


5 Transformieren von Daten

Fortsetzung der Seiten in der 9. Auflage

5.1. Berechnen neuer Variablen

Funktionen. Die Option „Berechnen“ stellt eine umfangreiche Reihe von Funktionen zur Verwendung in numerischen Ausdrücken zur Verfügung. Sie sind alle im Auswahlfeld „Funktionen“ enthalten. Um zur gesuchten Funktion zu gelangen, muss man zunächst eine „Funktionsgruppe“ auswählen und dann im Auswahlfeld „Funktionen und Sondervariablen:“ die gewünschte Funktion suchen. U. U. muss man in diesem Auswahlfeld scrollen (beim Eintippen eines Buchstabens springt der Cursor auf die erste Funktion mit diesem Buchstaben als Anfangsbuchstaben). Die Funktion überträgt man in das Feld „Numerischer Ausdruck“, indem man sie zuerst markiert und dann  anklickt (oder durch Doppelklicken auf die Funktionsbezeichnung). Gegebenenfalls müssen noch Werte, Variablen etc. in die Funktion eingesetzt werden. Die Funktionen sind im Folgenden dargestellt. Die kursiv gedruckte Angabe zeigt das Format der Ausgabevariablen an. Ab Version 13 sind die Funktionen im Auswahlfeld in Gruppen aufgeteilt (neben der Gruppe: Alle). Dies erleichtert die Orientierung, wenn auch die Bezeichnung der Gruppen oft unglücklich gewählt ist und nicht mit der Gliederung im Hilfetext und den Handbüchern übereinstimmt. Im Folgenden wird bei der Beschreibung der Funktionen auch die Bezeichnung der Gruppe, der sie zugeordnet sind, angeführt.

① *Arithmetische Funktionen* (Gruppe: Arithmetisch).

ABS(numAusdr). *Numerisch.* Ergibt den Absolutbetrag eines numerischen Ausdrucks. *Beispiel:* ABS(-5-8) ergibt 13.

RND(numAusdr). *Numerisch.* Gibt es in drei Versionen, durch eine Zahl in Klammern unterschieden). Die erste rundet zur nächstgelegenen ganzen Zahl. *Beispiel:* 3.8 ergibt 4. Bei der zweiten kann als zweites Argument angegeben werden, wie groß der Rundungssprung sein soll. *Beispiel:* (3.85, 0.1) ergibt 3,9. Bei der dritten Variante kann zusätzlich die Zahl der verwendeten Fuzzibist angegeben werden.

TRUNC(numAusdr). *Numerisch.* Schneidet Dezimalstellen ab. Die Funktion gibt es ebenfalls in drei Versionen. Die Argumente sind die gleichen wie bei der Funktion RND.

MOD(numAusdr,modulus). *Numerisch.* Der Rest einer Division eines Arguments durch ein zweites (Modulus). *Beispiel:* MOD(930,100) ergibt 30. Die Argumente werden durch Komma getrennt.

SQRT(numAusdr). *Numerisch.* Quadratwurzel des Ausdrucks.

EXP(numAusdr). *Numerisch.* Exponentialfunktion, gleich $e^{(\text{numAusdr.})}$.

LG10(numAusdr). *Numerisch.* Logarithmus zur Basis 10.

LN(Numausdr). *Numerisch.* Natürlicher Logarithmus, Basis e.

LNGAMMA(numAusdr) *Numerisch*. Ergibt den Logarithmus der Gammafunktion. Das Argument NumAusdr muss numerisch und positiv sein.

ARSIN(numAusdr). *Numerisch*. Arkussinus. Ergebnisse in Bogenmaß.

ARTAN(numAusdr). *Numerisch*. Arkustangens. Ergebnisse in Bogenmaß.

SIN(radiant). *Numerisch*. Sinus. Argumente müssen in Bogenmaß eingegeben werden.

COS(radiant). *Numerisch*. Kosinus. Argumente müssen in Bogenmaß eingegeben werden.

Bei den numerischen Ausdrücken kann es sich um einzelne Zahlen, aber auch komplexe Ausdrücke handeln. Gewöhnlich werden auch Variablen in ihnen enthalten sein.

