)	Überprüft, ob die mit Suchtext angegebene Zeichenfolge in einer anderen Zeichenfolge vorkommt.
FINV(**Wahrsch**; **Freiheitsgrade1**; **Freiheitsgrade2**)	Liefert Quantile der F-Verteilung.
FISHER(**x**)	Liefert die Fisher-Transformation eines Korrelationskoeffizienten.
FISHERINV(**y**)	Umkehrfunktion zu FISHER()
FORMELTEXT(**Bezug**)	Gibt die in der mit Bezug angegebenen Zelle enthaltene Formel als Zeichenfolge zurück.
FTEST(**Matrix1**; **Matrix2**)	Liefert die Wahrscheinlichkeit der Übereinstimmung zweier Stichproben hinsichtlich ihrer Varianzen.
FVERT(**x**; **Freiheitsgrade1**; **Freiheitsgrade2**)	Liefert Werte der Verteilungsfunktion (1-Alpha) einer F-verteilten Zufallsvariablen.
G.TEST(**Matrix**; **x**; Sigma)	Liefert die einseitige Wahrscheinlichkeit für einen Gauß-Test.
GAMMA(**x**)	Liefert den Wert der Gammafunktion.
GAMMA.INV(**Wahrsch**; **Alpha**; **Beta**)	Liefert Quantile der Gammaverteilung.
GAMMA.VERT(**x**; **Alpha**; **Beta**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für eine gammaverteilte Zufallsvariable.
GAMMAINV(**Wahrsch**; **Alpha**; **Beta**)	Liefert Quantile der Gammaverteilung.
GAMMALN(**x**)	Liefert den natürlichen Logarithmus zur Gammafunktion.
GAMMALN.GENAU(**x**)	Liefert den natürlichen Logarithmus zur Gammafunktion.
GAMMAVERT(**x**; **Alpha**; **Beta**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für eine gammaverteilte Zufallsvariable.
GANZZAHL(**Zahl**)	Reduziert einen numerischen Ausdruck auf die nächstkleinere ganze Zahl.
GAUSS(**x**)	Gibt 0,5 weniger als die kumulierte Normalverteilung zurück.

Funktion	Beschreibung
GAUSSF.GENAU(**x**)	Liefert Werte des Gaußschen Fehlerintegrals.
GAUSSFEHLER(**Untere_Grenze**; Obere_Grenze)	Liefert Werte des Gaußschen Fehlerintegrals.
GAUSSFKOMPL(**Untere_Grenze**)	Liefert komplementäre Werte des Gaußschen Fehlerintegrals.
GAUSSFKOMPL.GENAU(**x**)	Liefert komplementäre Werte des Gaußschen Fehlerintegrals.
GDA(**Ansch_Wert**; **Restwert**; **Nutzungsdauer**; **Periode**; Faktor)	Berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der geometrisch-degressiven Methode.
GDA2(**Ansch_Wert**; **Restwert**; **Nutzungsdauer**; **Periode**; Monate)	Berechnet die Abschreibungsrate für einen bestimmten Abschreibungszeitraum nach der geometrisch-degressiven Methode.
GEOMITTEL(**Zahl1**; Zahl2; ...)	Berechnet das geometrische Mittel.
GERADE(**Zahl**)	Rundet Zahlen zur nächsten geraden Ganzzahl.
GESTUTZTMITTEL(**Matrix**; **Prozent**)	Berechnet das arithmetische Mittel, wobei Extremwerte nicht berücksichtigt werden.
GGANZZAHL(**Zahl**; Schritt)	Prüft, ob der mit Zahl angegebene Wert den mit Schritt angegebenen Schwellenwert erreicht oder überschreitet.
GGT(**Zahl1**; Zahl2; ...)	Liefert den größten gemeinsamen Teiler von ganzen Zahlen.
GLÄTTEN(**Text**)	Entfernt führende Leerzeichen und überzählige Leerzeichen.
GRAD(**Winkel**)	Rechnet aus Bogenmaß in Grad um.
GROSS(**Text**)	Wandelt alle Buchstaben aus der mit Text angegebenen Zeichenfolge in Großbuchstaben um.

Funktion	Beschreibung
GROSS2(**Text**)	Wandelt den ersten Buchstaben in Großbuchstaben, den Rest in Kleinbuchstaben um.
GTEST(**Matrix**; **x**; Sigma)	Liefert die einseitige Wahrscheinlichkeit für einen Gauß-Test.
HARMITTEL(**Zahl1**; Zahl2; ...)	Berechnet das harmonische Mittel.
HÄUFIGKEIT(**Daten**; **Klassen**)	Liefert die Häufigkeit innerhalb der durch Klassen definierten Intervalle.
HEUTE()	Liefert das Systemdatum.
HEXINBIN(**Zahl**; Stellen)	Liefert den Binärwert einer hexadezimalen Zahl.
HEXINDEZ(**Zahl**)	Liefert den Dezimalwert einer hexadezimalen Zahl.
HEXINOKT(**Zahl**; Stellen)	Liefert den Oktalwert einer hexadezimalen Zahl.
HYPERLINK(**Hyperlink_Adresse**; Freundlicher_Name)	Erstellt eine Verknüpfung zu der angegebenen Adresse.
HYPGEOM.VERT(**Erfolge_S**; **Umfang_S**; **Erfolge_G**; **Umfang_G**; **Kumuliert**)	Berechnet Wahrscheinlichkeiten einer hypergeometrisch verteilten Zufallsvariablen.
HYPGEOMVERT(**Erfolge_S**; **Umfang_S**; **Erfolge_G**; **Umfang_G**)	Berechnet Wahrscheinlichkeiten einer hypergeometrisch verteilten Zufallsvariablen.
IDENTISCH(**Text1**;**Text2**)	Vergleicht zwei Zeichenfolgen daraufhin, ob sie exakt übereinstimmen.
IKV(**Werte**; Schätzwert)	Berechnet den internen Ertragszins einer Investition.
IMABS(**Komplexe_Zahl**)	Liefert den Absolutwert (Modul) einer komplexen Zahl.
IMAGINÄRTEIL(**Komplexe_Zahl**)	Liefert den imaginären Anteil einer komplexen Zahl.
IMAPOTENZ(**Komplexe_Zahl**; **Potenz**)	Liefert die Potenz einer komplexen Zahl.
IMARGUMENT(**Komplexe_Zahl**)	Liefert das Argument einer komplexen Zahl.

Funktion	Beschreibung
IMCOS(**Komplexe_Zahl**)	Liefert den Kosinus einer komplexen Zahl.
IMCOSEC(**Komplexe_Zahl**)	Liefert den Kosekans einer komplexen Zahl.
IMCOSECHYP(**Komplexe_Zahl**)	Liefert den hyperbolischen Kosekans einer komplexen Zahl.
IMCOSHYP(**Komplexe_Zahl**)	Liefert den hyperbolischen Kosinus einer komplexen Zahl.
IMCOT(**Komplexe_Zahl**)	Liefert den Kotangens einer komplexen Zahl.
IMDIV(**Komplexe_Zahl1**; **Komplexe_Zahl2**)	Division komplexer Zahlen
IMEXP(**Komplexe_Zahl**)	Potenzierung von e (Eulersche Zahl) mit einer komplexen Zahl
IMKONJUGIERTE(**Komplexe_Zahl**)	Liefert zu einer komplexen Zahl das konjugierte komplexe Komplement.
IMLN(**Komplexe_Zahl**)	Liefert den natürlichen Logarithmus einer komplexen Zahl.
IMLOG10(**Komplexe_Zahl**)	Liefert den dekadischen Logarithmus einer komplexen Zahl.
IMLOG2(**Komplexe_Zahl**)	Liefert den binären Logarithmus einer komplexen Zahl.
IMPRODUKT(**Komplexe_Zahl1**; Komplexe_Zahl2, ...)	Berechnet das Produkt komplexer Zahlen.
IMREALTEIL(**Komplexe_Zahl**)	Liefert den reellen Anteil einer komplexen Zahl.
IMSEC(**Komplexe_Zahl**)	Liefert den Sekans einer komplexen Zahl.
IMSECHYP(**Komplexe_Zahl**)	Liefert den hyperbolischen Sekans einer komplexen Zahl.
IMSIN(**Komplexe_Zahl**)	Liefert den Sinus einer komplexen Zahl.
IMSINHYP(**Komplexe_Zahl**)	Liefert den hyperbolischen Sinus einer komplexen Zahl.
IMSUB(**Komplexe_Zahl1**; **Komplexe_Zahl2**)	Subtraktion zweier komplexer Zahlen

Funktion	Beschreibung
IMSUMME(**Komplexe_Zahl1**; Komplexe_Zahl2; ...)	Addition komplexer Zahlen
IMTAN(**Komplexe_Zahl**)	Liefert den Tangens einer komplexen Zahl.
IMWURZEL(**Komplexe_Zahl**)	Liefert die Quadratwurzel einer komplexen Zahl.
INDEX(**Bezug**; Zeile; Spalte; Bereich)	Liefert einen Wert aus einem Bereich.
INDEX(**Matrix**; Zeile; Spalte)	Liefert einen Wert aus einer Matrix.
INDIREKT(**Bezug**; A1)	Ermittelt den Inhalt einer Zelle, auf die in einer anderen Zelle verwiesen wird.
INFO(**Typ**)	Liefert Informationen über die Betriebssystemumgebung.
ISOKALENDERWOCHE(**Datum**)	Liefert Zahl der Kalenderwoche nach dem ISO-Standard.
ISPMT(**Rate**; **Per**; **Nper**; **Pv**)	Berechnet die Zinsen, die während einer bestimmten Periode gezahlt werden.
ISTBEZUG(**Wert**)	Testet, ob Wert ein gültiger Bereichsname oder eine Bereichsadresse ist.
ISTFEHL(**Wert**)	Testet, ob Wert ein Fehlerwert ungleich #NV ist.
ISTFEHLER(**Wert**)	Testet, ob Wert ein Fehlerwert ist.
ISTFORMEL(**Bezug**)	Testet, ob die mit Bezug angegebene Zelle eine Formel enthält.
ISTGERADE(**Zahl**)	Testet, ob Wert durch 2 ganzzahlig teilbar ist.
ISTKTEXT(**Wert**)	Testet, ob in einer Zelle kein Text vorhanden ist.
ISTLEER(**Wert**)	Testet, ob eine Zelle leer ist.
ISTLOG(**Wert**)	Überprüft, ob das Argument Wert einen Wahrheitswert liefert.
ISTNV(**Wert**)	Überprüft, ob eine Zelle den Fehlerwert #NV enthält.
ISTTEXT(**Wert**)	Prüft, ob Wert eine Zeichenfolge liefert.

Funktion	Beschreibung
ISTUNGERADE(**Zahl**)	Testet, ob Wert nicht durch 2 ganzzahlig teilbar ist.
ISTZAHL(**Wert**)	Prüft, ob Wert eine Zahl liefert.
JAHR(**Zahl**)	Liefert das Jahr für das eingegebene Datum.
JETZT()	Liefert das Systemdatum und die Uhrzeit.
KALENDERWOCHE(**Fortlaufende_Zahl**; Zahl_Typ)	Liefert für das angegebene Datum die entsprechende Kalenderwoche.
KAPZ(**Zins**; **Zr**; **Zzr**; **Bw**; Zw; F)	Berechnet den Tilgungsanteil für die Abzahlung eines Annuitätendarlehens.
KGRÖSSTE(**Matrix**; **k**)	Gibt den k-größten Wert zurück.
KGV(**Zahl1**; Zahl2; ...)	Liefert das kleinste gemeinsame Vielfache.
KKLEINSTE(**Matrix**; **k**)	Gibt den k-kleinsten Wert zurück.
KLEIN(**Text**)	Wandelt alle Buchstaben in Kleinbuchstaben um.
KOMBINATIONEN(**n**; **k**)	Gibt den Wert des Binomialkoeffizienten aus.
KOMBINATIONEN2(**Zahl**; **gewählte_Zahl**)	Berechnet die Anzahl der Kombinationen mit Wiederholung für eine angegebene Anzahl von Elementen.
KOMPLEXE(**Realteil**; **Imaginärteil**; Suffix)	Erzeugt eine komplexe Zahl aus zwei reellen Zahlen.
KONFIDENZ(**Alpha**; **Standabwn**; **Umfang_S**)	Berechnet das Konfidenzintervall.
KONFIDENZ.NORM(**Alpha**; **Standardabwn**; **Umfang**)	Berechnet das Konfidenzintervall.
KONFIDENZ.T(**Alpha**; **Standardabwn**; **Umfang**)	Berechnet das Konfidenzintervall einer t-verteilten Grundgesamtheit.
KORREL(**Matrix1**; **Matrix2**)	Liefert den Korrelationskoeffizienten zweier Datenreihen.
KOVAR(**Matrix1**; **Matrix2**)	Liefert ein Maß für den Zusammenhang zwischen den Daten zweier Datenreihen.

Funktion	Beschreibung
KOVARIANZ.P(**Array1**; **Array2**)	Liefert ein Maß für den Zusammenhang zwischen den Daten zweier Datenreihen.
KOVARIANZ.S(**Array1**; **Array2**)	Liefert für Stichproben ein Maß für den Zusammenhang zwischen den Daten zweier Datenreihen.
KRITBINOM(**Versuche**; **Erfolgswahrsch**; **Alpha**)	Liefert die kleinste Anzahl erfolgreicher Versuche, für die die kumulierten Wahrscheinlichkeiten größer oder gleich einer Grenzwahrscheinlichkeit sind.
KUMKAPITAL(**Zins**; **Zzr**; **Bw**; **Zeitraum_Anfang**; **Zeitraum_Ende**; **F**)	Berechnet den Betrag, der für die Tilgung eines Annuitätendarlehens in einem Zeitraum aufgebracht wird.
KUMZINSZ(**Zins**; **Zzr**; **Bw**; **Zeitraum_Anfang**; **Zeitraum_Ende**; **F**)	Berechnet den für die Zinsen aufgebrachten Betrag.
KURS(**Abrechnung**; **Fälligkeit**; **Zins**; **Rendite**; **Rückzahlung**; **Häufigkeit**; Basis)	Liefert den Kurswert eines festverzinslichen Wertpapiers.
KURSDISAGIO(**Abrechnung**; **Fälligkeit**; **Disagio**; **Rückzahlung**; Basis)	Berechnet den Ausgabekurs eines unverzinslichen Wertpapiers.
KURSFÄLLIG(**Abrechnung**; **Fälligkeit**; **Emission**; **Zins**; **Rendite**; Basis)	Berechnet den Kurswert eines festverzinslichen Wertpapiers.
KURT(**Zahl1**; Zahl2; ...)	Liefert die Kurtosis (Wölbung) einer Häufigkeitsverteilung.
KÜRZEN(**Zahl**; Anzahl_Stellen)	Kürzt auf die angegebene Stellenzahl.
LÄNGE(**Text**)	Liefert die Zahl der Zeichen in der mit Text angegebenen Zeichenfolge.
LIA(**Ansch_Wert**; **Restwert**; **Nutzungsdauer**)	Berechnet die Abschreibungsrate nach der linearen Methode (AFA).
LINKS(**Text**; Anzahl_Zeichen)	Liefert die mit Anzahl_Zeichen festgelegte Zeichenzahl vom Beginn der Zeichenfolge an.
LN(**Zahl**)	Liefert den natürlichen Logarithmus.
LOG(**Zahl**; Basis)	Liefert den Logarithmus zur angegebenen Basis.
LOG10(**Zahl**)	Liefert den dekadischen Logarithmus.

Funktion	Beschreibung
LOGINV(**Wahrsch**; **Mittelwert**; **Standabwn**)	Liefert Quantile einer logarithmischen Normalverteilung.
LOGNORM.INV(**Wahrsch**; **Mittelwert**; **Standabwn**)	Liefert das Quantil einer logarithmischen Normalverteilung.
LOGNORM.VERT(**x**; **Mittelwert**; **Standabwn**; **Kumuliert**)	Liefert den Wert der Wahrscheinlichkeitsverteilung für eine logarithmische Normalverteilung.
LOGNORMVERT(**x**; **Mittelwert**; **Standabwn**)	Liefert die Wahrscheinlichkeitsverteilung für eine logarithmische Normalverteilung.
MAX(**Zahl1**; Zahl2; ...)	Liefert den größten Wert.
MAXA(**Wert1**; Wert2; ...)	Liefert den größten Wert, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
MDET(**Matrix**)	Liefert die Determinante der angegebenen Matrix.
MDURATION(**Abrechnung**; **Fälligkeit**; **Coupon**; **Rendite**; **Häufigkeit**; Basis)	Berechnet die modifizierte Duration.
MEDIAN(**Zahl1**; Zahl2; ...)	Ermittelt den Median einer Reihe von Daten.
MEINHEIT(**Größe**)	Liefert die Einheitsmatrix für die angegebene Größe.
MIN(**Zahl1**; Zahl2; ...)	Liefert den kleinsten Wert.
MINA(**Wert1**; Wert2; ...)	Liefert den kleinsten Wert, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
MINUTE(**Zahl**)	Berechnet den Minutenanteil eines Zeitwerts.
MINV(**Matrix**)	Bildet die Inverse zu einer Matrix.
MITTELABW(**Zahl1**; Zahl2; ...)	Liefert die mittlere lineare Abweichung.
MITTELWERT(**Zahl1**; Zahl2; ...)	Liefert das arithmetische Mittel.
MITTELWERTA(**Wert1**; Wert2; ...)	Liefert das arithmetische Mittel, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.

