


Diese Leseprobe haben Sie beim
 edv buchversand.de heruntergeladen.
Das Buch können Sie online in unserem
Shop bestellen.
[Hier zum Shop](#)

Kapitel 8

Wichtige Kalkulationsfunktionen für Controller

Kalkulationsfunktionen bilden das Herzstück von Excel. Mit jeder neuen Version des Programms lässt sich das Entwicklungsteam in Redmond etwas Neues einfallen. Das ist auch in der aktuellen Version von Excel nicht anders. Aber, keine Sorge, es ist nicht notwendig, alle Kalkulationsfunktionen zu kennen. Mit denen, die Ihnen die folgenden Seiten vorstellen, sollten Sie im Controlling allerdings schon sehr weit kommen.

Auch in der aktuellen Excel-Version gibt es einige neue Kalkulationsfunktionen. In Controlling und Datenanalyse werden sich sicherlich die hier genannten als Schwergewichte erweisen:

- ▶ **XVERWEIS()** – diese Neuerung ist ein ernst zu nehmender Herausforderer für die wahrscheinlich im Controlling nach wie vor populärste Verweisfunktion, den **SVERWEIS()**. Der neue **XVERWEIS()** erweitert und vereinfacht das Zusammenführen von Daten aus verschiedenen Tabellen und entledigt sich aller technischen Limitationen des Vorgängers. Abschnitt 8.3 zu dieser neuen Funktion ist ein *Must-read!*
- ▶ **LET()** – Bereichsnamen besitzen das Potenzial, komplexe Kalkulationen übersichtlicher zu gestalten. Der Wechsel zwischen **NAMENS-MANAGER** und Tabellenblatt ist dennoch vielen Nutzern ein Buch mit sieben Siegeln. Mit **LET()** können Sie nun Namen für einzelne Kalkulationsschritte direkt in eine Formel schreiben. Welche Vorteile die neue Funktion perspektivisch sonst noch mit sich bringt, lesen Sie in Abschnitt 8.11.
- ▶ **LAMBDA()** – zum Zeitpunkt der Erstellung dieses Skripts, im April 2022, steckt diese neue Funktion noch in einer sehr frühen Entwicklungsphase. Doch sie gibt bereits zu erkennen, welches gewaltige Potenzial in ihr steckt. Power User werden in der Lage sein, mit ihr benutzerdefinierte Funktionen zu schreiben, selbst die Programmierung von Berechnungsschleifen sind mit **LAMBDA()** möglich. Nimmt man die Fähigkeiten von Power Query bei der Datenbereinigung und die von **LAMBDA()** bei der Definition benutzerdefinierter Berechnungen und Prozeduren im Tabellenblatt zusammen, bleibt für VBA-Anwendungen nur noch ein sehr schmaler Handlungspfad. Auch hier finden Sie eine Beschreibung des *State of the art* in Abschnitt 8.12 am Ende dieses Kapitels.



- Dynamische Matrixfunktionen waren bereits Gegenstand in der vorangegangenen Ausgabe dieses Buches. Da sie aber noch nicht lange den Eingang in die regulären Excel-Versionen gefunden haben, für Controller sehr viele Optionen bei der Optimierung von Berechnungen und Prozessen bieten und auf der gleichen Roadmap zur Vereinfachung der Analyse großer Datenmengen liegen, möchte ich diese Funktionsgruppe hier ebenfalls noch einmal nennen. Ihnen ist das folgende Kapitel gewidmet.

In diesem Kapitel möchte ich Ihnen allerdings neben diesen neuen Funktionen auch die etablierten vorstellen, die meiner Erfahrung nach das unverzichtbare Grundgerüst für Lösungen im Controlling darstellen. Ich werde Ihnen ihre Funktionsweise kurz beschreiben und ihre Verwendung an ebenso kurzen Beispielen veranschaulichen, bevor sich die folgenden Kapitel dann mit komplexeren Anwendungen aus der Praxis ausführlicher mit den Funktionen befassen.

Bei meinem Vorhaben orientiere ich mich nicht durchgängig an den Kategorien des Funktionsassistenten. Meine thematische Gliederung ist stattdessen folgende:

- Rechnen mit Datum und Zeit
- Verweise und Matrizen
- dynamischer Zugriff auf Tabellen
- Bildung und Berechnung von Rangfolgen
- Rundung und Mittelwerte
- logische Funktionen und Fehlerunterdrückung
- die alten Matrixfunktionen und die neuen Matrixfunktionen seit Excel 2019

8.1 Berechnungen mit Datumsbezug

Zeitliche Analysen von Daten gehören im Controlling zum Alltag. Der Funktionsassistent hält in der Kategorie DATUM & ZEIT einige Funktionen bereit, die in diesem Zusammenhang hilfreich sind. Grundsätzlich haben Sie auf unterschiedlichen Wegen Zugang zu den Datums- und allen anderen Funktionen. Da ist zunächst der Funktionsassistent, den Sie über die Schaltfläche FUNKTION EINFÜGEN direkt neben der Editierleiste oder mit  +  aktivieren. Im Menü FORMELN • FUNKTIONSBIBLIOTHEK finden Sie die nach Kategorien geordnete Übersicht der Funktionen. Die Kategorie DATUM U. UHRZEIT listet die Funktionen auf, um die es in diesem Abschnitt geht (Abbildung 8.1).

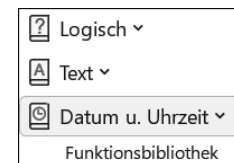


Abbildung 8.1 Kategorie »Datum u. Uhrzeit« im Menü »Formeln«

Datumsbereich

Bevor wir uns mit der Ermittlung von so speziellen Daten wie Nettoarbeitstagen beschäftigen, ist ein wenig Grundlagenarbeit zu leisten. In der Arbeitsmappe *O8_Datum_Grundlagen_01.xlsx* habe ich wesentliche Informationen zur Verwendung von Datumswerten in Excel zusammengetragen. Mit diesen Basisinformationen sollten wir uns zu Beginn auseinandersetzen.

Das Tabellenblatt *Datumsbereich* veranschaulicht Ihnen den Datumsbereich von Excel, der vom 01.01.1900 bis zum 31.12.9999 reicht. In den beiden Zellen A4 und A19 wird deutlich, was geschieht, wenn Sie einen Wert eingeben, der außerhalb dieses Bereichs liegt: Die Eingabe wird als Text interpretiert, was Sie an der linksbündigen Ausrichtung unschwer erkennen (Abbildung 8.2).

| Serieller Datumswert und Datumsformat | | | | |
|--|----------------|--|-----------|------------|
| Eingabe des Datums und Anzeige des seriellen Werts | | Eingabe des seriellen Werts und Formatierung als Datum | | |
| Datum | serieller Wert | serieller Wert | Datum | |
| 31-12-1899 | 31-12-1899 | -1 | ##### | |
| | 1900-01-01 | 1 | | 1900-01-01 |
| | 1913-09-08 | 5000 | 5.000 | 1913-09-08 |
| | 1927-05-18 | 10000 | 10.000 | 1927-05-18 |
| | 1941-01-24 | 15000 | 15.000 | 1941-01-24 |
| | 1954-10-03 | 20000 | 20.000 | 1954-10-03 |
| | 1968-06-11 | 25000 | 25.000 | 1968-06-11 |
| | 1982-02-18 | 30000 | 30.000 | 1982-02-18 |
| | 1995-10-28 | 35000 | 35.000 | 1995-10-28 |
| | 2009-07-06 | 40000 | 40.000 | 2009-07-06 |
| | 2036-11-21 | 50000 | 50.000 | 2036-11-21 |
| | 2105-05-04 | 75000 | 75.000 | 2105-05-04 |
| | 2173-10-14 | 100000 | 100.000 | 2173-10-14 |
| | 4637-11-26 | 1000000 | 1.000.000 | 4637-11-26 |
| | 9999-12-31 | 2958465 | 2.958.465 | 9999-12-31 |
| 01.01.10000 | 01.01.10000 | | 2.958.466 | ##### |

Abbildung 8.2 Datumsbereich in Excel

Wertemäßig entspricht das Datum 01.01.1900 der 1 und der 31.12.9999 der Zahl 2.958.465. Jedem Datumswert ist folglich ein Zahlenwert zugeordnet. Und diese Werte bilden die eigentliche Basis für sämtliche Berechnungen, die in Excel auf Grundlage des Datums möglich sind. Der Datumsbereich definiert aber auf besondere Art auch, was nicht möglich ist. Und das sind Kalkulationen mit negativen Datums- oder auch Zeitwerten.

Die Zeichenkette ##### in Zelle E4 resultiert nicht aus einer mangelnden Spaltenbreite im Tabellenblatt, sondern aus dem Versuch, den Wert -1 aus Zelle C4 über eine Datumsformatierung als Datum anzuzeigen. Um Probleme dieser Art bei Kalkulationen zu verhindern, verfügt Excel über eine Option, den Beginn des internen Kalenders vom Jahr 1900 auf das Jahr 1904 zu verschieben. Sie gewinnen dadurch quasi vier Jahre oder genau 1.463 Tage, um auch mit negativen Datums- und Zeitdifferenzen, etwa bei der Arbeitszeiterfassung, arbeiten zu können. Dieses Verfahren beschreibe ich in Abschnitt 8.2, »Berechnungen mit Zeitangaben«.

Datumsformate und ISO 8601:2000

Wenn Sie einen Datumswert aus dem gültigen Bereich vor sich haben, stehen Ihnen – und das wird im Tabellenblatt *Datumsformat* deutlich – zum Teil sehr unterschiedliche Datumsformate zur Verfügung (Abbildung 8.3). Wenn Sie mit einer Formatierung über **START • ZAHL • ZAHLENFORMAT • ZAHLEN** oder **[Strg] + [1]** in die Kategorie **DATUM** wechseln, stoßen Sie auf eine Liste, die Datumsformate nach dem **GEBIETSSCHEMA** von Afrikaans über Grönländisch und Maori bis Zulu anbietet. Was hingegen fehlt, ist das aus früheren Excel-Versionen bekannte Schema **INTERNATIONAL**, das das Datum nach ISO 8601:2000 formatierte. Dieser internationale Standard definiert die Schreibweise des Datums in der Form einer vierstelligen Jahresangabe, des zweistelligen Monats und der ebenfalls zweistelligen Tagesangabe, wobei alle Datumsteile durch einen Bindestrich getrennt werden: **JJJJ-MM-TT**.

| | A | B |
|---|-----------------------------------|------------------------|
| 1 | Datumsformate | |
| 2 | ingegebenes Ausgangsdatum: | 19.06.2016 |
| 3 | Ausgabeformat: | |
| 4 | serieller Wert | 42540 |
| 5 | ISO 8601:2000 | 2016-06-19 |
| 6 | Wochentag-Tag-Monat-Jahr | Sonntag, 19. Juni 2016 |
| 7 | Tag-Monat-Jahr | 19. Juni 2016 |
| 8 | benutzerdefiniert | 2016-Juni-Sonntag |

Abbildung 8.3 Formatierung eines Wertes mit unterschiedlichen Datumsformaten

Den Verlust der Kategorie **INTERNATIONAL** – so verwunderlich er angesichts der Tatsache ist, dass die ISO 8601:2000 EU-weiter Standard und auch in anderen Regionen der Welt weit verbreitet ist – können Sie auf zwei Arten kompensieren:

- ▶ Erstellen Sie ein benutzerdefiniertes Datumsformat mit dem Aufbau **JJJJ-MM-TT** in der Kategorie **BENUTZERDEFINIERT**.
- ▶ Wählen Sie ein anderes Gebietsschema, z. B. **AFRIKAANS**, in dem das Datum nach dem Schema **JJJJ-MM-TT** verwendet wird.

Als Nebeneffekt dieser nachträglichen Anpassung nehmen Sie aber immerhin mit, dass die Definition von Datumsteilen über die drei Buchstaben **J**, **M** und **T** erfolgt, wie es Tabelle 8.1 veranschaulicht.

| Platzhalter im Format | Datum |
|-----------------------|------------------------------|
| M | Monat (einstellig) |
| MM | Monat (zweistellig) |
| MMM | Monat (Wort, abgekürzt) |
| MMMM | Monat (Wort, ausgeschrieben) |

Tabelle 8.1 Optionen für die Erstellung benutzerdefinierter Datumsformate

| Platzhalter im Format | Datum |
|-----------------------|-------------------------------------|
| T | Tag (einstellig) |
| TT | Tag (zweistellig) |
| TTT | Tag (Wort, abgekürzt) |
| TTTT | Tag (Wort, ausgeschrieben) |
| J bis JJJJ | Jahresangabe (ein- bis vierstellig) |

Tabelle 8.1 Optionen für die Erstellung benutzerdefinierter Datumsformate (Forts.)

Datumsberechnungen

Die Kalkulationsmöglichkeiten auf Grundlage von Datumswerten sind vielfältig und reichen von einfacher Addition und Subtraktion, beispielsweise bei der Berechnung von Zahlungszielen, bis hin zu filigran ineinander verschachtelten Funktionen, etwa der Berechnung der Kalenderwoche nach ISO 8601:2000. Einige Kostproben liefert Ihnen das Tabellenblatt *Datumsberechnung* der Beispieldatei (Abbildung 8.4).

| | A | B | C | D |
|---|---|----|-------------------|----------------|
| 1 | Datumsberechnungen | | | |
| 2 | ingegebenes Ausgangsdatum: | | 2016-06-19 | Sonntag |
| 3 | einfache Addition von n Tagen | 15 | 2016-07-04 | Montag |
| 4 | einfache Subtraktion von n Tagen | 15 | 2016-06-04 | Samstag |
| 5 | ingegebenes Ausgangsdatum: | | 2016-06-19 | Sonntag |

Abbildung 8.4 Addition und Subtraktion von Werten zu bzw. von einem Datum

Zum Ausgangsdatum in Zelle C3 können Sie mit der Formel **=C2+B3** eine in Zelle B3 festgelegte Anzahl von Tagen hinzuzählen, und zwar so, wie Sie in C4 mit **=C2-B4** eine bestimmte Anzahl subtrahieren können. Kein Problem!

8.1.1 Dynamische Datumslisten ohne Wochenenden

Ein wenig komplizierter können jedoch auf Ebene der Datumsfunktionen selbst einfache Fragestellungen aussehen. Im Zellbereich C6 bis C15 wird dies an einer Liste berechneter Nachfolgetermine deutlich, bei der alle Tage, die auf ein Wochenende fallen, ausgespart werden sollen. In diesem Fall kommen wir schon nicht mehr ohne eine Verschachtelung mehrerer Funktionen aus.

Die Logik, die dieses Beispiel bestimmt, lautet: Zähle zum letzten genannten Datum drei Tage hinzu, wenn es auf einen Freitag fällt. Zwei Tage sind hinzuzuzählen, wenn das letzte Datum auf einen Samstag fällt; ansonsten ist immer nur ein Tag zum letzten Datumswert hinzuzuaddieren. Logisch! Ja, um die Liste ohne Wochenenden in Excel umzusetzen, greifen Sie deshalb auch auf eine Funktion aus der Kategorie **LOGIK** zurück (Abbildung 8.5):

=WENN(Prüfung; Dann-Wert; Sonst-Wert)

In Zelle C6 lautet die Funktion, bezogen auf das Ausgangsdatum in Zelle C5:

=WENN(WOCHENTAG(C5;2)=5;C5+3;WENN(WOCHENTAG(C5;2)=6;C5+2;C5+1))

| | | |
|--|---|-----------------|
| einggegebenes Ausgangsdatum: | 2016-06-19 | Sonntag |
| Folgetermine (ohne Wochenenden) | =WENN(WOCHENTAG(C5;2)=5;C5+3;WENN(WOCHENTAG(C5;2)=6;C5+2;C5+1)) | |
| | WENN(Wahrheitstest; [Wert_wenn_wahr]; [Wert_wenn_falsch]) | Dienstag |

Abbildung 8.5 Berechnung einer Datumsliste ohne Wochenendtermine

Die Prüfung bezieht sich hier auf den Wochentag in Zelle C5. Mit WOCHENTAG(C5;2)=5 finden Sie heraus, ob das Datum auf den fünften Tag der Woche fällt. Das Argument 2 sorgt dafür, dass der Wochenbeginn auf Montag gesetzt wird. Ist das Datum in Zelle C5 ein Freitag, gibt die Funktion WENN() ein WAHR zurück, und die DANN-Anweisung kann ausgeführt werden. Zum Freitag werden drei Tage hinzuaddiert, und die Liste wird somit mit dem Datum des folgenden Montags fortgesetzt.

Hinsichtlich der SONST-Anweisung verbleiben nun zwei Alternativen: Wenn das geprüfte Datum nicht auf einen Freitag fällt, könnte es sich entweder um einen Samstag oder um einen anderen Wochentag handeln. Dies muss herausgefunden werden, weil auch beim Samstag ein Tag, nämlich der nachfolgende Sonntag, in der Datumsliste übersprungen werden muss. Es bleibt Ihnen also nichts anderes übrig, als die SONST-Anweisung mit einem weiteren WENN() zu füllen:

WENN(WOCHENTAG(C5;2)=6;C5+2;C5+1)

8.1.2 Berechnung der Kalenderwoche nach ISO 8601:2000 und des Quartals

Die ISO-Norm 8601:2000 definiert nicht nur das Erscheinungsbild einer Datumsangabe. Grundlegende Aussagen trifft diese Norm auch zu der Frage, welche überhaupt die erste Woche des Jahres ist. Dabei gilt: Die Woche beginnt generell mit dem Montag, und die erste Kalenderwoche des Jahres enthält immer den Donnerstag der Woche. Mit anderen Worten: Beginnt das neue Jahr mit einem Freitag, wird die Woche dem vorangegangenen Jahr als KW 53 zugeschlagen.

Seit der Version 2000 verfügt Excel über die Funktion KALENDERWOCHE(Bezug, Typ), die erst ab Version 2010 das eigentlich nicht allzu komplizierte Regelwerk der ISO 8601:2000 beherrscht. Im ersten Argument müssen Sie das Datum angeben. Das zweite Argument sollte den Wert 21 enthalten. Dies entspricht den Vorgaben, dass die Woche mit dem Montag zu beginnen hat und mindestens vier Tage haben muss, um zum neuen Jahr zu zählen. Nehmen Sie hingegen Typ 2, fiel der zweite Teil der Regel weg, und die Berechnung wäre nicht ISO-konform.

Seit Excel 2013 gibt es zusätzlich die Funktion: ISOKALENDERWOCHE(). Sie erwartet nur noch die Angabe des Datums, dessen Kalenderwoche Sie berechnen möchten, und wendet automatisch das ISO-Regelwerk an (Abbildung 8.6).

| Kalenderwochen | | |
|----------------|-------------------------|--------------------------|
| Datum | ISOKALENDERWOCHE() | KALENDERWOCHE() - Typ 21 |
| 01.01.2009 | =ISOKALENDERWOCHE(G3) | 1 |
| 01.01.2010 | ISOKALENDERWOCHE(Datum) | 53 |
| 01.01.2011 | | 52 |
| 01.01.2012 | | 52 |
| 01.01.2013 | | 1 |
| 01.01.2014 | | 1 |
| 01.01.2015 | | 1 |
| 01.01.2016 | | 53 |
| 01.01.2017 | | 52 |
| 01.01.2018 | | 1 |
| 01.01.2019 | | 1 |
| 01.01.2020 | | 1 |

Abbildung 8.6 Zwei Funktionen ermöglichen seit Excel 2013 die ISO-konforme Berechnung der Kalenderwoche.

Arbeiten Sie mit einer älteren Excel-Version, wird es gleich etwas komplizierter. In Zelle C17 sehen Sie eine verschachtelte Funktion, die in allen Versionen das richtige Ergebnis ermittelt:

=KÜRZEN((C16-DATUM(JAHR(C16+3-REST(C16-2;7)));1;REST(C16-2;7)-9))/7)

Nicht ganz so kompliziert geht es zu, wenn Sie ein Quartal berechnen. In Zelle C18 bedarf es aber immer noch einiger Handarbeit, um eine Funktion nachzubilden, die es in Excel nicht gibt – die Berechnung des Quartals auf Basis eines gegebenen Datums. Die verschachtelte Funktion lautet hier:

=AUFRUNDEN(MONAT(C16)/3;0)&" . Quartal"

Fazit: Verhältnismäßig banale Tatbestände bei der Kalkulation von Datumswerten setzen in Excel ein gewisses Fingerspitzengefühl und eine gesunde kritische Grundhaltung gegenüber dem Funktionskatalog des Programms voraus.

8.1.3 Berechnung von Nettoarbeitstagen

Dass die Addition und Subtraktion einer Anzahl von Tagen zu bzw. von einem vorgegebenen Datumswert problemfrei funktioniert, haben Sie bereits erkennen können. Auch die Berechnung der Differenz zwischen zwei Datumswerten ist umstandslos möglich. Ziehen Sie z. B. vom 07.09.2022 den 01.01.2022 ab, erhalten Sie das Ergebnis 249. In den meisten Fällen wird Sie allerdings nicht die Anzahl der Kalendertage zwischen zwei Datumswerten interessieren, sondern die Anzahl der Arbeitstage.

| | A | B | C | D | E | F | G | H | I |
|---|-----------|----------|----------|----------------|-----------------------|--------------|------------------|-------------|--------------------|
| 1 | Pers. Nr. | Name | Vorname | Vertragsbeginn | Laufzeit (in Monaten) | Vertragsende | Nettoarbeitstage | verbleibend | Arbeitszeit (Std.) |
| 2 | 210-001 | Thewes | Paul | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 3 | 210-002 | Piel | Luis | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 4 | 210-003 | Lohmeyer | Herbert | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 5 | 210-004 | Umbert | Hanno | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 83 |
| 6 | 210-005 | da Silva | Everaldo | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 7 | 210-006 | Wolsch | Lydia | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 8 | 210-007 | Ballert | Susanne | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |

Abbildung 8.7 Berechnung der Nettoarbeitstage in einer Personalliste

Dazu steht Ihnen die Funktion NETTOARBEITSTAGE(Ausgangsdatum; Enddatum; Freie_Tage) zur Verfügung. In der Arbeitsmappe *08_Datum_Nettoarbeitstage_01.xlsx* wird die Anzahl der Arbeitstage berechnet, die zwischen einem Vertragsbeginn und -ende unter Berücksichtigung einer Liste von freien Tagen liegen (Abbildung 8.7).

| Arbeitsfreie Tage 2016 | Datum |
|------------------------|------------|
| Neujahr | 01.01.2016 |
| Hl. 3 Könige | 06.01.2016 |
| Karfreitag | 25.03.2016 |
| 1. Mai | 01.05.2016 |
| Himmelfahrt | 05.05.2016 |
| Pfingstmontag | 16.05.2016 |
| Fronleichnam | 26.05.2016 |
| Ostermontag | 28.03.2016 |
| Mariä Himmelfahrt | 15.08.2016 |
| Nationalfeiertag | 03.10.2016 |
| Reformationstag | 31.10.2016 |
| Allerheiligen | 01.11.2016 |
| 1. Weihnachtstag | 25.12.2016 |
| 2. Weihnachtstag | 26.12.2016 |

Abbildung 8.8 Liste berechneter Feiertage und sonstiger arbeitsfreier Tage

Dies bedeutet, dass Sie zunächst einmal in einem Tabellenblatt die Liste der arbeitsfreien Tage – Feiertage, Betriebsferien, Fortbildungstage etc. – erfassen müssen (Abbildung 8.8). Im Tabellenblatt *Arbeitsfreie Tage* ist dies bereits für einen Zeitraum von drei Jahren geschehen. Die Liste muss aus einem zusammenhängenden Zellbereich bestehen, der auch nicht durch etwaige Texte wie Überschriften für die einzelnen Jahre unterbrochen werden darf. In der Beispieldatei habe ich dem Zellbereich B5 bis B63 den Bereichsnamen *ArbeitsfreieTage* zugewiesen.

| | A | B | C | D | E | F | G | H | I |
|---|-----------|----------|----------|----------------|-----------------------|--------------|---|-------------|--------------------|
| 1 | Pers. Nr. | Name | Vorname | Vertragsbeginn | Laufzeit (in Monaten) | Vertragsende | Nettoarbeitstage | verbleibend | Arbeitszeit (Std.) |
| 2 | 210-001 | Thewes | Paul | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 3 | 210-002 | Piel | Luis | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 4 | 210-003 | Lohmeyer | Herbert | 01.01.2016 | 24 | 31.01.2018 | =NETTOARBEITSTAGE(D4;F4;ArbeitsfreieTage) | | |
| 5 | 210-004 | Umbert | Hanno | 01.01.2016 | 24 | 31.01.2018 | NETTOARBEITSTAGE(Ausgangsdatum; Enddatum; [Freie_Tage]) | 3 | |
| 6 | 210-005 | da Silva | Everaldo | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |
| 7 | 210-006 | Wolsch | Lydia | 01.01.2016 | 24 | 31.01.2018 | 518 | -1017 | 167 |

Abbildung 8.9 Verwendung der Funktion »NETTOARBEITSTAGE()«

In Zelle G2 können Sie nun die Anzahl der Arbeitstage ohne Wochenenden, Feiertage und sonstige arbeitsfreie Tage berechnen (Abbildung 8.9):

=NETTOARBEITSTAGE(D2;F2;ArbeitsfreieTage)

Die Funktion kopieren Sie dann nach unten, um auch für die anderen Mitarbeiter und Verträge die gewünschten Ergebnisse zu erhalten.

NETTOARBEITSTAGE() gibt es seit Excel 2010 in einer weiteren Version mit der Bezeichnung NETTOARBEITSTAGE.INTL(). Bei dieser internationalen Version der Funktion können Sie mit dem Argument *Wochenende* bestimmen, welche Tage der Woche innerhalb der Kalkulation als Wochenende gelten sollen.

8.1.4 Berechnung der verbleibenden Tage bis zum Monats- oder Projektende

Der Leitgedanke *From here to eternity!* zählt im Controlling bekanntlich verhältnismäßig wenig. Die Anzahl der verbleibenden Tage vom heutigen Datum bis zum Monatsende oder bis zum Ende eines definierten Projekts liegt schon eher im Erkenntnisinteresse des Controllers. Kein Wunder also, dass Excel für Letzteres auch einige Berechnungsfunktionen anbietet.

So können Sie sich die Eingabe des Vertragsendes in Spalte F sparen, indem Sie es mit der Funktion MONATSENDE(Ausgangsdatum; Monate) von Excel berechnen lassen und die Funktion dann wieder aus der Ausgangszelle F2 nach unten kopieren:

=MONATSENDE(D2;E2)

Als Ergebnis erhalten Sie immer den kalendarisch letzten Tag eines Monats, der um die angegebene Anzahl von Monaten hinter dem Ausgangsdatum liegt. Ups! Das ist nicht ganz richtig! Denn wenn das Argument *Monate* einen negativen Wert (z. B. -6) enthält, können Sie das Monatsende auch für vorangegangene Perioden berechnen.

=MONATSENDE(HEUTE();0) liefert Ihnen das Enddatum des aktuellen Monats. Die Funktion HEUTE() ist Ihr Garant für die Verwendung des aktuellen Tagesdatums im Tabellenblatt. Sie enthält keine weiteren Argumente. Doch da sich beide Funktionen wunderbar miteinander kombinieren lassen, errechnen Sie mit den hier vorgestellten Bausteinen auch die Anzahl der Nettoarbeitstage bis zum Ende des aktuellen Monats:

=NETTOARBEITSTAGE(HEUTE();MONATSENDE(HEUTE();0);ArbeitsfreieTage)

Bezogen auf die Projektdauer lautet die Funktion:

=NETTOARBEITSTAGE(HEUTE();F2;ArbeitsfreieTage)

Hier wird vorausgesetzt, dass in Zelle F2 das Datum des Projektendes eingegeben wurde.



»ARBEITSTAG.INTL()« und »NETTOARBEITSTAGE.INTL()«

In Excel 2010 wurden zwei neue Funktionen in der Kategorie DATUM & ZEIT etabliert. ARBEITSTAG.INTL(Ausgangsdatum; Tage; Wochenende; freie_Tage) enthält das zusätzliche Argument Wochenende. Über einen Code können Sie hier vorgeben, an welchen Tagen der Woche das reguläre Wochenende ist. Der Code 2 definiert das Wochenende beispielsweise auf Sonntag und Montag. Außerdem ist eine Wochenendzeichenfolge möglich, bei der 1 für einen arbeitsfreien Tag steht, 0 für einen Arbeitstag. Die Woche beginnt bei solchen Zeichenfolgen grundsätzlich mit einem Montag. Die Zeichenfolge 0011000 lieferte das Resultat, dass das Wochenende auf Mittwoch und Donnerstag fällt. Die Zeichenfolge 1111111 ist übrigens unzulässig. Schade!