Beispiel: Die Werte der Variablen EINK sollen logarithmiert und in der neuen Variablen LOGEINK gespeichert werden:

Zielvariable:	Numerischer Ausdruck:
Logeink	= LG10(Eink)

Abb. 5.30. Rechnen mit einer arithmetischen Funktion

- ▷ Tragen Sie in das Eingabefeld „Zielvariable“ den neuen Variablennamen LOGEINK ein.
- ▷ Markieren Sie im Feld Funktionen die Funktion LG10(numAusdr) und übertragen Sie sie in das Feld „Numerischer Ausdruck“. Es erscheint LG10(?).
- ▷ Markieren Sie das Fragezeichen in der Funktion, markieren Sie in der Quellvariablenliste EINK und übertragen Sie die Variable in den Ausdruck.
- ▷ Bestätigen Sie mit „OK“.

Eine Funktion verlangt immer das Einsetzen von *Argumenten*. Per Voreinstellung enthält sie bei Übertragung so viele Fragezeichen wie die Mindestzahl der Argumente beträgt. Argumente trägt man ein, indem man das Fragezeichen markiert und das Argument danach eingibt. Wird (bei statistischen und logischen Funktionen) mehr als die Mindestzahl an Argumenten verwendet, fügt man die zusätzlichen Argumente durch Komma getrennt in die Argumentliste ein.

© *Statistische Funktionen* (Gruppe: Statistisch).

SUM(numAusdr,numAusdr,...). *Numerisch*. Summe der Werte über die Argumente. *Beispiel:* SUM(kredit1,kredit2) ergibt die gesamte Kreditsumme für die beiden Kredite.

MEAN(numAusdr,numAusdr,...). *Numerisch*. Arithmetisches Mittel über die Argumente.

SD(numAusdr,numAusdr,...). *Numerisch*. Standardabweichung über die Argumente.

VARIANCE(numAusdr,numAusdr,...). *Numerisch*. Varianz über die Argumente.

CFVAR(numAusdr,numAusdr,...). *Numerisch*. Variationskoeffizient über die Argumente.

MIN(wert,wert,...). *Numerisch oder String.* Kleinster Wert über alle Argumente. (Bei Stringvariablen der in alphabetischer Reihenfolge erste Wert.)

MAX(wert,wert,...). *Numerisch oder String.* Größter Wert über alle Argumente. (Bei Stringvariablen der in alphabetischer Reihenfolge letzte Wert.)

Alle Ausdrücke haben mindestens zwei durch Komma getrennte Argumente. Gewöhnlich ergeben sich diese Argumente aus Variablenwerten verschiedener Variablen. Im Unterschied zur Berechnung statistischer Maßzahlen zur Beschreibung eindimensionaler Häufigkeitsverteilungen, geht es hier um die Zusammenfassung mehrerer Argumente/Variablen jeweils eines Falles, sei es in Form einer Summe, eines arithmetischen Mittels, einer Standardabweichung usw. Deshalb sollte man genau prüfen, inwiefern dies nützlich ist. In unserem Beispiel aus der Datei VZ.SAV kann das für die Summenbildung bejaht werden. Der Gesamtbetrag mehrerer Kredite ist eine wichtige Information. Bei allen anderen Maßzahlen wäre das fraglich. Was können wir aus dem arithmetischen Mittel zweier Kredite oder aus Streuungsmaßen, die sich auf nur zwei Kredite beziehen, entnehmen? Eher nutzbringende Verarbeitungsmöglichkeiten wären schon für die Ergebnisse der Funktion MIN und MAX denkbar. Haben wir dagegen zahlreiche Messungen einer latenten Variablen, etwa eine Testbatterie bei psychologischen Tests, vorliegen, kann durchaus der Durchschnittswert ein sinnvoller zusammenfassender Wert sein. Varianz, Standardabweichung und Variationskoeffizient sind vielleicht brauchbare Maße für die Homogenität bzw. Heterogenität der verschiedenen Messungen.