Funktion	Beschreibung
MITTELWERTWENN(**Bereich**; **Kriterien**; Mittelwert_Bereich)	Liefert das arithmetische Mittel von Werten, die einem Kriterium entsprechen.
MITTELWERTWENNS(**Mittelwert_Bereich**; **Kriterien_Bereich1**; **Kriterien1**; Kriterien_Bereich2; Kriterien2 ...)	Liefert das arithmetische Mittel von Werten, die mehreren Kriterien entsprechen.
MMULT(**Array1**; **Array2**)	Liefert das Produkt zweier Matrizen.
MODALWERT(**Zahl1**; Zahl2; ...)	Liefert den in einer Datenreihe am häufigsten vorkommenden Wert.
MODUS.EINF(**Zahl1**; Zahl2; ...)	Liefert den in einer Datenreihe am häufigsten vorkommenden Wert.
MODUS.VIELF(**Zahl1**; Zahl2; ...)	Liefert die in einer Datenreihe am häufigsten vorkommenden Werte.
MONAT(**Zahl**)	Liefert die Monatszahl eines Datumswerts.
MONATSENDE(**Ausgangsdatum**; **Monate**)	Liefert den letzten Tag des Monats, in den das um Monate verschobene Datum fällt.
MTRANS(**Matrix**)	Tauscht die Zeilen und Spalten in einer Matrix.
N(**Wert**)	Gibt Wert als einen in eine Zahl gewandelten Wert zurück.
NBW(**Zins**; **Wert1**; Wert2; ...)	Berechnet den Nettokapitalwert periodischer Cashflows.
NEGBINOM.VERT(**Zahl_Misserfolge**; **Zahl_Erfolge**; **Erfolgswahrsch**; **Kumuliert**)	Berechnet, mit welcher Wahrscheinlichkeit ein zusammengesetztes Ereignis eintritt.
NEGBINOMVERT(**Zahl_Misserfolge**; **Zahl_Erfolge**; **Erfolgswahrsch**)	Berechnet, mit welcher Wahrscheinlichkeit ein zusammengesetztes Ereignis eintritt.
NETTOARBEITSTAGE(**Ausgangsdatum**; **Enddatum**; Freie_Tage)	Berechnet die Arbeitstage zwischen zwei Daten.
NETTOARBEITSTAGE.INTL (**Ausgangsdatum**; **Enddatum**; Wochenende; Freie_Tage)	Berechnet die Arbeitstage zwischen zwei Daten, wobei die Wochenendtage frei gewählt werden können.
NICHT(**Wahrheitswert**)	Kehrt den Wert von Wahrheitswert um.

Funktion	Beschreibung
NOMINAL(**Effektiver_Zins**; **Perioden**)	Berechnet die jährliche Nominalverzinsung.
NORM.INV(**Wahrsch**; **Mittelwert**; **Standabwn**)	Liefert Quantile der Normalverteilung.
NORM.S.INV(**Wahrsch**)	Liefert Quantile der Standardnormalverteilung.
NORM.S.VERT(**z**; **Kumuliert**)	Gibt die Wahrscheinlichkeit dafür an, dass eine Zufallsvariable aus der Standardnormalverteilung den Wert z oder kleiner annimmt.
NORM.VERT(**x**; **Mittelwert**; **Standabwn**; **Kumuliert**)	Liefert Werte einer Normalverteilung.
NORMINV(**Wahrsch**; **Mittelwert**; **Standabwn**)	Liefert Quantile der Normalverteilung.
NORMVERT(**x**; **Mittelwert**; **Standabwn**; **Kumuliert**)	Liefert Werte einer Normalverteilung.
NOTIERUNGBRU(**Zahl**; **Teiler**)	Berechnet einen als Bruch interpretierbaren Ausdruck einer Dezimalzahl.
NOTIERUNGDEZ(**Zahl**; **Teiler**)	Konvertiert einen Bruch in eine Dezimalzahl.
NV()	Trägt den Fehlerwert #NV in die Zelle ein.
OBERGRENZE(**Zahl**; **Schritt**)	Rundet eine Zahl auf das kleinste Vielfache von Schritt auf.
OBERGRENZE.MATHEMATIK(**Zahl**; Schritt; Modus)	Rundet eine Zahl auf das kleinste Vielfache von Schritt auf.
ODER(**Wahrheitswert1**; Wahrheitswert2; ...)	Liefert WAHR, wenn ein Argument WAHR ist, sonst FALSCH.
OKTINBIN(**Zahl**; Stellen)	Liefert den Binärwert einer oktalen Zahl.
OKTINDEZ(**Zahl**)	Liefert den Dezimalwert einer oktalen Zahl.
OKTINHEX(**Zahl**; Stellen)	Liefert den hexadezimalen Wert einer oktalen Zahl.

Funktion	Beschreibung
PDURATION(**Zins**; **Bw**; **Zw**)	Berechnet die Anzahl der Perioden, bis eine Investition den angegebenen Zielwert erreicht hat.
PEARSON(**Matrix1**; **Matrix2**)	Liefert den Pearsonschen Korrelationskoeffizienten.
PHI(**x**)	Liefert den Wert der Dichtefunktion für eine Standardnormalverteilung.
PI()	Liefert den numerischen Wert von PI.
PIVOTDATENZUORDNEN(**Datenfeld**; **PivotTable**; Feld1; Element1; Feld2; Element2; ...)	Gibt Werte aus einer Pivot-Tabelle zurück.
POISSON(**x**; **Mittelwert**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für Zufallsvariablen, die zu einer Poisson-Verteilung gehören.
POISSON.VERT(**x**; **Mittelwert**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für Zufallsvariablen, die zu einer Poisson-Verteilung gehören.
POLYNOMIAL(**Zahl1**; Zahl2; ...)	Liefert die Fakultät der Summe der Argumente geteilt durch das Produkt der Fakultäten.
POTENZ(**Zahl**; **Potenz**)	Liefert die angegebene Potenz einer Zahl.
POTENZREIHE(**x**; **n**; **m**; **Koeffizienten**)	Liefert eine Summe von Potenzen der Zahl x.
PRODUKT(**Zahl1**; Zahl2; ...)	Multipliziert alle angegebenen Argumente miteinander.
QIKV(**Werte**; **Investition**; **Reinvestition**)	Berechnet die interne Ertragsrate einer Reihe von Ein- und Auszahlungen.
QUADRATESUMME(**Zahl1**; Zahl2; ...)	Berechnet die Summe der Quadrate der angegebenen Werte.
QUANTIL(**Matrix**; **Alpha**)	Liefert denjenigen Wert einer Datenreihe, unterhalb dessen ein mit Alpha angegebener Bruchteil der Daten liegt.

Funktion	Beschreibung
QUANTIL.EXKL(**Array**; **k**)	Liefert denjenigen Wert einer Datenreihe, unterhalb dessen ein mit k (Wertebereich zwischen 0 und 1 und ausschließlich 0 und 1) angegebener Bruchteil der Daten liegt.
QUANTIL.INKL(**Array**; **k**)	Liefert denjenigen Wert einer Datenreihe, unterhalb dessen ein mit k (Wertebereich von 0 bis 1) angegebener Bruchteil der Daten liegt.
QUANTILSRANG(**Matrix**; **x**; Genauigkeit)	Liefert die Angabe des Anteils von Daten, die unterhalb des angegebenen Werts liegen.
QUANTILSRANG.EXKL(**Array**; **x**; Genauigkeit)	Liefert die Angabe des Anteils von Daten, die unterhalb des angegebenen Werts (Wertebereich zwischen 0 und 1 und ausschließlich 0 und 1) liegen.
QUANTILSRANG.INKL(**Array**; **x**; **Genauigkeit**)	Liefert die Angabe des Anteils von Daten, die unterhalb des angegebenen Werts (Wertebereich von 0 bis 1) liegen.
QUARTILE(**Matrix**; **Quartil**)	Unterteilt die Daten von Matrix in vier Bereiche mit je gleichen Anteilen von Daten.
QUARTILE.EXKL(**Array**; **Quartile**)	Unterteilt – basierend auf Perzentilwerten (Wertebereich zwischen 0 und 1 und ausschließlich 0 und 1) – die Daten von Array in Bereiche mit je gleichen Anteilen von Daten.
QUARTILE.INKL(**Array**; **Quartile**)	Unterteilt – basierend auf Perzentilwerten (Wertebereich von 0 bis 1) – die Daten von Array in Anteilsbereiche.
QUOTIENT(**Zähler**; **Nenner**)	Liefert das ganzzahlige Ergebnis einer Division.
RANG(**Zahl**; **Bezug**; Reihenfolge)	Liefert den Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt.
RANG.GLEICH(**Zahl**; **Bezug**; Reihenfolge)	Liefert den Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt.

Funktion	Beschreibung
RANG.MITTELW(**Zahl**; **Bezug**; Reihenfolge)	Liefert den gemittelten Rang, den ein Wert in einer Datenreihe in Bezug auf seine Größe einnimmt.
RECHTS(**Text**; Anzahl_Zeichen)	Liefert die mit Anzahl_Zeichen festgelegte Zeichenzahl vom Ende der Zeichenfolge gerechnet.
RENDITE(**Abrechnung**; **Fälligkeit**; **Zins**; **Kurs**; **Rückzahlung**; **Häufigkeit**; Basis)	Berechnet die jährliche Rendite eines Wertpapiers.
RENDITEDIS(**Abrechnung**; **Fälligkeit**; **Kurs**; **Rückzahlung**; Basis)	Berechnet die jährliche Rendite eines unverzinslichen Wertpapiers.
RENDITEFÄLL(**Abrechnung**; **Fälligkeit**; **Emission**; **Zins**; **Kurs**; Basis)	Berechnet die jährliche Rendite eines Wertpapiers, dessen Zinsen zum Fälligkeitstermin ausgezahlt werden.
REST(**Zahl**; **Divisor**)	Liefert den Restbetrag (Modulo) bei einer Division.
RGP(**Y_Werte**; X_Werte; Konstante; Stats)	Liefert Kennziffern zur linearen Regression.
RKP(**Y_Werte**; X_Werte; Konstante; Stats)	Liefert Kennziffern zur exponentiellen Regression.
RMZ(**Zins**; **Zzr**; **Bw**; Zw; F)	Berechnet die periodische Zahlung für eine Annuität.
RÖMISCH(**Zahl**; Typ)	Wandelt Zahlen in die römische Zahlendarstellung um.
RTD(**ProgID**; **Server**; **Topic1**; Topic2; ...)	Empfängt Echtzeitdaten eines registrierten Add-Ins.
RUNDEN(**Zahl**; **Anzahl_Stellen**)	Rundet Werte auf oder ab.
SÄUBERN(**Text**)	Entfernt aus der mit dem Argument Text gelieferten Zeichenfolge alle nicht druckbaren Zeichen.
SCHÄTZER(**x**; **Y_Werte**; **X_Werte**)	Liefert für den vorgegebenen Wert x einen Schätzwert von y anhand einer linearen Regression.
SCHIEFE(**Zahl1**; Zahl2; ...)	Liefert ein Maß für die Asymmetrie der Häufigkeitsverteilung einer Stichprobe.

Funktion	Beschreibung
SCHIEFE.P(**Zahl1**; Zahl2; ...)	Liefert ein Maß für die Asymmetrie der Häufigkeitsverteilung einer Grundgesamtheit.
SEC(**Zahl**)	Liefert den Sekans eines Winkels.
SECHYP(**Zahl**)	Liefert den hyperbolischen Sekans eines Winkels.
SEKUNDE(**Zahl**)	Liefert den Sekundenanteil eines Zeitwerts.
SIN(**Zahl**)	Berechnet den Sinus des angegebenen Winkels.
SINHYP(**Zahl**)	Liefert den hyperbolischen Sinus des angegebenen Werts.
SPALTE(**Bezug**)	Liefert die Spaltennummer eines Bereichs.
SPALTEN(**Matrix**)	Liefert die Anzahl der Spalten eines Bereichs oder einer Matrix.
STABW(**Zahl1**; Zahl2; ...)	Schätzt die Standardabweichung auf der Basis einer Stichprobe.
STABW.N(**Zahl1**; Zahl2; ...)	Berechnet die Standardabweichung auf der Basis der Grundgesamtheit.
STABW.S(**Zahl1**; Zahl2; ...)	Schätzt die Standardabweichung auf der Basis einer Stichprobe.
STABWA(**Wert1**; Wert2; ...)	Schätzt die Standardabweichung auf der Basis einer Stichprobe, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
STABWN(**Zahl1**; Zahl2; ...)	Berechnet die Standardabweichung auf der Basis der Grundgesamtheit.
STABWNA(**Wert1**; Wert2; ...)	Berechnet die Standardabweichung auf der Basis der Grundgesamtheit, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
STANDARDISIERUNG(**x**; **Mittelwert**; **Standabwn**)	Berechnet Werte einer Normalverteilung in Werte einer Standardnormalverteilung um.

Funktion	Beschreibung
STANDNORMINV(**Wahrsch**)	Liefert Quantile der Standardnormalverteilung.
STANDNORMVERT(**z**)	Gibt die Wahrscheinlichkeit dafür an, dass eine Zufallsvariable aus einer Standardnormalverteilung den Wert z oder kleiner annimmt.
STEIGUNG(**Y_Werte**; **X_Werte**)	Liefert die Steigung für eine Regressionsgerade.
STFEHLERYX(**Y_Werte**; **X_Werte**)	Liefert den Standardschätzfehler bei linearer Regression.
STUNDE(**Zahl**)	Liefert den Stundenanteil eines Zeitwerts.
SUCHEN(**Suchtext**; **Text**; Erstes_Zeichen)	Überprüft, ob die mit Suchtext angegebene Zeichenfolge in einer anderen Zeichenfolge vorkommt.
SUMME(**Zahl1**; Zahl2; ...)	Berechnet die Summe der angegebenen Argumente.
SUMMENPRODUKT(**Array1**; Array2; Array3; ...)	Bewirkt, dass die Elemente der Matrizen miteinander multipliziert und anschließend die Multiplikationsergebnisse summiert werden.
SUMMEWENN(**Bereich**; **Suchkriterien**; Summe_Bereich)	Berechnet die Summe der Werte, die das angegebene Kriterium erfüllen.
SUMMEWENNS(**Summe_Bereich**; **Kriterium_Bereich1**; **Kriterien1**; Kriterium_Bereich2; Kriterien2 ...)	Berechnet die Summe der Werte, die mehrere Kriterien erfüllen.
SUMMEX2MY2(**Matrix_x**; **Matrix_y**)	Subtrahiert die Summen der quadrierten x-Werte und der quadrierten y-Werte.
SUMMEX2PY2(**Matrix_x**; **Matrix_y**)	Addiert die Summen der quadrierten x-Werte und der quadrierten y-Werte.
SUMMEXMY2(**Matrix_x**; **Matrix_y**)	Bewirkt, dass die Differenz zwischen x-Werten und y-Werten ausquadriert wird und dann die einzelnen Werte aufsummiert werden.
SUMQUADABW(**Zahl1**; Zahl2; ...)	Gibt die Summe der quadratischen Abweichungen aus.

Funktion	Beschreibung
SVERWEIS(**Suchkriterium**; **Matrix**; **Spaltenindex**; Bereich_Verweis)	Ermittelt, ausgehend von einer Zelle in einer Matrix, den Inhalt der Zelle in derselben Zeile einer anderen Spalte.
T(**Wert**)	Überprüft, ob das mit Wert angegebene Argument eine Zahl oder eine Zeichenfolge liefert.
T.INV(**Wahrsch**; **Freiheitsgrade**)	Liefert linksseitige Quantile der t-Verteilung.
T.INV.2S(**Wahrsch**; **Freiheitsgrade**)	Liefert zweiseitige Quantile der t-Verteilung.
T.TEST(**Matrix1**; **Matrix2**; **Seiten**; **Typ**)	Liefert Werte für den Vergleich zweier Stichproben.
T.VERT(**x**; **Freiheitsgrade**; **Kumuliert**)	Liefert linksseitige Wahrscheinlichkeitswerte einer t-verteilten Zufallsvariablen.
T.VERT.2S(**x**; **Freiheitsgrade**)	Liefert zweiseitige Wahrscheinlichkeitswerte einer t-verteilten Zufallsvariablen.
T.VERT.RE(**x**; **Freiheitsgrade**)	Liefert rechtsseitige Wahrscheinlichkeitswerte einer t-verteilten Zufallsvariablen.
TAG(**Zahl**)	Gibt die Tagesnummer eines Datumswerts aus.
TAGE(**Zieldatum**; **Ausgangsdatum**)	Gibt die Anzahl der Tage zwischen zwei Datumswerten aus.
TAGE360(**Ausgangsdatum**; **Enddatum**; Methode)	Gibt die Anzahl der Tage zwischen zwei verschiedenen Daten aus, wobei das Jahr mit 360 Tagen gerechnet wird.
TAN(**Zahl**)	Berechnet den Tangens des angegebenen Winkels.
TANHYP(**Zahl**)	Liefert den hyperbolischen Tangens.
TBILLÄQUIV(**Abrechnung**; **Fälligkeit**; **Abzinsungssatz**)	Berechnet die vergleichsweise jährliche Verzinsung eines Schatzwechsels (Treasury Bill).
TBILLKURS(**Abrechnung**; **Fälligkeit**; **Abzinsungssatz**)	Berechnet den Ausgabekurs für einen Schatzwechsel.
TBILLRENDITE(**Abrechnung**; **Fälligkeit**; **Pr**)	Berechnet die Rendite eines Schatzwechsels.