In gleicher Weise können Sie seit Excel 2010 die neue Funktion NETTOARBEITSTAGE.INTL(Ausgangsdatum; Enddatum; Wochenende; freie_Tage) verwenden.

8.1.5 Feiertage berechnen

Für das vorangegangene Thema lässt sich festhalten, dass es manchmal selbstverständlich praktischer und schneller ist, einen Blick in den Kalender zu werfen und dort die Tage einfach abzuzählen, um sie anschließend in Excel einzugeben, als mit einer komplexen verschachtelten Funktion die Anzahl der Tage zwischen – sagen wir – dem 27.07. und 31.07. aufwendig zu berechnen. Umgekehrt gilt in gleichem Maße für die bereits dargestellten wie die nun folgenden Beispiele, dass Datumsberechnungen in manchen Tabellenblättern unabdingbar sind, um dynamische Auswertungen überhaupt erst zu ermöglichen. Es kommt also immer auf das Augenmaß und den konkreten Anwendungsbereich an.

Die Liste der arbeitsfreien Tage in der letzten Beispieldatei enthielt bereits Elemente zur Berechnung von beweglichen Feiertagen. In der Arbeitsmappe *08_Datum_Feiertage_01.xlsx* ist dieser Aufgabe ein größerer Raum gewidmet.

Im Mittelpunkt steht dabei immer der Ostersonntag, von dem aus die weiteren Feiertage bestimmt werden können, sofern das konkrete Jahr angegeben wird (Abbildung 8.10). Die Jahresangabe steht in der Beispieldatei in Zelle B1. In Zelle B2 ist somit die Berechnung des Ostersonntags mit der folgenden Funktion möglich:

```
=DM((TAG(MINUTE(B1/38)/2+55)&" .4."&B1)/7;)*7-6
```

Diese phänomenale Funktion stammt von Norbert Hetterich, der sie im Rahmen eines Internetwettbewerbs um die kürzeste Funktion zur Berechnung des Ostersonntags entwickelte ... und den Wettbewerb gewann. Kleiner Haken: Die Funktionsverkettung liefert nur das richtige Ergebnis, wenn die Datumswerte in den Excel-Optionen mit dem Jahr 1900 beginnen.

| | A | B |
|----|---------------------------------|--------------|
| 1 | Jahr: | 2016 |
| 2 | berechneter Ostermontag: | 28.03.2016 |
| 3 | | |
| 4 | Arbeitsfreie Tage 2016 | Datum |
| 5 | Neujahr | 01.01.2016 |
| 6 | Hl. 3 Könige | 06.01.2016 |
| 7 | Karfreitag | 25.03.2016 |
| 8 | 1. Mai | 01.05.2016 |
| 9 | Himmelfahrt | 05.05.2016 |
| 10 | Pfingstmontag | 16.05.2016 |
| 11 | Fronleichnam | 26.05.2016 |

Abbildung 8.10 Berechnung der beweglichen Feiertage

Setzen Sie in einer Arbeitsmappe hingegen 1904-DATUMSWERTE im Rahmen der Funktion DATEI • OPTIONEN • ERWEITERT ein, wird die Funktion zur Berechnung des Ostersonntags etwas länger:

```
=DATUM(B1;3;28)+REST(24-REST(B1;19)*10,63;29)-REST(KÜRZEN(B1*5/4)+  
REST(24-REST(B1;19)*10,63;29)+1;7)
```

Die vom Ostersonntag abhängigen beweglichen Feiertage erhalten Sie, indem Sie sich auf das berechnete Datum beziehen und die entsprechende Tagesanzahl hinzuzählen. Beispiel: Den Pfingstmontag ermitteln Sie in Zelle B10 durch die Formel =B2+50. Die restlichen Feiertage ergeben sich aus der Anwendung der Funktion DATUM(Jahr; Monat; Tag). Dies lässt sich am Beispiel des ersten Weihnachtsfeiertages in Zelle B17 gut nachvollziehen: =DATUM(B1;12;25).

8.1.6 Dynamischer Kalender für alle Bundesländer

Der notwendige nächste Schritt bei der Dynamisierung von Datumsberechnungen liegt in der Einbeziehung regionaler Unterschiede. Da sich die Feiertagsregelungen in den Bundesländern erheblich unterscheiden, kann es nicht nur eine Liste von arbeitsfreien Tagen geben. Zu den mindestens 16 Listen der Bundesländer treten nochmals drei weitere hinzu, da für die Bundesländer Bayern, Saarland und Thüringen zusätzliche Feiertage in Gemeinden mit überwiegend katholischer Bevölkerung üblich sind.

In der Arbeitsmappe *08_Datum_Kalender_01.xlsx* enthält das Tabellenblatt *Berechneter Kalender mit KW* einen Jahreskalender, in dem Sie durch die Auswahl des Jahres und des Bundeslandes in den Zellen B1 und D1 die Anzeige der betreffenden Feiertage im Kalender steuern können (Abbildung 8.11). Beide Listen basieren auf der Funktion DATENÜBERPRÜFUNG im Menü DATEN • DATENTOOLS und greifen auf jeweils einen Bereichsnamen zu.

Sie erreichen dies mit einer logischen Funktion. Für einen nicht beweglichen Feiertag wie Neujahr gelingt die Anzeige beispielsweise in Zelle W6 mit:

```
=WENN(V6="x";DATUM(JAHR(V2);1;1);DATUM(1900;1;1))
```

Bei beweglichen Feiertagen wie dem Pfingstmontag verwenden Sie in Zelle W11 stattdessen:

```
=WENN(V11="x";V2+50;DATUM(1900;1;1))
```

Formatierung des Kalenders

Um die Wochenenden, die Feiertage und das aktuelle Datum im Kalender zu kennzeichnen, verwenden Sie am besten die *bedingte Formatierung*. Die einzusetzenden Funktionen sehen Sie in Tabelle 8.2.

| Funktion | Formatierung |
|--|-----------------------------------|
| =B3=HEUTE() | Zeigt das aktuelle Datum rot an. |
| =WOCHENTAG(B3;2)=6 | Markiert die Samstage hellgrau. |
| =WOCHENTAG(B3;2)=7 | Markiert die Sonntage dunkelgrau. |
| =SVERWEIS(B3;C.ber.feiertage;1;FALSCH) | Zeigt Feiertage dunkelrot an. |

Tabelle 8.2 Funktionen der bedingten Formatierung

Wenn ein Feiertag auf ein Wochenende fällt, ist die Reihenfolge der Regeln für die bedingte Formatierung dafür ausschlaggebend, ob der Tag im Kalender grau oder dunkelrot gekennzeichnet wird. Setzen Sie die *BEDINGTE FORMATIERUNG* mit der Funktion *SVERWEIS()* an die Spitze der Regelliste, wenn Sie die Feiertage auch an den Wochenenden gekennzeichnet sehen möchten (Abbildung 8.14).

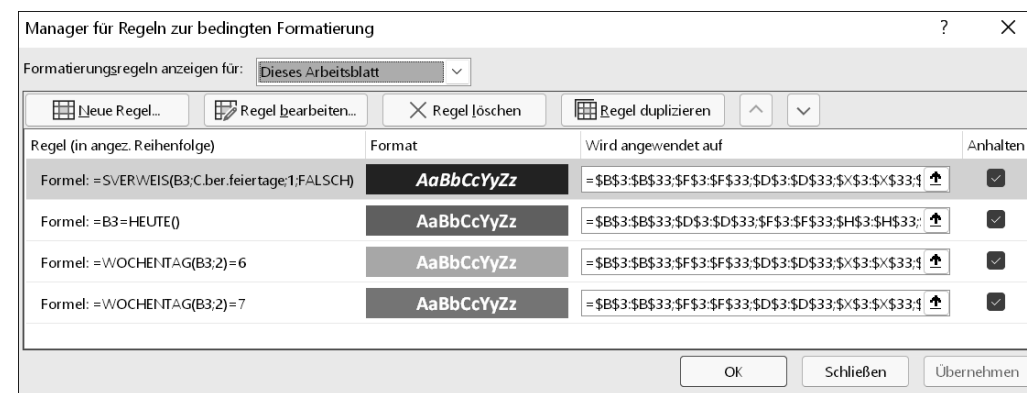


Abbildung 8.14 Prioritätensetzung der Formatierungsregeln für den Kalender

8.1.7 Berechnung des Enddatums für Vorgänge

Lassen Sie uns nach diesem notwendigen Exkurs in das Gebiet der Feiertagsberechnung in Excel zu unserem ursprünglichen Thema, der Berechnung von Zeitintervallen, zurückkehren. Dort ist es uns eben gelungen, aus zwei Datumsvorgaben die Anzahl der Nettoarbeitstage zwischen diesen Eckwerten zu ermitteln. Von einer vergleichbaren Überlegung werden Sie geleitet, wenn Sie das Enddatum eines Vorgangs berechnen möchten, dessen Startdatum und Dauer Sie kennen.

Auch in einem solchen Fall wird Sie nicht die Einbeziehung der Wochenenden und arbeitsfreien Tage in die Berechnung von *aktuelles Datum + x* interessieren. Sie benötigen, wie schon bei der Kalkulation der Nettoarbeitstage, eine spezielle Funktion und eine Liste der arbeitsfreien Tage. In der Arbeitsmappe *08_Datum_Arbeitstag_01.xlsx* finden Sie beides.

Die Liste der arbeitsfreien Zeiten befindet sich in dieser Beispieldatei im Tabellenblatt *Arbeitsfreie Tage*. Es wird erneut der Bereichsname *ArbeitsfreieTage* für den notwendigerweise zusammenhängenden Zellbereich verwendet. Das Startdatum für die Berechnung wurde in Zelle G1 des Tabellenblattes *Bühnenaufbau* eingegeben.

In D1 wird das Abschlussdatum für die eintägige Tätigkeit mit der Funktion *=ARBEITSTAG(G1;C2-1;ArbeitsfreieTage)* ermittelt. Wenn Sie diese Funktion in Zelle D2 in *=ARBEITSTAG(D2;C3;ArbeitsfreieTage)* abwandeln und nach unten kopieren, schließt jeder nachfolgende Vorgang nahtlos an den bereits abgeschlossenen Vorgänger an. Doch, wie gut zu erkennen ist, liegen zwischen den Vorgängen *Beladung Transporter* und *Anfahrt* nicht weniger als zwei Tage, da sie durch das Wochenende unterbrochen werden (Abbildung 8.15).

| | A | B | C | D | E | F | G |
|---|------------|-----------------------|--------------|-------------------------|---|--------------------|------------|
| 1 | Nr. | Arbeitsschritt | Dauer | Abgeschlossen am | | Startdatum: | 17.06.2016 |
| 2 | 1 | Beladung Transporter | 1 | 17.06.2016 | | | |
| 3 | 2 | Anfahrt | 1 | 20.06.2016 | | | |
| 4 | 3 | Bühnenbau | 5 | 27.06.2016 | | | |
| 5 | 4 | Licht & Dekoration | 3 | 30.06.2016 | | | |
| 6 | 5 | Funktionstest | 1 | 01.07.2016 | | | |

Abbildung 8.15 Berechnung des Enddatums

8.1.8 Berechnung von Datumsdifferenzen mit »DATEDIF()«

Eine Übersicht über wichtige Datumsfunktionen kann nicht ohne einen echten Exoten unter den Excel-Funktionen abgeschlossen werden. Die Funktion *DATEDIF(Startdatum; Enddatum; Zeiteinheit)* fristet ein Schattendasein, da sie weder im Funktionsassistenten noch in der Hilfe von Excel aufgeführt wird. Aus Kompatibilitätsgründen mit *Lotus 1-2-3* vor langer Zeit in Excel integriert, leistet sie verlässliche Dienste bei der Berechnung unterschiedlicher Datumsdifferenzen, wenn man von ihrer Existenz weiß. Denn da die Funktion nicht aufgelistet wird, kann sie ausschließlich per Tastatur in die Zellen des Tabellenblattes eingegeben werden.

Die bereits in einem vorherigen Beispiel verwendete Vertragsübersicht ist in der Arbeitsmappe *08_Datum_DATEDIF_01.xlsx* Grundlage für die Berechnung einer Datumsdifferenz. Lassen Sie uns annehmen, Sie möchten die Anzahl der Jahre, die ein Vertrag läuft, berechnen, weil davon bestimmte Zuschlagszahlungen an Angestellte abhängen.

In Zelle G2 verwenden Sie in diesem Fall die Funktion `=DATEDIF(D2;E2;"Y")` und kopieren sie wie gewohnt nach unten. In Zelle F2 erreichen Sie die Ausgabe der Monatsanzahl zwischen Vertragsbeginn und -ende mit `=DATEDIF(D2;E2;"M")`. Sie erkennen unschwer, dass dem Argument Zeiteinheit die Rolle eines Schalters bei der Auswahl der Ergebnisanzeige zukommt (Abbildung 8.16). Die verfügbaren Optionen für dieses Argument sehen Sie in Tabelle 8.3.

| | A | B | C | D | E | F |
|---|------------------|-------------|----------------|-----------------------|---------------------|------------------------------|
| 1 | Pers. Nr. | Name | Vorname | Vertragsbeginn | Vertragsende | Laufzeit (in Monaten) |
| 2 | 210-001 | Thewes | Paul | 01.01.2016 | 31.01.2018 | =DATEDIF(D2;E2;"M") |

Abbildung 8.16 Anzahl der Monate zwischen Vertragsbeginn und -ende, berechnet mit »DATEDIF()«

| Option | Berechnung |
|--------|---|
| "D" | Anzahl der Tage zwischen zwei Datumswerten |
| "M" | Anzahl der Monate zwischen zwei Datumswerten |
| "Y" | Anzahl der Jahre zwischen zwei Datumswerten |
| "MD" | Ignoriert Monate und Jahre und bildet die Differenz der Tage zwischen Anfang- und Enddatum. |
| "YM" | Berechnet die Differenz der Monate; Tage und Jahre werden ignoriert. |
| "YD" | Differenz der Tage wird berechnet, Jahre aber werden ignoriert. |

Tabelle 8.3 Optionen der Funktion »DATEDIF()«

8.1.9 Weitere nützliche Funktionen in der Kategorie »Datum & Zeit«

Im Controlling spielen zeitliche Betrachtungen und Analysen stets eine bedeutende Rolle. Auch die in Tabelle 8.4 dargestellten Datumsfunktionen können dabei äußerst nützlich sein.

| Funktion | Beschreibung |
|----------|---|
| HEUTE() | Gibt das veränderliche Tagesdatum auf Basis der Systemzeit des Computers aus. |
| EDATUM() | Addiert zu einem Ausgangsdatum die im zweiten Argument Monate angegebene Anzahl an Monaten. |

Tabelle 8.4 Nützliche Datumsfunktionen in der Übersicht

| Funktion | Beschreibung |
|----------------------------|--|
| JAHR() MONAT() TAG() | Die drei Funktionen dienen dazu, aus einem vorgegebenen Datumswert Teile wie Jahr, Monat oder Tag zu isolieren. Die Ergebnisse werden häufig z. B. beim Sortieren, Filtern oder bei der Bildung von Teilergebnissen weiterverwendet. |
| BRTEILJAHRE() | Auf der Basis eines Start- und Enddatums berechnet Excel die Differenz in ganzen Tagen. Das Ergebnis wird in Bruchteile von Jahren umgewandelt, wobei Sie im Argument Basis zwischen verschiedenen Systemen unterscheiden können, wie z. B. USA (NASD) oder tagesgenauer Abrechnung für die Berechnung der Zinstage. Die Funktion dient der Verbesserung der Vergleichbarkeit von Forderungen und Verbindlichkeiten. |

Tabelle 8.4 Nützliche Datumsfunktionen in der Übersicht (Forts.)

8.2 Berechnungen mit Zeitangaben

Den Einstieg in den Themenbereich der Berechnungen auf Grundlage von Zeitangaben möchte ich analog zu den Datumsberechnungen beginnen. Auch bei den Zeitwerten in Excel bilden der Wertebereich, die Formatierung und die Berechnung einen Dreiklang, mit dem sich – in kalkulatorischer Hinsicht – eine Menge zum Klingen bringen lässt.

In der Arbeitsmappe *08_Zeit_Grundlagen_01.xlsx* sehen Sie ein in Einzelheiten vertrautes Bild. Einer in Spalte A eingegebenen formatierten Uhrzeit entspricht in Spalte B ein numerischer Wert. Der Uhrzeitbereich reicht von 0, also 00:00 Uhr, bis 1, dem Dezimalwert für 24:00 Uhr. Rutschen Sie bei der Eingabe oder Berechnung von Uhrzeiten in einen negativen Wertebereich – in Zelle A3 ist dies durch den Wert `-0,125` geschehen –, erhalten Sie die Fehleranzeige `#####` (Abbildung 8.17).

| | A | B |
|----|-------------------|----------------|
| 1 | Grundlagen | |
| 2 | Zeit | Dezimal |
| 3 | ##### | -0,125 |
| 4 | 00:00 | 0 |
| 5 | 03:00 | 0,125 |
| 6 | 06:00 | 0,25 |
| 7 | 09:00 | 0,375 |
| 8 | 12:00 | 0,5 |
| 9 | 15:00 | 0,625 |
| 10 | 18:00 | 0,75 |
| 11 | 21:00 | 0,875 |
| 12 | 23:59 | 0,999306 |

Abbildung 8.17 Zeitbereich in Excel



Umgang mit negativen Zeitangaben

Das Problem der negativen Uhrzeiten entsteht häufig bei der Erfassung und Berechnung von Arbeitszeiten. Nehmen Sie an, ein Mitarbeiter hat von 8 Uhr 30 bis 15 Uhr gearbeitet, dann entspricht das 6,5 Stunden, die in Excel in der Form 06:30 angezeigt werden.

Beträgt die Soll-Arbeitszeit hingegen 7 Stunden, ergibt sich bei der Subtraktion ein negativer Wert von -0,5, der als Uhrzeit in Excel allerdings nicht darstellbar ist.

Um das Problem zu lösen, wechseln Sie in die OPTIONEN von Excel und rufen dort das Register ERWEITERT auf. In der Rubrik BEIM BERECHNEN DIESER ARBEITSMAPPE aktivieren Sie die Option 1904-DATUMSWERTE VERWENDEN. Dadurch gewinnen Sie einen Puffer von vier Jahren, der für das skizzierte Problem bei der Arbeitszeiterfassung ausreicht.

8.2.1 Formatierung von Uhrzeiten

Die Formatierung der Zeitwerte erfolgt in Excel nach dem Schema *hh:mm:ss*. Die unterschiedlichen Formate übernehmen Sie mit **[Strg] + [1]** aus der Kategorie UHRZEIT der Dialogbox ZELLEN FORMATIEREN. Dies klingt alles wenig aufregend. Eine kleine Tücke bieten die Uhrzeitformate aber doch: Sie offenbart sich, wenn Sie Zeiten addieren möchten und das Ergebnis dabei die 24-Stunden-Marke überschreitet. Excel beginnt in diesem Fall wieder bei 0, was Sie im Tabellenblatt *Zeit – Format* in der Zelle D8 sehr gut erkennen (Abbildung 8.18).

Erst die Umstellung des Zeitformats in Zelle E8 von *hh:mm* auf *[hh]:mm* führt zum korrekten Ergebnis. Die eckigen Klammern sind also mehr als reiner Schmuck – sie befähigen Excel quasi, sich das Ergebnis des Vortages zu merken und darauf aufbauend weiterzurechnen.

| | A | B | C | D | E |
|---|----------------------|---------------|-------------|---------------------------------|-----------------------------------|
| 1 | Arbeitszeiten | | | | |
| 2 | Tag | Beginn | Ende | Dauer (Format hh:mm) | Dauer (Format [hh]:mm) |
| 3 | Montag | 06:30 | 15:45 | 09:15 | 09:15 |
| 4 | Dienstag | 07:15 | 16:35 | 09:20 | 09:20 |
| 5 | Mittwoch | 08:30 | 16:00 | 07:30 | 07:30 |
| 6 | Donnerstag | 08:20 | 16:20 | 08:00 | 08:00 |
| 7 | Freitag | 08:30 | 15:25 | 06:55 | 06:55 |
| 8 | | | | 17:00 | 41:00 |

Abbildung 8.18 Addition von Zeitangaben

8.2.2 Umrechnung von Dezimal- in Industriezeit

Für das Umrechnen von Dezimalzeit in Industriezeit gibt es in Excel keine eingebaute Funktion. Sie müssen zurück zu den Wurzeln und sich vor Augen führen, dass die Werte, die Ihnen in dezimaler Form vorliegen, durch die Dauer eines Tages, sprich 24 Stunden, geteilt werden müssen und dass eine Stunde aus 60 Minuten besteht.

In Zelle B3 des Tabellenblattes *Dezimal- und Industriezeit* der Arbeitsmappe *08_Zeit_Industriezeit_01.xlsx* wird mit der einfachen Formel $=A3/24/60$ gearbeitet, um den Wert 68,1 in die Industriezeit 1:08:06 – also in 1 Stunde, 8 Minuten und 6 Sekunden – zu konvertieren (Abbildung 8.19). Wenn Sie die Formel nach unten kopieren, werden auch alle anderen Werte aus Spalte A entsprechend umgerechnet und dargestellt, vorausgesetzt, in Spalte B wurde mit *[h]:mm:ss* auch das gewünschte Uhrzeitformat aktiviert.

| | A | B | C | D | E |
|---|-----------------------------------|------------------|---|------------------|-----------------|
| 1 | Dezimal- und Industriezeit | | | | |
| 2 | dezimal (Minuten) | Industrie | | Industrie | dezimal |
| 3 | 68,1 | 1:08:06 | | 0:45:30 | 0,7583333 |
| 4 | 57,3 | 0:57:18 | | 2:05:45 | 2,0958333 |
| 5 | 1250 | 20:50:00 | | 16:40:00 | 16,666667 |
| 6 | 1400 | 23:20:00 | | 25:00:00 | 25 |
| 7 | 1700 | 28:20:00 | | 33:20:00 | 33,333333 |
| 8 | | 74:35:24 | | | 77,85417 |

Abbildung 8.19 Umrechnung von Dezimal- in Industriezeit und umgekehrt

Möchten Sie hingegen von einer Uhrzeit im Industriezeitformat in eine dezimale Darstellung umrechnen, wie es in Spalte E der Fall ist, dann reicht es aus, den Wert in Spalte D mit 24 zu multiplizieren, um das korrekte Ergebnis zu erhalten. Auch hier müssen Sie gegebenenfalls das Zahlenformat auf STANDARD stellen.

8.2.3 Berechnung von Arbeitszeiten bei Schichtbetrieb

Eine letzte mögliche Hürde bei der Anwendung von Kalkulationen im Bereich der Zeiterfassung und -auswertung ist die Problematik von Arbeitsbeginn und -ende bei Schichtbetrieb. Wie errechnet man die Anzahl der geleisteten Arbeitsstunden, wenn ein Mitarbeiter um 19:30 Uhr mit seiner Arbeit begonnen und diese um 04:09 Uhr beendet hat?

Die einfache Subtraktion würde hier erneut zu einem negativen Ergebniswert führen, der zu allem Überfluss falsch wäre, wenn Sie die 1904-Datumswerte verwenden. Die Lösung ist in diesem Beispiel die Verwendung der Funktion $=REST(C3-B3;1)$ in Zelle D3 (Abbildung 8.20).

| | A | B | C | D |
|---|----------------------|---------------|-------------|---------------|
| 1 | Schichtzeiten | | | |
| 2 | Tag | Beginn | Ende | Gesamt |
| 3 | Montag | 19:30 | 04:09 | 08:39 |
| 4 | Dienstag | 20:10 | 04:00 | 07:50 |
| 5 | Mittwoch | 20:15 | 04:07 | 07:52 |
| 6 | Donnerstag | 20:00 | 04:12 | 08:12 |
| 7 | Freitag | 20:05 | 03:50 | 07:45 |
| 8 | | | | 40:18 |

Abbildung 8.20 Berechnung von Arbeitszeiten bei Schichtdienst

Sie subtrahiert den Wert aus Zelle B3 von C3 und teilt das Ergebnis durch den Divisor 1, was einer Umwandlung des negativen in einen positiven Wert gleichkommt, bevor der Wert überhaupt in die Zelle geschrieben wird.

Die Formel kopieren Sie wie gewohnt nach unten. Dann setzen Sie für den Zellbereich D2 bis D7 das Uhrzeitformat *hh:mm* und für die Ergebniszeile D8 auf *[hh]:mm*, um alle Berechnungen korrekt abzuschließen.

8.3 Arbeiten mit Verweisen und Matrizen

Das Arbeiten mit Verweisen auf Tabellen ist in Excel äußerst populär. Nur zu oft werden nach dem Import von Daten für die Weiterverarbeitung nötige Werte aus Referenztabellen den Basisdaten über Verweise hinzugefügt. Auch bei durchgestalteten Tabellen, seien es Liquiditätspläne oder Produktkalkulationen, kommen Verweisfunktionen oft zum Einsatz. Gäbe es eine Top 10 der am häufigsten eingesetzten Funktionen in Excel, würde der `SVERWEIS()` mit großer Wahrscheinlichkeit einen der vorderen Ränge belegen.

Würden wir hingegen nur einige Seiten zurückblättern und zum Arbeitsbeispiel des dynamischen Kalenders zurückkehren, ist beinahe anzunehmen, dass die Schwester dieses Prominenten, der `WVERWEIS()`, schon wesentlich weniger Bekanntheit besitzt. Da Verweisfunktionen aber eine wichtige Rolle bei der Zusammenführung und Umgestaltung von bereits vorhandenen Daten spielen, sollten wir dieser Kategorie eine angemessene Aufmerksamkeit widmen. Für die aktuelle Excel-Version müssen Sie dabei eine wichtige Änderung berücksichtigen. `SVERWEIS()` und `WVERWEIS()` wiesen über Jahre mehrere Schwächen auf. Ihre Performance ließ bei der Bearbeitung größerer Datenmengen sehr zu wünschen übrig. Beide Funktionen waren außerdem stark limitiert, beispielsweise hinsichtlich der Suchrichtung. Die Performancenachteile wurden zwischenzeitlich beseitigt. Viel wichtiger ist allerdings, dass es mit `XVERWEIS()` nun eine dritte Funktion gibt, die die beiden anderen zusammenfasst und in ihren Möglichkeiten weit hinter sich lässt.

Doch damit nicht genug. Die konventionellen Verweisfunktionen haben mit der seit Excel 2013 verfügbaren Funktion der *Datenmodelle* und mit *Power Pivot* starke Konkurrenten bekommen. Über die Datenmodelle ist Excel erstmalig in der Lage, Pivottabellen auf Basis mehrerer Tabellen zu erstellen. Verweisfunktionen adieu! Mithilfe von *Power Pivot* lassen sich ebenso über logische Beziehungen Tabellen unterschiedlicher Provenienz zusammenführen und mit leistungsstarken DAX-Funktionen berechnen. Adios Verweisfunktionen! Beide Alternativen bestechen vor allem durch ihre Performancevorteile bei größeren Datenmengen. Streng genommen lassen sich drei Nutzungsszenarien unterscheiden, bei denen Daten aus verschiedenen Tabellen zusammengeführt werden müssen:

Im ersten Szenario müssen große Datenmengen in verschiedenen Tabellen verarbeitet werden, und *Power Pivot* ist in Excel verfügbar. Aufgrund der besseren Performance fahren

Anwender mit den logischen Beziehungen eines Datenmodells in solchen Fällen deutlich besser als mit Verweisfunktionen.

- ▶ Im zweiten Szenario steht *Power Pivot* nicht zur Verfügung, obwohl größere Datenmengen zusammengeführt werden müssen und/oder spezielle Anforderungen an die Verweisfunktion gestellt werden. Solche Anforderungen ergeben sich daraus, dass sich die Ergebnisspalte links von der Suchspalte befindet oder dass die Daten von unten nach oben durchsucht werden müssen. In diesem Fall ist der neue `XVERWEIS()` wegen seiner enormen Flexibilität das Mittel der Wahl.
- ▶ Zuletzt bleiben die Fälle übrig, bei denen kleinere Datenmengen von mehreren hundert Zeilen durchsucht werden sollen und eine Anordnung der Such- und Ergebnisspalte von links nach rechts sowie die Suchrichtung von oben nach unten gegeben ist. In diesem Szenario spricht nichts gegen die weitere Verwendung der Funktion `SVERWEIS()`.