Bei den Funktionen SUM bis MAX können Sie auch eine Mindestzahl gültiger Argumente angeben. Wird diese Zahl unterschritten, setzt SPSS in der Ergebnisvariablen einen System-Missing-Wert ein. Dazu werden zwischen den Funktionsnamen und der ersten öffnenden Klammer ein Punkt und die gewünschte Mindestzahl gesetzt. *Beispiel:* Sum.2(kredit1,kredit2,kredit3) berechnet nur dann eine Summe, wenn mindestens für zwei Kredite ein gültiger Wert vorliegt. Ansonsten wird ein System-Missing-Wert eingesetzt.

③ *Suchfunktionen* (Gruppe: Suchen)

RANGE(test,min,max[,min,max...]). *Logisch.* Dient dazu zu prüfen, ob ein Wert innerhalb eines oder mehrerer Bereiche liegt. „Test“ steht gewöhnlich für einen Variablennamen, „min“ und „max“ für die Grenzen des Bereiches. *Beispiel:* Es soll geprüft werden, ob jemand in die Gruppen der „Armen“ fällt. Dies sei der Fall bei einem Einkommen von 0 bis 2000 DM. Entsprechend gälte die logische Funktion: Range(EINK,0,2000). Wahr ergibt den Wert 1, nicht wahr eine 0.

ANY(test,wert,wert,...). *Logisch.* Kann die Werte wahr (= 1) und nicht wahr (= 0) annehmen. Ist wahr, wenn der Wert des ersten Arguments (gewöhnlich eines Tests oder einer Variablen) mit irgendeinem der folgenden Argumente übereinstimmt. Das erste Argument (Test) ist gewöhnlich ein Variablennamen. *Beispiel:* Es sollen aus der Datei einer Wahlumfrage auf Basis der Angaben in Variable PART_91 alle die Fälle ausgewählt werden, die irgendeine konservative Partei wählen wollen. Diese seien CDU = 2, REP = 5, DVU = 8. Entsprechend gälte: ANY(part_91,2,5,8).

INDEX(heuhaufen,nadel). *Numerisch.* Bezeichnung: Char.Index(2). Diese Funktion hilft dabei, einen bestimmten Ausdruck (Nadel) in einer Stringvariablen (Heuhaufen) zu finden. *Beispiel:* in der Stringvariablen LAND sei Deutschland in

unterschiedlicher Weise gespeichert, z.B. als „Bundesrepublik Deutschland“ und „Deutschland“. Die Funktion INDEX(„Land“,„Deutschland“) gibt in einer neuen numerischen Variablen für jeden Fall einen numerischen Wert aus, bei dem der String „Deutschland“ in der Variablen LAND an irgendeiner Stelle vorkommt. Der numerische Wert beträgt z.B. 16, wenn der gesamte String „Bundesrepublik Deutschland“ lautet. Die 16 besagt, dass das gesuchte Wort Deutschland an der 16. Stelle im gefundenen String beginnt. Alle Fälle, in denen der String nicht vorkommt, erhalten den Wert 0 zugewiesen. Die Ergebnisvariable muss numerisch sein. Bei allen Heuhaufen- Nadel-Argumenten muss der Stringausdruck „Nadel“ in Anführungszeichen eingegeben werden.

INDEX(heuhaufen,nadel,teiler). *Numerisch.* Bezeichnung: Char.Index(3). Wie die vorhergehende Funktion. Das Argument „Teiler“ kann wahlweise verwendet werden. Mit ihm kann der String „nadel“ in einzelne zu suchende Teilstrings unterteilt werden. Der ganzzahlige Wert von „teiler“ muss den String „nadel“ so teilen, dass kein Rest verbleibt.

RINDEX(heuhaufen,nadel). *Numerisch.* Bezeichnung: Char.Rindex(2). Diese Funktion hilft wie die Funktion Index dabei, einen bestimmten Ausdruck (nadel) in einer Stringvariablen (heuhaufen) zu finden. Sie ergibt einen ganzzahligen Wert für das letzte Auftreten des Strings „nadel“ im String „heuhaufen“. Das ganzzahlige Ergebnis gibt die Stelle des ersten Zeichens von „nadel“ in „heuhaufen“ an. Ergibt 0, wenn „nadel“ nicht in „heuhaufen“ vorkommt. Die Funktion RINDEX(„Land“,„Deutschland“) gibt z.B. in einer neuen numerischen Variablen für jeden Fall einen numerischen Wert aus, bei dem der String „Deutschland“ in der Variablen LAND vorkommt. Kommt Deutschland mehrmals vor, dann wird die Stelle des ersten Buchstabens vom letzten Vorkommen zum numerischen Wert. „Lieferland Deutschland Empfangsland Deutschland“ z.B. ergibt den Wert 37, weil das letzte Auftreten von Deutschland“ an Position 37 beginnt.