Funktion	Beschreibung
TEIL(**Text**; **Erstes_Zeichen**; **Anzahl_Zeichen**)	Schneidet aus der Zeichenfolge eine Unterzeichenfolge aus.
TEILERGEBNIS(**Funktion**; **Bezug1**; Bezug2; ...)	Gibt ein Teilergebnis aus einer Tabelle oder Datenbank zurück.
TEXT(**Wert**; **Textformat**)	Wandelt einen numerischen Wert in einen Text um.
TINV(**Wahrsch**; **Freiheitsgrade**)	Liefert Quantile der t-Verteilung.
TREND(**Y_Werte**; X_Werte; Neue_x_Werte; Konstante)	Berechnet auf der Basis der linearen Regression geschätzte y-Werte.
TTEST(**Matrix1**; **Matrix2**; **Seiten**; **Typ**)	Liefert Werte für den Vergleich zweier Stichproben.
TVERT(**x**; **Freiheitsgrade**; **Seiten**)	Liefert Wahrscheinlichkeiten für t-verteilte Zufallsvariablen.
TYP(**Wert**)	Liefert den Datentyp von Wert.
UMWANDELN(**Zahl**; **Von_Maßeinheit**; **In_Maßeinheit**)	Liefert Umrechnungen zwischen verschiedenen Maßeinheiten.
UND(**Wahrheitswert1**; Wahrheitswert2; ...)	Liefert nur WAHR, wenn alle Argumente WAHR sind, sonst FALSCH.
UNGERADE(**Zahl**)	Rundet Zahlen zur nächsten ungeraden Ganzzahl.
UNICODE(**Text**)	Liefert den Code für das erste Zeichen in Text.
UNIZEICHEN(**Zahl**)	Liefert das Unicode-Zeichen für den angegebenen Code.
UNREGER.KURS(**Abrechnung**; **Fälligkeit**; **Emission**; **Erster_Zinstermin**; **Zins**; **Rendite**; **Rückzahlung**; **Häufigkeit**; Basis)	Liefert den Kurswert eines festverzinslichen Wertpapiers mit abweichender erster Zinsperiode.
UNREGER.REND(**Abrechnung**; **Fälligkeit**; **Emission**; **Erster_Zinstermin**; **Zins**; **Kurs**; **Rückzahlung**; **Häufigkeit**; Basis)	Liefert die Rendite eines festverzinslichen Wertpapiers mit abweichender erster Zinsperiode.
UNREGLE.KURS(**Abrechnung**; **Fälligkeit**; **Letzter_Zinstermin**; **Zins**; **Rendite**; **Rückzahlung**; **Häufigkeit**; Basis)	Liefert den Kurswert eines festverzinslichen Wertpapiers mit abweichender letzter Zinsperiode.

Funktion	Beschreibung
UNREGLE.REND(**Abrechnung**; **Fälligkeit**; **Letzter_Zinstermin**; **Zins**; **Kurs**; **Rückzahlung**; **Häufigkeit**; Basis)	Liefert die Rendite eines festverzinslichen Wertpapiers mit abweichender letzter Zinsperiode.
UNTERGRENZE(**Zahl**; **Schritt**)	Rundet eine Zahl auf das nächste Vielfache von Schritt.
UNTERGRENZE.MATHEMATIK(**Zahl**; Schritt; Modus)	Rundet Werte auf das nächste Vielfache von Schritt ab.
URLCODIEREN(**Text**)	Konvertiert Text in eine URL-kodierte Zeichenfolge.
VAR.P(**Zahl1**; **Zahl2**; ...)	Berechnet die Varianz auf der Basis der Grundgesamtheit.
VAR.S(**Zahl1**; **Zahl2**; ...)	Schätzt die Varianz der Grundgesamtheit auf der Basis einer Stichprobe.
VARIANZ(**Zahl1**; Zahl2; ...)	Schätzt die Varianz der Grundgesamtheit auf der Basis einer Stichprobe.
VARIANZA(**Wert1**; Wert2; ...)	Schätzt die Varianz der Grundgesamtheit auf der Basis einer Stichprobe, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
VARIANZEN(**Zahl1**; Zahl2; ...)	Berechnet die Varianz auf der Basis der Grundgesamtheit.
VARIANZENA(**Wert1**; Wert2; ...)	Berechnet die Varianz auf der Basis der Grundgesamtheit, wobei auch Zellen mit Texten oder Wahrheitswerten berücksichtigt werden.
VARIATION(**Y_Werte**; X_Werte; Neue_x_Werte; Konstante)	Berechnet Werte, die sich aus einem exponentiellen Trend ergeben.
VARIATIONEN(**n**; **k**)	Berechnet die Reihe von geordneten Folgen, die mit den angegebenen Argumenten möglich sind.
VARIATIONEN2(**Zahl**; **Gewählte_Zahl**)	Berechnet die Anzahl der Permutationen für die angegebene Anzahl von Objekten (mit Wiederholungen), die aus der Gesamtmenge der Objekte ausgewählt werden können.

Funktion	Beschreibung
VDB(**Ansch_Wert**; **Restwert**; **Nutzungs-dauer**; **Anfang**; **Ende**; Faktor; Nicht_wechseln)	Berechnet die Abschreibungsrate nach der variabel-degressiven Methode.
VERGLEICH(**Suchkriterium**; **Suchmatrix**; Vergleichstyp)	Durchsucht eine Suchmatrix bzw. einen Bereich nach einem Suchkriterium und gibt die relative Position aus.
VERKETTEN(**Text1**; Text2; ...)	Verknüpft Zeichenfolgen miteinander.
VERWEIS(**Suchkriterium**; **Matrix**)	Durchsucht die Werte einer Matrix.
VERWEIS(**Suchkriterium**; **Suchvektor**; **Ergebnisvektor**)	Durchsucht die Werte eines Vektors.
VORZEICHEN(**Zahl**)	Ergibt bei positiven Zahlen 1, bei negativen Zahlen –1 und bei Null 0.
VRUNDEN(**Zahl**; **Vielfaches**)	Rundet Werte auf das nächste erreichbare Vielfache.
WAHL(**Index**; **Wert1**; Wert2; ...)	Liefert einen Wert aus einer Liste von Werten.
WAHR()	Trägt den Wahrheitswert WAHR in eine Zelle ein.
WAHRSCHBEREICH(**Beob_Werte**; **Beob_Wahrsch**; **Untergrenze**; Obergrenze)	Berechnet die Wahrscheinlichkeit dafür, dass ein Beobachtungswert in das angegebene Intervall fällt.
WEBDIENST(**URL**)	Gibt Daten von einem Webdienst zurück.
WECHSELN(**Text**; **Alter_Text**; **Neuer_Text**; Ntes_Auftreten)	Ersetzt alten Text durch neuen Text in der Zeichenfolge.
WEIBULL(**x**; **Alpha**; **Beta**; **Kumuliert**)	Liefert Wahrscheinlichkeiten für eine Weibull-verteilte Zufallsvariable.
WEIBULL.VERT(**x**; **Alpha**; **Beta**; **Kumuliert**)	Liefert die Wahrscheinlichkeiten für eine Weibull-verteilte Zufallsvariable.
WENN(**Prüfung**; **Dann_Wert**; Sonst_Wert)	Liefert den Dann_Wert, wenn Wahrheitsprüfung WAHR ergibt, sonst den Sonst_Wert.
WENNFEHLER(**Wert**; **Wert_falls_Fehler**)	Erlaubt es, im Fehlerfall benutzerdefinierte Fehleranzeigen auszugeben.

Funktion	Beschreibung
WENNNV(**Wert**; **Wert_bei_NV**)	Gibt den ersten Wert zurück, wenn der Ausdruck nicht den Fehlerwert #NV liefert, andernfalls den zweiten Wert.
WERT(**Text**)	Wandelt die Zeichenfolge in einen numerischen Wert um, sofern die Zeichen ein gültiges Zahlenformat ergeben.
WIEDERHOLEN(**Text**; **Multiplikator**)	Wiederholt in der Zelle die angegebene Zeichenfolge n-mal.
WOCHENTAG(**Zahl**; Typ)	Liefert die Wochentagsnummer eines Datumswerts.
WURZEL(**Zahl**)	Ermittelt die Quadratwurzel einer Zahl.
WURZELPI(**Zahl**)	Liefert die Quadratwurzel aus (Zahl * PI).
WVERWEIS(**Suchkriterium**; **Matrix**; **Zeilenindex**; Bereich_Verweis)	Ermittelt, ausgehend von einer Zelle in einer Matrix, den Inhalt der Zelle in derselben Spalte.
XINTZINSFUSS(**Werte**; **Zeitpkte**; Schätzwert)	Berechnet den internen Zinssatz für eine Reihe von unregelmäßigen Zahlungsvorgängen im unterjährigen Bereich.
XKAPITALWERT(**Zins**; **Werte**; **Zeitpkte**)	Berechnet den Nettokapitalwert für eine Reihe von unregelmäßigen Zahlungsvorgängen im unterjährigen Bereich.
XMLFILTERN(**xml**; **xpath**)	Liefert aus XML-Inhalten die mit dem XPath-Ausdruck bestimmten Daten.
XODER(**Wahrheitswert1**; Wahrheitswert2; ...)	Gibt ein ausschließliches Oder aller Argumente zurück.
ZÄHLENWENN(**Bereich**; **Suchkriterien**)	Gibt die Anzahl der nicht leeren Zellen im Bereich, die dem angegebenen Kriterium entsprechen, an.
ZÄHLENWENNS(**Kriterienbereich1**; **Kriterien1**; Kriterienbereich2; Kriterien2 ...)	Gibt die Anzahl der nicht leeren Zellen im Bereich, die mehreren Kriterien entsprechen, an.
ZAHLENWERT(**Text**; Dezimaltrennzeichen; Gruppentrennzeichen)	Konvertiert unabhängig vom aktuellen Gebietsschema Text in Zahlen.
ZEICHEN(**Zahl**)	Liefert das der Codezahl entsprechende Zeichen.

Funktion	Beschreibung
ZEILE(**Bezug**)	Liefert die Zeilennummer eines Bezugs.
ZEILEN(**Matrix**)	Liefert die Anzahl der Zeilen eines Bereichs oder einer Matrix.
ZEIT(**Stunde**; **Minute**; **Sekunde**)	Berechnet eine serielle Zahl für die eingegebene Uhrzeit.
ZEITWERT(**Zeit**)	Wandelt die Zeichenfolge in einen Zeitwert um.
ZELLE(**Infotyp**; Bezug)	Gibt verschiedene Informationen über eine Zelle aus.
ZINS(**Zzr**; **Rmz**; **Bw**; Zw; F; Schätzwert)	Berechnet den Zinssatz einer Investition bei regelmäßigen Auszahlungen.
ZINSSATZ(**Abrechnung**; **Fälligkeit**; **Anlage**; **Rückzahlung**; Basis)	Berechnet den (jährlichen) Zinssatz für eine Investition, bei der zwischen Abrechnung und Rückzahlung keine Zinsen ausgeschüttet werden.
ZINSTERMNZ(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet das Datum der ersten Zinsausschüttung.
ZINSTERMTAGE(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet die Anzahl der Tage in derjenigen Zinsperiode, in die der Abrechnungszeitpunkt fällt.
ZINSTERMTAGNZ(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet die Tage bis zum ersten Zinstermin.
ZINSTERMTAGVA(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet die Tage vom letzten Zinstermin bis zur Abrechnung.
ZINSTERMVZ(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet das Datum des letzten Zinstermins.
ZINSTERMZAHL(**Abrechnung**; **Fälligkeit**; **Häufigkeit**; Basis)	Berechnet die Zahl der Zinstermine zwischen Kaufdatum und Fälligkeitsdatum.
ZINSZ(**Zins**; **Zr**; **Zzr**; **Bw**; Zw; F)	Berechnet den Zinsanteil für die Abzahlung eines Darlehens.
ZSATZINVEST(**Zzr**; **Bw**; **Zw**)	Berechnet den effektiven Jahreszins für den Wertzuwachs einer Investition.
ZUFALLSBEREICH(**Untere_Zahl**; **Obere_Zahl**)	Erzeugt ganzzahlige Zufallszahlen zwischen Untere_Zahl und Obere_Zahl.

Funktion	Beschreibung
ZUFALLSZAHL()	Produziert Zufallszahlen zwischen 0 und 1.
ZW(**Zins**; **Zzr**; **Rmz**; Bw; F)	Berechnet den zukünftigen Wert auf der Grundlage regelmäßiger Zahlungen.
ZW2(**Kapital**; **Zinsen**)	Berechnet den Endwert eines Kapitals, das mit wechselnden Zinsen verzinst wird.
ZWEIFAKULTÄT(**Zahl**)	Liefert die Doppelfakultät einer positiven Zahl.
ZZR(**Zins**; **Rmz**; **Bw**; Zw; F)	Berechnet die Zahl der Zahlungsperioden für die Tilgung eines Darlehens.

B Funktionenliste Deutsch–Englisch/ Englisch–Deutsch

B.1 Deutsch–Englisch

Deutsch	Englisch
ABS()	ABS()
ACHSENABSCHNITT()	INTERCEPT()
ADRESSE()	ADDRESS()
AGGREGAT()	AGGREGATE()
AMORDEGRK()	AMORDEGRC()
AMORLINEARK()	AMORLINC()
ANZAHL()	COUNT()
ANZAHL2()	COUNTA()
ANZAHLLEEREZELLEN()	COUNTBLANK()
ARABISCH()	ARABIC()
ARBEITSTAG()	WORKDAY()
ARBEITSTAG.INTL()	WORKDAY.INTL()
ARCCOS()	ACOS()
ARCCOSHYP()	ACOSH()
ARCCOT()	ACOT()
ARCCOTHYP()	ACOTH()
ARCSIN()	ASIN()
ARCSINHYP()	ASINH()
ARCTAN()	ATAN()
ARCTAN2()	ATAN2()
ARCTANHYP()	ATANH()

Deutsch	Englisch
AUFGELZINS()	ACCRINT()
AUFGELZINSF()	ACCRINTM()
AUFRUNDEN()	ROUNDUP()
AUSZAHLUNG()	INTRATE()
BAHTTEXT()	BAHTTEXT()
BASIS()	BASE()
BEREICH.VERSCHIEBEN()	OFFSET()
BEREICHE()	AREAS()
BESSELI()	BESSELI()
BESSELJ()	BESSELJ()
BESSELK()	BESSELK()
BESSELY()	BESSELY()
BESTIMMTHEITSMASS()	RSQ()
BETA.INV()	BETA.INV()
BETA.VERT()	BETA.DIST()
BETAINV()	BETAINV()
BETAVERT()	BETADIST()
BININDEZ()	BIN2DEC()
BININHEX()	BIN2HEX()
BININOKT()	BIN2OCT()
BINOM.INV()	BINOM.INV()
BINOM.VERT()	BINOM.DIST()
BINOM.VERT.BEREICH()	BINOM.DIST.RANGE()
BINOMVERT()	BINOMDIST()
BITLVERSCHIEB()	BITLSHIFT()
BITODER()	BITOR()
BITRVERSCHIEB()	BITRSHIFT()
BITUND()	BITAND()
BITXODER()	BITXOR()

Deutsch	Englisch
BLATT ()	SHEET()
BLÄTTER()	SHEETS()
BOGENMASS()	RADIANS()
BRTEILJAHRE()	YEARFRAC()
BW()	PV()
CHIINV()	CHIINV()
CHIQU.INV()	CHISQ.INV()
CHIQU.INV.RE()	CHISQ.INV.RT()
CHIQU.TEST()	CHISQ.TEST()
CHIQU.VERT()	CHISQ.DIST()
CHIQU.VERT.RE()	CHISQ.DIST.RT()
CHITEST()	CHITEST()
CHIVERT()	CHIDIST()
CODE()	CODE()
COS()	COS()
COSEC(Zahl)	CSC()
COSECHYP()	CSCH()
COSHYP()	COSH()
COT()	COT()
COTHYP()	COTH()
CUBEELEMENT()	CUBEMEMBER()
CUBEELEMENTEIGENSCHAFT()	CUBEMEMBERPROPERTY()
CUBEKPIELEMENT()	CUBEKPIMEMBER()
CUBEMENGE()	CUBESET()
CUBEMENGENANZAHL()	CUBESETCOUNT()
CUBERANGE-ELEMENT()	CUBERANKEDMEMBER()
CUBEWERT()	CUBEVALUE()
DATEDIF()	DATEDIF()
DATUM()	DATE()

Deutsch	Englisch
DATWERT()	DATEVALUE()
DBANZAHL()	DCOUNT()
DBANZAHL2()	DCOUNTA()
DBAUSZUG()	DGET()
DBMAX()	DMAX()
DBMIN()	DMIN()
DBMITTELWERT()	DAVERAGE()
DBPRODUKT()	DPRODUCT()
DBSTDABW()	DSTDEV()
DBSTDABWN()	DSTDEVP()
DBSUMME()	DSUM()
DBVARIANZ()	DVAR()
DBVARIANZEN()	DVARP()
DELTA()	DELTA()
DEZIMAL()	DECIMAL()
DEZINBIN()	DEC2BIN()
DEZINHEX()	DEC2HEX()
DEZINOKT()	DEC2OCT()
DIA()	SYD()
DISAGIO()	DISC()
DM()	DOLLAR()
DURATION()	DURATION()
EDATUM()	EDATE()
EFFEKTIV()	EFFECT()
ERSETZEN()	REPLACE()
EXP()	EXP()
EXPON.VERT()	EXPON.DIST()
EXPONVERT()	EXPONDIST()
F.INV()	F.INV()

Deutsch	Englisch
F.INV.RE()	F.INV.RT()
F.TEST()	F.TEST()
F.VERT()	F.DIST()
F.VERT.RE()	F.DIST.RT()
FAKULTÄT()	FACT()
FALSCH()	FALSE()
FEHLER.TYP()	ERROR.TYPE()
FEST()	FIXED()
FINDEN()	FIND()
FINV()	FINV()
FISHER()	FISHER()
FISHERINV()	FISHERINV()
FORMELTEXT()	FORMULATEXT()
FTEST()	FTEST()
FVERT()	FDIST()
G.TEST()	G.TEST()
GAMMA(x)	GAMMA()
GAMMA.INV()	GAMMA.INV()
GAMMA.VERT()	GAMMA.DIST()
GAMMAINV()	GAMMAINV()
GAMMALN()	GAMMALN()
GAMMALN.GENAU()	GAMMALN.PRECISE()
GAMMAVERT()	GAMMADIST()
GANZZAHL()	INT()
GAUSS()	GAUSS()
GAUSSF.GENAU()	ERF.PRECISE()
GAUSSFEHLER()	ERF()
GAUSSFKOMPL()	ERFC()
GAUSSFKOMPL.GENAU()	ERFC.PRECISE()