8.3.1 Die Vorteile von »XVERWEIS()« gegenüber anderen Verweisfunktionen

Die Neuerungen bei der Funktion `XVERWEIS()` sind so umfangreich, dass es sich lohnt, zunächst ein paar Highlights dem nachfolgenden Arbeitsbeispiel voranzustellen:

- ▶ Die genaue Übereinstimmung zur Ermittlung eines Ergebniswertes ist bei `XVERWEIS()` die Standardeinstellung. Nur wenn auch der nächstgrößere oder nächstkleinere Wert zu einem Rückgabewert führen sollen, muss dies ausdrücklich angegeben werden. Da in den meisten Fällen mit genauer Übereinstimmung gearbeitet wird, fällt somit die Eingabe eines Funktionsarguments häufig weg. Die neue Funktion beherrscht zudem gleich drei verschiedene Vergleichsmodi. Daten werden bei genauer Übereinstimmung gefunden. Aber auch das gesuchte oder das nächstkleinere Element können gefunden werden. Die Liste der Möglichkeiten wird durch genaue Übereinstimmung oder das nächstgrößere Element vervollständigt.
- ▶ Die neue Verweisfunktion kann endlich Ergebniswerte liefern, die sich sowohl rechts als auch links von der Suchspalte befinden. Damit muss nicht mehr auf die Kombination von `INDEX()` und `VERGLEICH()` ausgewichen oder die Tabellenstruktur angepasst werden, wenn die Ergebniswerte links von der Suchspalte gespeichert sind.
- ▶ Letzteres ist dadurch möglich, dass die Ergebnisspalte nicht mehr mit einem numerischen Wert, dem sogenannten Spaltenindex, benannt wird, sondern stattdessen ein Zellbereich angegeben werden muss. Als Folge davon müssen Such- und Ergebnisbereich auch nicht mehr zwangsläufig in einem zusammenhängenden Zellbereich angeordnet sein. Zellbereiche können zudem horizontal definiert werden. Dadurch ersetzt `XVERWEIS()` auch die Funktion `WVERWEIS()`.
- ▶ Da das Argument `Wenn_nicht_gefunden` in `XVERWEIS()` integriert ist, fällt das Einbetten einer Verweisfunktion in eine logische Funktion wie `WENNFEHLER()`, `WENN()` oder `WENNS()` weg. Wird kein Ergebnis für das festgelegte Suchkriterium gefunden, kann ein angegebener alternativer Wert zurückgegeben werden.

- ▶ Die Suchrichtung kann in XVERWEIS() definiert werden. Es ist somit möglich, Daten wahlweise von oben nach unten oder in umgekehrter Richtung zu durchsuchen und die entsprechenden Ergebnisse auszulesen.
- ▶ Eine weitere Option betrifft die Verwendung von Wildcards, die aktivierbar oder deaktivierbar ist. Dadurch kann festgelegt werden, ob die Zeichen * oder ? als Teil einer Zeichenkette oder als Platzhalter für ein einzelnes oder mehrere Zeichen verstanden werden sollen.

Wow! Das ist eine Menge an Neuerungen! Wir müssen uns diese nun im Detail ansehen!

8.3.2 »SVERWEIS()« durch »XVERWEIS()« ersetzen

Öffnen Sie die Arbeitsmappe *O8_Verweise_XVERWEIS_Grundlagen.xlsx*, um sich die grundlegende Bedienung der neuen Verweisfunktion anzusehen. Im Tabellenblatt XVERWEIS() finden Sie ein erstes Beispiel, das zugegebenermaßen auch mit SVERWEIS() umsetzbar wäre (Abbildung 8.21). Eine Datentabelle im Zellbereich B3 bis D9 dient als Matrix. Spalte B wird durchsucht. Die potenziellen Ergebnisse befinden sich rechts davon in Spalte D. Das Suchkriterium ist in Zelle F4 zu finden.

| | A | B | C | D | E | F |
|---|---|---------|------------|-------|---|---------|
| 1 | | | | | | |
| 2 | | ACT | | | | |
| 3 | | Product | Date | Value | | Product |
| 4 | | ABC123 | 01.01.2016 | 100 | | DEF234 |
| 5 | | ABC123 | 01.02.2016 | 120 | | |
| 6 | | DEF234 | 01.01.2016 | 90 | | Value |
| 7 | | DEF234 | 01.02.2016 | 110 | | 90 |
| 8 | | ABC123 | 01.03.2016 | 135 | | |
| 9 | | DEF234 | 01.03.2016 | 125 | | |

Abbildung 8.21 Das angezeigte Ergebnis wäre auch mit »SVERWEIS()« realisierbar gewesen, ...

In Zelle F7 wird das Ergebnis mithilfe von

=XVERWEIS(F4;ACT_8[Product];ACT_8[Value])

zurückgegeben. Die Berechnung verwendet drei Argumente: das Suchkriterium aus Zelle F4, eine Suchmatrix in Form eines strukturierten Bezugs der Datentabelle (ACT_8[Product]) und eine Rückgabematrix, ebenfalls in Form eines strukturierten Bezugs (ACT_8[Value]). Klarer Fall, die beiden letzten Argumente könnten auch als konventionelle Zellbezüge (B3:B9 und D3:D9) verwendet werden. Wie arbeitet XVERWEIS() in diesem Beispiel? Die Funktion durchläuft die erste Spalte, wird beim Suchkriterium DEF234 fündig und liefert als Resultat aus der dritten Spalte der Tabelle den Ergebniswert 90.

Von den insgesamt sechs verfügbaren Argumenten von XVERWEIS() werden in diesem konkreten Fall lediglich drei verwendet, wie in Abbildung 8.22 erkennbar ist.

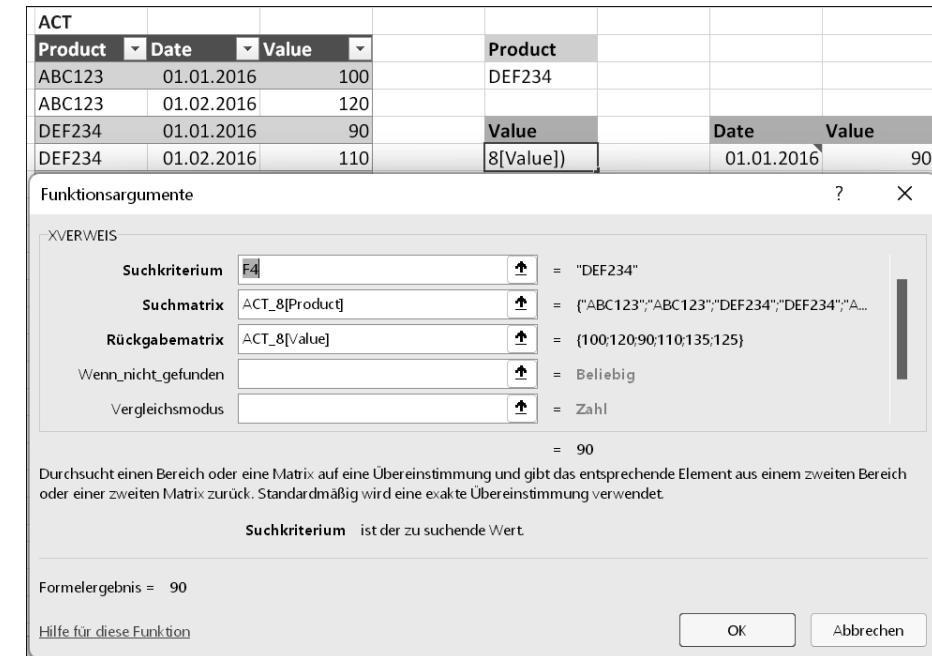


Abbildung 8.22 ... es wurde allerdings mit der neuen Funktion »XVERWEIS()« ermittelt.

Nichtsdestotrotz sollten Sie sich bereits an dieser Stelle einen Überblick über alle Argumente verschaffen (Tabelle 8.5).

| Funktionsargument | Beschreibung |
|-------------------|---|
| Suchkriterium | Hier wird der Wert angegeben, nach dem gesucht werden soll. Dies kann ein Text, eine Zahl oder eine Kombination daraus sein. Ideal ist es, das Suchkriterium aus einer Zelle zu übernehmen, auf die dann in diesem Funktionsargument verwiesen wird. Das Suchkriterium kann auch die Platzhalter ? oder * enthalten. Im vorletzten Argument Suchmodus wird festgelegt, wie mit Platzhaltern bei der Suche verfahren wird. |
| Suchmatrix | In diesem Argument wird angegeben, welcher Bereich des Tabellenblattes anhand des Suchkriteriums durchsucht werden soll. Die Suchmatrix kann mit normalen Zellbezügen, aber auch mit strukturierten Bezügen einer Datentabelle angegeben werden. Strukturierte Bezüge bieten hier den Vorteil, dass sich der Suchbereich automatisch erweitert, sobald die Datentabelle vergrößert wird. Die Suchmatrix kann sowohl eine vertikale als auch eine horizontale Zellanordnung besitzen. |

Tabelle 8.5 Funktionsargumente von »XVERWEIS()«

| Funktionsargument | Beschreibung |
|---------------------|---|
| Rückgabematrix | Dies ist der Zellbereich, aus dem die Ergebniswerte übernommen werden, sofern die Suchbedingungen erfüllt sind. Die für die Suchmatrix genannten Regeln und Überlegungen gelten auch für die Rückgabematrix. |
| Wenn_nicht_gefunden | Das Argument ermöglicht die Ausgabe eines Textes, einer Zahl oder einer Berechnung, für den Fall, dass das Suchkriterium nicht gefunden wurde. |
| Vergleichsmodus | <p>Dieses Argument kann über vier numerische Eingaben konfiguriert werden:</p> <ul style="list-style-type: none"> ▶ 0 gibt ausdrücklich an, dass eine exakte Übereinstimmung mit der Rückgabe eines Ergebniswertes vorausgesetzt wird. Dies ist gleichzeitig die Standardeinstellung von XVERWEIS(), die auch dann ausgeführt wird, wenn das Argument leer gelassen wird. ▶ -1 legt fest, dass eine exakte Übereinstimmung mit dem Suchkriterium vorliegen muss oder das nächstkleinere Element für die Ermittlung des Ergebnisses herangezogen werden soll. Wird der Suchwert 100 beispielsweise nicht gefunden, kann das Ergebnis auf Basis des Wertes 99 zurückgegeben werden. Neu ist auch, dass dazu die Suchtabelle weder auf- noch absteigend sortiert sein muss. ▶ Die 1 als Eingabe folgt der gleichen Logik. Allerdings wird durch sie eine exakte Übereinstimmung auf Basis des Suchkriteriums erzwungen oder der nächstgrößere Wert als Ergebnis geliefert. Liefert das Suchkriterium 100 keinen Treffer, kann stattdessen 101 das Ergebnis liefern. ▶ Durch Eingabe von 2 wird die sogenannte Platzhalterzeichenübereinstimmung aktiviert. Enthält das Suchkriterium das Zeichen ?, so wird dieses als Platzhalter für ein beliebiges anderes Zeichen interpretiert (sch?n kann also für schon oder schön stehen). Ein * ersetzt eine Zeichenkette (sch*n steht dann unter anderem für schein oder schulen). |

Tabelle 8.5 Funktionsargumente von »XVERWEIS()« (Forts.)

| Funktionsargument | Beschreibung |
|-------------------|---|
| Suchmodus | Argument Nummer sechs erlaubt die Festlegung der Suchrichtung, wobei 1 als Standardvorgabe für die Suche von oben nach unten steht. -1 führt dazu, dass die Suchmatrix von unten nach oben durchsucht wird. Die beiden weiteren Optionen 2 Binärsuche (sortiert in aufsteigender Reihenfolge) und -2 Binärsuche (sortiert in absteigender Reihenfolge) verwenden die althergebrachten Suchmodi von SVERWEIS(). Hierbei müssen die zu durchsuchenden Daten allerdings auf- oder absteigend sortiert sein. Tests haben ergeben, dass diese Einstellung zu keinen Performanceverbesserungen der Verweisfunktion führt. Sollte die Tabelle allerdings nicht sortiert sein, führt die Nutzung dieser beiden Optionen zu fehlerhaften Ergebnissen. Es empfiehlt sich also, die beiden Optionen schlichtweg zu ignorieren. |

Tabelle 8.5 Funktionsargumente von »XVERWEIS()« (Forts.)

Doch zurück zur Beispieldatei. Abbildung 8.23 zeigt die Verwendung der Option Suchmodus:

=XVERWEIS(F14;ACT_17[Product];ACT_17[Value];;-1)

Der Parameter -1 erzwingt die Umkehrung der Suchrichtung von unten nach oben. Folgerichtig wird bei der Suche nach dem Wert 1237 die erste Fundstelle von unten ermittelt, was zur Angabe des Wertes 135 (und nicht 110) führt.

| ACT_1 | | | | |
|---------|------------|-------|--|---------|
| Product | Date | Value | | Product |
| 1234 | 01.01.2016 | 100 | | 1237 |
| 1234 | 01.02.2016 | 120 | | |
| 1236 | 01.01.2016 | 90 | | Value |
| 1237 | 01.02.2016 | 110 | | 135 |
| 1237 | 01.03.2016 | 135 | | |
| 1238 | 01.03.2016 | 125 | | |

Abbildung 8.23 Ergebnis der Suche mit »XVERWEIS()« von unten nach oben

Die Datentabelle ACT_1 bildet auch die Grundlage für eine weitere Anpassung der Suchoptionen. In Abbildung 8.24 wird nach dem Wert 1235 gesucht, der allerdings in der Spalte PRODUCT nicht vorhanden ist. Wird kein Parameter im Funktionsargument Vergleichsmodus gesetzt, was dem Parameter 0 entspricht, führt die Berechnung zu einem Fehlerwert #NV. Dies lässt sich jedoch durch den Parameter 1 leicht ändern. =XVERWEIS(F14;ACT_17[Product];ACT_17[Value];;1;-1) gibt in Zelle G17 den Wert 90 als Ergebnis an. Dieser ist in der Matrix dem nächsthöheren Element (1236) zugeordnet.

| ACT_1 | | | | | |
|---------|------------|-------|---------|-------|-------|
| Product | Date | Value | Product | | |
| 1234 | 01.01.2016 | 100 | 1235 | | |
| 1234 | 01.02.2016 | 120 | | | |
| 1236 | 01.01.2016 | 90 | Value | Value | Value |
| 1237 | 01.02.2016 | 110 | #NV | 90 | 120 |
| 1237 | 01.03.2016 | 135 | | | |
| 1238 | 01.03.2016 | 125 | | | |

Abbildung 8.24 Suchergebnisse mit fehlender exakter Übereinstimmung

Mit dem Parameter -1 springt der Zeiger für das Ergebnis hingegen zur Zeile mit dem Wert 1234. Ergebnis: 120. Warum? Weil in Zelle H17 die Suchrichtung im letzten Argument mit -1 von unten nach oben definiert wurde:

```
=XVERWEIS(F14;ACT_17[Product];ACT_17[Value];;-1;-1)
```

Setzen Sie hingegen diesen Parameter auf 1, erhalten Sie den Wert 100 vom 01.01.2016 als Resultat, da von oben nach unten gesucht wird.

| ACT_3 | | | | | |
|----------|------------|-------|-------|--|--|
| Name | Date | Value | Name | | |
| Muster | 01.01.2016 | 100 | T?st | | |
| Beispiel | 01.02.2016 | 120 | | | |
| Tast | 01.01.2016 | 90 | Value | | |
| Test | 01.02.2016 | 110 | 90 | | |
| Mister | 01.03.2016 | 135 | | | |
| Beispiel | 01.03.2016 | 125 | | | |

Abbildung 8.25 Suche in »XVERWEIS()« mit Platzhalterzeichen

Eine weitere Variante zur Verwendung der Suchkriterien zeigt schließlich Abbildung 8.25. In Zelle F24 wird ein Suchkriterium mit dem Platzhalter ? verwendet. Die Suche nach der Zeichenkette T?st verlief im Sande. Dies ändert sich, wenn mit dem Parameter 2 im letzten Funktionsargument Suchmodus die Platzhalterübereinstimmung aktiviert wird:

```
=XVERWEIS(F24;ACT_310[Name];ACT_310[Value];;2)
```

Nun werden die beiden Namen *Tast* und *Test* als mögliche Treffer berücksichtigt. Da der Standardwert für die Suchrichtung in diesem Beispiel aktiviert ist, wird der erste Wert (*Tast*) als Fundstelle und 90 als Ergebnis ermittelt. Eine Umkehrung der Suchrichtung (-1) führte unweigerlich zu einem Treffer beim Namen *Test* und der Rückgabe des Wertes 110 als Ergebnis.

8.3.3 »WVERWEIS()« und »INDEX()/VERGLEICH()« durch »XVERWEIS()« ersetzen

Mit den bis hierhin erworbenen Kenntnissen ist die Geschichte der Funktion WVERWEIS() schnell erzählt. Da die beiden ersten Argumente von XVERWEIS() nicht nur vertikal, sondern

eben auch horizontal ausgerichtet sein können, lassen sich Tabellen auch zeilenweise durchsuchen (Abbildung 8.26).

| Date | 01.01.2016 | 01.02.2016 | 01.01.20162 | 01.02.20163 | 01.03.2016 | 01.03.20164 | Product | |
|---------|------------|------------|-------------|-------------|------------|-------------|--|--|
| Value | 100 | 120 | 90 | 110 | 135 | 125 | DEF345 | |
| Product | ABC123 | ABC123 | DEF234 | DEF234 | ABC123 | DEF345 | | |
| | | | | | | | Value | |
| | | | | | | | =XVERWEIS(AA4;T5:Y5;T4:Y4) | |
| | | | | | | | XVERWEIS(Suchkriterium; Suchmatrix; Rückgabematrix; [wenn_nicht_gefunden]) | |

Abbildung 8.26 Verweis auf eine horizontal orientierte Tabellenstruktur

Der zu findende Wert in Zelle AA4 ist DEF345. Gesucht wird er im Zellbereich T5 bis Y5. Das Resultat soll dem Bereich T4 bis Y4 entnommen werden. Ergebniswert: 125. Alle weiter oben beschriebenen Optionen lassen sich auch auf die horizontale Struktur der Matrix anwenden: Änderung der Genauigkeit bei der Übereinstimmung mit dem Suchwert, Anpassung der Suchrichtung (von rechts nach links anstelle der Standardrichtung von links nach rechts), alles kein Problem. XVERWEIS() erstellt auch bei alphanumerischen Inhalten einer Suchmatrix, wie sie im Beispiel mit Codierungen wie ABC123 oder DEF234 vorliegen, zunächst im Hintergrund eine sortierte alphanumerische Liste. Darauf aufbauend können nächstgrößere oder nächstkleinere Elemente recht mühelos aus beiden Suchrichtungen identifiziert werden.

Doch wie sieht es mit dem »SVERWEIS() nach links« aus? Vielleicht ist Ihnen die Problemstellung bekannt, und Sie erkennen sie in Abbildung 8.27 auf Anhieb wieder. Durchsucht werden soll die Spalte PRODUCT (strukturierter Bezug ACT412[Product]) mittels Suchkriterium DEF345, das sich in Zelle Q4 befindet. Allerdings sind die gewünschten Rückgabewerte in der Spalte VALUE links davon gespeichert (strukturierter Bezug ACT_412[Value]). In der guten alten Vor-XVERWEIS()-Zeit musste nun eine verschachtelte Funktion ran. Kurz gesagt: Es wurde mithilfe von VERGLEICH() die Zeile der Fundstelle in PRODUCT ermittelt (rechte Spalte der Matrix) und dann der zugehörige Wert aus VALUE (linke Spalte der Matrix) mit INDEX() ausgelesen. Verschachtelte Funktionen? Das lässt nicht nur so manchem Excel-Nutzer im Allgemeinen eiskalte Schauer über den Rücken laufen. Die beiden Möglichkeiten, mit SVERWEIS() oder mit INDEX()/VERGLEICH() Daten zuzuweisen, spaltete gar die gesamte Excel-Community in zwei miteinander fremdelnde Lager. Während das eine Lager SVERWEIS() für seine Einfachheit liebte, begeisterte sich das andere an der tollen Flexibilität von INDEX() und Co.

| ACT_4 | | | | | |
|------------|-------|---------|---|--|--|
| Date | Value | Product | Product | | |
| 01.01.2016 | 100 | ABC123 | DEF345 | | |
| 01.02.2016 | 120 | ABC123 | | | |
| 01.01.2016 | 90 | DEF234 | Value | | |
| 01.02.2016 | 110 | DEF345 | =XVERWEIS(Q4;ACT_412[Product];ACT_412[Value]) | | |
| 01.03.2016 | 135 | ABC123 | XVERWEIS(Suchkriterium; Suchmatrix; Rückgabematrix; | | |
| 01.03.2016 | 125 | DEF234 | | | |

Abbildung 8.27 Den »SVERWEIS()« nach links kann »XVERWEIS()« auch.

Und nun? Come together as one and have fun! Denn die linke-Spalte-rechte-Spalte-Thematik kann mit folgender Funktion gelöst werden:

```
=XVERWEIS(Q4;ACT_412[Product];ACT_412[Value])
```

Da es keinen zusammenhängenden Matrixbereich mehr gibt, können Such- und Rückgabematrix an beliebigen Stellen des Tabellenblattes angeordnet sein. Die Rückgabematrix kann sich folglich – wie im Beispiel gezeigt – links von der Suchmatrix in der gleichen Datentabelle befinden. Das Ergebnis *110* ist in diesem Fall korrekt. Ein Umbau der Tabelle ist durch die Anwendung von XVERWEIS() nicht mehr nötig.

8.3.4 »XVERWEIS()« im Kontext einer Kalkulationsfunktion

Die bislang beschriebenen Nutzungsmöglichkeiten entsprechen den gängigsten Szenarien der Anwendung von XVERWEIS(). Doch dahinter verbergen sich noch einige weitere sehr nützliche Funktionalitäten. An erster Stelle wäre die besondere Fähigkeit der Verweisfunktion zu nennen, nicht einen Ergebniswert, sondern einen Zellbezug zurückzugeben, wenn XVERWEIS() in eine andere Kalkulationsfunktion eingebettet wird.

| ACT_5 | | | Startdatum | Enddatum |
|------------|-------|---------|------------|------------|
| Date | Value | Product | | |
| 01.01.2016 | 100 | 1234 | 01.02.2016 | 01.03.2016 |
| 01.02.2016 | 120 | 1234 | | |
| 01.02.2016 | 90 | 1236 | Value | |
| 01.03.2016 | 110 | 1237 | | 320 |
| 01.04.2016 | 135 | 1237 | | |
| 01.06.2016 | 125 | 1238 | | |

Abbildung 8.28 Hier ermittelt »XVERWEIS()« die Summe aller Werte eines Datumsbereichs.

Diese Chamäleon-Fähigkeit besitzen einige Kalkulationsfunktionen in Excel, z. B. INDEX() und WAHL(). Nützlich ist diese verborgene Fähigkeit in den meisten Fällen, weil sie es erlaubt, dynamische Datenbereiche zu definieren, ohne den Nachteil der Volatilität von BEREICH.VERSCHIEBEN() in Kauf zu nehmen. Volatilität ist in umfangreichen Arbeitsmappen ein Performancekiller. Wie können Sie sich in XVERWEIS() also auf einen dynamischen Zellbereich beziehen?

Nehmen wir an, in Zelle Q17 wäre lediglich die folgende Funktion eingegeben worden:

```
=XVERWEIS(R14;ACT_513[Date];ACT_513[Value])
```

Der Rückgabewerte lautete dann *120*. Denn dies ist der Zelleninhalt in der Spalte VALUE für den ersten Wert des Suchkriteriums *01.02.2016* aus Zelle R14. Mit Bezug auf das Suchkriterium *01.03.2016* in Zelle S14 könnte folgende Funktion verwendet werden:

```
XVERWEIS(S14;ACT_513[Date];ACT_513[Value])
```

Daraus ergäbe sich der Ergebniswert *110*, also erneut ein Zelleninhalt.

Packen wir diese beiden Verweisfunktionen jedoch in eine Zusammenfassungsfunktion wie SUMME(), dann verhält sich XVERWEIS() anders:

```
=SUMME(
XVERWEIS(R14;ACT_513[Date];ACT_513[Value])
:
XVERWEIS(S14;ACT_513[Date];ACT_513[Value]))
```

Anstelle der Zelleninhalte *120* und *110* gibt die Funktion dann die Zelladressen *\$N\$15* und *\$N\$17* zurück. Werden diese beiden Zelladressen an SUMME() übergeben, wird die Summenfunktion das Ergebnis des Zellbereichs ermitteln. XVERWEIS() wird somit nicht direkt in einer Zelle, sondern im Kontext einer anderen Kalkulationsfunktion verwendet. In Zelle R17 steht folglich die Zahl *320*, das Ergebnis aus den drei Einzelwerten dieses Zellbereichs (Abbildung 8.28). Ändern Sie die beiden Datumswerte im Tabellenblatt, erhalten Sie ein aktualisiertes Ergebnis. Kleine Fußnote: Würden Sie die beiden Datumswerte über ein Listenfeld, etwa eine DATENPRÜFUNG, oder eine Berechnung erstellen, hätten Sie sogar eine vollständig dynamische Summenberechnung. Cool!

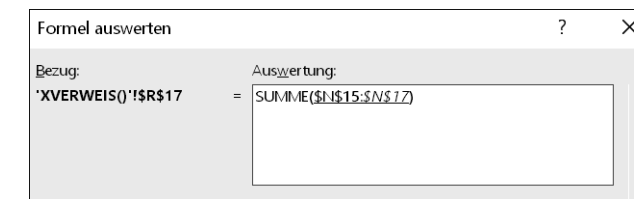


Abbildung 8.29 Die Formelauswertung veranschaulicht die Umwandlung von Zelleninhalten in Zellbezüge.

Wenn Sie SUMME() und XVERWEIS() in Zelle R17 ein wenig bei der Zusammenarbeit zuschauen möchten, starten Sie einfach FORMELN • FORMELÜBERWACHUNG • FORMELAUSWERTUNG. Danach klicken Sie sich über die Schaltfläche AUSWERTEN durch die einzelnen Berechnungsschritte. Die Umwandlung der Zelleninhalte in Zelladressen ist in dieser Ansicht sehr gut nachvollziehbar (Abbildung 8.29).

8.3.5 Überlaufbereiche der Ergebnisspalten bei der Verwendung von »XVERWEIS()«

Apropos Dynamik! Im Jahr 2019 wurde die Funktionsbibliothek von Excel um die Gruppe der dynamischen Matrixfunktionen erweitert, Funktionen, die so neuartig und leistungsstark sind, dass ihnen in diesem Buch ein eigenes Kapitel gewidmet wird (Kapitel 9, »Neue dynamische Matrixfunktionen in Excel für Microsoft 365«). Ohne an dieser Stelle die Katze vollständig aus dem Sack zu lassen: Eine Stärke der Funktionen besteht darin, Berechnungsbereiche automatisch dem zugrunde liegenden Rohdatenbestand anzupassen. Dieses für

Excel neuartige Verhalten wird auch als *Überlaufen* bezeichnet. Exakt diese Fähigkeit besitzt auch der XVERWEIS().

| ACT | | | | | |
|---------|------------|-------|---------|------------|-------|
| Product | Date | Value | Product | | |
| ABC123 | 01.01.2016 | 100 | DEF234 | | |
| ABC123 | 01.02.2016 | 120 | | | |
| DEF234 | 01.01.2016 | 90 | Value | Date | Value |
| DEF234 | 01.02.2016 | 110 | 90 | 01.01.2016 | 90 |
| ABC123 | 01.03.2016 | 135 | | | |
| DEF234 | 01.03.2016 | 125 | | | |


Abbildung 8.30 Das Ergebnis für »Value« wurde durch Überlauf aus der angrenzenden Zelle »Date« ermittelt.