RINDEX(heuhaufen,nadel,teiler). *Numerisch.* Bezeichnung: Char.Rindex(3). Wie die vorhergehende Funktion. Das optionale dritte Argument „teiler“ wird von SPSS verwendet, um den String „nadel“ in einzelnen zu suchende Teilstrings zu unterteilen. Der ganzzahlige Wert von „teiler“ muss den String „nadel“ so teilen, dass kein Rest verbleibt.

Außerdem befinden sich in dieser Gruppe die Funktionen Max und Min, die auch in der Gruppe Statistisch angeboten werden und Replace, die auch unter den String-Funktionen angeboten werden.

④ *Funktionen für fehlende Werte* (Gruppe: Fehlende Werte) .

Man kann damit festlegen, dass die fehlenden Werte ignoriert werden sollen oder auch gerade nutzerdefinierte Missing-Werte oder System-Missing-Werte herausuchen. Schließlich kann man über eine Argumentliste (Variablenliste) die Zahl der fehlenden Werte oder der gültigen Werte auszählen.

VALUE(variable). *Numerisch oder String.* Überträgt die Werte einer Variablen in eine neue und löscht dabei die Definition fehlender Werte. Wurde z.B. 5 = Sonstige in der alten Variable als fehlender Wert behandelt, ist das in der neuen Variablen ein gültiger Wert.

MISSING(variable). *Logisch.* Setzt die fehlenden Werte der Argumentvariablen 1, alle anderen 0. *Beispiel:* War 0 als fehlender Wert deklariert, erhalten Fälle mit 0 eine 1, alle anderen eine 0.

SYSMIS(numvar). *Logisch.* Setzt für die System-Missing-Werte der Argumentvariablen eine 1, für alle anderen eine 0. Geht nur, wenn die Argumentvariable numerisch ist.

\$SYSMIS. Ergibt eine Variable mit systemdefinierten fehlenden Werten.

NMISS(variable,...). *Numerisch.* Zählt aus, wievielmals ein fehlender Wert in den Argumentvariablen auftritt. Minimal kann eine Variable als Argument benutzt werden, sinnvoll ist der Einsatz allerdings nur bei mehreren Argumenten. Sind z.B. drei Variablen als Argumente eingesetzt, so können 0, 1, 2, oder 3-mal Missing-Werte auftreten.

NVALID(variable,...). *Numerisch.* Zählt umgekehrt aus, wie viel Argumentvariablen einen gültigen Wert haben.

⑤ Funktionen für Datums- und Zeitvariablen.

Es gibt mehrere Gruppen von Funktionen, die speziell zum Arbeiten mit Datums und/oder Zeitangaben vorgesehen sind. Man kann aus den auf verschiedene Variablen Informationen über ein Datum/eine Zeit eine einzige neue Datums-/Zeitvariable bilden (Gruppen: Datumserstellung/Zeiterstellung) oder umgekehrt aus einer in einer Variablen gespeicherten komplexen Datums-/Zeitinformation einen Teil ausziehen und in einer neuen Variable speichern (Datums- und Zeitextraktionsfunktionen). Es ist weiter möglich aus einer Datums-/Zeitangabe den Zeitraum seit einem Referenzdatum/einer Referenzzeit ermitteln (Datums- und Zeitkonvertierungsfunktionen; ebenfalls Gruppen: Datums- oder Zeitextraktion). Schließlich kann die Differenz zwischen zwei Datums-/Zeitangaben ermittelt werden und es ist möglich zu einem Ausgangsdatum/einer Ausgangszeit eine bestimmte Zeitdauer hinzu zu addieren (Gruppe: Datumsarithmetik).

⑤ a) Datums- und Zeitaggregationsfunktionen.

Dienen dazu, in unterschiedlichen Variablen gespeicherte Datumsangaben in einer Datumsvariablen zusammenzufassen.