Deutsch	Englisch
GDA()	DDB()
GDA2()	DB()
GEOMITTEL()	GEOMEAN()
GERADE()	EVEN()
GESTUTZTMITTEL()	TRIMMEAN()
GGANZZAHL()	GESTEP()
GGT()	GCD()
GLÄTTEN()	TRIM()
GRAD()	DEGREES()
GROSS()	UPPER()
GROSS2()	PROPER()
GTEST()	ZTEST()
HARMITTEL()	HARMEAN()
HÄUFIGKEIT()	FREQUENCY()
HEUTE()	TODAY()
HEXINBIN()	HEX2BIN()
HEXINDEZ()	HEX2DEC()
HEXINOKT()	HEX2OCT()
HYPERLINK()	HYPERLINK()
HYPGEOM.VERT()	HYPGEOM.DIST()
HYPGEOMVERT()	HYPGEOMDIST()
IDENTISCH()	EXACT()
IKV()	IRR()
IMABS()	IMABS()
IMAGINÄRTEIL()	IMAGINARY()
IMAPOTENZ()	IMPOWER()
IMARGUMENT()	IMARGUMENT()
IMCOS()	IMCOS()
IMCOSEC()	IMCSC()

Deutsch	Englisch
IMCOSECHYP()	IMCSCH()
IMCOSHYP()	IMCOSH()
IMCOT()	IMCOT()
IMDIV()	IMDIV()
IMEXP()	IMEXP()
IMKONJUGIERTE()	IMCONJUGATE()
IMLN()	IMLN()
IMLOG10()	IMLOG10()
IMLOG2()	IMLOG2()
IMPRODUKT()	IMPRODUCT()
IMREALTEIL()	IMREAL()
IMSEC()	IMSEC()
IMSECHYP()	IMSECH()
IMSIN()	IMSIN()
IMSINHYP()	IMSINH()
IMSUB()	IMSUB()
IMSUMME()	IMSUM()
IMTAN()	IMTAN()
IMWURZEL()	IMSQRT()
INDEX()	INDEX()
INDIREKT()	INDIRECT()
INFO()	INFO()
ISOKALENDERWOCHE()	ISOWEEKNUM()
ISPMT()	ISPMT()
ISTBEZUG()	ISREF()
ISTFEHL()	ISERR()
ISTFEHLER()	ISERROR()
ISTFORMEL()	ISFORMULA()
ISTGERADE()	ISEVEN()

Deutsch	Englisch
ISTKTEXT()	ISNONTEXT()
ISTLEER()	ISBLANK()
ISTLOG()	ISLOGICAL()
ISTNV()	ISNA()
ISTTEXT()	ISTEXT()
ISTUNGERADE()	ISODD()
ISTZAHL()	ISNUMBER()
JAHR()	YEAR()
JETZT()	NOW()
KALENDERWOCHE()	WEEKNUM()
KAPZ()	PPMT()
KGRÖSSTE()	LARGE()
KGV()	LCM()
KKLEINSTE()	SMALL()
KLEIN()	LOWER()
KOMBINATIONEN()	COMBIN()
KOMBINATIONEN2(Zahl;	COMBINA()
KOMPLEXE()	COMPLEX()
KONFIDENZ()	CONFIDENCE()
KONFIDENZ.NORM()	CONFIDENCE.NORM()
KONFIDENZ.T()	CONFIDENCE.T()
KORREL()	CORREL()
KOVAR()	COVAR()
KOVARIANZ.P()	COVARIANCE.P()
KOVARIANZ.S()	COVARIANCE.S()
KRITBINOM()	CRITBINOM()
KÜRZEN()	TRUNC()
KUMKAPITAL()	CUMPRINC()
KUMZINSZ()	CUMIPMT()

Deutsch	Englisch
KURS()	PRICE()
KURSDISAGIO()	PRICEDISC()
KURSFÄLLIG()	PRICEMAT()
KURT()	KURT()
LÄNGE()	LEN()
LIA()	SLN()
LINKS()	LEFT()
LN()	LN()
LOG()	LOG()
LOG10()	LOG10()
LOGINV()	LOGINV()
LOGNORM.INV()	LOGNORM.INV()
LOGNORM.VERT()	LOGNORM.DIST()
LOGNORMVERT()	LOGNORMDIST()
MAX()	MAX()
MAXA()	MAXA()
MDET()	MDETERM()
MDURATION()	MDURATION()
MEDIAN()	MEDIAN()
MEINHEIT()	MUNIT()
MIN()	MIN()
MINA()	MINA()
MINUTE()	MINUTE()
MINV()	MINVERSE()
MITTELABW()	AVEDEV()
MITTELWERT()	AVERAGE()
MITTELWERTA()	AVERAGEA()
MITTELWERTWENN()	AVERAGEIF()
MITTELWERTWENNS()	AVERAGEIFS()

Deutsch	Englisch
MMULT()	MMULT()
MODALWERT()	MODE()
MODUS.EINF()	MODE.SNGL()
MODUS.VIELF()	MODE.MULT()
MONAT()	MONTH()
MONATSENDE()	EOMONTH()
MTRANS()	TRANSPOSE()
N()	N()
NBW()	NPV()
NEGBINOM.VERT()	NEGBINOM.DIST()
NEGBINOMVERT()	NEGBINOMDIST()
NETTOARBEITSTAGE()	NETWORKDAYS()
NETTAARBEITSTAGE.INTL()	NETWORKDAYS.INTL()
NICHT()	NOT()
NOMINAL()	NOMINAL()
NORM.INV()	NORM.INV()
NORM.S.INV()	NORM.S.INV()
NORM.S.VERT()	NORM.S.DIST()
NORM.VERT()	NORM.DIST()
NORMINV()	NORMINV()
NORMVERT()	NORMDIST()
NOTIERUNGBRU()	DOLLARFR()
NOTIERUNGDEZ()	DOLLARDE()
NV()	NA()
OBERGRENZE()	CEILING()
OPERGRENZE.MATHEMATIK()	CEILING.PRECISE()
ODER()	OR()
OKTINBIN()	OCT2BIN()
OKTINDEZ()	OCT2DEC()

Deutsch	Englisch
OKTINHEX()	OCT2HEX()
PDURATION (Zins; Bw;	PDURATION()
PEARSON()	PEARSON()
PHI(x)	PHI()
PI()	PI()
PIVOTDATENZUORDNEN()	GETPIVOTDATA()
POISSON()	POISSON()
POISSON.VERT()	POISSON.DIST()
POLYNOMIAL()	MULTINOMIAL()
POTENZ()	POWER()
POTENZREIHE()	SERIESSUM()
PRODUKT()	PRODUCT()
QIKV()	MIRR()
QUADRATESUMME()	SUMSQ()
QUANTIL()	PERCENTILE()
QUANTIL.EXKL()	PERCENTILE.EXC()
QUANTIL.INKL()	PERCENTILE.INC()
QUANTILSRANG()	PERCENTRANK()
QUANTILSRANG.EXKL()	PERCENTRANK.EXC()
QUANTILSRANG.INKL()	PERCENTRANK.INC()
QUARTILE()	QUARTILE()
QUARTILE.EXKL()	QUARTILE.EXC()
QUARTILE.INKL()	QUARTILE.INC()
QUOTIENT()	QUOTIENT()
RANG()	RANK()
RANG.GLEICH()	RANK.EQ()
RANG.MITTELW()	RANK.AVG()
RECHTS()	RIGHT()
RENDITE()	YIELD()

Deutsch	Englisch
RENDITEDIS()	YIELDDISC()
RENDITEFÄLL()	YIELDMAT()
REST()	MOD()
RGP()	LINEST()
RKP()	LOGEST()
RMZ()	PMT()
RÖMISCH()	ROMAN()
RTD()	RTD()
RUNDEN()	ROUND()
SÄUBERN()	CLEAN()
SCHÄTZER()	FORECAST()
SCHIEFE()	SKEW()
SCHIEFE.P()	SKEW.P()
SEC()	SEC()
SECHYP()	SECH()
SEKUNDE()	SECOND()
SIN()	SIN()
SINHYP()	SINH()
SPALTE()	COLUMN()
SPALTEN()	COLUMNS()
STABW()	STDEV()
STABW.N()	STDEV.P()
STABW.S()	STDEV.S()
STABWA()	STDEVA()
STABWN()	STDEVP()
STABWNA()	STDEVPA()
STANDARDISIERUNG()	STANDARDIZE()
STANDNORMINV()	NORMSINV()
STANDNORMVERT()	NORMSDIST()

Deutsch	Englisch
STEIGUNG()	SLOPE()
STFEHLERYX()	STEYX()
STUNDE()	HOUR()
SUCHEN()	SEARCH()
SUMME()	SUM()
SUMMENPRODUKT()	SUMPRODUCT()
SUMMEWENN()	SUMIF()
SUMMEWENNS()	SUMIFS()
SUMMEX2MY2()	SUMX2MY2()
SUMMEX2PY2()	SUMX2PY2()
SUMMEXMY2()	SUMXMY2()
SUMQUADABW()	DEVSQ()
SVERWEIS()	VLOOKUP()
T()	T()
T.INV()	T.INV()
T.INV.2S()	T.INV.2T()
T.TEST()	T.TEST()
T.VERT()	T.DIST()
T.VERT.2S()	T.DIST.2T()
T.VERT.RE()	T.DIST.RT()
TAG()	DAY()
TAGE()	DAYS()
TAGE360()	DAYS360()
TAN()	TAN()
TANHYP()	TANH()
TBILLÄQUIV()	TBILLEQ()
TBILLKURS()	TBILLPRICE()
TBILLRENDITE()	TBILLYIELD()
TEIL()	MID()

Deutsch	Englisch
TEILERGEBNIS()	SUBTOTAL()
TEXT()	TEXT()
TINV()	TINV()
TREND()	TREND()
TTEST()	TTEST()
TVERT()	TDIST()
TYP()	TYPE()
UMWANDELN()	CONVERT()
UND()	AND()
UNGERADE()	ODD()
UNICODE()	UNICODE()
UNIZEICHEN()	UNICHAR()
UNREGER.KURS()	ODDFPRICE()
UNREGER.REND()	ODDFYIELD()
UNREGLE.KURS()	ODDLPRICE()
UNREGLE.REND()	ODDLYIELD()
UNTERGRENZE()	FLOOR()
UNTERGRENZE.MATHEMATIK()	FLOOR.PRECISE()
URLCODIEREN(Text)	ENCODEURL()
VAR.P()	VAR.P()
VAR.S()	VAR.S()
VARIANZ()	VAR()
VARIANZA()	VARA()
VARIANZEN()	VARP()
VARIANZENA()	VARPA()
VARIATION()	GROWTH()
VARIATIONEN()	PERMUT()
VARIATIONEN2()	PERMUTATIONA()
VDB()	VDB()

Deutsch	Englisch
VERGLEICH()	MATCH()
VERKETTEN()	CONCATENATE()
VERWEIS()	LOOKUP()
VORZEICHEN()	SIGN()
VRUNDEN()	MROUND()
WAHL()	CHOOSE()
WAHR()	TRUE()
WAHRSCHBEREICH()	PROB()
WEBDIENST(URL)	WEBSERVICE()
WECHSELN()	SUBSTITUTE()
WEIBULL()	WEIBULL()
WEIBULL.VERT()	WEIBULL.DIST()
WENN()	IF()
WENNFEHLER()	IFERROR()
WENNNV()	IFNA()
WERT()	VALUE()
WIEDERHOLEN()	REPT()
WOCHENTAG()	WEEKDAY()
WURZEL()	SQRT()
WURZELPI()	SQRTPI()
WVERWEIS()	HLOOKUP()
XINTZINSFUSS()	XIRR()
XKAPITALWERT()	XNPV()
XMLFILTERN()	FILTERXML()
XODER()	XOR()
ZÄHLENWENN()	COUNTIF()
ZÄHLENWENNS()	COUNTIFS()
ZAHLENWERT()	NUMBERVALUE()
ZEICHEN()	CHAR()

Deutsch	Englisch
ZEILE()	ROW()
ZEILEN()	ROWS()
ZEIT()	TIME()
ZEITWERT()	TIMEVALUE()
ZELLE()	CELL()
ZINS()	RATE()
ZINSSATZ()	RECEIVED()
ZINSTERMNZ()	COUPNCD()
ZINSTERMTAGE()	COUPDAYS()
ZINSTERMTAGNZ()	COUPDAYSNC()
ZINSTERMTAGVA()	COUPDAYBS()
ZINSTERMVZ()	COUPPCD()
ZINSTERMZAHL()	COUPNUM()
ZINSZ()	IPMT()
ZSATZINVEST()	RRI()
ZUFALLSBEREICH()	RANDBETWEEN()
ZUFALLSZAHL()	RAND()
ZW()	FV()
ZW2()	FVSCHEDULE()
ZWEIFAKULTÄT()	FACTDOUBLE()
ZZR()	NPER()

B.2 Englisch–Deutsch

Englisch	Deutsch
ABS()	ABS()
ACCRINT()	AUFGELZINS()
ACCRINTM()	AUFGELZINSF()
ACOS()	ARCCOS()
ACOSH()	ARCCOSHYP()
ACOT()	ARCCOT()
ACOTH()	ARCCOTHYP()
ADDRESS()	ADRESSE()
AGGREGATE()	AGGREGAT()
AMORDEGRC()	AMORDEGRK()
AMORLINC()	AMORLINEARK()
AND()	UND()
ARABIC()	ARABISCH()
AREAS()	BEREICHE()
ASIN()	ARCSIN()
ASINH()	ARCSINHYP()
ATAN()	ARCTAN()
ATAN2()	ARCTAN2()
ATANH()	ARCTANHYP()
AVEDEV()	MITTELABW()
AVERAGE()	MITTELWERT()
AVERAGEA()	MITTELWERTA()
AVERAGEIF()	MITTELWERTWENN()
AVERAGEIFS()	MITTELWERTWENNS()
BAHTTEXT()	BAHTTEXT()
BASE()	BASIS()
BESSELI()	BESSELI()

Englisch	Deutsch
BESSELJ()	BESSELJ()
BESSELK()	BESSELK()
BESSELY()	BESSELY()
BETA.DIST()	BETA.VERT()
BETA.INV()	BETA.INV()
BETADIST()	BETAVERT()
BETAINV()	BETAINV()
BIN2DEC()	BININDEZ()
BIN2HEX()	BININHEX()
BIN2OCT()	BININOKT()
BINOM.DIST()	BINOM.VERT()
BINOM.DIST.RANGE()	BINOM.VERT.BEREICH()
BINOM.INV()	BINOM.INV()
BINOMDIST()	BINOMVERT()
BITAND()	BITUND()
BITLSHIFT()	BITLVERSCHIEB()
BITOR()	BITODER()
BITRSHIFT()	BITRVERSCHIEB()
BITXOR()	BITXODER()
CEILING()	OBERGRENZE()
CEILING.PRECISE()	OPERGRENZE.MATHEMATIK()
CELL()	ZELLE()
CHAR()	ZEICHEN()
CHIDIST()	CHIVERT()
CHIINV()	CHIINV()
CHISQ.DIST()	CHIQU.VERT()
CHISQ.DIST.RT()	CHIQU.VERT.RE()
CHISQ.INV()	CHIQU.INV()
CHISQ.INV.RT()	CHIQU.INV.RE()

Englisch	Deutsch
CHISQ.TEST()	CHIQU.TEST()
CHITEST()	CHITEST()
CHOOSE()	WAHL()
CLEAN()	SÄUBERN()
CODE()	CODE()
COLUMN()	SPALTE()
COLUMNS()	SPALTEN()
COMBIN()	KOMBINATIONEN()
COMBINA()	KOMBINATIONEN2(Zahl;
COMPLEX()	KOMPLEXE()
CONCATENATE()	VERKETTEN()
CONFIDENCE()	KONFIDENZ()
CONFIDENCE.NORM()	KONFIDENZ.NORM()
CONFIDENCE.T()	KONFIDENZ.T()
CONVERT()	UMWANDELN()
CORREL()	KORREL()
COS()	COS()
COSH()	COSHYP()
COT()	COT()
COTH()	COTHYP()
COUNT()	ANZAHL()
COUNTA()	ANZAHL2()
COUNTBLANK()	ANZAHLLEEREZELLEN()
COUNTIF()	ZÄHLENWENN()
COUNTIFS()	ZÄHLENWENNS()
COUPDAYBS()	ZINSTERMTAGVA()
COUPDAYS()	ZINSTERMTAGE()
COUPDAYSNC()	ZINSTERMTAGNZ()
COUPNCD()	ZINSTERMNZ()

Englisch	Deutsch
COUPNUM()	ZINSTERMZAHL()
COUPPCD()	ZINSTERMVZ()
COVAR()	KOVAR()
COVARIANCE.P()	KOVARIANZ.P()
COVARIANCE.S()	KOVARIANZ.S()
CRITBINOM()	KRITBINOM()
CSC()	COSEC(Zahl)
CSCH()	COSECHYP()
CUBEKPIMEMBER()	CUBEKPIELEMENT()
CUBEMEMBER()	CUBEELEMENT()
CUBEMEMBERPROPERTY()	CUBEELEMENTEIGENSCHAFT()
CUBERANKEDMEMBER()	CUBERANGE-ELEMENT()
CUBESET()	CUBEMENGE()
CUBESETCOUNT()	CUBEMENGENANZAHL()
CUBEVALUE()	CUBEWERT()
CUMIPMT()	KUMZINSZ()
CUMPRINC()	KUMKAPITAL()
DATE()	DATUM()
DATEDIF()	DATEDIF()
DATEVALUE()	DATWERT()
DAVERAGE()	DBMITTELWERT()
DAY()	TAG()
DAYS()	TAGE()
DAYS360()	TAGE360()
DB()	GDA2()
DCOUNT()	DBANZAHL()
DCOUNTA()	DBANZAHL2()
DDB()	GDA()
DEC2BIN()	DEZINBIN()