In Abbildung 8.30 ist dies bereits an der dünnen blauen Umrahmung der Zelle I7 erkennbar. Ein solcher Rahmen macht kenntlich, dass der betreffende Zellbereich nicht durch manuelle Formeleingabe oder einen händischen Kopiervorgang gefüllt wurde. Stattdessen können Sie den Ergebnisbereich folgendermaßen erstellen:

1. In die beiden Zellen H6 und I6 schreiben Sie die Überschriften »Date« und »Value«.
2. Anschließend geben Sie in Zelle H7 die Kalkulationsfunktion ein:

```
=XVERWEIS(F4;ACT_8[Product];ACT_8[[Date]:[Value]])
```

Dabei ist auffällig, dass die Rückgabematrix in diesem Fall mehrspaltig ist ([Date]:[Value]).

3. Betätigen Sie nun .

Die Funktion XVERWEIS() liefert nun ein mehrspaltiges Ergebnis und kopiert die Funktion automatisch in Zelle I7. Stellen Sie sich vor, die Rohdatentabelle enthielte die Umsätze mehrerer Jahre, Quartale oder verschiedener Produktgruppen, dann würde das soeben getestete Vorgehen in einem Arbeitsgang sämtliche benötigten Ergebnisse liefern. Das leidige Kopieren der Verweisfunktion entfällt somit.



Besonderheiten bei der Verwendung von »XVERWEIS()« als Überlauffunktion

Bei XVERWEIS() sind die gängigen Besonderheiten dynamischer Matrixfunktionen zu beobachten:

- ▶ Die Kalkulationsfunktion wird nur in die erste Zelle der Ergebnismatrix geschrieben. Im Beispiel ist dies H7.
- ▶ Nur in dieser ursprünglichen Eingabezelle kann die Funktion überarbeitet werden.
- ▶ Alle anderen Zellen des Überlaufbereichs (z. B. I7) enthalten eine ausgegraute und nicht bearbeitbare Kopie der Funktion.
- ▶ Um den gesamten Ergebnisbereich zu löschen, muss die ursprüngliche Funktion in Zelle H7 entfernt werden.

8.3.6 Summe der Suchwerte einer Referenztable direkt bilden

Eine gängige Technik bei der Anwendung des erweiterten Filters in Excel besteht in der Möglichkeit, ein berechnetes Feld als Filterkriterium zu verwenden. Dies hat den Vorteil, dass man sich Hilfsspalten sparen und direkt auf das gefilterte Ergebnis zugreifen kann. Eine weitere Neuerung, die sich aus der Verwendung von XVERWEIS() ergibt, erinnert mich an genau diesen Vorteil. In Abbildung 8.31 sehen Sie ein Beispiel dazu. Die Datei dazu finden Sie unter dem Namen *O8_Verweise_XVERWEIS()_weitere_Funktionen.xlsx*.

| | A | B | C | D | E | F |
|----|-------|-------------|------------------|---|-------------|------------|
| 1 | KdNr | Umsatz | Kontrollrechnung | | Umsatz | Gutschrift |
| 2 | KD001 | 12.000,00 € | 120,00 € | | 0,00 € | 0,00 € |
| 3 | KD002 | 24.000,00 € | 240,00 € | | 5.000,00 € | 50,00 € |
| 4 | KD003 | 6.000,00 € | 50,00 € | | 10.000,00 € | 120,00 € |
| 5 | KD004 | 9.500,00 € | 50,00 € | | 15.000,00 € | 190,00 € |
| 6 | KD005 | 18.200,00 € | 190,00 € | | 20.000,00 € | 240,00 € |
| 7 | KD006 | 22.000,00 € | 240,00 € | | | |
| 8 | KD007 | 2.000,00 € | 0,00 € | | | |
| 9 | KD008 | 1.900,00 € | 0,00 € | | | |
| 10 | KD009 | 5.000,00 € | 50,00 € | | | |
| 11 | KD010 | 12.300,00 € | 120,00 € | | | |
| 12 | | | 1.060,00 € | | 1.060,00 € | |

Abbildung 8.31 Die Berechnung der Summe aller Gutschriften auf Basis von Umsätzen ohne Hilfsspalte macht »XVERWEIS()« möglich.

In den Spalten A und B sind Kundennummern und die dazugehörigen Umsätze gespeichert. Abhängig von vorgegebenen Umsatzklassen sollen nun Gutschriften zugewiesen und die Summe der Gutschriften berechnet werden. Dies klingt zunächst einmal nach einer Hilfsspalte, wie sie in Spalte C auch ausgegraut als Kontrollrechnung angezeigt wird. Doch diese zusätzliche Spalte ist nicht notwendig, wenn man folgende Kalkulationsfunktion in Zelle E12 verwendet:

```
=SUMME(XVERWEIS(B2:B11;E2:E6;F2:F6;;-1))
```

Der Unterschied zu allen anderen bislang dargestellten Beispielen besteht darin, dass das Suchkriterium hier keine einzelne Zelle ist, sondern ein Zeilenbereich zum Einsatz kommt (B2:B11). XVERGLEICH() vergleicht nun jeden der zehn Werte aus Spalte B mit den Angaben aus Spalte E. In dieser Spalte sind die Untergrenzen für die geplanten Gutschriften definiert. Im Zellbereich F2:F6 befinden sich hingegen die zu zahlenden Beträge. Dass eine Gutschrift nur erfolgen soll, wenn der Umsatz die Untergrenze erreicht hat, wird über den Vergleichsmodus -1 festgelegt. Ist der Umsatz niedriger, kommt die vorherige Gutschriftklasse zur Anwendung.

XVERWEIS() verhält sich hier ähnlich einer klassischen Matrixfunktion. Es erzeugt eine Liste an Kriterien, vergleicht diese zeilenweise mit einer Suchmatrix, merkt sich, ob das Kriterium in der betreffenden Zeile erfüllt ist oder nicht und ist anschließend in der Lage, mit diesen

Zwischenergebnissen weiterzuarbeiten. Im konkreten Fall werden die Ergebnisse nach dem Muster WAHR oder FALSCH an die Funktion SUMME() übergeben. Bei einem WAHR wird die Summe gebildet, bei einem FALSCH nicht.

8.3.7 Zweidimensionale Verweise bei der Nutzung von »XVERWEIS()«

Die spezifische Fähigkeit, während der Berechnung Tabellenauszüge zu bilden und weiterzuverwenden, ist auch im letzten Beispiel zum Thema XVERWEIS() erkennbar. In ihm geht es darum, zwei Suchkriterien zu verwenden, von denen sich das erste auf Zeilen und das zweite auf Spalten einer Tabelle bezieht.

| | A | B | C | D | E | F | G | H |
|----|-------|---------|---------|---------|---------|---|---------|-------|
| 1 | KdNr | Nord | Süd | Ost | West | | | |
| 2 | KD001 | 7.436 € | 7.159 € | 5.680 € | 2.727 € | | Kunde: | KD003 |
| 3 | KD002 | 4.233 € | 7.577 € | 6.659 € | 7.063 € | | Region: | Süd |
| 4 | KD003 | 5.250 € | 5.660 € | 2.293 € | 5.116 € | | | |
| 5 | KD004 | 5.272 € | 3.874 € | 5.699 € | 6.193 € | | | |
| 6 | KD005 | 5.983 € | 4.149 € | 4.648 € | 4.263 € | | | |
| 7 | KD006 | 5.563 € | 2.225 € | 5.751 € | 6.769 € | | | |
| 8 | KD007 | 3.302 € | 5.042 € | 5.045 € | 2.290 € | | | |
| 9 | KD008 | 3.472 € | 6.951 € | 5.793 € | 4.904 € | | | |
| 10 | KD009 | 7.840 € | 7.221 € | 3.922 € | 4.705 € | | | |
| 11 | KD010 | 7.961 € | 3.739 € | 5.998 € | 7.628 € | | 5.660 € | |

Abbildung 8.32 Auslesen eines Wertes auf Basis von zeilen- und spaltenbezogenen Kriterien

Um auf einen Wert der Gesamttabelle zuzugreifen, müssen normalerweise das Suchkriterium, die zu durchsuchende Spalte und die Spalte für den Rückgabewert statisch definiert werden. Um die Daten der Region Süd zu verwenden, müsste die Funktion lauten:

=XVERWEIS(H2;A2:A11;C2:C11)

Die Kundennummer, die gesucht werden soll, kann dadurch veränderlich sein, da in Zelle H2 jeder beliebige Wert als Suchkriterium möglich ist. Dies trifft jedoch nicht auf die Auswahl der Region zu, die durch C2:C11 sehr starr festgelegt wird. Allerdings kann auch dieser starre Rückgabebereich durch einen weiteren XVERWEIS() ersetzt werden:

XVERWEIS(H3;B1:E1;B2:E11)

Ineinander verschachtelt entsteht so folgende Funktion:

=XVERWEIS(H2;A2:A11;XVERWEIS(H3;B1:E1;B2:E11))

Der innere XVERWEIS() liefert eine Liste von Werten – die Daten der in Zelle H3 angegebenen Region Süd. Dieser Zellbereich wird an den äußeren XVERWEIS() übergeben. Der erste Teil der verschachtelten Funktion XVERWEIS(H3;B1:E1;...) ermittelt nun den Umsatzwert des ausgewählten Kunden (5660; Abbildung 8.32).

Betrachtet man den Aufbau und die Anwendungsmöglichkeiten von XVERWEIS(), dann stellt man fest, welche immense Bedeutung dem Schritt beigemessen werden muss, die ursprünglich bei SVERWEIS() und WVERWEIS() zusammenhängende Matrix in zwei physikalisch nicht miteinander verbundene Such- und Rückgabebereiche zu teilen. Dadurch ergibt sich unter anderem die Möglichkeit, während der Berechnung Tabellenauszüge wie im vorangegangenen Beispiel zu bilden. Und dies macht die neue Funktion für den Einsatz in dynamischen Reports besonders interessant.

8.3.8 Tabellen spalten- und zeilenweise mit »SVERWEIS()« und »WVERWEIS()« durchsuchen

Es wäre nur zu logisch, wenn sich der XVERWEIS() im Laufe der Jahre zum neuen Standard bei der Nutzung von Verweisfunktionen entwickeln würde. Aber bis dahin wird es noch ältere Versionen im Einsatz geben. Auch zahlreiche Auswertungen und Reports haben eine lange Lebensdauer, unter anderem deshalb, weil schlichtweg die Zeit fehlt, funktionierende Dateien aus Sicht der Kalkulationstechniken auf den neuesten Stand zu bringen. Deshalb ist es sinnvoll, auch den beiden Klassikern SVERWEIS() und WVERWEIS() die gebührende Aufmerksamkeit zu schenken. In der Arbeitsmappe *O8_Verweise_SVERWEIS_01.xlsx* ist das Arbeitsprinzip der Funktion SVERWEIS(Prüfung; Matrix; Spaltenindex; Bereich_Verweis) exemplarisch dargestellt (Abbildung 8.33). Die Datei enthält eine Referenztable im Zellbereich D1 bis E6. Die erste Spalte stellt für den Benutzer die wohl am besten les- und erinnerbare Information bereit: eine Liste mit Bezeichnungen. Um die Kostenanalyse im Zellbereich A1 bis B5 durchzuführen, wäre es am angenehmsten, eine der Bezeichnungen einzugeben, um die davon abhängigen Berechnungen der Anzahl und Kosten in den Zellen B4 und B5 zu starten.

| | A | B | C | D | E |
|---|-------------------------------|--|---|-------------|-------|
| 1 | Kostenanalyse | | | Bezeichnung | Konto |
| 2 | ausgewählte Bezeichnung: | Bewirtung | | Pflanzen | 1234 |
| 3 | | =SVERWEIS(B2;D1:E6;2;FALSCH) | | | 2100 |
| 4 | Anzahl in Kategorie Bewirtung | SVERWEIS(Suchkriterium; Matrix; Spaltenindex; [Bereich_Verweis]) | | | |
| 5 | Kosten (Pauschale): | 1.355,00 € | | Fahrtkosten | 2150 |
| 6 | | | | Dekoration | 2500 |
| 7 | | | | | |

Abbildung 8.33 Suchen in einer Matrix mit »SVERWEIS()«

Genau das funktioniert jedoch nicht, weil die Liste der Kosten im Zellbereich G1 bis I16 diese Bezeichnung nicht enthält, sondern lediglich die Konten, die in der Referenztable die zweite Spalte bilden.

Erste Spalte mit »SVERWEIS()« durchsuchen

Die Funktion =SVERWEIS(B2;D1:E6;2;FALSCH) hilft Ihnen in diesem konkreten Beispiel mit einer Übersetzungsarbeit. Wird der in B3 eingetragenen Funktion eine Bezeichnung überge-

ben (B2), durchsucht sie die erste Spalte der Matrix (D1:E5) auf eine Übereinstimmung und gibt das zugehörige Konto aus der zweiten Spalte zurück (2), sofern eine hundertprozentige Übereinstimmung zwischen Suchbegriff und Fundstelle besteht (FALSCH). Der senkrechten Suchrichtung verdankt die Funktion ihren Anfangsbuchstaben: SVERWEIS().

Wichtig ist in diesem Zusammenhang die Bedeutung des Arguments `Bereich_Verweis`. Ist es auf FALSCH oder 0 gesetzt, wird eine genaue Entsprechung von Gesuchtem und Gefundenem erzwungen. Dies umfasst auch die Möglichkeit, dass kein korrespondierender Wert gefunden wird und der Fehlerwert #NV statt z. B. eines Kontos zurückgegeben wird. In der Folge kann dies den Benutzer wiederum dazu zwingen, den möglichen Fehlerwert mit Funktionen wie WENNFehler() zu unterdrücken. Doch dazu später mehr. Möchten Sie hingegen ausdrücklich den Fehlerwert #NV abfangen, andere Fehlerwerte hingegen nicht, steht Ihnen auch die Funktion WENNNV() zur Verfügung.

Nimmt das Argument hingegen den Wert WAHR an oder wird es einfach weggelassen, gibt sich Excel bei einer aufsteigend sortierten Liste bereits mit einer Ähnlichkeit zwischen Suchkriterium und Fundstück zufrieden. Der Zeiger stoppt in der ersten Spalte bei dem Wert, der am nächsten beim Suchbegriff liegt, und Excel liest den entsprechenden Spaltenindex aus.

In der Beispieldatei `08_Verweise_SVERWEIS_01.xlsx` wird die Variante, bei der die genaue Entsprechung erzwungen wird, verwendet. Das Resultat bildet schließlich die Grundlage für zwei bedingte Kalkulationen in den Zellen B4 und B5. Dort kann nun durch das Heraussuchen des Kontos aus der Referenzliste mit `=ZÄHLENWENN(H2:H16;B3)` die Anzahl der Buchungen und mit `=SUMMEWENN(H2:H16;B3;I2:I16)` auch deren Gesamtsumme ermittelt werden.

Erste Zeile mit »WVERWEIS()« durchsuchen

Wie Sie im Beispiel des dynamischen und regionalen Kalenders bereits gesehen haben, funktioniert die Suche auch in einer anderen Richtung. Wird die erste Zeile einer Matrix auf ein Suchkriterium hin untersucht, ist für diese waagerechte Suche die Funktion WVERWEIS() verantwortlich. Ihr Funktionsprinzip unterscheidet sich ansonsten in keiner Weise von SVERWEIS(). Davon können Sie sich in der Beispieldatei `08_Verweise_WVERWEIS_01.xlsx` einmal mehr überzeugen (Abbildung 8.34).

| | A | B | C | D | E |
|---|---------------------------------|--------------------------------|---|--------------------|--------------|
| 1 | Kostenanalyse | | | Bezeichnung | Konto |
| 2 | ausgewählte Bezeichnung: | Fahrtkosten | | Pflanzen | 1234 |
| 3 | | =WVERWEIS(B2;A19:F20;2;FALSCH) | | | 2100 |
| 4 | Anzahl in Kategorie Fahrtkosten | | | | 1200 |
| 5 | Kosten (Pauschale): | 1.187,00 € | | Fahrtkosten | 2150 |
| 6 | | | | Dekoration | 2500 |

Abbildung 8.34 Durchsuchen einer horizontalen Matrix mit »WVERWEIS()«

Im Zellbereich A19 bis F20 befindet sich erneut eine Referenztablelle. Doch diesmal ist die Liste horizontal ausgerichtet. Wollen Sie nach einer Bezeichnung suchen, um ein Konto zu finden, muss die Funktion in Zelle B3 diesmal `=WVERWEIS(B2;A19:F20;2;FALSCH)` lauten.

Transponieren einer Matrix

Sicherlich ist Ihnen aufgefallen, dass die Referenztablelle in diesem Beispiel gleich zweimal im Tabellenblatt vorkommt. Neben dem eben benutzten Bereich A19 bis F20 befindet sie sich noch einmal im Zellbereich D1 bis E5. Die untere der beiden Tabellen ist einfach gedreht oder – wie es in Excel heißt – *transponiert* worden. Sie können eine Tabelle auf zweierlei Arten transponieren:

- ▶ **Manuell:** Markieren Sie die Daten, und kopieren Sie sie mit `[Strg] + [C]` in die Zwischenablage. Danach bewegen Sie den Cursor an die Zielstelle und fügen den Inhalt der Zwischenablage mit `START • ZWISCHENABLAGE • EINFÜGEN • TRANSPONIEREN` wieder ein.
- ▶ **Per Funktion:** Verwenden Sie die Funktion `MTRANS(Matrix)` aus dem Funktionsassistenten. Markieren Sie einen Zielbereich im Tabellenblatt, der mindestens die Größe der zu transponierenden Tabelle hat, starten Sie die Funktion dann aus dem Funktionsassistenten, und schließen Sie die Eingabe mit `[Strg] + [↕] + [↶]` ab, da es sich um eine Matrixfunktion handelt.

Die Vorteile von `MTRANS()` bei der Neuordnung von Basisdaten liegen gegenüber dem manuellen Drehen via Zwischenablage auf der Hand: Die Funktion ist dynamisch. Aktualisieren Sie Ihre Basisdaten, wird auch der transponierte Bereich angepasst. Bei der manuellen Variante müssten Sie nach jedem Ändern der Basisdaten die Tabelle auch wieder manuell transponieren.

`MTRANS()` macht aber auch in einem anderen Zusammenhang der Überschrift dieses Abschnitts alle Ehre: Es ist eine dezidierte *Matrixfunktion*. Das erkennen Sie an einigen typischen Merkmalen:

- ▶ Anders als normale Funktionen werden Matrixfunktionen häufig nicht in eine Zielzelle eingegeben, sondern gleich in einen zusammenhängenden Zellbereich.
- ▶ Sie werden nicht mit `[↶]`, sondern mit `[Strg] + [↕] + [↶]` abgeschlossen.
- ▶ In der Editierzeile erkennen Sie Matrixfunktionen an den geschweiften Klammern, die Anfang und Ende des Funktionstextes umschließen.
- ▶ Sollten Sie versuchen, einen Teil des Ergebnisbereichs einer Matrixfunktion zu überarbeiten oder zu entfernen, wird Ihnen dies nicht gelingen; Änderungen sind nur für den gesamten zusammenhängenden Bereich zulässig.

Matrixfunktionen

Zwar gibt es eine Kategorie `MATRIX` im Funktionsassistenten, doch sind die hier gemeinten Matrixfunktionen über verschiedene Kategorien verteilt. Und auch »normale« Funktionen, beispielsweise `SUMME()`, können als Matrixfunktionen in Excel eingesetzt werden. Suchen Sie nach einem gemeinsamen Merkmal der Matrixfunktionen, ist dies die Art und Weise, mit der sie ihre Aufgaben erledigen. Sie durchlaufen einen Zellbereich nicht einmal von oben nach unten, wie es beispielsweise bei der Berechnung der Summe geschieht, um dann das Ergeb-



nis in eine Zelle zu schreiben. Stattdessen durchlaufen sie den definierten Zellbereich mehrmals, speichern bei jedem Durchlauf die ermittelten Zwischenergebnisse ab und sind in der Lage, das Endergebnis oder die Endergebnisse abschließend in eine oder mehrere Zellen zu schreiben.

Typische und wichtige Matrixfunktionen sind:

- ▶ MTRANS(Matrix) dient dem Transponieren von Zellbereichen.
- ▶ TREND(Y_Werte; X_Werte; Neue_X_Werte; Konstante) berechnet einen linearen Trend.
- ▶ HÄUFIGKEIT(Daten; Klassen) berechnet eine Häufigkeitsverteilung.

Ein Beispiel für die Verwendung von SUMME() als Matrixfunktion:

```
{=SUMME((A2:A10="Mai")*(B2:B10="Nord")*D2:D10)}
```

Durchsucht wird der Zellbereich in Spalte A nach der Bedingung Mai, der Bereich in Spalte B wird auf das Suchkriterium Nord hin überprüft. Die Werte aus Spalte D, die die beiden Bedingungen erfüllen, werden anschließend addiert.

Den Möglichkeiten von solchen bedingten Kalkulationen, bei denen auch Matrixfunktionen eine wichtige Rolle spielen, ist Kapitel 10, »Bedingte Kalkulationen in Datenanalysen«, gewidmet. Informieren Sie sich dort über Matrixfunktionen, wie z. B. SUMMENPRODUKT().

8.3.9 Finden des letzten Eintrags einer Spalte oder Zeile

Diese Fragestellung ist Ihnen vielleicht beim Erstellen eines Soll-Ist-Vergleichs schon einmal begegnet: Sie hängen an eine bestehende Tabelle kontinuierlich Zeilen oder Spalten an, benötigen aber immer nur den letzten, den aktuellsten Wert der Tabelle, um ihn mit einem anderen Wert, der Soll-Vorgabe, zu vergleichen.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|--------------|----------|----------|--|----------|----------|----------|----------|----------|---|-------------|----------|
| 1 | Soll | 19.500 € | 17.500 € | 20.000 € | 18.000 € | 17.500 € | 18.500 € | 17.000 € | 20.000 € | | Maximalwert | 24.298 € |
| 2 | Abweichung | -38,5% | -9,4% | -32,3% | 9,3% | 22,3% | 17,5% | -23,5% | 1,2% | | | |
| 3 | Letzter Wert | 12.000 € | 15.859 € | =VERWEIS(\$L\$1;E5:E16) | | | 21.735 € | 13.010 € | 20.245 € | | | |
| 4 | | Nord | West | VERWEIS(Suchkriterium: Suchvektor, [Ergebnisvektor]) | | | | Nordost | Mitte | | | |
| 5 | Status 1 | 17.263 € | 15.886 € | VERWEIS(Suchkriterium: Matrix) | | | | 15.068 € | 12.179 € | | | |
| 6 | Status 2 | 18.943 € | 21.265 € | 21.379 € | 13.975 € | 15.276 € | 12.086 € | 12.337 € | 19.407 € | | | |
| 7 | Status 3 | 13.533 € | 17.440 € | 15.529 € | 13.472 € | 14.491 € | 12.774 € | 18.819 € | 20.752 € | | | |
| 8 | Status 4 | 19.473 € | 19.423 € | 16.651 € | 23.340 € | 21.846 € | 17.518 € | 24.297 € | | | | |
| 9 | Status 5 | 22.951 € | 19.100 € | 20.184 € | 19.782 € | 21.410 € | 15.287 € | 12.417 € | 17.962 € | | | |
| 10 | Status 6 | 18.570 € | 15.859 € | 17.998 € | 13.534 € | | 12.693 € | 13.010 € | 20.285 € | | | |
| 11 | Status 7 | 14.631 € | | 23.961 € | 19.680 € | | 16.319 € | | 20.245 € | | | |
| 12 | Status 8 | 15.109 € | | 13.535 € | | | 21.735 € | | | | | |
| 13 | Status 9 | 12.000 € | | | | | | | | | | |
| 14 | Status 10 | | | | | | | | | | | |
| 15 | Status 11 | | | | | | | | | | | |
| 16 | Status 12 | | | | | | | | | | | |

Abbildung 8.35 Den aktuellen Wert für einen Soll-Ist-Vergleich finden

In der Arbeitsmappe O8_Verweise_VERWEIS_01.xlsx habe ich dieses Beispiel aufgegriffen (Abbildung 8.35). Es liefert eine einfache Lösung für das beschriebene Problem und ist eine

kleine Hommage an Bill Jelen, besser bekannt unter dem Namen *Mr. Excel*, der eine ähnliche Vorgehensweise in einem seiner lohnenswerten Excel-Podcasts vorstellte. Versäumen Sie es nicht, auf www.mrexcel.com vorbeizuschauen und den einen oder anderen Podcast zu genießen. Großes Excel-Kino im ganz kleinen Format!

Alternative 1: »SVERWEIS()«

Wenn Sie einige Informationen zusammenfassen, die Sie bezüglich der Verweisfunktionen bereits besitzen, dann kommen Sie unter Umständen auf die Idee, dass ein SVERWEIS() in der Lage wäre, die gestellte Aufgabe zu lösen. Die Funktion könnte beispielsweise den Zellbereich B5 bis B16 durchsuchen. Wonach? Nach einem möglichst hohen Wert, der in diesem Zellbereich garantiert nicht vorkommt. Wäre das Argument Bereich_Verweis nicht oder auf WAHR gesetzt, würde die Funktion bis zum letzten Eintrag der Liste suchen und nicht fündig werden. Sie gäbe den letzten Wert des durchsuchten Bereichs zurück, vorausgesetzt, der Spaltenindex wäre 1, Such- und Ergebnisspalte wären also identisch.

Alternative 2: »VERWEIS()«

Diese Lösung würde funktionieren. Sie hätte aber einen ästhetischen sowie einen didaktischen Mangel:

- ▶ Rein ästhetisch wäre zu bemängeln, dass es eine andere Funktion gibt, bei der wir uns die Eingabe von zwei Argumenten sparen können.
- ▶ Didaktisch betrachtet entginge uns durch den Gebrauch der altbekannten Funktion eine neue wichtige Stütze bei der Analyse von Matrizen – die Funktion VERWEIS().

Diese Funktion, die es in einer Vektor- und in einer Matrixausführung gibt, wird hier in der Matrixvariante benutzt. In Zelle B3 lauten die Argumente =VERWEIS(\$L\$1;B5:B16). Grundannahmen bei der Verwendung der Funktion sind:

- ▶ Es wird eine Matrix anhand eines Suchkriteriums durchsucht.
- ▶ Besitzt die Matrix mehr Zeilen als Spalten oder sind Spalten- und Zeilenzahl identisch, wird die erste Spalte durchsucht; umgekehrt wird die erste Zeile durchsucht, wenn mehr Spalten als Zeilen vorhanden sind.
- ▶ Wird eine Übereinstimmung mit dem Suchkriterium in der ersten Spalte bzw. Zeile festgestellt, gibt die Funktion den korrespondierenden Wert aus der letzten Spalte bzw. Zeile zurück.
- ▶ Wird hingegen keine Übereinstimmung mit dem Suchkriterium gefunden, fällt der Zeiger der Funktion um eine Position zurück und wählt den nächstkleineren Wert in der Matrix.
- ▶ Letzteres kann nur funktionieren, wenn die Matrix auf Basis der Spalte, die durchsucht wird, aufsteigend sortiert ist.

Diese hier beschriebenen Grundannahmen werden gleich in drei Punkten bei der Anwendung der Funktion zum Auffinden des letzten Eintrags in einer Spalte nicht erfüllt: Erstens

ist die uns vorliegende Liste nicht sortiert. Zweitens ist die Spalte, die durchsucht wird, mit der Ergebnisspalte identisch; die Matrix ist also einspaltig. Drittens wird mit dem Ergebnis der Funktion `=MAX(B5:I16)+1` eine Zahl gesucht, die genau um den Wert 1 über dem Maximalwert liegt und deshalb unmöglich gefunden werden kann.

Doch genau diese Verfremdungen der Argumente haben zur Folge, dass die Funktion `VERWEIS()` bis zum letzten Eintrag einer jeden unsortierten Spalte den Suchvorgang erfolglos fortsetzt. Danach fällt der Zeiger der Funktion auf den letzten geprüften Wert zurück. Und dies ist der letzte, also aktuellste, Wert in der jeweiligen Spalte.

8.4 Funktionen zur Dynamisierung von Tabellen

Die Ausgangslage der folgenden Beschreibung ist Ihnen sicherlich auch bekannt: Sie beziehen in regelmäßigen Abständen aktuelle Datenbestände aus anderen Programmen. Dann beginnen Sie damit, die Daten zu analysieren und zu verdichten. Am Ende des Arbeitsprozesses möchten Sie über eine Reihe aussagekräftiger Tabellen und Diagramme verfügen. Eigentlich ganz einfach!

Erschwert wird das Datenmanagement jedoch zumeist durch die schiere Menge an Auswertungen, Dimensionen und Betrachtungsweisen. Gingen Sie von lediglich fünf Vertriebsgebieten und zehn darin vertretenen Produkten aus, kämen Sie in der Einzelbetrachtung bereits auf 50 Tabellen und ebenso viele Diagramme. Hinzuzuzählen wären noch die regionalen oder produktspezifischen Vergleiche und die zeitliche Analyse der Daten.