Englisch	Deutsch
DEC2HEX()	DEZINHEX()
DEC2OCT()	DEZINOKT()
DECIMAL()	DEZIMAL()
DEGREES()	GRAD()
DELTA()	DELTA()
DEVSQ()	SUMQUADABW()
DGET()	DBAUSZUG()
DISC()	DISAGIO()
DMAX()	DBMAX()
DMIN()	DBMIN()
DOLLAR()	DM()
DOLLARDE()	NOTIERUNGDEZ()
DOLLARFR()	NOTIERUNGBRU()
DPRODUCT()	DBPRODUKT()
DSTDEV()	DBSTDABW()
DSTDEVP()	DBSTDABWN()
DSUM()	DBSUMME()
DURATION()	DURATION()
DVAR()	DBVARIANZ()
DVARP()	DBVARIANZEN()
EDATE()	EDATUM()
EFFECT()	EFFEKTIV()
ENCODEURL()	URLCODIEREN(Text)
EOMONTH()	MONATSENDE()
ERF()	GAUSSFEHLER()
ERF.PRECISE()	GAUSSF.GENAU()
ERFC()	GAUSSFKOMPL()
ERFC.PRECISE()	GAUSSFKOMPL.GENAU()
ERROR.TYPE()	FEHLER.TYP()

Englisch	Deutsch
EVEN()	GERADE()
EXACT()	IDENTISCH()
EXP()	EXP()
EXPON.DIST()	EXPON.VERT()
EXPONDIST()	EXPONVERT()
F.DIST()	F.VERT()
F.DIST.RT()	F.VERT.RE()
F.INV()	F.INV()
F.INV.RT()	F.INV.RE()
F.TEST()	F.TEST()
FACT()	FAKULTÄT()
FACTDOUBLE()	ZWEIFAKULTÄT()
FALSE()	FALSCH()
FDIST()	FVERT()
FILTERXML()	XMLFILTERN()
FIND()	FINDEN()
FINV()	FINV()
FISHER()	FISHER()
FISHERINV()	FISHERINV()
FIXED()	FEST()
FLOOR()	UNTERGRENZE()
FLOOR.PRECISE()	UNTERGRENZE.MATHEMATIK()
FORECAST()	SCHÄTZER()
FORMULATEXT()	FORMELTEXT()
FREQUENCY()	HÄUFIGKEIT()
FTEST()	FTEST()
FV()	ZW()
FVSCHEDULE()	ZW2()
G.TEST()	G.TEST()

Englisch	Deutsch
GAMMA()	GAMMA(x)
GAMMA.DIST()	GAMMA.VERT()
GAMMA.INV()	GAMMA.INV()
GAMMADIST()	GAMMAVERT()
GAMMAINV()	GAMMAINV()
GAMMALN()	GAMMALN()
GAMMALN.PRECISE()	GAMMALN.GENAU()
GAUSS()	GAUSS()
GCD()	GGT()
GEOMEAN()	GEOMITTEL()
GESTEP()	GGANZZAHL()
GETPIVOTDATA()	PIVOTDATENZUORDNEN()
GROWTH()	VARIATION()
HARMEAN()	HARMITTEL()
HEX2BIN()	HEXINBIN()
HEX2DEC()	HEXINDEZ()
HEX2OCT()	HEXINOKT()
HLOOKUP()	WVERWEIS()
HOUR()	STUNDE()
HYPERLINK()	HYPERLINK()
HYPGEOM.DIST()	HYPGEOM.VERT()
HYPGEOMDIST()	HYPGEOMVERT()
IF()	WENN()
IFERROR()	WENNFEHLER()
IFNA()	WENNNV()
IMABS()	IMABS()
IMAGINARY()	IMAGINÄRTEIL()
IMARGUMENT()	IMARGUMENT()
IMCONJUGATE()	IMKONJUGIERTE()

Englisch	Deutsch
IMCOS()	IMCOS()
IMCOSH()	IMCOSHYP()
IMCOT()	IMCOT()
IMCSC()	IMCOSEC()
IMCSCH()	IMCOSECHYP()
IMDIV()	IMDIV()
IMEXP()	IMEXP()
IMLN()	IMLN()
IMLOG10()	IMLOG10()
IMLOG2()	IMLOG2()
IMPOWER()	IMAPOTENZ()
IMPRODUCT()	IMPRODUKT()
IMREAL()	IMREALTEIL()
IMSEC()	IMSEC()
IMSECH()	IMSECHYP()
IMSIN()	IMSIN()
IMSINH()	IMSINHYP()
IMSQRT()	IMWURZEL()
IMSUB()	IMSUB()
IMSUM()	IMSUMME()
IMTAN()	IMTAN()
INDEX()	INDEX()
INDIRECT()	INDIREKT()
INFO()	INFO()
INT()	GANZZAHL()
INTERCEPT()	ACHSENABSCHNITT()
INTRATE()	AUSZAHLUNG()
IPMT()	ZINSZ()
IRR()	IKV()

Englisch	Deutsch
ISBLANK()	ISTLEER()
ISERR()	ISTFEHL()
ISERROR()	ISTFEHLER()
ISEVEN()	ISTGERADE()
ISFORMULA()	ISTFORMEL()
ISLOGICAL()	ISTLOG()
ISNA()	ISTNV()
ISNONTEXT()	ISTKTEXT()
ISNUMBER()	ISTZAHL()
ISODD()	ISTUNGERADE()
ISOWEEKNUM()	ISOKALENDERWOCHE()
ISPMT()	ISPMT()
ISREF()	ISTBEZUG()
ISTEXT()	ISTTEXT()
KURT()	KURT()
LARGE()	KGRÖSSTE()
LCM()	KGV()
LEFT()	LINKS()
LEN()	LÄNGE()
LINEST()	RGP()
LN()	LN()
LOG()	LOG()
LOG10()	LOG10()
LOGEST()	RKP()
LOGINV()	LOGINV()
LOGNORM.DIST()	LOGNORM.VERT()
LOGNORM.INV()	LOGNORM.INV()
LOGNORMDIST()	LOGNORMVERT()
LOOKUP()	VERWEIS()

Englisch	Deutsch
LOWER()	KLEIN()
MATCH()	VERGLEICH()
MAX()	MAX()
MAXA()	MAXA()
MDETERM()	MDET()
MDURATION()	MDURATION()
MEDIAN()	MEDIAN()
MID()	TEIL()
MIN()	MIN()
MINA()	MINA()
MINUTE()	MINUTE()
MINVERSE()	MINV()
MIRR()	QIKV()
MMULT()	MMULT()
MOD()	REST()
MODE()	MODALWERT()
MODE.MULT()	MODUS.VIELF()
MODE.SNGL()	MODUS.EINF()
MONTH()	MONAT()
MROUND()	VRUNDEN()
MULTINOMIAL()	POLYNOMIAL()
MUNIT()	MEINHEIT()
N()	N()
NA()	NV()
NEGBINOM.DIST()	NEGBINOM.VERT()
NEGBINOMDIST()	NEGBINOMVERT()
NETWORKDAYS()	NETTOARBEITSTAGE()
NETWORKDAYS.INTL()	NETTAARBEITSTAGE.INTL()
NOMINAL()	NOMINAL()

Englisch	Deutsch
NORM.DIST()	NORM.VERT()
NORM.INV()	NORM.INV()
NORM.S.DIST()	NORM.S.VERT()
NORM.S.INV()	NORM.S.INV()
NORMDIST()	NORMVERT()
NORMINV()	NORMINV()
NORMSDIST()	STANDNORMVERT()
NORMSINV()	STANDNORMINV()
NOT()	NICHT()
NOW()	JETZT()
NPER()	ZZR()
NPV()	NBW()
NUMBERVALUE()	ZAHLENWERT()
OCT2BIN()	OKTINBIN()
OCT2DEC()	OKTINDEZ()
OCT2HEX()	OKTINHEX()
ODD()	UNGERADE()
ODDFPRICE()	UNREGER.KURS()
ODDFYIELD()	UNREGER.REND()
ODDLPRICE()	UNREGLE.KURS()
ODDLYIELD()	UNREGLE.REND()
OFFSET()	BEREICH.VERSCHIEBEN()
OR()	ODER()
PDURATION()	PDURATION (Zins; Bw;
PEARSON()	PEARSON()
PERCENTILE()	QUANTIL()
PERCENTILE.EXC()	QUANTIL.EXKL()
PERCENTILE.INC()	QUANTIL.INKL()
PERCENTRANK()	QUANTILSRANG()

Englisch	Deutsch
PERCENTRANK.EXC()	QUANTILSRANG.EXKL()
PERCENTRANK.INC()	QUANTILSRANG.INKL()
PERMUT()	VARIATIONEN()
PERMUTATIONA()	VARIATIONEN2()
PHI()	PHI(x)
PI()	PI()
PMT()	RMZ()
POISSON()	POISSON()
POISSON.DIST()	POISSON.VERT()
POWER()	POTENZ()
PPMT()	KAPZ()
PRICE()	KURS()
PRICEDISC()	KURSDISAGIO()
PRICEMAT()	KURSFÄLLIG()
PROB()	WAHRSCHBEREICH()
PRODUCT()	PRODUKT()
PROPER()	GROSS2()
PV()	BW()
QUARTILE()	QUARTILE()
QUARTILE.EXC()	QUARTILE.EXKL()
QUARTILE.INC()	QUARTILE.INKL()
QUOTIENT()	QUOTIENT()
RADIANS()	BOGENMASS()
RAND()	ZUFALLSZAHL()
RANDBETWEEN()	ZUFALLSBEREICH()
RANK()	RANG()
RANK.AVG()	RANG.MITTELW()
RANK.EQ()	RANG.GLEICH()
RATE()	ZINS()

Englisch	Deutsch
RECEIVED()	ZINSSATZ()
REPLACE()	ERSETZEN()
REPT()	WIEDERHOLEN()
RIGHT()	RECHTS()
ROMAN()	RÖMISCH()
ROUND()	RUNDEN()
ROUNDDOWN()	ABRUNDEN()
ROUNDUP()	AUFRUNDEN()
ROW()	ZEILE()
ROWS()	ZEILEN()
RRI()	ZSATZINVEST()
RSQ()	BESTIMMTHEITSMASS()
RTD()	RTD()
SEARCH()	SUCHEN()
SEC()	SEC()
SECH()	SECHYP()
SECOND()	SEKUNDE()
SERIESSUM()	POTENZREIHE()
SHEET()	BLATT ()
SHEETS()	BLÄTTER()
SIGN()	VORZEICHEN()
SIN()	SIN()
SINH()	SINHYP()
SKEW()	SCHIEFE()
SKEW.P()	SCHIEFE.P()
SLN()	LIA()
SLOPE()	STEIGUNG()
SMALL()	KKLEINSTE()
SQRT()	WURZEL()

Englisch	Deutsch
SQRTPI()	WURZELPI()
STANDARDIZE()	STANDARDISIERUNG()
STDEV()	STABW()
STDEV.P()	STABW.N()
STDEV.S()	STABW.S()
STDEVA()	STABWA()
STDEVP()	STABWN()
STDEVPA()	STABWNA()
STEYX()	STFEHLERYX()
SUBSTITUTE()	WECHSELN()
SUBTOTAL()	TEILERGEBNIS()
SUM()	SUMME()
SUMIF()	SUMMEWENN()
SUMIFS()	SUMMEWENNS()
SUMPRODUCT()	SUMMENPRODUKT()
SUMSQ()	QUADRATESUMME()
SUMX2MY2()	SUMMEX2MY2()
SUMX2PY2()	SUMMEX2PY2()
SUMXMY2()	SUMMEXMY2()
SYD()	DIA()
T()	T()
T.DIST()	T.VERT()
T.DIST.2T()	T.VERT.2S()
T.DIST.RT()	T.VERT.RE()
T.INV()	T.INV()
T.INV.2T()	T.INV.2S()
T.TEST()	T.TEST()
TAN()	TAN()
TANH()	TANHYP()

Englisch	Deutsch
TBILLEQ()	TBILLÄQUIV()
TBILLPRICE()	TBILLKURS()
TBILLYIELD()	TBILLRENDITE()
TDIST()	TVERT()
TEXT()	TEXT()
TIME()	ZEIT()
TIMEVALUE()	ZEITWERT()
TINV()	TINV()
TODAY()	HEUTE()
TRANSPOSE()	MTRANS()
TREND()	TREND()
TRIM()	GLÄTTEN()
TRIMMEAN()	GESTUTZTMITTEL()
TRUE()	WAHR()
TRUNC()	KÜRZEN()
TTEST()	TTEST()
TYPE()	TYP()
UNICHAR()	UNIZEICHEN()
UNICODE()	UNICODE()
UPPER()	GROSS()
VALUE()	WERT()
VAR()	VARIANZ()
VAR.P()	VAR.P()
VAR.S()	VAR.S()
VARA()	VARIANZA()
VARP()	VARIANZEN()
VARPA()	VARIANZENA()
VDB()	VDB()
VLOOKUP()	SVERWEIS()

Englisch	Deutsch
WEBSERVICE()	WEBDIENST(URL)
WEEKDAY()	WOCHENTAG()
WEEKNUM()	KALENDERWOCHE()
WEIBULL()	WEIBULL()
WEIBULL.DIST()	WEIBULL.VERT()
WORKDAY()	ARBEITSTAG()
WORKDAY.INTL()	ARBEITSTAG.INTL()
XIRR()	XINTZINSFUSS()
XNPV()	XKAPITALWERT()
XOR()	XODER()
YEAR()	JAHR()
YEARFRAC()	BRTEILJAHRE()
YIELD()	RENDITE()
YIELDDISC()	RENDITEDIS()
YIELDMAT()	RENDITEFÄLL()
ZTEST()	GTEST()

C Funktionen im Überblick

A

ABRUNDEN() 311
ABS() 312
ACHSENABSCHNITT() 465
ADRESSE() 668
AGGREGAT() 313
AMORDEGRK() 184
AMORLINEARK() 186
ANZAHL() 466
ANZAHL2() 468
ANZAHLLEEREZELLEN() 471
ARABISCH() 316
ARBEITSTAG.INTL() 271
ARBEITSTAG() 269
ARCCOS() 317
ARCCOSHYP() 319
ARCCOT() 320
ARCCOTHYP() 320
ARCSIN() 321
ARCSINHYP() 323
ARCTAN() 324
ARCTAN2() 325
ARCTANHYP() 326
AUFGELZINS() 187
AUFGELZINSF() 189
AUFRUNDEN() 327
AUSZAHLUNG() 190

B

BAHTTEXT() 747
BASIS() 328
BEREICH.VERSCHIEBEN() 670
BEREICHE() 673
BESSELI() 401
BESSELJ() 402
BESSELK() 403
BESSELY() 404
BESTIMMTHEITSMASS() 473
BETA.INV() 474
BETA.VERT() 475
BETAINV() 618
BETAVERT() 619
BININDEZ() 404
BININHEX() 405
BININOKT() 406
BINOM.INV() 477
BINOM.VERT.BEREICH() 480
BINOM.VERT() 478
BINOMVERT() 620
BITLVERSCHIEB() 407
BITODER() 408
BITRVERSCHIEB() 408
BITUND() 409
BITXODER() 410
BLATT() 801
BLÄTTER() 802
BOGENMASS() 329
BRTEILJAHRE() 273
BW() 191

C

CHIINV() 621
CHIQU.INV.RE() 482
CHIQU.INV() 481
CHIQU.TEST() 482
CHIQU.VERT.RE() 487

CHIQU.VERT() 484
CHITEST() 622
CHIVERT() 624
CODE() 748
COS() 330
COSEC() 332
COSECHYP() 333
COSHYP() 334
COT() 335
COTHYP() 337
CUBEELEMENT() 739
CUBEELEMENTEIGENSCHAFT() 740
CUBEKPIELEMENT() 740
CUBEKPIELEMENT() 740
CUBEMENGE() 742
CUBEMENGE() 742
CUBEMENGENANZAHL() 743
CUBEMENGENANZAHL() 743
CUBERANGELEMENT() 744
CUBERANGELEMENT() 744
CUBEWERT() 744
CUBEWERT() 744

D

DATEDIF() 274
DATUM() 275
DATWERT() 277
DBANZAHL() 707
DBANZAHL2() 708
DBAUSZUG() 708
DBMAX() 709
DBMIN() 710
DBMITTELWERT() 711
DBPRODUKT() 712
DBSTDABW() 713
DBSTDABWN() 714
DBSUMME() 714
DBVARIANZ() 715
DBVARIANZEN() 716
DELTA() 410
DEZIMAL() 337
DEZINBIN() 411
DEZINHEX() 412
DEZINOKT() 413
DIA() 194
DISAGIO() 195
DM() 750
DURATION() 196

E

EDATUM() 278
EFFEKTIV() 198
ERSETZEN() 751
EXP() 339
EXPON.VERT() 488
EXPONVERT() 625

F

F.INV.RE() 491
F.INV() 490
F.TEST() 492
F.VERT.RE() 495
F.VERT() 493
FAKULTÄT() 340
FALSCH() 790
FEHLER.TYP() 803
FEST() 752
FINDEN() 753
FINV() 626
FISHER() 496
FISHERINV() 497
FORMELTEXT() 675
FTEST() 627
FVERT() 628

G

G.TEST() 498
GAMMA.INV() 501
GAMMA.VERT() 502

GAMMA() 499
GAMMAINV() 629
GAMMALN.GENAU() 504
GAMMALN() 504
GAMMAVERT() 629
GANZZAHL() 341
GAUSS() 505
GAUSSF.GENAU() 414
GAUSSFEHLER() 414
GAUSSFKOMPL.GENAU() 416
GAUSSFKOMPL() 415
GDA() 199
GDA2() 200
GEOMITTEL() 506
GERADE() 341
GESTUTZTMITTEL() 508
GGANZZAHL() 416
GGT() 342
GLÄTTEN() 754
GRAD() 343
GROSS() 755
GROSS2() 756
GTEST() 631

H

HARMITTEL() 511
HÄUFIGKEIT() 509
HEUTE() 279
HEXINBIN() 417
HEXINDEZ() 419
HEXINOKT() 419
HYPERLINK() 676
HYPGEOM.VERT() 512
HYPGEOMVERT() 632

I

IDENTISCH() 757
IKV() 202
IMABS() 420
IMAGINÄRTEIL() 421
IMAPOTENZ() 421
IMARGUMENT() 422
IMCOS() 422
IMCOSEC() 423
IMCOSECHYP() 423
IMCOSHYP() 424
IMCOT() 424
IMDIV() 425
IMEXP() 425
IMKONJUGIERTE() 426
IMLN() 427
IMLOG10() 427
IMLOG2() 428
IMPRODUKT() 428
IMREALTEIL() 429
IMSEC() 429
IMSECHYP() 430
IMSIN() 430
IMSINHYP() 431
IMSUB() 431
IMSUMME() 431
IMTAN() 432
IMWURZEL() 432
INDEX() 678
INDIREKT() 680
INFO() 804
ISOKALENDERWOCHE() 280
ISPMT() 203
ISTBEZUG() 806
ISTFEHL() 806
ISTFEHLER() 806
ISTFORMEL() 807
ISTGERADE() 807
ISTKTEXT() 808
ISTLEER() 808
ISTLOG() 809
ISTNV() 810
ISTTEXT() 810
ISTUNGERADE() 811
ISTZAHL() 812

J

JAHR() 281
JETZT() 282

K

KALENDERWOCHE() 283
KAPZ() 204
KGRÖSSTE() 514
KGV() 344
KKLEINSTE() 515
KLEIN() 758
KOMBINATIONEN() 345
KOMBINATIONEN2() 346
KOMPLEXE() 433
KONFIDENZ.NORM() 517
KONFIDENZ.T() 519
KONFIDENZ() 633
KORREL() 521
KOVAR() 635
KOVARIANZ.P() 523
KOVARIANZ.S() 524
KRITBINOM() 636
KUMKAPITAL() 205
KUMZINSZ() 206
KURS() 207
KURSDISAGIO() 208
KURSFÄLLIG() 210
KURT() 524
KÜRZEN() 347

L

LÄNGE() 759
LIA() 211
LINKS() 759
LN() 348
LOG() 348
LOG10() 350
LOGINV() 637
LOGNORM.INV() 526
LOGNORM.VERT() 527
LOGNORMVERT() 638

M

MAX() 528
MAXA() 529
MDET() 351
MDURATION() 212
MEDIAN() 529
MEINHEIT() 352
MIN() 531
MINA() 532
MINUTE() 286
MINV() 353
MITTELABW() 532
MITTELWERT() 534
MITTELWERTA() 536
MITTELWERTWENN() 538
MITTELWERTWENNS() 540
MMULT() 354
MODALWERT() 639
MODUS.EINF() 542
MODUS.VIELF() 543
MONAT() 286
MONATSENDE() 288
MTRANS() 681

N

N() 813
NBW() 213
NEGBINOM.VERT() 545
NEGBINOMVERT() 640
NETTOARBEITSTAGE.INTL() 290
NETTOARBEITSTAGE() 289
NICHT() 791
NOMINAL() 215
NORM.INV() 546
NORM.S.INV() 547

NORM.S.VERT() 548
NORM.VERT() 549
NORMINV() 641
NORMVERT() 642
NOTIERUNGBRU() 215
NOTIERUNGDEZ() 216
NV() 814

O

OBERGRENZE.MATHEMATIK() 355
OBERGRENZE() 644
ODER() 791
OKTINBIN() 434
OKTINDEZ() 435
OKTINHEX() 435

P

PDURATION() 217
PEARSON() 553
PHI() 554
PI() 356
PIVOTDATENZUORDNEN() 682
POISSON.VERT() 555
POISSON() 645
POLYNOMIAL() 357
POTENZ() 358
POTENZREIHE() 359
PRODUKT() 360

Q

QIKV() 218
QUADRATESUMME() 361
QUANTIL.EXKL() 557
QUANTIL.INKL() 558
QUANTIL() 646
QUANTILSRANG.EXKL() 559
QUANTILSRANG.INKL() 561
QUANTILSRANG() 647
QUARTILE.EXKL() 562
QUARTILE.INKL() 563
QUARTILE() 649
QUOTIENT() 362

R

RANG.GLEICH() 564
RANG.MITTELW() 565
RANG() 650
RECHTS() 761
RENDITE() 220
RENDITEDIS() 221
RENDITEFÄLL() 222
REST() 363
RGP() 566
RKP() 569
RMZ() 223
RÖMISCH() 365
RTD() 684
RUNDEN() 366

S

SÄUBERN() 764
SCHÄTZER() 572
SCHIEFE.P() 575
SCHIEFE() 574
SEC() 367
SECHYP() 368
SEKUNDE() 292
SIN() 369
SINHYP() 371
SPALTE() 684
SPALTEN() 686
STABW.N() 577
STABW.S() 578
STABW() 651
STABWA() 579
STABWN() 652
STABWNA() 581
STANDARDISIERUNG() 581

STANDNORMINV() 652
STANDNORMVERT() 653
STEIGUNG() 583
STFEHLERYX() 584
STUNDE() 293
SUCHEN() 764
SUMME() 372
SUMMENPRODUKT() 374
SUMMEWENN() 374
SUMMEWENNS() 376
SUMMEX2MY2() 377
SUMMEX2PY2() 378
SUMMEXMY2() 379
SUMQUADABW() 585
SVERWEIS() 687

T

T.INV.2S() 587
T.INV() 587
T.TEST() 590
T.VERT.2S() 594
T.VERT.RE() 595
T.VERT() 592
T() 765
TAG() 294
TAGE() 294
TAGE360() 295
TAN() 380
TANHYP() 382
TBILLÄQUIV() 225
TBILLKURS() 226
TBILLRENDITE() 227
TEIL() 766
TEILERGEBNIS() 383
TEXT() 768
TINV() 654
TREND() 595
TTEST() 656
TVERT() 657
TYP() 815

U

UMWANDELN() 436
UND() 793
UNGERADE() 384
UNICODE() 769
UNIZEICHEN() 770
UNREGER.KURS() 228
UNREGER.REND() 229
UNREGLE.KURS() 230
UNREGLE.REND() 232
UNTERGRENZE.MATHEMATIK() 385
UNTERGRENZE() 658
URLCODIEREN() 822

V

VAR.P() 598
VAR.S() 600
VARIANZ() 659
VARIANZA() 601
VARIANZEN() 660
VARIANZENA() 602
VARIATION() 603
VARIATIONEN() 605
VARIATIONEN2() 606
VDB() 233
VERGLEICH() 688
VERKETTEN() 771
VERWEIS() 690
VORZEICHEN() 386
VRUNDEN() 387

W

WAHL() 691
WAHR() 794
WAHRSCHBEREICH() 607
WEBDIENST() 825
WECHSELN() 772
WEIBULL.VERT() 609

WEIBULL() 661
WENN() 795
WENNFEHLER() 796
WENNNV() 797
WERT() 774
WIEDERHOLEN() 775
WOCHENTAG() 297
WURZEL() 388
WURZELPI() 389
WVERWEIS() 694

X

XINTZINSFUSS() 234
XKAPITALWERT() 235
XMLFILTERN() 825
XODER() 797

Z

ZÄHLENWENN() 610
ZÄHLENWENNS() 612
ZAHLENWERT() 776
ZEICHEN() 777
ZEILE() 695
ZEILEN() 696
ZEIT() 299
ZEITWERT() 300
ZELLE() 817
ZINS() 236
ZINSSATZ() 239
ZINSTERMNZ() 240
ZINSTERMTAGE() 241
ZINSTERMTAGNZ() 242
ZINSTERMTAGVA() 243
ZINSTERMVZ() 244
ZINSTERMZAHL() 245
ZINSZ() 246
ZSATZINVEST() 247
ZUFALLSBEREICH() 390
ZUFALLSZAHL() 391
ZW() 248
ZW2() 250
ZWEIFAKULTÄT() 392
ZZR() 251

D Funktionstasten, Tasten und Tastenkombinationen

D.1 Allgemeine Tastenfunktionen im Überblick

In der folgenden Tabelle finden Sie eine Reihe von einzelnen Tasten, Funktionstasten und Tastenkombinationen, mit denen Sie zahlreiche Aufgaben in Excel erledigen können, ohne das Menüband oder die Schnellzugriffsleiste bemühen zu müssen. Das ist oft eine große Hilfe, insbesondere für Leute, die sich durch zu viel Mausarbeit Empfindlichkeiten an den Sehnenscheiden zugezogen haben.

Für Tastenkombinationen werden mehrere Tasten entweder einzeln oder gemischt verwendet: Strg, ⇧, Alt, manchmal auch zwei davon gleichzeitig. In all diesen Fällen müssen alle Tasten der Kombination gleichzeitig gedrückt werden, anders also als bei den Tastenreihen, mit denen Befehle des Menübands ebenfalls aufgerufen werden können. Es sei noch der Hinweis erlaubt, dass in der Hilfe zu Excel einige Angaben zu den Tastenkombinationen für die deutsche Version nicht zutreffen.

Eine in früheren Versionen gerne verwendete Tastenkombination zum Anzeigen der Formeln anstelle der Werte – Strg + # – wird in der deutschen Version zwar zu dem entsprechenden Befehl in **Formeln • Formelüberwachung • Formel anzeigen** in der QuickInfo angezeigt, steht aber nicht zur Verfügung. Am einfachsten ist es, den Befehl über das Kontextmenü in die Schnellzugriffsleiste zu übernehmen. Dann steht er ständig zur Verfügung.

Taste/Tastenkombination	Beschreibung
Alt	Zeigt die Tasteninfos für den Zugriff auf die Befehle im Menüband.
Alt + =	Fügt die Funktion `=SUMME()` ein.
Alt + F1	Erstellt aus den markierten Daten ein eingebettetes Diagramm mit dem Standarddiagrammtyp.
Alt + F4	Beendet Excel.
Alt + F8	Zeigt das Dialogfeld **Makro** an.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen

Taste/Tastenkombination	Beschreibung
Alt + F11	Öffnet das Fenster des VBA-Editors.
Alt + ⇧ + F1	Fügt ein neues Tabellenblatt in die Arbeitsmappe ein.
↵	Bestätigt eine Eingabe oder einen Befehl.
Esc	Bricht eine Eingabe oder einen Befehl ab.
F1	Ruft die Hilfe auf.
F2	Öffnet die Zelle zur Bearbeitung.
F4	Wiederholt eine Aktion oder einen Befehl.
F5	Öffnet den Dialog **Gehe zu**.
F6	Wechselt zwischen Fenster, Menüband und Aufgabenbereich oder springt zum nächsten Fensterausschnitt, wenn das Fenster geteilt ist.
F7	Aktiviert die Rechtschreibprüfung.
F8	Schaltet den Erweiterungsmodus ein oder aus.
F9	Startet die Neuberechnung in allen offenen Arbeitsmappen, falls die automatische Neuberechnung abgeschaltet ist.
F10	Zeigt die Tasteninfos für den Zugriff auf die Befehle im Menüband.
F11	Erstellt aus den aktuell markierten Tabellendaten ein Diagramm auf einem eigenen Blatt mit dem Standarddiagrammtyp.
F12	Öffnet den Dialog **Speichern unter**.
Strg + .	Fügt das aktuelle Systemdatum in die Zelle oder das Eingabefeld ein.
Strg + :	Fügt die aktuelle Systemzeit in die Zelle oder das Eingabefeld ein.
Strg + ,	Kopiert die Formel der Zelle über der aktiven Zelle in die aktive Zelle.
Strg + -	Entspricht dem Befehl **Zellen löschen**.
Strg + +	Öffnet den Dialog **Zellen einfügen**.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen (Forts.)

Taste/Tastenkombination	Beschreibung
Strg + 1	Öffnet den Dialog **Zellen formatieren**.
Strg + 2	Formatiert die Auswahl fett oder hebt die Formatierung auf.
Strg + 3	Formatiert die Auswahl kursiv oder hebt die Formatierung auf.
Strg + 4	Unterstreicht die Auswahl oder hebt die Formatierung auf.
Strg + 5	Streicht die Auswahl durch oder hebt die Formatierung auf.
Strg + 6	Blendet grafische Objekte ein oder aus.
Strg + 7	Blendet die Gliederungssymbole ein oder aus.
Strg + 8	Blendet die Spalten aus, in der die aktuell markierten Zellen liegen, wobei auch eine Mehrfachauswahl möglich ist.
Strg + 9	Blendet die Zeilen aus, in der die aktuell markierten Zellen liegen, wobei auch eine Mehrfachauswahl möglich ist.
Strg + Alt + V	Öffnet den Dialog **Inhalte einfügen**.
Strg + Bild ↑	vorheriges Blatt
Strg + Bild ↓	nächstes Blatt
Strg + A	Markiert das gesamte Tabellenblatt.
Strg + C	Kopiert die aktuelle Auswahl in die Zwischenablage.
Strg + F	Öffnet den Dialog **Suchen und Ersetzen** mit dem Register **Suchen**.
Strg + F1	Blendet das Menüband aus und ein.
Strg + F10	Schaltet für die aktive Arbeitsmappe die maximierte Ansicht ein und aus.
Strg + F11	Fügt ein neues Blatt für Excel 4-Makros in die Arbeitsmappe ein.
Strg + F2	Zeigt die Druckvorschau an.
Strg + F3	Öffnet den Dialog des Namens-Managers.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen (Forts.)

Taste/Tastenkombination	Beschreibung
Strg + F4	Entspricht dem Befehl **Datei • Schließen**.
Strg + F5	Stellt die gewählte Größe des Arbeitsmappenfensters wieder her, wenn es maximiert worden ist.
Strg + F6 oder Strg + ⇆	Wechselt zur nächsten offenen Arbeitsmappe, falls mehrere Mappen gleichzeitig geöffnet sind.
Strg + F7	Erlaubt das Verschieben des Arbeitsmappenfensters, falls dieses nicht maximiert ist.
Strg + F8	Erlaubt das Ändern der Größe des Arbeitsmappenfensters, falls dieses nicht maximiert ist.
Strg + F9	Reduziert das Arbeitsmappenfenster auf ein Symbol.
Strg + G	Öffnet den Dialog **Gehe zu**.
Strg + H	Öffnet den Dialog **Suchen und Ersetzen** mit dem Register **Ersetzen**.
Strg + K	Öffnet den Dialog **Hyperlink einfügen**.
Strg + L	Öffnet den Dialog **Tabelle erstellen**.
Strg + Leertaste	Wählt die Spalte der aktiven Zelle aus.
Strg + N	Fügt eine neue Arbeitsmappe ein.
Strg + O	Öffnet den Dialog **Öffnen**.
Strg + P	Öffnet das Register **Datei** mit der Seite **Drucken**.
Strg + R	Kopiert in einem mehrspaltigen Bereich den Inhalt und das Format der ersten Spalte in die rechts benachbarten Zellen, entspricht also dem Befehl **Füllbereich • Rechts**.
Strg + S	Öffnet den Dialog **Speichern unter**.
Strg + ⇆	Wechselt zur nächsten Arbeitsmappe, wenn mehrere Mappen geöffnet sind.
Strg + U	Kopiert in einem mehrzeiligen Bereich den Inhalt und das Format der ersten Zeile in die darunter liegenden Zellen, entspricht also dem Befehl **Füllbereich • Unten**.
Strg + ⇧ + +	Markiert den kompletten Datenblock, in dem die aktive Zelle liegt.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen (Forts.)

Taste/Tastenkombination	Beschreibung
Strg + ⇧ + _	Umgibt die Markierung mit einem Außenrahmen.
Strg + ⇧ + :	Kopiert den Wert der Zelle über der aktiven Zelle in die aktive Zelle, bei einer Formel also das Ergebnis, nicht die Formel.
Strg + ⇧ + /	Wählt das komplette Array aus, in dem die aktive Zelle liegt.
Strg + ⇧ + F12	Öffnet das Register **Datei** mit der Seite **Drucken**.
Strg + ⇧ + F3	Öffnet den Dialog **Namen aus Auswahl erstellen**, um Werte mit den benachbarten Beschriftungen zu benennen.
Strg + ⇧ + Leertaste	Wählt das ganze Tabellenblatt aus.
Strg + ⇧ + 0	Wählt alle Zellen aus, für die Kommentare vorhanden sind.
Strg + ⇧ + ⇆	Wechselt zur vorherigen Arbeitsmappe, wenn mehrere Mappen geöffnet sind.
Strg + ⇧ + (	Blendet ausgeblendete Spalten in dem markierten Bereich ein.
Strg + ⇧ +)	Blendet ausgeblendete Zeilen in dem markierten Bereich ein.
Strg + ⇧ + $	Weist der Markierung ein zweistelliges Währungsformat zu.
Strg + ⇧ + %	Weist der Markierung ein Prozentformat zu.
Strg + ⇧ + !	Weist der Markierung ein zweistelliges Zahlenformat zu.
Strg + ⇧ + *	Markiert den aktuellen Zellbereich, der um die aktive Zelle herum vorhanden ist.
Strg + ⇧ + U	Unterstreicht die Auswahl oder hebt die Unterstreichung auf.
Strg + V	Fügt die aktuelle Auswahl aus der Zwischenablage ein.
Strg + W	Schließt das ausgewählte Arbeitsmappenfenster.
Strg + X	Schneidet die aktuelle Auswahl aus und übernimmt sie in die Zwischenablage.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen (Forts.)