In der Praxis sind diese Teildatenbestände durch verschiedene Funktionen untereinander verknüpft, was es noch schwieriger macht, den Überblick zu bewahren. Der Aufwand für die Pflege und Datenaktualisierung bei der Verwendung solcher Spaghetti-Lösungen ist immens. Ganz zu schweigen von den anschwellenden Dateigrößen, die zumeist erheblich auf die Arbeitsgeschwindigkeit von Excel drücken.

Vor dem Hintergrund dieses Szenarios spielen Funktionen, mit denen Sie dynamische Tabellen und Diagramme generieren können, eine wichtige Rolle. Sie bilden neben den Pivottabellen als Ad-hoc-Analysewerkzeuge und der VBA-Programmierung als Tool für die Entwicklung ganzer Anwendungen die dritte Säule bei der flexiblen und wiederkehrenden Auswertung von großen Datenmengen. Von den Pivottabellen unterscheiden sie sich durch ihre fast unbeschränkte Formatierbarkeit, die klare Benutzerführung und die Möglichkeit der problemlosen Weiterverarbeitung einmal generierter Daten. Der Unterschied zur VBA-Programmierung liegt für den Controller vor allem darin, dass er keine Programmierkenntnisse benötigt, um solche dynamischen Reports zu erstellen. Er kann sich stattdessen aller Mittel im Funktionsassistenten auf der Oberfläche des Tabellenblattes bedienen, um seine Ziele zu erreichen.

Zielführend ist dabei vor allem die systematische Nutzung einiger kombinierter Excel-Werkzeuge. In Kapitel 7, »Dynamische Reports erstellen«, bin ich darauf bereits ausführlich einge-

gangen. Besonders wichtig sind dabei die dynamischen Datentabellen, die Sie mit `[Strg] + [T]` erstellen. Sie bilden das Werkzeug Nummer eins zur Dynamisierung. Doch es gibt auch Konstellationen, in denen Sie eine andere Lösung als eine komplette dynamische Datentabelle benötigen. Deshalb werde ich Ihnen einige Funktionen zeigen, die ebenjenes dynamische Potenzial besitzen, das Ihnen die tägliche Arbeit erheblich erleichtern kann – `INDEX()`, `INDIREKT()`, `VERGLEICH()`, `BEREICH.VERSCHIEBEN()` und Co.

8.4.1 Dynamischen Summenbereich mit »BEREICH.VERSCHIEBEN()« erstellen

In der Arbeitsmappe `O8_Dynamisierung_BEREICH.VERSCHIEBEN_01.xlsx` ist das Problem der sich verändernden Zellbereiche bei der Nutzung von Kalkulationsfunktionen zunächst an einem sehr überschaubaren Beispiel beschrieben. Im Zellbereich B2 bis B6 des Tabellenblattes *dynamische Summe I* wurden einige Werte erfasst. In Zelle G2 wurde aus ihnen mit der Funktion `=SUMME(B2:B6)` das Gesamtergebnis gebildet. Tragen Sie nun zu einem späteren Zeitpunkt in B7 einen weiteren Wert in Spalte B ein, erkennt Excel zwar, dass Daten hinzugekommen sind, doch das Programm bezieht den neuen Wert nicht in die Bildung der Summe ein.

Stattdessen zeigt Ihnen das Programm durch ein kleines grünes Dreieck in der Ecke links oben in der Summenzeile an, dass eventuell ein Problem vorliegt. Der Hinweistext beim Klicken auf das Ausrufezeichen lautet: **DIE FORMEL SCHLIESST NICHT ALLE ANGRENZENDEN ZELLEN EIN** (Abbildung 8.36). Um das Problem zu umgehen, müssten Sie einen dynamischen Bereich definieren, bei dem erkannt wird, wenn ein oder mehrere Werte im Zellbereich ergänzt worden sind.

| 1 | Produkt | Wert | C | D | E | F | G | H | I | J | K | L |
|----|-----------|------|---|---|---|-------|-----|---|---|---|---|---|
| 2 | Produkt 1 | 100 | | | | Summe | 472 | | | | | |
| 3 | Produkt 2 | 120 | | | | | | | | | | |
| 4 | Produkt 3 | 124 | | | | | | | | | | |
| 5 | Produkt 4 | 128 | | | | 1. | | | | | | |
| 6 | Produkt 5 | 130 | | | | 2. | | | | | | |
| 7 | | | | | | 3. | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |

Abbildung 8.36 Hinweis auf Zellen, die an einen berechneten Zellbereich angrenzen

Beim Aufbau eines dynamischen Bereichs zählen Sie in einem ersten Schritt, wie viele Werte in Spalte B der Tabelle überhaupt vorhanden sind. Dazu setzen Sie die Funktion `=ANZAHL2($B:$B)` ein. Mit ihrer Hilfe ermitteln Sie die Anzahl der nicht leeren Zellen in der gesamten Spalte B, unabhängig davon, ob es sich um Textüberschriften oder Zahlen handelt. Da Sie im Vorfeld nicht wissen können, wie viele Werte in der Spalte zukünftig stehen werden, ist es ratsam, den Bereich mit `$B:$B` anzugeben. Dadurch wird die gesamte Spalte von der ersten bis zur letzten Zeile untersucht.

Das Ergebnis des Zählens muss nun an eine Funktion übergeben werden, die daraus einen dynamischen Bereich erstellen kann. `BEREICH.VERSCHIEBEN()` ist dazu in der Lage. Die Funktion bewegt, ausgehend von einem definierten Startpunkt, einen Zellbereich auf dem Tabellenblatt an eine bestimmte Stelle. Die Größe des Zellbereichs bestimmen Sie, indem Sie z. B. die Größe des Zellbereichs mit `ANZAHL2()` berechnen lassen.

Die beiden Funktionen scheinen perfekt zusammenzupassen. Mit dem Ausdruck `=BEREICH.VERSCHIEBEN(B1;;;ANZAHL2($B:$B);1)` testen Sie das beispielsweise in Zelle G14 der Beispieldatei. Nehmen Sie den kleinen Rückschlag, dass Ihnen Excel den Fehlerwert `#WERT!` präsentiert, gelassen. Dies bedeutet nicht, dass Sie etwas Fehlerhaftes eingegeben haben. Die Funktion ist lediglich an dieser Stelle nicht brauchbar. Und glücklicherweise müssen wir die Funktion dort auch nicht einsetzen. Der Fehlerwert verschwindet aber schlagartig, wenn Sie die verschachtelte Funktion als Bereichsangabe bei der Berechnung der Gesamtsumme verwenden.

In Zelle G2 steht dann folgende Funktion (Abbildung 8.37):

```
=SUMME(BEREICH.VERSCHIEBEN($B$1;;;ANZAHL2($B:$B);1))
```

Was ist in diesem Beispiel genau geschehen?

- ▶ Sie haben mit dem ersten Argument Bezug einen Startpunkt mit der Zelle `B1` festgelegt.
- ▶ Danach wurden zwei Argumente, die sich auf das Verschieben eines Zellbereichs für diesen Startpunkt beziehen, einfach übersprungen.
- ▶ Um im vierten Argument die Höhe des Bereichs zu benennen, wurde die Funktion `ANZAHL2()` eingesetzt, woraus sich zwangsläufig eine variable Größe der Tabelle ergibt.
- ▶ Zuletzt wurde die Breite des Zellbereichs mit dem Wert `1` als einspaltiger Zellbereich definiert.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|-----------|------|---|---|---|--------|--|---|---|---|---|---|
| 1 | Produkt | Wert | | | | | | | | | | |
| 2 | Produkt 1 | 100 | | | | Summe: | =SUMME(BEREICH.VERSCHIEBEN(\$B\$1;;;ANZAHL2(\$B:\$B);1)) | | | | | |
| 3 | Produkt 2 | 120 | | | | | SUMME(Zahl1; [Zahl2] ...) | | | | | |
| 4 | Produkt 3 | 124 | | | | | | | | | | |
| 5 | Produkt 4 | 128 | | | | 1. | =ANZAHL2(\$B:\$B) | | | | | |
| 6 | Produkt 5 | 130 | | | | 2. | =BEREICH.VERSCHIEBEN(\$B\$1;;;ANZAHL2(\$B:\$B);1) | | | | | |
| 7 | Produkt 6 | 145 | | | | 3. | =SUMME(BEREICH.VERSCHIEBEN(\$B\$1;;;ANZAHL2(\$B:\$B);1)) | | | | | |
| 8 | | | | | | | | | | | | |

Abbildung 8.37 Berechnung der Summe für einen dynamischen Bereich

Wird diese Funktion in einer Kalkulationsfunktion als Zellbereich verwendet, erweitert sich der Kalkulationsbereich automatisch, wenn Werte den bestehenden Daten angehängt werden. Voraussetzung: Die neuen Werte müssen unmittelbar den bereits vorhandenen Daten angefügt werden. Die Funktionen `ANZAHL2()` oder `ANZAHL()` sollten sich deshalb immer auf eine Spalte beziehen, in der obligatorische Werte stehen (z. B. Produkt-, Kunden- oder Personalnummern).

Streng genommen ist diese Nutzungsform von `BEREICH.VERSCHIEBEN()` am ehesten zu ersetzen, und zwar durch eine dynamische Datentabelle und strukturierte Bezüge. Nehmen wir an, die Produktliste wäre in eine Datentabelle umgewandelt und mit dem Namen `Umsatz` versehen worden, dann hätte der strukturierte Bezug in `SUMME()` ebenfalls eine dynamische Erweiterbarkeit zur Folge:

```
=SUMME(Umsatz[Wert])
```

Die entscheidende Überlegung, welche der beiden Möglichkeiten angewandt wird, sollte stets die Auswirkungen auf die Performance von Excel mit in Erwägung ziehen. Einige Berechnungen mit `BEREICH.VERSCHIEBEN()` mögen unkritisch sein. Wird diese volatile Funktion jedoch hundert- oder tausendfach eingesetzt, bleibt das nicht ohne negative Auswirkungen auf die Rechengeschwindigkeit.

Im Sinne einer Vereinheitlichung der eigenen Werkzeuge und Methoden spricht Weiteres für die Verwendung von strukturierten Bezügen. Power Pivot als wichtiges Analysetool benutzt diese Bezugsform. Möchten Sie also Arbeitsmappenfunktionen und DAX-Funktionen angleichen, erreichen Sie dies nur über dynamische Datentabellen. Und die Vereinheitlichung von Methoden und Mitteln ist eine Grundforderung von M, wie Model, im xlSMILE-Konzept.

Produkt-, Regions- oder Periodendaten mit einem dynamischen Bereich markieren

Doch mit `BEREICH.VERSCHIEBEN()` ist noch mehr möglich. Im Tabellenblatt *dynamische Summe II* der Beispieldatei können Sie sich davon überzeugen. Das Tabellenblatt enthält eine einfache Liste, in der Daten zu unterschiedlichen Produkten dargestellt werden. Die Zielsetzung ist einfach: Es soll für jedes Produkt die Summe der Ergebnisse aus den vier angegebenen Regionen gebildet werden. Die Summenbildung soll auf Knopfdruck des Benutzers erfolgen.

Technisch bedeutet dies, dass mit der Funktion `SUMME()` die Werte aus den Spalten C bis F addiert werden müssen. Die Zeile, deren Werte summiert werden sollen, muss jedoch flexibel angesteuert werden. Excel bietet verschiedene Funktionen an, mit denen Sie Zellbezüge über das Tabellenblatt wandern lassen können. Doch erneut ist die Funktion `BEREICH.VERSCHIEBEN()` die erste Wahl bei der Lösung dieser Aufgabenstellung (Abbildung 8.38).

| H | I | J | K | L | M | N |
|---|---|-------------|---|---|---|---|
| | | | | | | |
| | | | Summe für Produkt | | | |
| 7 | | Produkt 413 | =SUMME(BEREICH.VERSCHIEBEN(\$C\$1;\$H\$3;;1;4)) | | | |
| | | | SUMME(Zahl1; [Zahl2] ...) | | | |

Abbildung 8.38 Berechnung einer Summe auf Basis einer Produktauswahl

Wie Sie bereits im vorherigen Beispiel gesehen haben, eignet sie sich für die hier skizzierte Aufgabe besonders, da sie einen Zellbereich, der in seiner Höhe und Breite flexibel bestimmt werden kann, von einem fest definierten Ausgangspunkt wie dem Anfang einer Datentabelle vertikal und/oder horizontal auf dem Tabellenblatt verlagern kann.

Lassen Sie uns jetzt einen etwas genaueren Blick auf die fünf Argumente der Funktion werfen, die Sie in Tabelle 8.6 finden.

| Argument | Funktion |
|----------|--|
| Bezug | Definiert den Startpunkt der Tabelle. In der Beispieldatei ist dies die Zelle \$C\$1, also die erste Zeile der ersten Spalte, in der sich Umsatzdaten befinden. Diesen Zellbezug sollten Sie immer absolut setzen. |
| Zeilen | Dieses Argument gibt an, um wie viele Zeilen der Zellbereich, bezogen auf den Startpunkt, verschoben werden soll. Dieser Wert muss in der Beispieldatei dynamisch bestimmt werden. |
| Spalten | Mit diesem Argument wird festgelegt, um wie viele Spalten der Bereich, bezogen auf den Startpunkt, verschoben werden soll. Da in der Beispieldatei die Berechnung immer in der ersten Spalte beginnen soll, wird hier kein Wert oder 0 eingegeben. |
| Höhe | Dieses Argument dient dazu, die Höhe des verschobenen Bereichs fest oder veränderlich zu bestimmen. Im Beispiel soll die Summe immer für ein Produkt berechnet werden. Da die Produktdaten eine Zeile umfassen, ist die Höhe mit 1 anzugeben. |
| Breite | Analog zum Argument Höhe legen Sie hiermit die Breite des zu verschiebenden Bereichs fest. Auch hier kann wahlweise eine feste Vorgabe oder eine flexible Berechnung erfolgen. In der Beispieldatei ist die Breite des Bereichs gleichbleibend mit vier Spalten anzugeben. |

Tabelle 8.6 Argumente der Funktion »BEREICH.VERSCHIEBEN()«

Den dynamischen Bereich an die Summenfunktion übergeben

Im Unterschied zum ersten Anwendungsbeispiel müssen wir nun also alle Argumente von BEREICH.VERSCHIEBEN() verwenden. Der variable Teil ist diesmal nicht die Höhe des zu verschiebenden Zellbereichs, sondern die Anzahl der Zeilen, um die der Bereich verschoben werden soll. Wenn Sie das Ergebnis für Produkt 627 sehen wollen, muss der Bereich um zwei Zeilen verschoben werden. Für Produkt 413 sind es schon sieben Zeilen.

Diesen variablen Teil der Funktion können Sie über den Wert in Zelle H3 steuern. Geben Sie dort den Wert, um den der Zellbereich nach unten verschoben werden soll, per Tastatur ein, oder wählen Sie die Zeile durch ein Formularfeld aus. In der Beispieldatei habe ich über das Menü ENTWICKLERTOOLS • STEUERELEMENTE • EINFÜGEN • FORMULARSTEUERELEMENTE ein

Kombinationsfeld in das Tabellenblatt eingefügt. Bei der Auswahl eines Produkts aus der Liste wird dessen Position in der Liste als Zahl in eine Verknüpfungszelle, z. B. H3, geschrieben. Somit haben Sie eine einfache Steuerung des veränderlichen Bezugs der Funktion BEREICH.VERSCHIEBEN().

Alle anderen Koordinaten des Zellbereichs bleiben hingegen unveränderlich. Der Startpunkt wird immer Zelle C1 sein; ein Verschieben der Spalten ist nicht notwendig. Die Höhe des Zellbereichs, den Sie berechnen möchten, wird immer 1 sein, seine Breite wird immer vier Spalten betragen. Daraus ergibt sich die folgende Funktion:

```
=BEREICH.VERSCHIEBEN($C$1;$H$3;0;1;4)
```

Möchten Sie die Summe zu diesem dynamischen Bereich in Zelle G3 ausgeben, verwenden Sie dort diese Funktion:

```
=SUMME(BEREICH.VERSCHIEBEN($C$1;$H$3;;1;4))
```

Wägen Sie den Einsatz von »BEREICH.VERSCHIEBEN()« ab

Bereits im vorangegangenen Kapitel habe ich den volatilen Charakter von BEREICH.VERSCHIEBEN() erwähnt. Die häufigen Neuberechnungen, die auch dann initiiert werden, wenn Sie an einer anderen Stelle der Arbeitsmappe Änderungen vornehmen, können Excel drastisch ausbremsen, wenn Sie diese Funktion häufig verwenden. Deshalb sollten Sie den Einsatz von BEREICH.VERSCHIEBEN() immer gegenüber dynamischen Datentabellen abwägen.

8.4.2 Zusammengesetzte Zellbezüge mit »INDIREKT()« erstellen

Zellbezüge setzen sich in Excel aus einem Buchstaben für die Spaltenbezeichnung und einer Zeilennummer zusammen. Dies wird auch als die *A1-Schreibweise* oder *A1-Methode* bezeichnet. Sie ist die gängigste Methode, Zellen zu adressieren. Die Adressierung einer Zelle oder eines Zellbereichs funktioniert im Normalfall immer dann, wenn Sie den Zellbereich direkt in die Formel schreiben. Er funktioniert jedoch nicht, wenn Sie einen Zellbezug, der als Text selbst in einer Zelle steht, in eine Formel oder Funktion übernehmen möchten.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|----------|-------------------|----------|----------|---|-----------|---------------------------------------|---|-----------------|-----------------------------------|---|------|
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | Kategorien | | | | | Auswahl: | C | A5 | 0 | | 5 |
| 4 | A | B | C | D | | Ergebnis: | =SUMME(INDIREKT(G3&5):INDIREKT(G3&7)) | | | | | 1011 |
| 5 | 100 | 200 | 310 | 401 | | | SUMME(Zahl1; [Zahl2]; ...) | | | | | |
| 6 | 120 | 210 | 312 | 410 | | | | | | | | |
| 7 | 130 | 215 | 319 | 420 | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | Falsch! | =SUMME(I3:I4) | | |
| 11 | | | | | | | | | Richtig! | =SUMME(INDIREKT(I3):INDIREKT(I4)) | | |

Abbildung 8.39 Verwendung von »INDIREKT()« bei der Bildung einer Summe

Im Tabellenblatt *Indirekt()* I der hier verwendeten Arbeitsmappe *08_Dynamisierung_INDIREKT_01.xlsx* wird der Versuch unternommen, aus den Zellen I3 und I4 zwei Zellbezüge zu übernehmen, um eine Summe in Zelle J3 zu bilden: =SUMME(I3:I4). Doch das funktioniert nicht. Das Ergebnis ist 0, obwohl im Zellbereich A5 bis A7 Zahlen stehen (Abbildung 8.39).

Damit der Inhalt der Zelle I3 – also der Text A5 – an die Summenfunktion als Zellbezug und nicht als Text übergeben wird, müssen Sie die Funktion INDIREKT(Bezug; A1) verwenden. Diese Funktion liest einen Zellinhalt aus und gibt ihn als Zellbezug an eine andere Funktion weiter. Die Funktion, die den Bezug entgegennimmt, z. B. die Summenfunktion, kann dann – auf indirektem Weg – mit dem Zellbezug ihre Aufgabe ausführen. Fazit: Verwenden Sie also in Zelle J4 die Funktion =SUMME(INDIREKT(I3):INDIREKT(I4)), bildet Excel wie beabsichtigt die Summe aus den Werten, die im Zellbereich A5 bis A7 stehen.

Spalten oder Zeilen flexibel ansteuern und berechnen

Die Funktion INDIREKT() ist in dynamischen Auswertungen enorm wichtig und vor allem unersetzlich, da nur sie in der Lage ist, diese spezielle Umwandlung von Texten in Bezüge zu realisieren. Zudem können Sie mit ihr Kombinationen aus festen Spaltenbezeichnungen und veränderlichen Zeilen oder – genau umgekehrt – aus veränderlichen Spaltenbezeichnungen und festen Zeilennummern erstellen.

Das zweite Beispiel im Tabellenblatt *Indirekt()* I zeigt, wie das funktioniert. In den Spalten A bis D werden die Daten zu vier Kategorien wiedergegeben. Die Summe der Daten für jede Kategorie muss jeweils aus den Werten in den Zeilen 5 bis 7 gebildet werden. Die Spaltenbezeichnung muss jedoch veränderlich sein (A5 bis A7, B5 bis B7 usw.). Wenn Sie nun in eine Zelle – im Beispiel ist es Zelle G3 – den Buchstaben der Spalte eingeben, deren Summe Sie berechnen möchten, können Sie Excel dazu veranlassen, mit der Funktion

```
=SUMME(INDIREKT(G3&5):INDIREKT(G3&7))
```

die Summe für die gewünschte Spalte/Kategorie zu bilden.

Das Verknüpfungszeichen & dient in diesem Fall dazu, den variablen Teil der Zelladresse, also die Spaltenangabe aus Zelle G3, mit einem fest vorgegebenen Bestandteil, der Zeilennummer, zu verbinden. Im Ergebnis haben Sie nun die Möglichkeit, die Berechnung in einem Tabellenblatt über eine Tastatureingabe zu steuern.

Das Verfahren sähe kaum anders aus, wenn die Berechnung der Summe nicht von Spalte zu Spalte, sondern zeilenweise verschoben werden sollte. In diesem Fall wäre die Spaltenbezeichnung als fester Bestandteil mit einer veränderlichen Zeilennummer kombinierbar. Dabei entstünde eine Funktion, die beispielsweise so aussieht:

```
=SUMME(INDIREKT("A"&L3):INDIREKT("D"&L3))
```

Der einzige beachtenswerte Unterschied besteht darin, dass Spaltenbezeichnungen als Text und somit in diesem Fall mit Anführungsstrichen eingegeben und verknüpft werden müssen.

Fehlervermeidung durch Eingabebeschränkungen

Das nächste Fallbeispiel im Tabellenblatt *Indirekt()* II geht in der Anwendung der Funktion lediglich einen kleinen Schritt weiter. Es zeigt Ihnen eine Kombination aus INDIREKT() und Datenüberprüfung (Abbildung 8.40). Denn das Risiko der Steuerung einer Funktion und Kalkulation über eine Dateneingabe in eine Zelle des Tabellenblattes liegt natürlich immer in der möglichen Fehleingabe durch den Benutzer.

| Produkt | ProdGruppe | Januar | Februar | März | April | Mai | Juni | Juli | August | September | Oktober | November | Dezember | Ist | Abweichung |
|-------------|------------|----------|----------|----------|----------|----------|----------|----------|--------|-----------|---------|----------|----------|-----------|------------|
| Produkt 159 | Ist B | 28.717 € | 34.919 € | 51.831 € | 45.371 € | | | | | | | | | 160.838 € | |
| | Soll | 25.000 € | 33.000 € | 53.000 € | 48.000 € | 45.000 € | 50.000 € | 56.000 € | | | | | | 204.000 € | 27% |
| Produkt 627 | Ist D | 30.633 € | 47.884 € | 23.089 € | 53.740 € | | | | | | | | | 155.336 € | |
| | Soll | 28.000 € | 50.000 € | 30.000 € | 34.000 € | 40.000 € | 42.000 € | 40.000 € | | | | | | 182.000 € | 17% |
| Produkt 116 | Ist C | 54.573 € | 25.598 € | 30.989 € | 23.045 € | | | | | | | | | 134.205 € | |
| | Soll | 55.000 € | 25.000 € | 32.000 € | 25.000 € | 34.000 € | 30.000 € | 35.000 € | | | | | | 171.000 € | 27% |
| Produkt 256 | Ist C | 48.273 € | 53.927 € | 36.514 € | 43.874 € | | | | | | | | | 182.588 € | |
| | Soll | 50.000 € | 54.000 € | 45.000 € | 53.000 € | 40.000 € | 23.000 € | 34.000 € | | | | | | 242.000 € | 33% |
| Produkt 247 | Ist D | 40.471 € | 36.920 € | 23.904 € | 39.258 € | | | | | | | | | 149.453 € | |
| | Soll | 45.000 € | 40.000 € | 30.000 € | 54.000 € | 34.000 € | 23.000 € | 43.000 € | | | | | | 208.000 € | 36% |

Abbildung 8.40 Die per Datenüberprüfung gewählte Spalte wird mit »INDIREKT()« weiterverarbeitet.

In diesem Tabellenblatt sollen die Plandaten mit den Ist-Daten verglichen werden. Ihre Plandaten stehen bereits für einen längeren Zeithorizont fest. Aber monatlich kommen neue Ist-Daten hinzu. Die Länge der Ist-Datenreihe verändert sich also kontinuierlich. Um eine fundierte Aussage bei Ihrem Soll-Ist-Vergleich zu erhalten, müssen Sie das Soll von Januar bis April mit dem Ist des gleichen Zeitraumes vergleichen. Sobald jedoch die Daten für Mai vorliegen, muss sich der Vergleich auf diesen Zeitraum beziehen.

Die dynamische Anpassung der Funktion erfolgt wieder durch die Eingabe des Spaltenbuchstabens und mithilfe der Funktion INDIREKT(). In Zelle P3 befindet sich die Funktion =SUMME(D3:INDIREKT(S1&3)). Die Spalte, bis zu der die Summe berechnet werden soll, wird aus Zelle S1 übernommen. Doch in S1 wird der Spaltenbuchstabe mit einer Datenüberprüfung, die über DATEN • DATENTOOLS • DATENÜBERPRÜFUNG eingefügt wurde, ausgewählt. So verhindern Sie, dass folgenschwere Fehleingaben in dieser Zelle möglich sind.

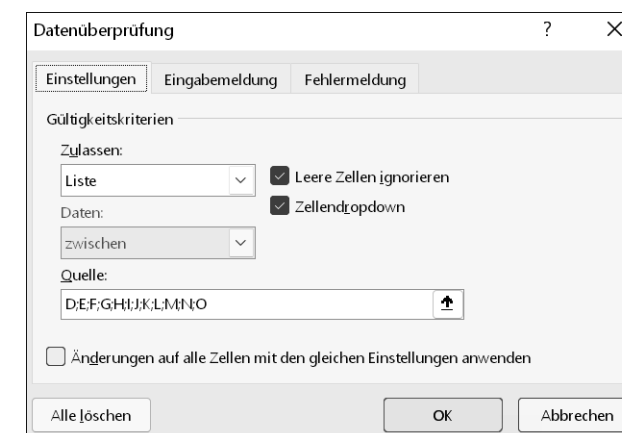


Abbildung 8.41 Eingabe der erlaubten Listeneinträge

Die Liste der erlaubten Spaltenbezeichnungen können Sie einfach in das Eingabefeld **QUELLE** der Dialogbox **DATENÜBERPRÜFUNG** (Abbildung 8.41), jeweils getrennt durch ein Semikolon, eingeben.

8.4.3 »INDIREKT()« zum Ansteuern von Zellen in anderen Tabellenblättern

Die Funktion **INDIREKT()** kann auch ein hervorragender Helfer sein, um durch eine Auswahl des Benutzers auf Zellen unterschiedlicher Tabellenblätter zuzugreifen. Das Tabellenblatt *Indirekt() III* der Beispieldatei *O8_Dynamisierung_INDIREKT_01.xlsx* zeigt das. Stellen Sie sich etwa vor, dass Sie in einem Dashboard mal die Monate des einen und dann wieder des anderen Monats auswählen möchten. Sie geben den jeweiligen Namen des Tabellenblattes, das die gewünschten Daten enthält, in eine Zelle ein oder wählen den Namen mit einer Datenüberprüfung aus. Und **INDIREKT()** kombiniert diesen Namen mit dem restlichen Zellbezug.