Taste/Tastenkombination	Beschreibung
Strg + Y	Wiederholt eine Aktion oder einen Befehl.
Strg + Z	Macht eine Aktion oder einen Befehl rückgängig.
⇧ + F10	Zeigt das Kontextmenü der aktuellen Auswahl an.
⇧ + F11	Fügt ein neues Tabellenblatt in die Arbeitsmappe ein.
⇧ + F2	Fügt einen neuen Kommentar hinzu oder öffnet einen Kommentar zur Bearbeitung.
⇧ + F3	Öffnet den Dialog **Funktion einfügen**.
⇧ + F5	Öffnet den Dialog **Suchen und Ersetzen**.
⇧ + F6	Rücksprung zwischen Fenster, Menüband und Aufgabenbereich oder Sprung zum nächsten Fensterausschnitt, wenn das Fenster geteilt ist
⇧ + F8	Schaltet den Modus **Hinzufügen** ein oder aus, um eine bestehende Auswahl zu erweitern.
⇧ + F9	Berechnet nur das aktive Tabellenblatt neu, falls die automatische Neuberechnung ausgeschaltet ist.
⇧ + F12	Öffnet den Dialog **Speichern unter**.
⇧ + Leertaste	Wählt die Zeile der markierten Zellen aus.
⇧ + ↵	Reduziert Auswahl auf aktive Zelle.

Tabelle D.1 Liste der Tasten und Tastenkombinationen, die bei der Arbeit mit Excel zur Verfügung stehen (Forts.)

D.2 Editiertasten und -tastenkombinationen

Die Tasten und Tastenkombinationen in der folgenden Tabelle können nur bei der Bearbeitung von Formeln oder anderen Zellinhalten verwendet werden.

Taste/Tastenkombination	Beschreibung
Alt + ↵	Beginnt eine neue Zeile in der Bearbeitungsleiste.
Ende	zum Ende des Eingabefelds

Tabelle D.2 Liste der Tasten und Tastenkombinationen, die beim Editieren in der Bearbeitungsleiste oder in der Zelle verwendet werden können

Taste/Tastenkombination	Beschreibung
F2	Wechselt zwischen den Modi **Bearbeiten** und **Eingeben**.
F3	Öffnet bei der Eingabe von Formeln den Dialog **Namen einfügen**.
F4	Wechsel zwischen relativen und absoluten Adressen
F9	Verwandelt eine Formel in ihr Ergebnis.
→ / ←	ein Zeichen nach rechts/links
Pos1	zum Anfang des Eingabefelds
Strg + → / ←	zum Anfang des nächsten/letzten Wortes oder zum Ende/Anfang einer Zahl
Strg + ⇧ + → / ←	Erweitert die Markierung nach rechts/links.
⇧ + → / ←	Erweitert die Markierung nach rechts/links.

Tabelle D.2 Liste der Tasten und Tastenkombinationen, die beim Editieren in der Bearbeitungsleiste oder in der Zelle verwendet werden können (Forts.)

D.3 Navigations- und Markierungstasten

Die Navigations- und Markierungstasten, mit denen Sie sich möglichst schnell auf dem Tabellenblatt bewegen und Bereiche auswählen können, wurden zwar in Kapitel 1, »Einstieg in Berechnungen mit Excel«, ab Seite 41 schon vorgestellt, sollen aber hier noch einmal beigefügt werden, damit Sie nicht an zwei Stellen suchen müssen.

Taste/Tastenkombination	Wirkung
Bild ↑	ein Fenster nach oben
Bild ↓	ein Fenster nach unten
← oder ⇧ + ↹	Spalte nach links
↑	Zeile nach oben
→ oder ↹	Spalte nach rechts
↓	Zeile nach unten
Pos1	Sprung zum Zeilenanfang

Tabelle D.3 Die Tastenfunktionen, mit denen Sie sich in der Tabelle bewegen können

Taste/Tastenkombination	Wirkung
Strg + Bild ↑	Sprung zum vorherigen Blatt
Strg + Bild ↓	Sprung zum nächsten Blatt
Strg + Ende	Sprung an das Ende des benutzten Tabellenbereichs
Strg + ←	Macht die aktive Zelle wieder sichtbar, wenn der Ausschnitt vorher mit einer Bildlaufleiste oder mit Rollen verschoben worden ist.
Strg + ←	Sprung in der Zeile nach links zum Anfang eines Datenblocks bzw. zum Ende des nächsten Datenblocks
Strg + ↑	Sprung in der Spalte nach oben zum Anfang eines Datenblocks bzw. zum Ende des nächsten Datenblocks
Strg + →	Sprung in der Zeile nach rechts zum Ende eines Datenblocks bzw. zum Anfang des nächsten Datenblocks
Strg + ↓	Sprung in der Spalte nach unten zum Ende eines Datenblocks bzw. zum Anfang des nächsten Datenblocks
Strg + Pos1	Sprung zum Tabellenanfang

Tabelle D.3 Die Tastenfunktionen, mit denen Sie sich in der Tabelle bewegen können (Forts.)

Tastenkombination	Wirkung
Strg + ⇧ + Richtungstaste	Ausweiten der Auswahl zum Ende/Anfang des Blocks in die Richtung der jeweiligen Taste bzw. zum Anfang/Ende des nächsten Datenblocks
Strg + ⇧ + Ende	Ausweiten der Auswahl bis an das Tabellenende
Strg + ⇧ + Leertaste	Auswählen der ganzen Tabelle
Strg + ⇧ + Pos1	Ausweiten der Auswahl bis zum Tabellenanfang
⇧ + Richtungstaste	Ausweiten der Auswahl um eine Zeile oder Spalte
⇧ + Bild ↑	Ausweiten der Auswahl um ein Fenster nach oben
⇧ + Bild ↓	Ausweiten der Auswahl um ein Fenster nach unten
⇧ + Leertaste	Auswählen der ganzen Zeile
⇧ + ←	Reduzieren der Auswahl auf die aktive Zelle
⇧ + Pos1	Ausweiten der Auswahl bis zum Zeilenanfang

Tabelle D.4 Liste der Tastenkombinationen für die Markierung von Zellbereichen

Index

####-Anzeige 44
#NV 63
.xlam 874
1904-Datumswert 127, 263
3D-Bezug 84

A

ABRUNDEN() 311
ABS() 312
Abschreibung 178
Abschreibungsrechnung 171
Absolute Bezüge 98
ACCRINT() 187
ACCRINTM() 189
ACHSENABSCHNITT() 465
Achsenoptionen 140
Achsenskalierung 139
Achsenspezifikation 725
ACOS() 317
ACOSH() 319
ACOT() 320
ACOTH() 320
Add-In 843
Addition 72
ADDRESS() 668
ADRESSE() 668
Adressen, Bezug 98
AGGREGAT() 304, 313
AGGREGATE() 313
Alpha 846
AMORDEGRC() 184
AMORDEGRK() 184
AMORLINC() 186
AMORLINEARK() 186
Analyse-Funktionen 843
  Einfaktorielle Varianzanalyse 846
  Exponentielles Glätten 853
  Fourieranalyse 855
  Gleitender Durchschnitt 858
  Histogramm 856
  Korrelation 851
  Kovarianz 852
  Populationskenngrößen 853
  Rang und Quantil 861
  Regression 862
  Stichprobenziehung 864
  Zufallszahlengenerierung 860
  Zweifaktorielle Varianzanalyse 848
  Zwei-Stichproben F-Test 855
  Zweistichproben t-Test 866
  Zwei-Stichprobentest bei bekannten Varianzen 868
AND() 793
Änderungswert, maximaler 127
Anova-Funktionen 843
ANZAHL() 466
ANZAHL2() 468
ANZAHLLEEREZELLEN() 471
Application-Objekt 874
ARABIC() 316
ARABISCH() 316
Arbeitsmappe
  Optionen 127
  Zuletzt verwendet 35
ARBEITSTAG() 269, 892
ARBEITSTAG.INTL() 271
ARCCOS() 317
ARCCOSHYP() 319
ARCCOT() 320
ARCCOTHYP() 320
ARCSIN() 321
ARCSINHYP() 323
ARCTAN() 324
ARCTAN2() 325
ARCTANHYP() 326
AREAS() 673
Arithmetische Formeln 69
ASIN() 321
ASINH() 323

ATAN() 324
ATAN2() 325
ATANH() 326
AUFGELZINS() 187
AUFGELZINSF() 189
AUFRUNDEN() 327
Ausgabeformat 43
Auswahl
Bereichsmarkierung 41
per Tastatur 40
AUSZAHLUNG() 190
AutoVervollständigen 618
AutoVervollständigen für Formeln 87
AVEDEV() 532
AVERAGE() 534
AVERAGEA() 536
AVERAGEIF() 538
AVERAGEIFS() 540

B

BAHTTEXT() 747
BASE() 328
BASIS() 328
Bearbeitungsleiste 79
Bedingte Formatierung 128
Benutzerdefinierte Formate 60
Benutzerdefiniertes Zahlenformat 268
Berechnen
periodische Datumsreihen 257
periodische Zeitreihen 260
Berechnung
Anzahl der Wiederholungen und minimale Abweichung 126
Arbeitszeiten 262
Kontrolle iterativer Berechnungen 126
Registerkarte 125
Berechnungsoptionen 125
BEREICH.VERSCHIEBEN() 670
BEREICHE() 673
Bereichsadresse 97
Bereichsmarkierung 41
BESSELI() 401
BESSELJ() 402
BESSELK() 403
BESSELY() 404
BESTIMMTHEITSMASS() 473
BETA.DIST() 475
BETA.INV() 474
BETA.VERT() 475
BETADIST() 619
BETAINV() 618
BETAVERT() 619
Betaverteilung 458
Bewegen in der Tabelle 969
Bezüge
absolute 98
eingeben 80
gemischte 98
Namen für Bezüge 103
relativ und absolut 97
Bildschirmtastatur 39
BIN2DEC() 404
BIN2HEX() 405
BIN2OCT() 406
BININDEZ() 404
BININHEX() 405
BININOKT() 406
BINOM.DIST.RANGE() 480
BINOM.DIST() 478
BINOM.INV() 477
BINOM.VERT.BEREICH() 480
BINOM.VERT() 478
BINOMDIST() 620
Binomialverteilung 457
BINOMVERT() 620
BI-Projekt 719
BITAND() 409
BITLSHIFT() 407
BITLVERSCHIEB() 407
BITODER() 408
BITOR() 408
BITRSHIFT() 408
BITRVERSCHIEB() 408
BITUND() 409
BITXODER() 410
BITXOR() 410
BLATT() 801
BLÄTTER() 802
Blattname 105
BlattName() 877
BOGENMASS() 329
BRTEILJAHRE() 273
BW() 191

C

CEILING() 644
CEILING.PRECISE() 355
CELL() 817
CHAR() 777
CHIDIST() 624
CHIINV() 621
CHIQU.INV() 481
CHIQU.INV.RE() 482
CHIQU.TEST() 482
CHIQU.VERT() 484
CHIQU.VERT.RE() 487
Chi-Quadrat-Verteilung 458
CHISQ.DIST() 484
CHISQ.DIST.RT() 487
CHISQ.INV() 481
CHISQ.INV.RT() 482
CHISQ.TEST() 482
CHITEST() 622
CHIVERT() 624
CHOOSE() 691
CLEAN() 764
CODE() 748
COLUMN() 684
COLUMNS() 686
COMBIN() 345
COMBINA() 346
COMPLEX() 433
CONCATENATE() 771
CONFIDENCE() 633
CONFIDENCE.NORM() 517
CONFIDENCE.T() 519
CONVERT() 436
CORREL() 521
COS() 330
COSEC() 332
COSECHYP() 333
COSH() 334
COSHYP() 334
COT() 335
COTH() 337
COTHYP() 337
COUNT() 466
COUNTA() 468
COUNTBLANK() 471
COUNTIF() 610
COUNTIFS() 612
COUPDAYBS() 243
COUPDAYS() 241
COUPDAYSNC() 242
COUPNCD() 240
COUPNUM() 245
COUPPCD() 244
COVAR() 635
COVARIANCE.P() 523
COVARIANCE.S() 524
CRITBINOM() 636
CSC() 332
CSCH() 333
CUBEELEMENT() 739
CUBEELEMENTEIGENSCHAFT() 740
Cube-Funktionen 718, 841
  Browser 724
  Datenquelle 721
  definieren 722
  Dimension 719
  Hierarchien 719
  Measure 719
  OLAP 718
  Sternschema 720
  verarbeiten 724
CUBEKPIELEMENT() 740
CUBEKPIMEMBER() 740
CUBEMEMBER() 739
CUBEMEMBERPROPERTY() 740
CUBEMENGE() 742
CUBEMENGENANZAHL() 743
CUBERANGEELEMENT() 744
CUBERANKEDMEMBER() 744
CUBESET() 742
CUBESETCOUNT() 743
CUBEVALUE() 744
CUBEWERT() 744
CUMIPMT() 206
CUMPRINC() 205

D

DATE() 275
DATEDIF() 274
Datenanalyse 843
Datenbanken 699
Datenmodell 829

Datenquellensicht 721
Datenschnitt 836
Datentabelle 159
Datentools 115, 159
Datentyp 799, 877
Datenüberprüfung 115
DATEVALUE() 277
Datum
  als numerischer Wert 45
  Datumsformat 55, 58
  Datumsfunktion 276
  Datumswert, 1904 127, 262
DATUM() 275
Datumsseriennummer 127
DATWERT() 277
DAVERAGE() 711
DAY() 294
DAYS () 294
DAYS360() 295
DB() 200
DBANZAHL() 707
DBANZAHL2() 708
DBAUSZUG() 708
DBMAX() 709
DBMIN() 710
DBMITTELWERT() 711
DBPRODUKT() 712
DBSTDABW() 713
DBSTDABWN() 714
DBSUMME() 714
DBVARIANZ() 715
DBVARIANZEN() 716
DCOUNT() 707
DCOUNTA() 708
DDB() 199
DEC2BIN() 411
DEC2HEX() 412
DEC2OCT() 413
DECIMAL() 337
DEGREES() 343
DELTA() 410
Desktop-Umgebung 33
DEVSQ() 585
DEZIMAL() 337
Dezimalstellen 61
Dezimalsystem 396
DEZINBIN() 411
DEZINHEX() 412
DEZINOKT() 413
DGET() 708
DIA() 194
Diagramme 134
Diagrammtyp ändern 141
Diagrammtypen 137
Dimension Tables 720
DISAGIO() 195
DISC() 195
Division 72
  durch null 73
DM() 750
DMAX() 709
DMIN() 710
DOLLAR() 750
DOLLARDE() 216
DOLLARFR() 215
Do-Loop 883
DPRODUCT() 712
DSTDEV() 713
DSTDEVP() 714
DSUM() 714
Dualsystem 396
DURATION() 196
DVAR() 715
DVARP() 716

E

EDATE() 278
EDATUM() 278
EFFECT() 198
EFFEKTIV() 198
EFFRZINS() 887
EFFZINS() 886
Eingabe
  von Bezügen 80
  von Zahlen 42
Eingabeformat 43
Empfohlene Diagramme 134
ENCODEURL() 822
Endkapital() 884
EOMONTH() 288
ERF() 414
ERF.PRECISE() 414
ERFC() 415

ERFC.PRECISE() 416
ERROR.TYPE() 803
ERSETZEN() 751
EVEN() 341
EXACT() 757
Excel-Add-In 874
Excel-Optionen 35
EXP() 339
EXPON.DIST() 488
EXPON.VERT() 488
EXPONDIST() 625
Exponentialverteilung 458
EXPONVERT() 625

F

F.DIST() 493
F.DIST.RT() 495
F.INV.RE() 491
F.INV.RT() 491
F.INV() 490
F.TEST() 492, 845
F.VERT() 493
F.VERT.RE() 495
Fact Tables 720
FACT() 340
FACTDOUBLE() 392
FAKULTÄT() 340
FALSCH() 790
FALSE() 790
FDIST() 628
FEHLER.TYP() 803
Fehlerindikatoren 121
Fehlersuche in Tabellen 122
Fehlerüberprüfung 120, 122
Fehlerwerte 120
FEST() 752
FILTERXML() 825
FIND() 753
FINDEN() 753
Fingereingabe-/Mausmodus 36
FINV() 626
FISHER() 496
FISHERINV() 497
FIXED() 752
FLOOR() 658
FLOOR.PRECISE() 385
FORECAST() 572
Format
  bedingte Formatierung 129
Formatbeschreibung 59
Formatcodes 61
Formatierung
  Löschen des Zellformats 53
  Zahlengröße und Spaltenbreite 44
Formatsymbole 54
Formeln
  arithmetische 69
  benannte Werte definieren 107
  Eingabe über Funktions-Assistenten 90
  externe Bezüge 84
  Fehler durch Werte 119
  logische 69, 74
  Namen anwenden 108
  Typen 69
  Verwendung von Bereichen 82
  Verwendung von Formeln 79
  Zeichenfolgen 69
FORMELTEXT() 675
Formelüberwachung 122, 125
FORMULATEXT() 675
For-Next 883
Fremdschlüssel 829
FREQUENCY() 509
FTEST() 627
Function 871
Funktionen 69, 76
  bearbeiten 92
  Eingabe 87
  verschachtelte 93
Funktions-Assistent 90, 93
FV() 248
FVERT() 628, 847
F-Verteilung 458
FVSCHEDULE() 250

G

G.TEST() 498, 868
GAMMA() 499
GAMMA.DIST() 502
GAMMA.INV() 501
GAMMA.VERT() 502
GAMMADIST() 629

GAMMAINV() 629
GAMMALN.GENAU() 504
GAMMALN.PRECISE() 504
GAMMALN() 504
GAMMAVERT() 629
Gammaverteilung 458
GANZZAHL() 341
GAUSS() 505
GAUSSF.GENAU() 414
GAUSSFEHLER() 414
GAUSSFKOMPL.GENAU() 416
GAUSSFKOMPL() 415
GCD() 342
GDA() 199
GDA2() 200
Gemischte Bezüge 98
Genauigkeit wie angezeigt 127
GEOMEAN() 506
GEOMITTEL() 506
GERADE() 341
GESTEP() 416
GESTUTZTMITTEL() 508
GETPIVOTDATA() 682
GGANZZAHL() 416
GGT() 342
Gitternetzlinien 143
GLÄTTEN() 754
Gleichung mit einer Unbekannten 147
Gleichungssysteme mit mehreren Unbekannten 150
GRAD() 343
Grenzwertbericht 157
GROSS() 755
GROSS2() 756
GROWTH() 603
Grundgesamtheit 447
Gruppe, Analyse 843
GTEST() 631
Gültigkeitsregeln 115