Ein Bezug auf ein Tabellenblatt *Mai* sähe in Excel so aus: `=Juni!B5`. Die Lösung zur Flexibilisierung des Tabellenblattbezugs würde folgendermaßen lauten (Abbildung 8.42):

`=INDIREKT(A2&"!B"&5)`

| | | | | |
|---|-------|---|-----------------------|---|
| | A | B | C | D |
| 1 | Monat | | Ergebnis | |
| 2 | Juni | | =@INDIREKT(A2&"!B"&5) | |
| 3 | | | INDIREKT(Bezug: [A1]) | |

Abbildung 8.42 Zugriff auf ein Tabellenblatt mit »INDIREKT()«

Voneinander abhängige Datenüberprüfungen mit »INDIREKT()« erstellen

Die Option Datenüberprüfung passt auch zum nächsten Beispiel, der Datei *O8_Dynamisierung_INDIREKT_Datenüberprüfung_01.xlsx*. In ihr sollen zwei Listen, die mit einer Datenüberprüfung abgerufen werden, zueinander in Beziehung gesetzt werden. Wird aus der ersten Liste z. B. das Vertriebsgebiet *Nord* ausgewählt, sollen in der zweiten Liste nur noch die zu diesem Gebiet gehörigen Orte zur Auswahl angeboten werden (Abbildung 8.43).

| | A | B | C | D | E | F | G | H | I |
|---|-----------------|-----------|---|---|-----------------|-----------|------------|------------|--------------|
| 1 | Vertriebsgebiet | Ort | | | Vertriebsgebiet | Nord | West | Nordost | Mitte |
| 2 | Nord | Bremen | | | Nord | Bremen | Aachen | Berlin | Braunschweig |
| 3 | | Bremen | | | West | Flensburg | Düsseldorf | Cottbus | Hannover |
| 4 | | Flensburg | | | Nordost | Hamburg | Essen | Eberswalde | Lerthe |
| 5 | | Hamburg | | | Mitte | Kiel | Köln | Potsdam | Kassel |
| 6 | | Kiel | | | Ost | Oldenburg | | | |
| | | Oldenburg | | | | | | | |

Abbildung 8.43 Steuerung voneinander abhängiger Listen mit »INDIREKT()«

Zu den bereits dargestellten Argumenten der Funktion **INDIREKT()** tritt in diesem Beispiel eine weitere Funktion: die Benutzung eines Bereichsnamens. Dieser bildet die Grundlage, um die beiden Listen miteinander zu verbinden.

1. Markieren Sie den Zellbereich E1 bis J6, in dem sich sowohl die Namen der Vertriebsgebiete als auch die Details zu diesen Gebieten befinden.
2. Wählen Sie die Funktion **FORMELN • DEFINIERTE NAMEN • AUS AUSWAHL ERSTELLEN**. Aktivieren Sie die Option **AUS OBERSTER ZEILE** für die Namenserstellung, und klicken Sie dann auf **OK**.
3. Ordnen Sie Zelle A2 eine Datenüberprüfung zu, und wählen Sie unter **ZULASSEN** die Option **LISTE**. Im Eingabefeld **QUELLE** drücken Sie **[F3]** und wählen den Bereichsnamen *Vertriebsgebiet* aus.
4. Danach legen Sie über **FORMELN • DEFINIERTE NAMEN • NAMENS-MANAGER • NEU** einen neuen Bereichsnamen mit der Bezeichnung *VGebiete* an. In der Eingabezelle **BEZIEHT SICH AUF:** der Option **NAMENS-MANAGER** geben Sie die Funktion `=INDIREKT(A2)` ein und beenden die Definition mit **OK**.
5. Zum Abschluss erstellen Sie eine weitere Datenüberprüfung für Zelle B2. Als **QUELLE** für die Datenauswahl bestimmen Sie den Bereichsnamen *VGebiete*.

Diese Verknüpfung von **INDIREKT()** mit einem Bereichsnamen hat den Effekt, dass Excel, sobald Sie Zelle A2 mithilfe der ersten Datenüberprüfung verändern, den für die zweite Datenüberprüfung notwendigen Bereichsnamen aktualisiert. **INDIREKT()** leitet eine Texteingabe diesmal nicht an eine Kalkulationsfunktion, sondern an den **NAMENS-MANAGER** weiter. Sämtliche Funktionen in dieser Arbeitsmappe, die den von der Aktualisierung betroffenen Namen verwenden – beispielsweise die Datenüberprüfung –, werden als logische Folge ebenfalls aktualisiert.

8.4.4 Finden und Berechnen von Daten mit »INDEX()« und »VERGLEICH()«

Aufgrund der Eigenschaft, Textwerte an andere Excel-Funktionen weiterzugeben, könnte man **INDIREKT()** geradezu als *Everybody's Darling* in Excel bezeichnen. Das Verhältnis zwischen zwei anderen Funktionen muss man hingegen als wesentlich inniger bezeichnen:

- Die Funktion **VERGLEICH(Suchkriterium; Suchmatrix; Vergleichstyp)** durchsucht eine Spalte oder Zeile und gibt die Position der Fundstelle als Zahl zurück; gesucht werden kann – je nach Vergleichstyp – nach einer genauen Übereinstimmung von Suchkriterium und Fundstelle oder der nächstgrößeren oder -kleineren Zahl.
- **INDEX(Matrix; Zeile; Spalte)** lokalisiert eine Zelle in einer Tabelle durch Angabe der genauen Zeile und Spalte in Form eines numerischen Wertes; mit anderen Worten, mit **INDEX()** verlassen Sie die strenge Logik der A1-Schreibweise.

Die Arbeitsmappe *O8_Dynamisierung_INDEX_VERGLEICH_01.xlsx* enthält einige Beispiele, die veranschaulichen, wie gut die beiden Funktionen zusammenpassen. Beginnen Sie im Tabellenblatt *INDEX() + VERGLEICH()*, um sich mit der Logik der Funktionen vertraut zu

machen. Im Zellbereich A2 bis D5 befindet sich eine einfache Tabelle, deren Zeilenbeschriftungen einige Produktbezeichnungen und deren Spaltenüberschriften verschiedene Kategorien enthalten (Abbildung 8.44).

| | A | B | C | D | E | F | G | H | I | J | K |
|---|------------|-----|-----|-----|---|------------------|--------|---|---|---|---|
| 1 | Kategorien | | | | | | | | | | |
| 2 | Produkte | A | B | C | | | Inhalt | Position | | | |
| 3 | ABC | 105 | 104 | 101 | | Produkt wählen: | ABC | =VERGLEICH(G3;A3:A5;0) | | | |
| 4 | DEF | 114 | 115 | 110 | | Kategorie wählen | A | VERGLEICH(Suchkriterium; Suchmatrix; [Vergleichstyp]) | | | |
| 5 | GHI | 110 | 107 | 107 | | Zellinhalt: | | 105 | | | |

Abbildung 8.44 Ansteuern einer Zelle mit »VERGLEICH()« und »INDEX()«

In Zelle G3 können Sie eine Produktbezeichnung eingeben. Dann erhalten Sie durch die Funktion =VERGLEICH(G3;A3:A5;0) die Information, in welcher Zeile der Matrix A3 bis A5 die gesuchte Bezeichnung zu finden ist. Auf gleiche Art und Weise verfahren Sie in Zelle G4, um in der Nachbarzelle mit =VERGLEICH(G4;B2:D2;0) zu erfahren, in welcher Spalte eine von Ihnen gesuchte Spaltenüberschrift steht.

Sie erhalten also die Koordinaten, die ein bestimmtes Produkt einer ausgewählten Kategorie in der Produkttabelle, Ihrer Matrix, besitzt. Wäre es nicht eine nützliche Sache, wenn es eine Funktion gäbe, mit der Sie diese Informationen verwerten könnten? Klar! Und die Funktion, mit der Sie die Koordinaten aufgreifen, um die konkrete Zelle ansteuern und ihren Inhalt nutzen zu können, ist INDEX(Matrix; Zeile; Spalte).

| | A | B | C | D | E | F | G | H | I | J |
|---|------------|-----|-----|-----|---|------------------|--------|---------------------|---|---|
| 1 | Kategorien | | | | | | | | | |
| 2 | Produkte | A | B | C | | | Inhalt | Position | | |
| 3 | ABC | 105 | 104 | 101 | | Produkt wählen: | ABC | 1 | | |
| 4 | DEF | 114 | 115 | 110 | | Kategorie wählen | A | 1 | | |
| 5 | GHI | 110 | 107 | 107 | | Zellinhalt: | | =INDEX(B3:D5;H3;H4) | | |

Abbildung 8.45 Kombination von »INDEX()« und »VERGLEICH()«

In Zelle H5 greift =INDEX(B3:D5;H3;H4) die Werte aus den Zellen H3 und H4 auf (Abbildung 8.45). Als Ergebnis wird für das Produkt ABC in der Kategorie der Wert 105 ausgegeben.

Dynamische Beschriftungen mit »INDEX()« erstellen

Die weiteren Tabellenblätter der Beispieldatei enthalten eine typische Anwendung für die beiden gerade beschriebenen Funktionen. Im Tabellenblatt *Produktdaten* befindet sich eine Liste mit Daten, wie Sie sie z. B. per Download aus einem anderen Programm erhalten. Es handelt sich um ein Beispiel aus dem Marketing, eine Auswertung der numerischen Distribution von Produkten in verschiedenen Teilmärkten. Die Spalte *Abweichung* zeigt Ihnen, wo Sie Ihre Kapazitäten noch nicht ausgereizt haben (Abbildung 8.46). In der Spalte *ID* werden zudem die verschiedenen Marktsegmente codiert.

| | A | B | C |
|----|-----|-----------|------------|
| 1 | ID | Produkt | Abweichung |
| 2 | 100 | Produkt 1 | 1 |
| 3 | 100 | Produkt 2 | 1 |
| 4 | 100 | Produkt 3 | 1 |
| 5 | 100 | Produkt 4 | 1 |
| 6 | 100 | Produkt 5 | 1 |
| 7 | 101 | Produkt 5 | 1 |
| 8 | 101 | Produkt 2 | 1 |
| 9 | 101 | Produkt 3 | 2 |
| 10 | 101 | Produkt 4 | 1 |
| 11 | 101 | Produkt 5 | 3 |

Abbildung 8.46 Ergebnis der Analyse der numerischen Distribution

Wechseln Sie in das Tabellenblatt *Prognose*, werden Ihnen dort die neuesten Daten einer Marktanalyse geliefert. Diese Werte zeigen Ihnen, welche zusätzlichen Umsätze Sie generieren könnten, wenn Sie die Potenziale, die als Abweichung in der vorherigen Tabelle ausgewiesen wurden, nutzen würden (Abbildung 8.47). Die Formel zur Berechnung der Potenziale wäre einfach zu bilden: *Summe der Abweichungen eines Produkts * Prognosewert pro Produkt = Gesamtpotenzial des Produkts.*

| | A | B | C |
|---|-------------|----------|-----------|
| 1 | Reihenfolge | Prognose | Produkt |
| 2 | 1 | 10 | Produkt 1 |
| 3 | 2 | 20 | Produkt 2 |
| 4 | 3 | 10 | Produkt 3 |
| 5 | 4 | 5 | Produkt 4 |
| 6 | 5 | 20 | Produkt 5 |

Abbildung 8.47 Liste der Marktpotenziale laut Marktanalyse

Doch es gibt einige technische Hürden bei der Berechnung des Potenzials. In der Tabelle *Ergebnis* müssen Sie erst einmal die Summe der Abweichungen pro ID und Produkt ermitteln. Dies ist an sich kein Problem. Wenn Sie in die Zeilen die ID schreiben und Ihre Produktbezeichnungen als Spaltenüberschriften eingeben, können Sie mit SUMMEWENNS() eine bedingte Summe auf Basis der zwei Bedingungen bilden.

Da Sie in regelmäßigen Abständen die gleiche Analyse aber mit aktualisierten Downloaddaten und den Ergebnissen von neuen Marktstudien durchzuführen gedenken, sollten so gut wie alle Elemente der Berechnung dynamisch veränderbar sein. Für die Überschriften in den Zellen B1 bis F1 erreichen Sie die angestrebte Dynamisierung mit der folgenden Funktion (Abbildung 8.48):

=INDEX(Prognose!\$C\$2:\$C\$6;SPALTE()-1;1)

Diese Funktion sorgt dafür, dass als Spaltenüberschriften immer die aktuellen und fehlerfreien Produktbezeichnungen in Ihrer Berechnungstabelle eingesetzt werden, die auch in der Prognosetabelle zum Einsatz kommen. Sie sparen auf diesem Weg einerseits die Arbeit

des Kopierens und vermeiden andererseits unnötige und nur mit großem Zeitaufwand zu findende Abweichungen in der Schreibweise der Daten.

| | A | B | C | D | E | F |
|---|-----------|---|---|---|---|------------------|
| 1 | ID | =@INDEX(Prognose!\$C\$2:\$C\$6;SPALTE()-1;1) | | | | Produkt 5 |
| 2 | 100 | INDEX(Matrix; Zeile; [Spalte]) | | 1 | 1 | 1 |
| 3 | 101 | INDEX(Bezug; Zeile; [Spalte]; [Bereich]) | | 2 | 1 | 4 |
| 4 | 102 | 0 | 3 | 1 | 1 | 8 |
| 5 | 103 | 0 | 0 | 1 | 2 | 1 |

Abbildung 8.48 Dynamische Beschriftung einer Tabelle mit »INDEX()«

Verknüpfungen von Berechnungen mit »INDEX()« und »VERGLEICH()«

Welchen Zwischenstand haben wir nun zu verbuchen? Erstens: Unsere Ausgangstabelle, in der die Produkte untereinander angeordnet waren, wurde mittlerweile gedreht. Zweitens: Um die Spaltenüberschriften werden wir uns zukünftig nicht mehr kümmern müssen, da sie ohne unser Zutun auch nach jeder Datenaktualisierung dynamisch aus den Basisdaten generiert werden. Es existiert also bereits eine grundsätzliche Dynamisierung der Daten.

Doch auch bei der eigentlichen Zielsetzung, die in der Berechnung der Potenziale pro Produkt liegt, können die beiden hier erprobten Funktionen einen wichtigen Beitrag leisten. Sie helfen dabei, ein Manko von SVERWEIS() in den Griff zu bekommen: Die Funktion SVERWEIS() kann immer nur die erste Spalte einer Matrix durchsuchen. Die auszulesende Spalte muss sich stets rechts von dieser Suchspalte befinden. VERGLEICH() kann hingegen eine beliebige Spalte durchsuchen, und mit INDEX() können Werte ausgelesen werden, die sich rechts oder auch links von der Suchspalte befinden. In der Beispieldatei ginge das so (Abbildung 8.49):

=INDEX(Prognose!\$B\$2:\$C\$6;VERGLEICH(Ergebnis!H\$1;Prognose!\$C\$2:\$C\$6;0);1)*B2

| | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|------------------|------------------|------------------|------------------|------------------|---|------------------|------------------|------------------|------------------|------------------|------------|---|
| 1 | Produkt 1 | Produkt 2 | Produkt 3 | Produkt 4 | Produkt 5 | | Produkt 1 | Produkt 2 | Produkt 3 | Produkt 4 | Produkt 5 | | |
| 2 | 1 | 1 | 1 | 1 | 1 | =INDEX(Prognose!\$B\$2:\$C\$6;VERGLEICH(Ergebnis!H\$1;Prognose!\$C\$2:\$C\$6;0);1)*B2 | | | | | | | |
| 3 | 0 | 1 | 2 | 1 | 4 | INDEX(Matrix; Zeile; [Spalte]) | 20 | 5 | 80 | 125 | | | |
| 4 | 0 | 3 | 1 | 1 | 8 | INDEX(Bezug; Zeile; [Spalte]; [Bereich]) | 10 | 5 | 160 | 235 | | | |
| 5 | 0 | 0 | 1 | 2 | 1 | | 0 | 0 | 10 | 20 | 40 | | |
| 6 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | | |
| 7 | 2 | 3 | 1 | 1 | 4 | | 20 | 60 | 10 | 5 | 80 | 175 | |
| 8 | | | | | | | 30 | 160 | 60 | 30 | 360 | 640 | |

Abbildung 8.49 »INDEX()/VERGLEICH()« funktionieren hier als »SVERWEIS()« von rechts nach links.

Auch hier wird der Zellbereich C2 bis C6 mittels Vergleich auf Übereinstimmung mit einer Produktbezeichnung hin untersucht. Die ermittelte Zeilennummer wird alsdann an INDEX() übergeben und die erste Spalte der Matrix, die sich diesmal links von der Suchspalte befindet, als weitere Koordinate bestimmt. Der damit lokalisierbare Prognosewert kann nun mit der Summe aus Zelle B2 des Tabellenblattes Ergebnis multipliziert werden.

Am Ende der einzelnen Schritte erhalten Sie das Marktpotenzial je Produkt und Marktsegment. Aus allen Einzelergebnissen, die sich mit dieser kopierbaren Funktion schnell errechnen lassen, bilden Sie die Zwischenergebnisse je Produkt und Marktsegment sowie das Gesamtpotenzial aller Produkte und Teilmärkte.

Fazit zur Verwendung von »INDEX()« und »VERGLEICH()«

Die Funktion INDEX() ist schwer zu ersetzen, wenn Sie über numerische Koordinaten gezielt auf die Zellen einer Matrix zugreifen möchten. Numerische Daten erhalten Sie immer dann, wenn Sie

- ▶ mit Steuerelementen wie Kombinationsfeldern oder Optionsfeldern arbeiten oder
- ▶ einen Tabellenbereich mit der Funktion VERGLEICH() durchsuchen.

INDEX()/VERGLEICH() sind in Kombination in der Lage, den SVERWEIS() zu ersetzen. Letzteres ist vor allem dann bedeutsam, wenn sich aufgrund der Datenstruktur die zu durchsuchende Spalte rechts von der Ergebnisspalte befindet und der SVERWEIS() aus diesem Grund nicht anwendbar ist. Wie bereits beschrieben, kann in den neueren Versionen von Excel auch die Funktion XVERWEIS() eingesetzt werden, da sie mit Rückgabebereichen links wie rechts vom Suchbereich mühelos zurechtkommt. Diese neue Funktion lässt sich allerdings nicht so einfach und flexibel mit Steuerelementen kombinieren.

Insgesamt lassen sich also folgende Vorteile von INDEX()/VERGLEICH() festhalten:

- ▶ höhere Rechengeschwindigkeit
- ▶ Nachschlagen von Werten in alle vier Richtungen
- ▶ einfache Kombinierbarkeit mit anderen Werkzeugen der Dynamisierung (z. B. Steuerelementen)

Datenüberprüfungen und dynamische Datentabellen

Im Laufe dieses Abschnitts habe ich ein Beispiel beschrieben, bei dem BEREICH.VERSCHIEBEN() mit einem Kombinationsfeld verbunden wurde, um einen Tabelleninhalt anzusteuern und das Ergebnis der darin gespeicherten Werte zu berechnen. Kombinationsfelder liefern durch die Auswahl eines Listeneintrags immer einen numerischen Ergebniswert, den Sie dann z. B. durch Funktionen wie BEREICH.VERSCHIEBEN() weiterverarbeiten können.

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|----------------|-------------------|-------------|-------------|-------------|-------------|---|---|---|--------------------------|------------------------------------|---|---|
| 1 | Produkt | ProdGruppe | Nord | West | Süd | Ost | | | | | | | |
| 2 | Produkt 159 | A | 28.717,00 € | 34.919,00 € | 51.831,00 € | 45.371,00 € | | | | | | | |
| 3 | Produkt 627 | D | 30.613,00 € | 47.884,00 € | 23.089,00 € | 53.740,00 € | | | | Summe für Produkt | | | |
| 4 | Produkt 116 | C | 54.573,00 € | 25.598,00 € | 30.989,00 € | 23.045,00 € | | | | Produkt 212 | =SUMME(BEREICH.VERSCHIEBEN(| | |
| 5 | Produkt 256 | C | 48.273,00 € | 53.927,00 € | 36.514,00 € | 43.874,00 € | | | | | INDIREKT("\$C\$"&VERGLEICH(\$J\$3; | | |
| | | | | | | | | | | | \$A\$1:\$A\$21;0));;1;4)) | | |

Abbildung 8.50 Auswahl von Daten mit einer Datenüberprüfung und dynamische Berechnung des gewählten Zellbereichs

Was ist jedoch zu tun, wenn keine numerischen Koordinaten vorliegen, ein Tabelleninhalt aber dennoch ausgewählt und berechnet werden soll? Die Problematik und eine mögliche Lösung lassen sich am Beispiel der Arbeitsmappe O8_Dynamisierung_INDIREKT_VERGLEICH_01.xlsx gut nachvollziehen.

In Zelle J3 befindet sich eine Datenüberprüfung, die ihre Werte aus dem Zellbereich A2 bis A21, also aus den Produktbezeichnungen, bezieht. Die Auswahl eines Eintrags aus der Liste führt nicht – wie bei Formularsteuerelementen – zur Anzeige eines numerischen Wertes in einer verknüpften Zelle. Stattdessen wird in der betreffenden Zelle der konkrete Zellinhalt, die Produktbezeichnung selbst, angezeigt (Abbildung 8.50).

Dies führt dazu, dass die Funktion INDEX() in diesem Beispiel nicht oder nur über Umwege anwendbar wäre. Eine Alternative zu dieser Funktion besteht jedoch in einer Kombination aus BEREICH.VERSCHIEBEN(), INDIREKT() und VERGLEICH(), da sich auch hier die Funktionen zur Dynamisierung von Tabellen wieder gegenseitig ergänzen. Im Mittelpunkt der Bestimmung eines veränderbaren Bereichs steht die folgende Kombination:

```
BEREICH.VERSCHIEBEN(INDIREKT("$C"&VERGLEICH($J$3;$A$1:$A$21;0));;1;4)
```

- ▶ Um den Startpunkt für den dynamischen Bereich zu definieren, wird mit INDIREKT() eine Kombination aus der Spaltenbezeichnung "\$C" und dem mit VERGLEICH() ermittelten Zeilenwert des ausgewählten Produkts gebildet.
- ▶ Diese Kombination wird an BEREICH.VERSCHIEBEN() übergeben.
- ▶ Die Höhe des veränderbaren Bereichs wird mit 1 angegeben.
- ▶ Die Breite ist ebenfalls konstant, nämlich 4 Spalten.

Wenn Sie diesen Ausdruck als Zellbezug von SUMME() verwenden, erhalten Sie eine benutzergesteuerte Berechnung der einzelnen Produkte. Die Beschreibung zur Erstellung des in diesem Beispiel verwendeten dynamischen Diagramms finden Sie in Kapitel 16, »Reporting mit Diagrammen und Tabellen«.

8.4.5 Auswahl von Berechnungsalternativen – »WAHL()« statt »WENN()«

Die Durchführung und Steuerung von alternativen Berechnungen in einem Tabellenblatt führt in den meisten Fällen zur Verwendung der Funktion WENN(Prüfung, Dann_Anweisung; Sonst_Anweisung). Liegen nur zwei Alternativen vor, ist die Benutzung dieser logischen Funktion auch weitestgehend unkritisch. Aber schon eine dritte Anweisungsalternative führt dazu, dass mehrere WENN()-Anweisungen ineinander verschachtelt werden müssen. Zwar sind seit Excel 2007 insgesamt bis zu 64 Ebenen der Verschachtelung von Funktionen möglich. Doch ist es niemandem zu wünschen, sich mit den Hunderten daraus resultierender Semikola und Klammern herumschlagen zu müssen.

Wo immer es möglich ist, Vereinfachungen einzuführen und Funktionsargumente zu reduzieren, sollten Sie diese Gelegenheit auch nutzen. Eine wesentliche Vereinfachung gegenüber verschachtelten WENN()-Funktionen bei der Ausführung von alternativen Berechnungen bietet die Funktion WAHL(Index; Wert1, Wert2 ...). Mit Index fragen Sie einen fortlaufenden numerischen Index ab, also z. B. die Abfolge der Zahlen von 1 bis 50. Für jeden der 50 Werte können Sie dann eine Anweisung definieren, die von der Funktion ausgeführt

wird. Da die Anweisungen nur durch ein Semikolon getrennt werden müssen, ist die Definition der Funktion erheblich leichter als eine WENN()-Funktion mit 49 Ebenen.

| | A | B | C | D | E | F | G | H | I |
|---|--------------------|----------------|----------------|--------------|-----------------|--------------------------------|---|-----------------|---|
| 1 | Personal-ID | Stunden | Honorar | Tarif | Zuschlag | Gesamt | | Zuschlag | |
| 2 | P001 | 167 | 4.175 € | 1 | 0 | 4.175 € | | 0 € | |
| 3 | P002 | 167 | 5.344 € | 2 | 25 | 5.369 € | | 25 € | |
| 4 | P003 | 84 | 1.764 € | 1 | 0 | 1.764 € | | 75 € | |
| 5 | P004 | 112 | 2.800 € | 1 | 0 | 2.800 € | | | |
| 6 | P005 | 112 | 2.464 € | 3 | 75 | =WAHL(D6;\$H\$2;\$H\$3;\$H\$4) | | | |
| 7 | P006 | 84 | 2.352 € | 2 | 25 | | | | |
| 8 | P007 | 167 | 5.344 € | 3 | 75 | 5.419 € | | | |

Abbildung 8.51 Drei und mehr Zuschlagsstufen können Sie mit »WAHL()« zuordnen.

Die Arbeitsmappe *08_Dynamisierung_WAHL_01.xlsx* beschreibt zwei typische Anwendungsbeispiele für die Funktion. Das Tabellenblatt *WAHL()*, das Sie in Abbildung 8.51 sehen, zeigt Teile einer Honorarliste und eine Auswahl von drei möglichen Zuschlagszahlungen, die abhängig von der jeweiligen Tarifgruppe gezahlt werden. In Zelle E2 ordnen Sie den Zuschlag mithilfe der Funktion =WAHL(D2;\$H\$2;\$H\$3;\$H\$4) einer Personal-ID zu. Die Aussage der Funktion ist simpel: Wenn die Tarifgruppe 1 gilt, dann verwende den Zuschlag aus Zelle H2; bei Tarifgruppe 2 benutze den in H3 stehenden Zuschlag; und wende schließlich den Zuschlag aus Zelle H4 an, wenn es sich um die Tarifgruppe 3 handelt. 254 dieser Argumente wären insgesamt möglich.

Selbstverständlich können Sie mit WAHL() nicht nur Zellinhalte zuweisen, sondern auch beliebige Berechnungen steuern. In Zelle F2 wird dies lediglich mit =WAHL(D2;C2+\$H\$2;C2+\$H\$3;C2+\$H\$4) angedeutet. Dem festgelegten Honorar aus Zelle C2 wird an dieser Stelle der von der Tarifgruppe abhängige Zuschlag hinzugefügt. In der Praxis können Berechnungen, die über WAHL() gesteuert werden, natürlich auch wesentlich komplexer sein.

»WAHL()« in Kombination mit Steuerelementen

Die definitive Voraussetzung für die Benutzung von WAHL() für die Berechnung von Alternativen ist das Vorhandensein eines Indexwertes. Diese Tatsache ist vor allem deshalb interessant, weil viele Steuerelemente, aber auch Funktionen wie VERGLEICH(), solche Indexwerte produzieren. Im Tabellenblatt *Soll-Ist* der Beispieldatei wird diese Überlegung aufgegriffen. Sie enthält einige Soll-Vorgaben in Spalte B und die dazu verfügbaren Ist-Werte in den Spalten C bis F. Um die Abweichung zwischen Soll und Ist nun für jede der vier Kalenderwochen zu ermitteln, benötigen wir vier Formeln: C2/B2-1 (Vergleich KW 1 mit Soll), D2/B2-1 (Vergleich KW 2 mit Soll), E2/B2-1 (Vergleich KW 3 mit Soll) und F2/B2-1 (Vergleich KW 4 mit Soll).

Auf konventionellem Weg würden Sie nun wahrscheinlich die vier Berechnungen in vier verschiedenen Spalten durchführen und daraus dann vier Diagramme erstellen. Mit der Funktion =WAHL(\$M\$2;C2/B2-1;D2/B2-1;E2/B2-1;F2/B2-1) in Zelle G2 können Sie die Ausgabe der Ergebnisse in einer Spalte zusammenfassen und aus den dort dargestellten Daten ein dynamisches Diagramm generieren (Abbildung 8.52), vorausgesetzt, in Zelle M2 befindet sich für die Berechnungen ein brauchbarer Indexwert.

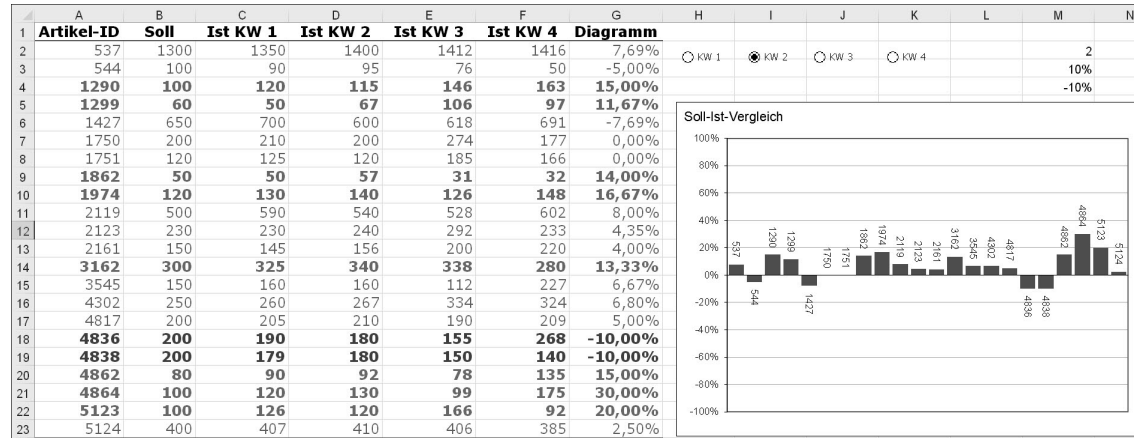


Abbildung 8.52 Auswahl von Kalenderwochen mit »WAHL()« und Optionsfeldern

Erzeugen von Indexwerten mit Steuerelementen

Diesen Indexwert können Sie natürlich in die betreffende Zelle einfach per Tastatur eingeben. Soll die erste Kalenderwoche mit dem Soll verglichen werden, tragen Sie den Wert 1 ein. Wird die zweite KW benötigt, ist es die 2. Doch auch hier sollten Sie wieder die Überlegung berücksichtigen, dass Fehleingaben zwangsläufig zu fehlerhaften Berechnungen führen und unbedingt vermieden werden müssen.

Der Einsatz von Optionsfeldern könnte sich unter diesem Gesichtspunkt lohnen (Abbildung 8.53). Sie wählen sie über ENTWICKLERTOOLS • STEUERELEMENTE • EINFÜGEN • FORMULARSTEUERELEMENTE aus (gegebenenfalls müssen Sie das Menü ENTWICKLERTOOLS zunächst in den Excel-Optionen aktivieren) und zeichnen sie in das Tabellenblatt. Wenn Sie das Steuerelement mit der rechten Maustaste anklicken, gelangen Sie unter STEUERELEMENT FORMATIEREN in das Register STEUERUNG und können dort als ZELLVERKNÜPFUNG eben die Zelle M2 angeben.



Abbildung 8.53 Optionsfelder zur Auswahl der Kalenderwochen

Das erste Optionsfeld schreibt den Wert 1 in die Verknüpfungszelle, das zweite Feld den Wert 2. Mit anderen Worten: Vier Optionsfelder reichen aus, um die vier Indexwerte in M2 zu generieren, die Sie zur Steuerung von vier alternativen Formeln in Zelle G2 benötigen.

8.5 Berechnung von Rangfolgen

Die Bildung von Rangfolgen in Excel-Arbeitsmapen kann gleich mehrere Hintergründe haben:

- ▶ Im Sinne von typischen Top-10-Listen ist es das Ziel, aus einer Fülle von Daten die Spitzenwerte – oder auch die niedrigsten Werte – auszulesen.
- ▶ Für die Benutzersteuerung mithilfe von Kombinationsfeldern stellen automatisch sortierte Listen für den Benutzer eine Erleichterung dar, wenn die Einträge der Auswahllisten nicht beliebig angeordnet sind, sondern automatisch sortiert wurden.
- ▶ Klassische Auswertungsmethoden wie die ABC-Analyse setzen die Sortierung und Bildung einer Rangfolge zwingend voraus.

Excel verfügt seinerseits über verschiedene Funktionen, die Sie bei der Bildung von Rangfolgen unterstützen. Das Angebot beginnt bereits beim Filtern von Daten. Wenn Sie die Funktion DATEN • SORTIEREN UND FILTERN • FILTERN aktivieren oder wahlweise **[Strg] + [⇅] + [L]** drücken und dann den Filter für eine Spalte setzen, die Zahlen enthält, werden Sie über die Option ZAHLENFILTER auch zur Auswahl TOP 10 gelangen (Abbildung 8.54).

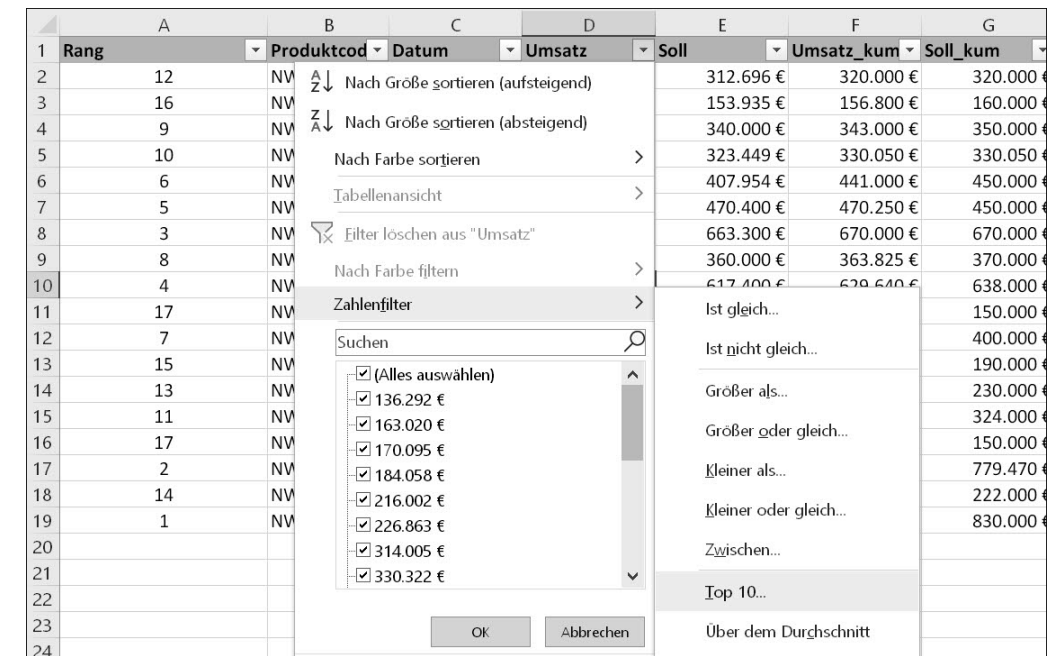


Abbildung 8.54 Top-10-Auswahl im AutoFilter

Als Ergebnis werden Sie eine Liste erhalten, die die obersten zehn Werte der Spalte enthält. Die Liste ist zunächst allerdings unsortiert. Durch die Angleichung der Benutzeroberfläche und Funktionalität von AutoFilter und Pivottabellen können Sie nach dem Erstellen einer Pivottable auf dem gleichen Weg auch dort einen Top-10-Filter nutzen.

8.5.1 Funktionen zur Bildung von Rangfolgen

Stoßen Sie fast zwangsläufig auf diese beiden Funktionen, wenn Sie Daten filtern oder zu Pivottabellen verarbeiten, sind einige der Funktionen des Funktionsassistenten, die ebenfalls bei der Bildung von Rangfolgen nützlich sind, versteckter und in der Folge auch weniger bekannt. Um diese Funktionen geht es an dieser Stelle. In der Arbeitsmappe *O8_Rangfolge_MIN_MAX_01.xlsx* werden die beiden wohl bekanntesten Funktionen dieser Art angewandt: die Funktionen zur Berechnung von Minimal- und Maximalwert.

Im Beispiel wird der Höchstwert in Zelle J1 auf Basis der Funktion `=MAX(F2:F19)` für die kumulierten Umsätze ermittelt. Auch die Berechnung des niedrigsten Wertes bedient sich dieses Wertebereiches: `=MIN(F2:F19)` (Abbildung 8.55).

| | A | B | C | D | E | F | G | H | I | J |
|----|------|------------|------------|-----------|-----------|------------|-----------|---|------------------|-----------|
| 1 | Rang | Produktcod | Datum | Umsatz | Soll | Umsatz_kum | Soll_kum | | Höchster Wert | 821.700 € |
| 2 | 12 | NWTB-1 | 31.01.2016 | 314.005 € | 312.696 € | 320.000 € | 320.000 € | | Niedrigster Wert | 148.500 € |
| 3 | 16 | NWTCO-3 | 31.01.2016 | 163.020 € | 153.935 € | 156.800 € | 160.000 € | | | |
| 4 | 9 | NWTCO-4 | 31.01.2016 | 333.185 € | 340.000 € | 343.000 € | 350.000 € | | | |
| 5 | 10 | NWTO-5 | 31.01.2016 | 339.411 € | 323.449 € | 330.050 € | 330.050 € | | | |
| 6 | 6 | NWTJP-6 | 31.01.2016 | 395.621 € | 407.954 € | 441.000 € | 450.000 € | | | |
| 7 | 5 | NWTDFN-7 | 31.01.2016 | 461.353 € | 470.400 € | 470.250 € | 450.000 € | | | |
| 8 | 3 | NWTS-8 | 31.01.2016 | 672.257 € | 663.300 € | 670.000 € | 670.000 € | | | |
| 9 | 8 | NWTDFN-14 | 31.01.2016 | 373.997 € | 360.000 € | 363.825 € | 370.000 € | | | |
| 10 | 4 | NWTFCV-17 | 31.01.2016 | 619.557 € | 617.400 € | 629.640 € | 638.000 € | | | |
| 11 | 17 | NWTBGM-19 | 31.01.2016 | 170.095 € | 148.500 € | 148.500 € | 150.000 € | | | |
| 12 | 7 | NWTJP-7 | 31.01.2016 | 401.251 € | 396.000 € | 400.000 € | 400.000 € | | | |
| 13 | 15 | NWTBGM-21 | 31.01.2016 | 184.058 € | 178.200 € | 190.000 € | 190.000 € | | | |
| 14 | 13 | NWTB-34 | 31.01.2016 | 216.002 € | 227.700 € | 230.000 € | 230.000 € | | | |
| 15 | 11 | NWTM-40 | 31.01.2016 | 330.322 € | 315.560 € | 320.265 € | 324.000 € | | | |
| 16 | 17 | NWTSO-41 | 31.01.2016 | 136.292 € | 148.500 € | 148.500 € | 150.000 € | | | |
| 17 | 2 | NWTB-43 | 31.01.2016 | 767.906 € | 779.470 € | 771.675 € | 779.470 € | | | |
| 18 | 14 | NWTCA-48 | 31.01.2016 | 226.863 € | 220.000 € | 222.000 € | 222.000 € | | | |
| 19 | 1 | NWTDFN-51 | 31.01.2016 | 812.975 € | 799.680 € | 821.700 € | 830.000 € | | | |

Abbildung 8.55 Minimal- und Maximalwert und nicht eindeutige Rangfolge

In Spalte A der Tabelle geht es dann jedoch nicht mehr um die beiden Werte am oberen bzw. unteren Ende der Skala. Hier soll stattdessen für jeden einzelnen Wert der Datenreihe die konkrete Position in der Rangfolge ermittelt werden. Um dies zu realisieren, nutzen Sie `RANG(Zahl; Bezug; Reihenfolge)`. In Zelle A2 führt dies zu den folgenden Argumenten (Abbildung 8.56):

`=RANG(F2;F2:F19)`

Bei einem Doppelklick auf Zelle A2 nach Eingabe der Bezüge wird schnell klar, was Excel zur Kalkulation der Rangfolge macht. Mit F2 wird der Wert benannt, dessen Rangfolge Sie bestimmen möchten. Durchsucht wird der gesamte Zellbereich, in dem sich Ihre kumulierten Umsatzdaten befinden. Das dritte Argument, Reihenfolge, ist optional. Wenn Sie es nicht ausdrücklich angeben oder null eingeben, wird Excel von der Rangfolge in einer absteigend sortierten Liste ausgehen. Der höchste Wert der Liste erhält somit den Wert 1. Bei Eingabe eines beliebigen anderen Wertes wird das Ergebnis auf Grundlage einer aufsteigend sortierten Liste ermittelt.

Achten Sie darauf, den Bezug auf den zu analysierenden Wert relativ und den auf den gesamten Wertebereich absolut zu setzen. Danach können Sie die Funktion einfach nach unten kopieren.

| | A | B | C | D | E | F | G |
|----|--------------------------------------|------------|------------|-----------|-----------|------------|-----------|
| 1 | Rang | Produktcod | Datum | Umsatz | Soll | Umsatz_kum | Soll_kum |
| 2 | =RANG(F2;\$F\$2:\$F\$19) | NWTB-1 | 31.01.2016 | 314.005 € | 312.696 € | 320.000 € | 320.000 € |
| 3 | RANG(Zahl; Bezug; [Reihenfolge]) >-3 | | 31.01.2016 | 163.020 € | 153.935 € | 156.800 € | 160.000 € |
| 4 | 9 | NWTCO-4 | 31.01.2016 | 333.185 € | 340.000 € | 343.000 € | 350.000 € |
| 5 | 10 | NWTO-5 | 31.01.2016 | 339.411 € | 323.449 € | 330.050 € | 330.050 € |
| 6 | 6 | NWTJP-6 | 31.01.2016 | 395.621 € | 407.954 € | 441.000 € | 450.000 € |
| 7 | 5 | NWTDFN-7 | 31.01.2016 | 461.353 € | 470.400 € | 470.250 € | 450.000 € |
| 8 | 3 | NWTS-8 | 31.01.2016 | 672.257 € | 663.300 € | 670.000 € | 670.000 € |
| 9 | 8 | NWTDFN-14 | 31.01.2016 | 373.997 € | 360.000 € | 363.825 € | 370.000 € |
| 10 | 4 | NWTFCV-17 | 31.01.2016 | 619.557 € | 617.400 € | 629.640 € | 638.000 € |
| 11 | 17 | NWTBGM-19 | 31.01.2016 | 170.095 € | 148.500 € | 148.500 € | 150.000 € |
| 12 | 7 | NWTJP-7 | 31.01.2016 | 401.251 € | 396.000 € | 400.000 € | 400.000 € |
| 13 | 15 | NWTBGM-21 | 31.01.2016 | 184.058 € | 178.200 € | 190.000 € | 190.000 € |
| 14 | 13 | NWTB-34 | 31.01.2016 | 216.002 € | 227.700 € | 230.000 € | 230.000 € |
| 15 | 11 | NWTM-40 | 31.01.2016 | 330.322 € | 315.560 € | 320.265 € | 324.000 € |
| 16 | 17 | NWTSO-41 | 31.01.2016 | 136.292 € | 148.500 € | 148.500 € | 150.000 € |
| 17 | 2 | NWTB-43 | 31.01.2016 | 767.906 € | 779.470 € | 771.675 € | 779.470 € |
| 18 | 14 | NWTCA-48 | 31.01.2016 | 226.863 € | 220.000 € | 222.000 € | 222.000 € |
| 19 | 1 | NWTDFN-51 | 31.01.2016 | 812.975 € | 799.680 € | 821.700 € | 830.000 € |

Abbildung 8.56 Datenbereich bei Verwendung der Funktion »RANG()«

8.5.2 Eindeutige Rangfolge bei identischen Werten der Liste

In dieser Beispieltabelle wird bereits ein charakteristisches Problem bei der Benutzung von `RANG()` offenbar: Die Liste kann gleichartige Werte enthalten. Ist dies der Fall, liefert Excel für diese Werte zwangsläufig auch den gleichen Rang. In den Zellen 11 und 16 ist dies erkennbar. In beiden Fällen beträgt der Wert der kumulierten Umsätze 148.500, was Rang 17 in der gesamten Liste entspricht.

Wenn dies auch rechnerisch korrekt ist, verursacht die Tatsache, dass keine eindeutig unterscheidbaren Werte vorliegen, bei der Weiterverarbeitung mit anderen Funktionen, wie z. B. `SVERWEIS()`, Probleme. Aus diesem Grund ist es notwendig, eine Eindeutigkeit der ursprünglichen Werte und damit auch der Rangfolge zu erzwingen.

Wie Sie dies umsetzen können, sehen Sie in der Arbeitsmappe *O8_Rangfolge_RANG_KGRÖSSTE_KKLEINSTE_01.xlsx*.

»RANG.GLEICH()« und »RANG.MITTELW()«

Diese beiden Funktionen gibt es bereits seit Excel 2010. `RANG.GLEICH(Zahl; Bezug; Reihenfolge)` entspricht dem bereits aus früheren Versionen bekannten `RANG()`. Kommt ein Wert zweimal in einer Liste vor, wird für jede Zahl derselbe Rang ausgegeben, z. B. Rang 14. Rang 15 entfiel dann zwangsläufig, und die Liste würde stattdessen mit 16 fortgesetzt.



Bei Verwendung der Funktion RANG.MITTELW(Zahl; Bezug; Reihenfolge) kommt hingegen ein Korrekturfaktor zur Anwendung. Aus Rang 14 wird dadurch 13,5. Bei einer Auswertung erkennen Sie so mühelos, dass dieser Rang zweimal belegt wurde. Auch in diesem Fall wird die Rangfolge jedoch mit 16 fortgesetzt.

Aus Gründen der Kompatibilität blieb RANG() in der Funktionsliste von Excel erhalten. Sollten Sie jedoch keine Dateien mit Nutzern älterer Versionen austauschen, rät Microsoft zur Verwendung von RANG.GLEICH().

Die kalkulatorische Bestimmung der eindeutigen Rangfolge von Werten kommt nicht ohne die Bildung einer Hilfsspalte aus. Das Verfahren ist jedoch einfach, da den ursprünglichen Werten ein eindeutiger Wert im Nachkommastellenbereich angehängt wird. Das klingt komplizierter, als es in der Praxis wirklich ist, da diese Ergänzung durch die Funktion ZEILE() automatisiert werden kann (Abbildung 8.57).

| | A | B | C | D |
|----|--------|-------------|------------|--------|
| 1 | Rang | Produktcode | Umsatz_kum | in % |
| 2 | 1,019 | NWTDFN-51 | 821.700 € | 11,78% |
| 3 | 2,017 | NWTB-43 | 771.675 € | 11,06% |
| 4 | 3,008 | NWTS-8 | 670.000 € | 9,60% |
| 5 | 4,010 | NWTCFV-17 | 629.640 € | 9,02% |
| 6 | 5,007 | NWTDFN-7 | 470.250 € | 6,74% |
| 7 | 6,006 | NWTJP-6 | 441.000 € | 6,32% |
| 8 | 7,012 | NWTJP-7 | 400.000 € | 5,73% |
| 9 | 8,009 | NWTDFN-14 | 363.825 € | 5,21% |
| 10 | 9,004 | NWTCO-4 | 343.000 € | 4,92% |
| 11 | 10,005 | NWTO-5 | 330.050 € | 4,73% |
| 12 | 11,015 | NWTCM-40 | 320.265 € | 4,59% |
| 13 | 12,002 | NWTB-1 | 320.000 € | 4,59% |
| 14 | 13,014 | NWTB-34 | 230.000 € | 3,30% |
| 15 | 14,018 | NWTCA-48 | 222.000 € | 3,18% |
| 16 | 15,013 | NWTBGM-21 | 190.000 € | 2,72% |
| 17 | 16,003 | NWTCO-3 | 156.800 € | 2,25% |
| 18 | 17,011 | NWTBGM-19 | 148.500 € | 2,13% |
| 19 | 17,016 | NWTSO-41 | 148.500 € | 2,13% |

Abbildung 8.57 Bildung einer eindeutigen Rangfolge mithilfe von »ZEILE()«

Diese Funktion liefert die Zeilennummer der aktuellen Zeile. Kopieren Sie sie nach unten, erhalten Sie eine fortlaufende Nummerierung. Teilen Sie das Ergebnis beispielsweise durch 10.000, in der Form =ZEILE()/10000, resultiert daraus ein eindeutiger Wert in der vierten Nachkommastelle, dem Sie als Unterscheidungsmerkmal den Originalwert hinzufügen.

8.5.3 Eindeutige Rangfolge berechnen

Es kommt auf die konkrete Situation und die Weiterverwendung der Daten an, ob Sie Ihren Umsatzzahlen den Ausdruck ZEILE()/10000 zuschlagen und dann die Rangfolge berechnen

oder erst das Ergebnis der Rangfolge mit der Funktion ZEILE()/10000 in einen eindeutigen Wert umwandeln. In der Beispielarbeitsmappe finden Sie beide Anwendungen. Im Tabellenblatt *transponiert* wird in Spalte A die Rangfolge auf Basis der kumulierten Umsätze gebildet. In den Zeilen 11 und 16 würde dies jeweils zum Rang 17 als Ergebnis führen, da der Wert 150.000 in Spalte G zweimal vorkommt.

In Zelle A2 können Sie nun mit =RANG(F2;\$F\$2:\$F\$19)+ZEILE()/1000 für eine eindeutige Rangfolge sorgen und die ursprünglichen Daten unverändert lassen. In den Zeilen 11 und 16 erhalten Sie in der Folge die Werte 17,011 und 17,016. Diese beiden Werte könnten problemlos in Verweisfunktionen wie dem SVERWEIS() ausgewertet werden. Es besteht keine Gefahr mehr, dass SVERWEIS() durch das mehrmalige Vorkommen von Rang 17 durcheinandergerät.

| | A | B | C | D |
|---|---|-------------|------------|--------|
| 1 | Rang | Produktcode | Umsatz_kum | in % |
| 2 | =KKLEINSTE(transponiert!\$A\$2:\$A\$19;ZEILE()-1) | | | |
| 3 | 2,017 | NWTB-43 | 771.675 € | 11,06% |
| 4 | 3,008 | NWTS-8 | 670.000 € | 9,60% |
| 5 | 4,010 | NWTCFV-17 | 629.640 € | 9,02% |
| 6 | 5,007 | NWTDFN-7 | 470.250 € | 6,74% |

Abbildung 8.58 Erstellen einer aufsteigenden Sortierung mit »KKLEINSTE()«

Im Tabellenblatt *sortiert* wird dies am Beispiel einer automatischen Sortierung auf Basis der berechneten eindeutigen Rangfolge sichtbar. Um die Liste auch nach dem Aktualisieren von Daten automatisch zu sortieren, benötigen Sie zunächst die gleiche Rangfolge wie in der Ursprungstabelle. Diese Werte der eindeutigen Rangfolge erhalten Sie am schnellsten, indem Sie die Daten nicht eingeben, sondern von Excel berechnen lassen:

=KKLEINSTE(transponiert!\$A\$2:\$A\$19;ZEILE()-1)

Die Funktion KKLEINSTE(Matrix; k) ermittelt einen spezifischen Wert aus einer angegebenen Matrix. An welcher Position der Wert stehen soll, wird durch das Argument k bestimmt. Möchten Sie also auf den niedrigsten Wert zugreifen, wäre das Argument k auf 1, für den zweitniedrigsten Wert auf 2 zu setzen. Um dieses Argument nicht in jede Zeile eingeben zu müssen, setzen Sie erneut die Funktion ZEILE() in die Funktion ein. Da Ihre Daten in der zweiten Zeile unterhalb der Überschrift beginnen, erhalten Sie mit ZEILE()-1 den Wert 1 für das Argument k (Abbildung 8.58). Die gesamte Funktion kopieren Sie dann wie gewohnt nach unten. Nun können Sie den Produktcode und die kumulierten Umsätze per SVERWEIS() zuordnen, ohne befürchten zu müssen, dass durch das Vorhandensein identischer Ränge die Zuordnung der Daten fehlerhaft ist. Die beiden Spalten B und C enthalten somit die Funktionen

=SVERWEIS(A2;transponiert!\$A\$1:\$G\$19;2;FALSCH)

und

=SVERWEIS(A2;transponiert!\$A\$1:\$G\$19;6;FALSCH)

8.5.4 Eindeutige Ursprungsdaten erzeugen

Das Pendant der Funktion `KKLEINSTE()` ist – und dies ist nicht schwer zu erraten – `KGRÖSSTE()`. Beide Funktionen sind ideal, um benutzerdefinierte und beliebig formatierbare Top-10-, Top-5-, Last-3-Listen und Ähnliches zu generieren. Wenn Ihnen also die ganz zu Beginn dieses Abschnitts vorgestellten Top-10-Funktionen im AutoFilter oder in der Pivottable nicht bei der Auswertung der Daten reichen, sind diese beiden Funktionen unersetzlich.

Im Tabellenblatt *Top 5 – Last 5* sind die fünf höchsten und die fünf niedrigsten Ergebnisse aus der Spalte der kumulierten Umsätze im Tabellenblatt *transponiert* aufgelistet. An den beiden Tabellen wird das praktische Problem sogleich sichtbar (Abbildung 8.59).

| | A | B | C | D | E | F | G |
|---|------------|--------------------|----------------------|---|-------------|--------------------|----------------------|
| 1 | Top | Produktcode | Umsatz (kum.) | | Flop | Produktcode | Umsatz (kum.) |
| 2 | 1 | NWTDFN-51 | 821.700 € | | 1 | NWTBGM-19 | 148.500 € |
| 3 | 2 | NWTB-43 | 771.675 € | | 2 | NWTSO-41 | 148.500 € |
| 4 | 3 | NWTS-8 | 670.000 € | | 3 | NWTCO-3 | 156.800 € |
| 5 | 4 | NWTFCV-17 | 629.640 € | | 4 | NWTBGM-21 | 190.000 € |
| 6 | 5 | NWTDFN-7 | 470.250 € | | 5 | NWTCA-48 | 222.000 € |

Abbildung 8.59 Top 5 und Last 5 mit »KGRÖSSTE()« und »KKLEINSTE()«

Im Zellbereich A2 bis A6 sind lediglich die Werte der Ränge angegeben, die Sie darstellen möchten. In C2 müssten Sie nun eigentlich mit `=KGRÖSSTE(transponiert!F2:F19; 'Top 5 – Last 5'!A2)` den höchsten kumulierten Umsatz finden. Die Rangfolge dazu basiert aber auf den Originalwerten der kumulierten Umsätze, und darin gibt es nun einmal leider Duplikate. Es bleibt Ihnen nicht viel anderes übrig, als bei den Umsatzdaten mit `ZEILE()/10000` wieder für Eindeutigkeit zu sorgen. Im Tabellenblatt *transponiert* müssen Sie in Zelle H2 die Funktion `=F2+ZEILE()/10000` einfügen und nach unten kopieren. Auf die eindeutigen Umsatzergebnisse können Sie anschließend mit `=KGRÖSSTE(transponiert!H2:H19; 'Top 5 – Last 5'!A2)` zugreifen.

Danach stellt sich erneut die bereits oben gestellte Frage, ob mit `SVERWEIS()` oder `INDEX()` weitergearbeitet werden soll. Denn wenn Sie die Produktcodierung in Spalte B angeben möchten, basiert diese Angabe auf den eindeutigen Umsatzwerten. Diese befinden sich in der Originaltabelle allerdings in einer Spalte links der kumulierten Ergebnisse. Es bleibt Ihnen, wenn Sie den `SVERWEIS()` ausführen möchten, keine andere Wahl, als die Produktcodierung in Spalte I des Tabellenblattes *transponiert* noch einmal zu erzeugen.

Möchten Sie die unschöne Redundanz vermeiden, sollten Sie `INDEX()` und `VERGLEICH()` einsetzen, mit denen der Verweis von rechts nach links und damit ohne Veränderung der Basisdaten möglich ist. Die alternativen Berechnungen in den Zellen B9 bis B13 im Tabellenblatt *Top 5 – Last 5* gründen auf der Funktion:

```
=INDEX(transponiert!$A$2:$H$19;VERGLEICH('Top 5 – Last 5'!C9;
transponiert!$H$2:$H$19;0);2)
```

`VERGLEICH()` bestimmt die genaue Position des mit `KGRÖSSTE()` bestimmten kumulierten Umsatzes. `INDEX()` nimmt das Resultat als Zeilenangabe auf und holt sich den Inhalt der zweiten Spalte der gesamten Matrix, also die Produktcodierung. Im Zellbereich F bis F13 verfahren Sie in der gleichen Weise mit der Bildung der Liste der fünf niedrigsten Werte.

8.6 Berechnung von Mittelwerten

Bereits in Excel 2007 wurde die Auswahl an Funktionen erweitert, die für bedingte Kalkulationen eingesetzt werden können. Zu den Neuerungen gehören auch die Funktionen `MITTELWERTWENN()` und `MITTELWERTWENNS()`, mit denen Sie bedingte Mittelwerte, wahlweise mit einer oder auch mit mehreren Bedingungen, ermitteln. Doch nicht nur diese beiden Funktionen lohnen die Beschäftigung mit dem Thema Mittelwerte.