H

HARMEAN() 511
HARMITTEL() 511
Häufigkeit 858
HÄUFIGKEIT() 509
HEUTE() 279
HEX2BIN() 417
HEX2DEC() 419
HEX2OCT() 419
Hexadezimalsystem 396
HEXINBIN() 417
HEXINDEZ() 419
HEXINOKT() 419
Hilfe bei der Eingabe von Funktionen 88
HLOOKUP() 694
Höchstzeit der Berechnung 156
HOUR() 293
Hypergeometrische Verteilung 458
HYPERLINK() 676
Hyperlink-Adresse 676
Hyperlink-Formel 677
HYPGEOM.DIST() 512
HYPGEOM.VERT() 512
HYPGEOMVERT() 632

I

IDENTISCH() 757
IF() 795
IFERROR() 796
IFNA() 797
If-Then 882
IKV() 202
IMABS() 420
IMAGINARY() 421
IMAPOTENZ() 421
IMARGUMENT() 422
IMCONJUGATE() 426
IMCOS() 422
IMCOSEC() 423
IMCOSECHYP() 423
IMCOSH() 424
IMCOSHYP() 424
IMCOT() 424
IMCSC() 423
IMCSCH() 423
IMDIV() 425
IMEXP() 425
IMKONJUGIERTE() 426
IMLN() 427
IMLOG10() 427
IMLOG2() 428
IMPOWER() 421
IMPRODUCT() 428
IMPRODUKT() 428

IMREAL() 429
IMREALTEIL() 429
IMSEC() 429
IMSECH() 430
IMSECHYP() 430
IMSIN() 430
IMSINH() 431
IMSINHYP() 431
IMSQRT() 432
IMSUB() 431
IMSUM() 431
IMSUMME() 431
IMTAN() 432
IMWURZEL() 432
INDEX() 678
INDIRECT() 680
INDIREKT() 680
INFO() 804
Informationsfunktionen 799
bedingte Berechnung 799
INT() 341
INTERCEPT() 465
INTRATE() 190
Investitionsrechnung 171
IPMT() 246
IRR() 202
ISBLANK() 808
ISERR() 806
ISERROR() 806
ISEVEN() 807
ISFORMULA() 807
ISLOGICAL() 809
ISNA() 810
ISNONTEXT() 808
ISNUMBER() 812
ISODD() 811
ISOKALENDERWOCHE() 280
ISOWEEKNUM() 280
ISPMT() 203
ISREF() 806
ISTBEZUG() 806
ISTEXT() 810
ISTFEHL() 806
ISTFEHLER() 806
ISTFORMEL() 807
ISTGERADE() 807
ISTKTEXT() 808
ISTLEER() 267, 808
ISTLOG() 809
ISTNV() 810
ISTTEXT() 810
ISTUNGERADE() 811
ISTZAHL() 812
Iterationszahl, maximale 127
Iterative Berechnungen 126

J

JAHR() 281
JETZT() 282

K

KALENDERWOCHE() 283, 890
KAPZ() 204
KGRÖSSTE() 514
KGV() 344
KKLEINSTE() 515
Klassen 857
KLEIN() 758
KOMBINATIONEN() 345
Kompatible Funktionen 615
Kompatible statistische Funktionen 615
KOMPLEXE() 433
KONFIDENZ() 633
KONFIDENZ.NORM() 517
KONFIDENZ.T() 519
Koordinatensystem 136
KORREL() 521, 851
Korrelationskoeffizient 451
KOVAR() 635
KOVARIANZ.P() 523
KOVARIANZ.S() 524
KPI 728
Kreisumfang 875
KRITBINOM() 636
KUMKAPITAL() 205
KUMZINSZ() 206
KURS() 207
KURSDISAGIO() 208
KURSFÄLLIG() 210
Kursrechnung 172
KURT() 524
KÜRZEN() 347

L

LÄNGE() 759
LARGE() 514
LCM() 344
LEFT() 759
LEN() 759
LIA() 211
LINEST() 566
LINKS() 759
LN() 348
LOG() 348
LOG10() 350
Logarithmische Normalverteilung 458
LOGEST() 569
LOGINV() 637
Logische Formeln 69, 74
LOGNORM.DIST() 527
LOGNORM.INV() 526
LOGNORM.VERT() 527
LOGNORMDIST() 638
LOGNORMVERT() 638
LOOKUP() 690
Löschen
Zellformat 53
Zellinhalt 53
LOWER() 758

M

MATCH() 688
MAX() 528
MAXA() 529
Maximaler Änderungswert 127
MDET() 351
MDETERM() 351
MDURATION() 212
MDX 725
FROM-Klausel 726
Funktionen 726
Mengenausdruck 727
SELECT 725
Tupel 727
WHERE-Klausel 726
Measures 720
MEDIAN() 529
Mehrfachoperationen 158
Merkmal 445
MID() 766
MIN() 531
MINA() 532
Minisymbolleiste 55
MINUTE() 286
MINV() 352, 353
MINVERSE() 352, 353
MIRR() 218
MITTELABW() 532
MITTELWERT() 534
MITTELWERTA() 536
MITTELWERTWENN() 538
MITTELWERTWENNS() 540
MMULT() 354
MOD() 363
MODALWERT() 639
MODE.MULT() 543
MODE.SNGL() 542
MODE() 639
MODUS.EINF() 542
MODUS.VIELF() 543
MONAT() 286
MONATSENDE() 288
MONTH() 286
MROUND() 387
MTRANS() 681
Multidimensional Expressions 725
MULTINOMIAL() 357
Multiplikation 72

N

N() 813
NA() 814
Namen
blattspezifisch 105
definieren 105
erstellen 105, 106
festlegen 103
übernehmen 108
Namens-Manager 107
NBW() 213
Negative Binomialverteilung 457
NEGBINOM.DIST() 545
NEGBINOM.VERT() 545
NEGBINOMDIST() 640
NEGBINOMVERT() 640
NETTOARBEITSTAGE.INTL() 290

NETTOARBEITSTAGE() 266, 289, 892
NETWORKDAYS.INTL() 290
NETWORKDAYS() 289
Neuberechnung, Berechnungsoptionen 125
NICHT() 791
NOMINAL() 215
NOMRZINS() 887
NORM.DIST() 549
NORM.INV() 546
NORM.S.DIST() 548
NORM.S.INV() 547
NORM.S.VERT() 548
NORM.VERT() 549
Normalverteilung 459
NORMDIST() 642
NORMINV() 641
NORMSDIST() 653
NORMSINV() 652
NORMVERT() 642
NOT() 791
NOTIERUNGBRU() 215
NOTIERUNGDEZ() 216
NOW() 282
NPER() 251
NPV() 213
Nullhypothese 846
NUMBERVALUE() 776
Numerische Werte
  Datum 45
  Uhrzeit 45
  Zahlen 42
NV() 814

O

OBERGRENZE() 644
OBERGRENZE.GENAU() 355
Objektkatalog 876
Objektmodell 876
OCT2BIN() 434
OCT2DEC() 435
OCT2HEX() 435
ODD() 384
ODDFPRICE() 228
ODDFYIELD() 229
ODDLPRICE() 230
ODDLYIELD() 232
ODER() 791
OFFSET() 670
Oktalsystem 396
OKTINBIN() 434
OKTINDEZ() 435
OKTINHEX() 435
OLAP 718
OLE DB 721
Operatoren 880
OR() 791
OSTERDATUM() 892

P

PDURATION() 217
PEARSON() 553
PERCENTILE() 646
PERCENTILE.EXC() 557
PERCENTILE.INC() 558
PERCENTRANK() 647
PERCENTRANK.EXC() 559
PERCENTRANK.INC() 561
PERMUT() 605
PERMUTATIONA() 606
Personal.xlsb 871
PHI() 554
PI() 356
PivotChart-Felder 839
PIVOTDATENZUORDNEN() 682
Pivot-Diagramm 839
Pivotieren 829
Pivot-Tabelle
  anlegen 831
  Layout 834
  Quelle 832
  Wertfeldeinstellungen 837
PivotTable 831
PivotTable-Felder 832
PivotTable-Format 835
PivotTable-Name 832
PivotTable-Szenariobericht 168
PivotTable-Tools 832
PMT() 223
POISSON() 645
POISSON.DIST() 555
POISSON.VERT() 555
Poisson-Verteilung 458
POLYNOMIAL() 357
POTENZ() 358

POTENZREIHE() 359
POWER() 358
PPMT() 204
PRICE() 207
PRICEDISC() 208
PRICEMAT() 210
PROB() 607
PRODUCT() 360
PRODUKT() 360
Programm-Kachel 33
PROPER() 756
PV() 191

Q

QIKV() 218
QUADRATESUMME() 361
QUANTIL() 646
QUANTIL.EXKL() 557
QUANTIL.INKL() 558
QUANTILSRANG() 647, 861
QUANTILSRANG.EXKL() 559
QUANTILSRANG.INKL() 561
QUARTILE() 649
QUARTILE.EXC() 562
QUARTILE.EXKL() 562
QUARTILE.INC() 563
QUARTILE.INKL() 563
QuickInfos, zu Funktionen 87
QUOTIENT() 362

R

RADIANS() 329
RAND() 391
RANDBETWEEN() 390
RANG.GLEICH() 564
RANG.MITTELW() 565
RANG() 650, 861
RANK() 650
RANK.AVG() 565
RANK.EQ() 564
RATE() 236
Ratenkredite 886
RECEIVED() 239
RECHTS() 761
RENDITE() 220
RENDITEDIS() 221
RENDITEFÄLL() 222
REPLACE() 751
REPT() 775
Residuen 864
REST() 363
RGP() 566
RIGHT() 761
RKP() 569
RMZ() 223
ROMAN() 365
RÖMISCH() 365
ROUND() 366
ROUNDDOWN() 311
ROUNDUP() 327
ROW() 695
ROWS() 696
RRI() 247
RSQ() 473
RTD() 684
RUNDEN() 366

S

SÄUBERN() 764
Säulendiagramm 138
SCHÄTZER() 572
SCHIEFE() 574
SCHIEFE.P() 575
Schnittmenge 75
Schnittmengenbezüge 75
SEARCH() 764
SECOND() 292
SEKUNDE() 292
Select Case 882
Sensitivitätsbericht 157
Serielle Zahl 255
SERIESSUM() 359
SHEET() 801
SHEETS() 802
SIGN() 386
SIN() 367, 368, 369
SINH() 371
SINHYP() 371
Skalierung 445
SKEW() 574
SKEW.P() 575

SLN() 211
SLOPE() 583
SMALL() 515
Solver 149
  Antwortbericht 156
  Nebenbedingungen 150
  Optionen 154
Sonderformat 59
SPALTE() 684
SPALTEN() 686
Sparklines 131
Sparklinetools 133
SQL Server Business Intelligence Development Studio 720
SQRT() 388
SQRTPI() 389
STABW() 651
STABW.N() 577
STABW.S() 578
STABWA() 579
STABWN() 652
STABWNA() 581
Standardabweichung 449
STANDARDISIERUNG() 581
STANDARDIZE() 581
Standardtastaturlayout 39
STANDNORMINV() 652
STANDNORMVERT() 653
Startbildschirm 33, 35
Statistische Funktionen 443
Statistischer Test 456
STDEV() 651
STDEV.P() 577
STDEV.S() 578
STDEVA() 579
STDEVP() 652
STDEVPA() 581
STEIGUNG() 583
STEYX() 584
STFEHLERYX() 584
Stichprobe 445, 447
Strukturierte Verweise 102
Student-t-Verteilung 459
STUNDE() 293
Stundenermittlung 267
SUBSTITUTE() 772
SUBTOTAL() 383
Subtraktion 72
SUCHEN() 764
SUM() 372
SUMIF() 374
SUMIFS() 376
SUMME() 372
SUMMENPRODUKT() 374
Summensymbol 98
SUMMEWENN() 374
SUMMEWENNS() 376
SUMMEX2MY2() 377
SUMMEX2PY2() 378
SUMMEXMY2() 379
SUMPRODUCT() 374
SUMQUADABW() 585
SUMSQ() 361
SUMX2MY2() 377
SUMX2PY2() 378
SUMXMY2() 379
SVERWEIS() 663, 687
SYD() 194
Syntaxprüfung 119
Szenario 163, 166
Szenariobericht 168
Szenario-Manager 163, 166

T

T() 765
T.DIST() 592
T.DIST.2T() 594
T.DIST.RT() 595
T.INV() 587
T.INV.2S() 587
T.INV.2T() 587
T.TEST() 590
T.VERT() 592
T.VERT.2S() 594
T.VERT.RE() 595
TAG() 294
TAGE() 294
TAGE360() 295
TAN() 380
TANH() 382
TANHYP() 382
Taskleiste 34
Tausenderabtrennung 44

TBILLÄQUIV() 225
TBILLEQ() 225
TBILLKURS() 226
TBILLPRICE() 226
TBILLRENDITE() 227
TBILLYIELD() 227
TDIST() 657
Technische Funktionen 394
TEIL() 766
TEILERGEBNIS() 383
Teilergebnisse 838
Text, verketten 74
TEXT() 768
Textformat 58
Textfunktionen 745
logische Werte aufnehmen 747
Sortiermöglichkeiten 746
Verknüpfung Text/Datum 747
Zahlen in Text umwandeln 746
Tilgungsrechnung 171
TIME() 299
TIMEVALUE() 300
TINV() 654
TODAY() 279
Touchscreen 36
TRANSPOSE() 681
TREND() 595
TRIM() 754
TRIMMEAN() 508
TRUE() 794
TRUNC() 347
TTEST() 656, 868
TVERT() 657
TYP() 815
TYPE() 815

U

Überwachungsfenster 123
Uhrzeit als numerischer Wert 45
UMWANDELN() 436
UND() 793
UNGERADE() 384
Ungleichungen 150
UNICHAR() 770
UNICODE() 769
UNIZEICHEN() 770
UNREGER.KURS() 228
UNREGER.REND() 229
UNREGLE.KURS() 230
UNREGLE.REND() 232
UNTERGRENZE() 658
UNTERGRENZE.GENAU() 385
Unvollständige Jahreszahlen 64
UPPER() 755
URLCODIEREN() 822
Urliste 446
URL-Kodierung 823

V

VALUE() 774
VAR() 659
VAR.P() 598
VAR.S() 600
VARA() 601
VARIANZ() 659, 847
VARIANZA() 601
Varianzanalyse 845
VARIANZEN() 660
VARIANZENA() 602
VARIATION() 603
VARIATIONEN() 605
VARIATIONEN2() 606
VARP() 660
VARPA() 602
VBA 871
VBA-Projekt 872
VDB() 233
VERGLEICH() 688
VERKETTEN() 771
Verknüpfte Bereiche und Schnittmengen 75
Verpackungsproblem 150
VERWEIS() 690
Verweisfunktionen 663
Visual Basic for Applications 871
VLOOKUP() 687
VORZEICHEN() 386
VRUNDEN() 387

W

WAHL() 691
WAHR() 794
Wahrheitswerte 781
WAHRSCHBEREICH() 607

Wahrscheinlichkeit 447
empirische 448
theoretische 448
Was-wäre-wenn-Analyse 165
WEBDIENST() 825
Webfunktionen 821
WEBSERVICE() 825
WECHSELN() 772
WEEKDAY() 297
WEEKNUM() 283
WEIBULL() 661
WEIBULL.DIST() 609
WEIBULL.VERT() 609
Weibull-Verteilung 459
WENN() 795, 796
WENNNV() 797
WERT() 774
Wertpapierverkauf 183
While 883
WIEDERHOLEN() 775
Windows 8 33
Windows RT 35
WOCHE() 890
WOCHENTAG() 297
WORKDAY.INTL() 271
WORKDAY() 269
WURZEL() 388
WURZELPI() 389
WVERWEIS() 663, 694

X

XINTZINSFUSS() 234
XIRR() 234
XKAPITALWERT() 235
XMLFILTERN() 825
XNPV() 235
XODER() 797
XOR() 797
XPath 826

Y

YEAR() 281
YEARFRAC() 273
YIELD() 220
YIELDDISC() 221
YIELDMAT() 222

Z

Zahlenformat 42, 54, 57
Bruch 57
Buchhaltung 57
Prozent 57
Währung 57
Zahlensystem 396
ZÄHLENWENN() 610
ZÄHLENWENNS() 612
ZAHLENWERT() 776
ZEICHEN() 777
Zeichenfolgen 42
Zeichenfolgen-Formeln 69
Zeigen-Modus 81
ZEILE() 695
ZEILEN() 696
ZEIT() 299
ZEITWERT() 300
ZELLE() 817
Zielzelle 149
Zins 172
ZINS() 236
Zinseszins 174
Zinsrechnung 170
ZINSSATZ() 239
ZINSTERMNZ() 240
ZINSTERMTAGE() 241
ZINSTERMTAGNZ() 242
ZINSTERMTAGVA() 243
ZINSTERMVZ() 244
ZINSTERMZAHL() 245
ZINSZ() 246
Zirkelbezüge 126
Zirkuläre Formeln 124
ZSATZINVEST() 247
ZTEST() 631
ZUFALLSBEREICH() 390
ZUFALLSZAHL() 391
ZW() 248, 871
ZW2() 250
ZWEIFAKULTÄT() 392
ZZR() 251
ZZWert() 871