In der Datei *08_Lageparameter_Diverse_01.xlsx* sind einige typische Berechnungen rund um die sogenannten *Lageparameter* zusammengefasst (Abbildung 8.60).

| | A | B | C | D | E | F |
|----|-----------------|---------------|---------------|---|-----------------------|----------|
| 1 | Standort | Kosten | Anzahl | | Lageparameter | |
| 2 | Standort 1 | 43.451 € | 2 | | Mittelwert | 66.835 € |
| 3 | Standort 2 | 1.748 € | 3 | | Median | 37.744 € |
| 4 | Standort 3 | 44.652 € | 1 | | Modalwert | 2 |
| 5 | Standort 4 | 25.384 € | 2 | | | |
| 6 | Standort 5 | 39.324 € | 2 | | Gestutzter Mittelwert | 49.450 € |
| 7 | Standort 6 | 43.121 € | 4 | | Gewogener Mittelwert | 45.775 € |
| 8 | Standort 7 | 24.435 € | 1 | | | |
| 9 | Standort 8 | 254.909 € | 1 | | | |
| 10 | Standort 9 | 27.199 € | 2 | | | |
| 11 | Standort 10 | 471 € | 3 | | | |
| 12 | Standort 11 | 26.439 € | 2 | | | |
| 13 | Standort 12 | 35.281 € | 4 | | | |
| 14 | Standort 13 | 37.744 € | 4 | | | |
| 15 | Standort 14 | 359.200 € | 1 | | | |
| 16 | Standort 15 | 39.164 € | 2 | | | |

Abbildung 8.60 Darstellung unterschiedlicher Lageparameter in Excel

8.6.1 Mittelwert, Median, Modalwert

Aus einer Datenreihe bilden Sie mithilfe von `MITTELWERT(Zahl1; Zahl2 ...)` den einfachen Durchschnitt. In Zelle F2 ist dies, bezogen auf den Zellbereich B2 bis B16, auch geschehen. Dieser Mittelwert zeichnet sich durch einige Besonderheiten aus:

- ▶ Er ist ein *künstlicher Wert*, da der Betrag 66.835 € in keinem der aufgelisteten Standorte erreicht wird.
- ▶ Er ist anfällig für Verzerrungen, da Datenreihen Ausreißer wie die Werte der Standorte 8 und 14, aber auch die von Standort 2 und 10 enthalten können.

Diese beiden Merkmale weist der in Zelle F3 berechnete `MEDIAN(Zahl1; Zahl2 ...)` nicht auf. Er teilt eine Datenreihe in zwei Hälften und ermittelt den Wert, der genau in der Mitte liegt: `=MEDIAN(B2:B16)`. Eine wichtige Aussage im Zusammenhang mit der hier verwendeten Kostenanalyse wäre beispielsweise, dass es genau so viele Standorte gibt, deren Kosten über 37.744 € liegen, wie es Standorte mit geringeren Kostenanteilen gibt. Zudem können Sie mit Fug und Recht behaupten, dass es einen Standort gibt, der exakt den ermittelten Kostenanteil aufweist. Dies eröffnet Ihnen völlig andere Denk- und Analyseansätze als bei der Berechnung des Mittelwertes. Sie könnten etwa den Standort genauer unter die Lupe nehmen, der den Median bildet, und durch einen Vergleich mit anderen Standorten die Faktoren bestimmen, die die Kosten insgesamt stark beeinflussen.

Der Modalwert bezieht sich in der Beispieldatei, wie Sie in Abbildung 8.61 erkennen, auf die Werte in Spalte C. Hier interessiert uns, welcher Wert in der Datenreihe am häufigsten vorkommt. Die Antwort liefert die Funktion `=MODALWERT(C2:C16)`.

| | A | B | C | D | E | F | G |
|----|-----------------|---------------|---------------|---|-----------------------|---|---|
| 1 | Standort | Kosten | Anzahl | | Lageparameter | | |
| 2 | Standort 1 | 43.451 € | 2 | | Mittelwert | 66.835 € | |
| 3 | Standort 2 | 1.748 € | 3 | | Median | 37.744 € | |
| 4 | Standort 3 | 44.652 € | 1 | | Modalwert | <code>=MODALWERT(C2:C16)</code> | |
| 5 | Standort 4 | 25.384 € | 2 | | | <code>MODALWERT(Zahl1; [Zahl2]; ...)</code> | |
| 6 | Standort 5 | 39.324 € | 2 | | Gestutzter Mittelwert | 49.450 € | |
| 7 | Standort 6 | 43.121 € | 4 | | Gewogener Mittelwert | 45.775 € | |
| 8 | Standort 7 | 24.435 € | 1 | | | | |
| 9 | Standort 8 | 254.909 € | 1 | | | | |
| 10 | Standort 9 | 27.199 € | 2 | | | | |
| 11 | Standort 10 | 471 € | 3 | | | | |
| 12 | Standort 11 | 26.439 € | 2 | | | | |
| 13 | Standort 12 | 35.281 € | 4 | | | | |
| 14 | Standort 13 | 37.744 € | 4 | | | | |
| 15 | Standort 14 | 359.200 € | 1 | | | | |
| 16 | Standort 15 | 39.164 € | 2 | | | | |

Abbildung 8.61 Häufigster Wert einer Datenreihe, berechnet mit »MODALWERT()«



»MODUS.EINF()« und »MODUS.VIELF()«

Bereits in Excel 2010 wurde der Funktionsumfang um `MODUS.VIELF(Zahl1; Zahl2 ...)` ergänzt. Damit ist es nun möglich, eine korrekte Berechnung des Modalwertes durchzuführen, auch wenn mehrere Werte an der Spitze die gleiche Häufigkeit haben. `MODUS.VIELF()` ist eine Matrixfunktion. Markieren Sie also mehrere Zellen, um der Möglichkeit Rechnung zu tragen, dass es mehrere häufigste Werte geben kann. Starten Sie die Funktion, und wählen Sie den Datenbereich aus, der analysiert werden soll. Schließen Sie dann die Auswahl mit `[Strg] + [↕] + [↩]` ab. Sie erhalten als Resultat die Liste der häufigsten Werte in der Liste.

`MODUS.EINF()` verfügt über die Funktionalität der aus früheren Versionen bekannten Funktion `MODALWERT()`, die aus Gründen der Kompatibilität im Funktionsassistenten erhalten wurde.

Die Datei `08_Lageparameter_MODUS.VIELF_01.xlsx` enthält ein Beispiel für die neue Funktion.

Im Tabellenblatt *Modalwert* wurde beispielhaft eine Häufigkeitsverteilung berechnet (Abbildung 8.62). Im Zellbereich F2 bis F5 befinden sich die vier in der Liste vorkommenden Werte. In Zelle G2 steht die Funktion `=SUMMENPRODUKT((C2: C16=F2)*1)`. Sie untersucht den Listenbereich auf eine Übereinstimmung mit dem Kriterium in F2 hin. Wird diese entdeckt, multipliziert Excel den Wahrheitswert WAHR, der mit dem Wert 1 gleichzusetzen ist, mit dem in der Funktion angegebenen Faktor 1 (*1). Wenn Sie die Funktion nach unten kopieren, erhalten Sie die Häufigkeit aller Werte und stellen in diesem Beispiel fest, dass sowohl der Wert 1 als auch der Wert 2 fünfmal in der Liste vorkommen (Abbildung 8.62). Über den Modalwert wäre dies nicht zu erkennen gewesen.

| | A | B | C | D | E | F | G |
|----|-----------------|---------------|---------------|---|-------------------|---|---|
| 1 | Standort | Kosten | Anzahl | | Häufigkeit | | |
| 2 | Standort 1 | 43.451 € | 2 | | Häufigkeit von 1: | 1 | 5 |
| 3 | Standort 2 | 1.748 € | 3 | | Häufigkeit von 2: | 2 | 5 |
| 4 | Standort 3 | 44.652 € | 1 | | Häufigkeit von 3: | 3 | 2 |
| 5 | Standort 4 | 25.384 € | 2 | | Häufigkeit von 4: | 4 | 3 |
| 6 | Standort 5 | 39.324 € | 1 | | | | |
| 7 | Standort 6 | 43.121 € | 4 | | Modalwert | | 2 |
| 8 | Standort 7 | 24.435 € | 1 | | | | |
| 9 | Standort 8 | 254.909 € | 1 | | | | |
| 10 | Standort 9 | 27.199 € | 2 | | | | |
| 11 | Standort 10 | 471 € | 3 | | | | |
| 12 | Standort 11 | 26.439 € | 2 | | | | |
| 13 | Standort 12 | 35.281 € | 4 | | | | |
| 14 | Standort 13 | 37.744 € | 4 | | | | |
| 15 | Standort 14 | 359.200 € | 1 | | | | |
| 16 | Standort 15 | 39.164 € | 2 | | | | |

Abbildung 8.62 Vergleich der Ergebnisse von »MODALWERT()« und Häufigkeit

8.6.2 Gestutzter Mittelwert

Die Problematik der Ausreißer innerhalb der gemessenen Daten habe ich bereits im Zusammenhang mit der Berechnung des einfachen Mittelwertes erwähnt. Den Median habe ich als einen Ausweg aus dem Dilemma beschrieben. Excel bietet aber eine weitere Funktion mit dem gestutzten Mittelwert, um den Einfluss von Ausreißern in einer Datenreihe zu reduzieren.

Mit `=GESTUTZTMITTEL(B2:B16;13,5%)` in Zelle F6 wurde bereits im Tabellenblatt *Lageparameter* entsprechend gegengesteuert. Im ersten Argument der Funktion geben Sie wie gewohnt die Matrix an, aus der Sie den Mittelwert ermitteln möchten. Mit dem Argument Prozent sind Sie dann aber in der Lage, den Anteil an Werten zu bestimmen, der bei der Berechnung ignoriert werden soll (Abbildung 8.63).

Bei einer Datenreihe mit 15 Werten, wie sie uns in der Beispieldatei *08_Lageparameter_Diverse_01.xlsx* vorliegt, entspräche der Prozentwert von 13,5 % in etwa zwei Werten in den Ausgangsdaten. Excel streicht als Konsequenz aus dieser Vorgabe je einen Wert am Anfang und am Ende der sortierten Datenreihe. In unserem Beispiel fallen die Werte 359.200 € und 471 € aus der Kalkulation. Das Ergebnis für den Mittelwert ist nun nicht mehr 66.835 €, sondern 49.450 €.

| | A | B | C | D | E | F | G | H |
|----|-------------|-----------|--------|---|-----------------------|---------------------------------|---|---|
| 1 | Standort | Kosten | Anzahl | | Lageparameter | | | |
| 2 | Standort 1 | 43.451 € | 2 | | Mittelwert | 66.835 € | | |
| 3 | Standort 2 | 1.748 € | 3 | | Median | 37.744 € | | |
| 4 | Standort 3 | 44.652 € | 1 | | Modalwert | 2 | | |
| 5 | Standort 4 | 25.384 € | 2 | | | | | |
| 6 | Standort 5 | 39.324 € | 2 | | Gestutzter Mittelwert | =GESTUTZTMITTEL(B2:B16;13,5%) | | |
| 7 | Standort 6 | 43.121 € | 4 | | Gewogener Mittelwert | GESTUTZTMITTEL(Matrix; Prozent) | | |
| 8 | Standort 7 | 24.435 € | 1 | | | | | |
| 9 | Standort 8 | 254.909 € | 1 | | | | | |
| 10 | Standort 9 | 27.199 € | 2 | | | | | |
| 11 | Standort 10 | 471 € | 3 | | | | | |
| 12 | Standort 11 | 26.439 € | 2 | | | | | |
| 13 | Standort 12 | 35.281 € | 4 | | | | | |
| 14 | Standort 13 | 37.744 € | 4 | | | | | |
| 15 | Standort 14 | 359.200 € | 1 | | | | | |
| 16 | Standort 15 | 39.164 € | 2 | | | | | |

Abbildung 8.63 Gestutzter Mittelwert mithilfe von »GESTUTZTMITTEL()«

8.6.3 Bedingte Mittelwerte

Öffnen Sie die Datei *08_Lageparameter_BedingterMittelwert_01.xlsx*, um sich mit der Funktion zur Berechnung des bedingten Mittelwertes mit einer oder mehreren Bedingungen vertraut zu machen. In Zelle F2 befindet sich die Bedingung für die erste Berechnung. Den gewünschten Wert geben Sie mit dem Vergleichsparameter – in diesem Fall > (größer) – ein. Daneben lässt sich der bedingte Mittelwert dann unschwer mit =MITTELWERTWENN(B2:B16;F2) errechnen.

Dabei wird die Syntax von der Funktion MITTELWERTWENN(Bereich; Kriterien; Mittelwert_Bereich) verwendet. Das Argument Mittelwert_Bereich ist optional. Da im Beispiel Kriterien- und Wertebereich identisch sind, muss es auch nicht eingesetzt werden.



»AGGREGAT()« als Tool zur Unterdrückung von Fehlerwerten

Wie andere Zusammenfassungsfunktionen reagiert auch der Mittelwert sensibel, wenn eine Datenreihe Fehlerwerte wie #NV! oder #DIV/0! enthält. Häufig müssen Sie solche Fehlerwerte deshalb mit WENNFehler() ausschalten. Bei großen Datenmengen kann dies wiederum mehr Arbeit für Sie und für Excel mehr Rechenarbeit bedeuten. Prüfen Sie deshalb immer eine alternative Berechnung mit der Funktion AGGREGAT(Funktion, Optionen, Array, k).

Neben vielen anderen Einsatzbereichen ist sie auch beim Umgang mit Fehlerwerten äußerst nützlich. Die Funktion =AGGREGAT(1;6;D4:D7) beispielsweise berechnet den Mittelwert (Funktion 1) unter Ausschluss aller Fehlerwerte (Optionen 6) für den Zellbereich D4 bis D7.

Anders sieht dies schon bei der Kalkulation des Mittelwertes mit mehreren Bedingungen aus. Die Argumente und die Syntax der Funktion lauten:

MITTELWERTWENNS(Mittelwert_Bereich; Kriterien_Bereich1; Kriterien1; ...)

Alle Argumente werden in diesem Fall auch tatsächlich benötigt. In den Zellen F4 und F5 werden die beiden Kriterien erwartet. Aufgegriffen werden diese Kriterien dann folgendermaßen (Abbildung 8.64):

=MITTELWERTWENNS(B2:B16;B2:B16;F4;C2:C16;F5)

| | A | B | C | D | E | F | G | H | I | J | K |
|----|-------------|-----------|--------|---|----------------------|--------|--|---|---|---|---|
| 1 | Standort | Kosten | Anzahl | | Bedingter Mittelwert | | | | | | |
| 2 | Standort 1 | 43.451 € | 2 | | Kosten größer als: | >40000 | 149.067 € | | | | |
| 3 | Standort 2 | 1.748 € | 3 | | | | | | | | |
| 4 | Standort 3 | 44.652 € | 1 | | Kosten höher als: | >40000 | | | | | |
| 5 | Standort 4 | 25.384 € | 2 | | und Anzahl gleich: | 4 | | | | | |
| 6 | Standort 5 | 39.324 € | 2 | | | | =MITTELWERTWENNS(B2:B16;B2:B16;F4;C2:C16;F5) | | | | |
| 7 | Standort 6 | 43.121 € | 4 | | | | MITTELWERTWENNS(Mittelwert_Bereich; Kriterien1; Kriterien1; [Kriterien_Bereich2; Kriterien2] [Kriterien... | | | | |
| 8 | Standort 7 | 24.435 € | 1 | | | | | | | | |
| 9 | Standort 8 | 254.909 € | 1 | | | | | | | | |
| 10 | Standort 9 | 27.199 € | 2 | | | | | | | | |
| 11 | Standort 10 | 471 € | 3 | | | | | | | | |
| 12 | Standort 11 | 26.439 € | 2 | | | | | | | | |
| 13 | Standort 12 | 35.281 € | 4 | | | | | | | | |
| 14 | Standort 13 | 37.744 € | 4 | | | | | | | | |
| 15 | Standort 14 | 359.200 € | 1 | | | | | | | | |
| 16 | Standort 15 | 39.164 € | 2 | | | | | | | | |

Abbildung 8.64 Mittelwert mit einer bzw. mehreren Bedingungen

Nachbemerkung: Auch bei dieser Funktion ist die Anzahl der maximal verwendbaren Kriterien sehr hoch. Möglich sind insgesamt 127 Bedingungen.

Nullwerte durch leere Zellen ersetzen

Eine weitere typische Problematik bei der Verwendung von Zusammenfassungsfunktionen sind Nullwerte bzw. scheinbar leere Zellen. Taucht der Wert 0 in einer Zelle auf, wird er häufig durch eine Funktion wie =WENN(A2=0;"", A2) oder =WENN(A2=0;;A2) ersetzt. Doch die scheinbar leere Zelle, die so entsteht, ist nicht leer. Sie enthält einen Text, da eine Formel oder Funktion schlichtweg unfähig ist, nichts zurückzugeben. Spätestens wenn Sie in dem Zellbereich, in dem Sie die Nullwerte getauscht haben, mit ANZAHL2() die Anzahl der nicht leeren Zellen ermitteln, stellen Sie fest, dass die Zellen nicht leer sind.

Abhilfe kann hier jenseits von einschlägigen VBA-Makros nur ein typischer Excel-Workaround schaffen. Dabei markieren Sie zunächst den Zellbereich, in dem Sie mit WENN() Nullwerte durch Text (z. B. "") ersetzt haben. Danach drücken Sie **F5** und klicken in der folgenden Dialogbox GEHE ZU auf INHALTE. Wählen Sie die Option FORMELN, und aktivieren Sie unterhalb der Option nur die Auswahl TEXT. Nachdem Sie die Suche gestartet haben, sind nur die Zellen markiert, die einen Text enthalten – nicht solche, in denen Zahlen stehen. Wenn Sie nun **Entf** drücken, sind die Zellen wirklich leer.

8.7 Runden von Daten

Die drei Funktionen zum Runden von Werten sind weitgehend selbsterklärend, daher an dieser Stelle nur eine kurze Zusammenfassung:



RUNDEN(Zahl; Anzahl_Stellen) rundet den Inhalt einer Zelle oder auch das Ergebnis einer Berechnung auf die Anzahl der angegebenen Nachkommastellen. Bis zum Wert 4 wird ab-, danach wird aufgerundet. In der Datei *O8_Runden_AUF_ABRUNDEN_01.xlsx* wird dies am Beispiel der Getränkebestellung für eine Veranstaltung dargestellt.

Im Zellbereich C8 bis C12 wurde die Anzahl der benötigten Tassen Kaffee auf Basis der gemeldeten Teilnehmerzahlen berechnet. Das Ergebnis beläuft sich in Zelle C8 beispielsweise auf 7,2 Tassen. Da eine Kanne maximal sechs Tassen Kaffee enthält, muss nun entschieden werden, ob die Zahl der zu bestellenden Kannen auf- oder abgerundet werden soll oder ob Sie diese Entscheidung dem Programm überlassen. Wenn Sie in Zelle E8 die Funktion =RUNDEN(C8/\$D\$3;0) einsetzen, wird mathematisch auf 0 Nachkommastellen, also ganze Kaffeekannen, gerundet (Abbildung 8.65).

| | A | B | C | D | E | F |
|----|----------------------------------|------------------------|--------------------|----------------------------|---------------------|------------------|
| 1 | Kaffee- und Teebestellung | | | | | |
| 2 | Element | Anteil | Preis in € | Tassen pro Kanne | | |
| 3 | Kaffee | 60% | 2,40 | 6 | | |
| 4 | Tee | 40% | 2,10 | | | |
| 5 | Service | 30% | | | | |
| 6 | | | | | | |
| 7 | Wochentag | Zahl der Teiln. | Kaffetassen | Teetassen | Kaffeekannen | Teekannen |
| 8 | Montag | 6 | 7,2 | 4,8 | 1 | 1 |
| 9 | Dienstag | 7 | 8,4 | 5,6 | 1 | 1 |
| 10 | Mittwoch | 12 | 14,4 | 9,6 | 2 | 1 |
| 11 | Donnerstag | 16 | 19,2 | 12,8 | 3 | 2 |
| 12 | Freitag | 9 | 10,8 | 7,2 | 2 | 1 |
| 13 | | | | Summe | 9 | 6 |
| 14 | | | | Preis | 21,60 € | 12,60 € |
| 15 | | | | Service | 6,48 € | 3,78 € |
| 16 | | | | Zw.-Summe | 28,08 € | 16,38 € |
| 17 | | | | Gesamtsumme: | | 44,46 € |
| 18 | | | | UST | 19% | 8,45 € |
| 19 | | | | Summe einschl. MwSt | | 51,57 € |

Abbildung 8.65 Runden von berechneten Ergebnissen

Anders ist das in Zelle F8. Dort wird AUFRUNDEN(D8/\$D\$3;0) als Teil einer WENN()-Funktion verwendet. Der Grund dafür: Bei einer geringen Teilnehmerzahl würden die Teetrinker leer ausgehen. Bei einem angenommenen Anteil von 40 % (Zelle B4) könnte es passieren, dass ihr Anteil auf null gerundet würde, wenn Sie die Rundung Excel überlassen. Die Folge ist, dass erst ab einer Teilnehmerzahl von mindestens acht Personen mit ABRUNDEN(D8/\$D\$3;0) auch wirklich abgerundet werden kann, ohne die Teetrinker zu verärgern.

Der vollständige Ausdruck in Zelle F8 lautet:

```
=WENN(B8=6;AUFRUNDEN(D8/$D$3;0);WENN(B8=7;AUFRUNDEN(D8/$D$3;0);ABRUNDEN(D8/$D$3;0)))
```

Die Werte 6 und 7 habe ich der Übersichtlichkeit halber in diesem Beispiel als Kriterium fest in die Funktion geschrieben. Im *realen Leben* sollten Sie diese Bedingungen aber wie

gewohnt über einen Zellbezug integrieren. Die gesamte Funktion können Sie wie üblich nach unten kopieren.

8.7.1 Runden auf ganze Zehner, Hunderter oder Tausender

Wenn Sie die Absicht haben, auf ein Vielfaches eines Ausgangswertes zu runden, stehen Ihnen in Excel gleich drei Möglichkeiten zur Verfügung:

- ▶ die Funktionen RUNDEN(), ABRUNDEN() oder AUFRUNDEN()
- ▶ die Funktionen OBERGRENZE() und UNTERGRENZE()
- ▶ die Funktion VRUNDEN()

| | A | B | C | D | E |
|----|-----------------|-------------------------|--------------------------|--------------------------|-----------------------------|
| 1 | Maschine | Produktionsmenge | gerundet (Fünfer) | gerundet (Zehner) | gerundet (Hunderter) |
| 2 | Maschine 1 | 45.297 | 45.295 | 45.300 | 45.300 |
| 3 | Maschine 2 | 36.155 | 36.155 | 36.160 | 36.200 |
| 4 | Maschine 3 | 51.214 | 51.215 | 51.210 | 51.200 |
| 5 | Maschine 4 | 46.687 | 46.685 | 46.690 | 46.700 |
| 6 | Maschine 5 | 50.410 | 50.410 | 50.410 | 50.400 |
| 7 | Maschine 6 | 44.840 | 44.840 | 44.840 | 44.800 |
| 8 | Maschine 7 | 42.019 | 42.020 | 42.020 | 42.000 |
| 9 | Maschine 8 | 73.965 | 73.965 | 73.970 | 74.000 |
| 10 | Maschine 9 | 60.959 | 60.960 | 60.960 | 61.000 |
| 11 | Maschine 10 | 56.374 | 56.375 | 56.370 | 56.400 |

Abbildung 8.66 Rundung auf ein Vielfaches am Beispiel von »RUNDEN()«

In der Arbeitsmappe *O8_Runden_Vielfaches_01.xlsx* wird im Tabellenblatt *RUNDEN()* zunächst die gleichnamige Funktion angewandt (Abbildung 8.66). Lassen Sie uns mit dem Runden auf volle Zehner beginnen. In Zelle D2 wird dies mit =RUNDEN(B2/10;0)*10 umstandslos erreicht. Teilen Sie die angegebene Produktionsmenge durch 10, entfernen Sie die Nachkommastellen, indem Sie das Argument Anzahl_Stellen auf 0 setzen, und multiplizieren Sie das Resultat wiederum mit 10. Schon erhalten Sie die Rundung auf volle Zehnerwerte.

Möchten Sie auf Hunderter runden, unterscheidet sich das Grundkonzept nicht, wie Sie in Zelle E2 der Beispieldatei (=RUNDEN(B2/100;0)*100) erkennen können. Auch die beiden Funktionen AUFRUNDEN() und ABRUNDEN() würden nach dem gleichen Muster arbeiten.

Ein wenig ungewöhnlich ist lediglich der Aufbau der Rundungsfunktion, wenn es darum geht, nicht auf ein Vielfaches von 10 zu runden. Bei =RUNDEN(B2*2;-1)/2 in Zelle C2, in der auf Fünfer gerundet werden soll, erscheint das Argument Anzahl_Stellen, das mit dem Wert -1 belegt ist, auf den ersten Blick unverständlich. Sie erreichen damit aber, dass Excel auf Zehnerpotenzen – man könnte auch sagen, nicht auf die Stellen rechts, sondern auf die links vom Komma – rundet. Die Multiplikation mit dem Faktor 2 gibt Ihnen den Anlass, das Ergebnis dann wiederum durch 2 zu teilen. Und bei der Division einer Zehnerpotenz durch 2 entsteht zwangsläufig ein Vielfaches von fünf.

8.7.2 »OBERGRENZE()« und »UNTERGRENZE()«

Der Charme der beiden Funktionen OBERGRENZE() und UNTERGRENZE() liegt – bei gleichartiger Fragestellung wie oben – in der Einheitlichkeit, mit der die Argumente verwendet werden. Es gilt für Runden auf ...

- ▶ ... Fünfer: =OBERGRENZE(B2;5)
- ▶ ... Zehner: =OBERGRENZE(B2;10)
- ▶ ... Tausender: =OBERGRENZE(B2;1000)

Der unter Zahl angegebene Wert – im Beispiel der Inhalt von Zelle B2 – wird auf das kleinste Vielfache des zweiten Arguments (Schritt) aufgerundet (Abbildung 8.67).

| | A | B | C | D |
|---|-----------------|-------------------------|-------------------------------|----------------------------|
| 1 | Maschine | Produktionsmenge | Obergrenze (Fünfer) | Obergrenze (Zehner) |
| 2 | Maschine 1 | 45.297 | =OBERGRENZE(B2;5) | 45.300 |
| 3 | Maschine 2 | 36.155 | OBERGRENZE(Zahl; Schritt) ,55 | 36.160 |
| 4 | Maschine 3 | 51.214 | 51.215 | 51.220 |

Abbildung 8.67 Verwendung von »OBERGRENZE()« zum Runden auf ein Vielfaches des Ausgangswertes

Im Tabellenblatt *Obergrenze – Untergrenze* sind auch die Berechnungen mit der Funktion UNTERGRENZE() enthalten (z. B. =UNTERGRENZE(B2;5) in Zelle F2), die nach dem gleichen Schema arbeitet und den Ausgangswert auf das nächste Vielfache abrundet.

8.7.3 Runden auf ein Vielfaches mit »VRUNDEN()«

Zu guter Letzt können Sie die Lösung der gleichen Aufgaben im Tabellenblatt *VRUNDEN()* mithilfe der Funktion testen, die das Runden auf ein Vielfaches bereits in ihrem Namen trägt (Abbildung 8.68).

| | A | B | C | D |
|---|-----------------|-------------------------|--------------------------------|------------------------------|
| 1 | Maschine | Produktionsmenge | VRUNDEN() (Fünfer) | VRUNDEN() (Tausender) |
| 2 | Maschine 1 | 45.297 | =VRUNDEN(B2;5) | 45.000 |
| 3 | Maschine 2 | 36.155 | VRUNDEN(Zahl; Vielfaches) ,155 | 36.000 |
| 4 | Maschine 3 | 51.214 | 51.215 | 51.000 |

Abbildung 8.68 Rundung mit »VRUNDEN()«

Auch diese Funktion verwendet lediglich zwei Argumente: Zahl und Vielfaches. Beziehen Sie sich auf den Wert in Zelle B2, erhalten Sie mit dem Wert 5 als Argument Vielfaches die Fünfer, mit 10 die Zehner und schließlich – wen wundert es? – mit 1000 die Tausender des Ursprungswertes.