α) Gruppe: Datumserstellung

DATE.DMY(tag,monat,jahr). *Numerisch im SPSS-Datumsformat.* Wenn Tag, Monat und Jahr in drei verschiedenen Variablen als Integerzahlen gespeichert sind, kann man sie damit in eine neue Variable mit Datumsformat überführen. Die Jahreszahl muss größer als 1528 und vierstellig oder zweistellig angegeben sein. Bei zweistelliger Angabe wird 19 ergänzt. *Beispiel:* Vom Geburtsdatum sind der Tag in der Variablen GBTAG, der Monat in der Variablen GBMONAT und das Jahr in der Variablen GBJAHR gespeichert. In einer Variablen fasst man sie zusammen mit dem Befehl DATE.MDY(GBTAG,GBMONAT,GBJAHR).

DATE.MDY(monat,tag,jahr). *Numerisch im SPSS-Datumsformat* (in manchen Versionen fälschlich als DATE.MD angezeigt). Wie vorher, jedoch mit anderer Reihenfolge der Eingabe von Monat, Tag und Jahr. Die neue Variable muss ebenfalls in das gewünschte Datumsformat umdefiniert werden.

DATE.MOYR(monat,jahr). *Numerisch im SPSS-Datumsformat.* Dasselbe, allerdings ohne Tagesangabe.

DATE.QYR(quarter,jahr). *Numerisch im SPSS-Datumsformat.* Gleiche Voraussetzungen wie bei den vorherigen Formaten. Eingegeben werden jedoch Quartal und Jahr.

DATE.WKYR(wochenum,jahr). *Numerisch im SPSS-Datumsformat.* Ebenso, jedoch Eingabe einer Wochennummer zwischen 1 und 52 und einer Jahreszahl.

DATE.YRDAY(jahr,tagnum). *Numerisch im SPSS-Datumsformat.* Ebenso, jedoch Eingabe einer Jahreszahl und einer Tagesnummer zwischen 1 und 366.

β) Gruppe: Erstellung der Zeitdauer

Die nächsten Funktionen dienen dazu, auf verschiedene Variablen verteilte Zeitanangaben zusammenzufassen.

TIME.HMS(std,min,sek). *Numerisch im SPSS-Zeitintervall-Format (Anzeige: Time.Hms(3)).* Wenn von einer Zeitangabe Stunden, Minuten und Sekundenangaben in verschiedenen Variablen als Integerzahlen gespeichert sind, können sie in einer Zeitvariablen zusammengefasst werden. Die Variable mit den Sekundenangaben kann auch Sekundenbruchteile als Nachkommastellen enthalten. *Beispiel:* Die Variable STUNDE enthält die Stunden-, die Variable MINUTE die Minuten- und SEKUNDE die Sekundenangabe. Die Zusammenfassung erfolgt mit TIME.HMS(STUNDE,MINUTE,SEKUNDE). Bei Ausgabe in eine neue Variable muss diese in ein passendes Zeitformat umdefiniert werden.

TIME.HMS(std,min) (Anzeige: Time.Hms(2)). Ebenso, jedoch werden nur Stunden und Minuten eingegeben.

TIME.HMS(std) (Anzeige: Time.Hms(1)). Ebenso, jedoch werden nur Stunden angegeben.

TIME.DAYS(Tage). *Numerisch im SPSS-Zeitintervall-Format.* Eine Tagesangabe wird in ein Zeitintervall umgerechnet. Die neue Variable muss in ein passendes Zeitformat umdefiniert werden.

Die Umwandlung einer einzigen Angabe wie einer Stunden- oder Tagesangabe kann aus verschiedenen Gründen sinnvoll sein. So kann das neue Format genauere Angaben, wie die Angabe von Sekundenbruchteilen zulassen. Außerdem werden Zeitintervalle angegeben. Tagesangaben werden etwa in Stunden seit dem Monatsbeginn umgewandelt. Die Datums- und Zeitvariablen können gut zur Differenzbildung benutzt werden, da sie intern immer von einem festen Referenzzeitpunkt aus gerechnet werden. Generell ist dies der wichtigste Vorteil der Speicherung in einer Datumsvariablen anstelle von getrennten Variablen für die Einzelangaben.

Ⓢ b) *Datums- und Zeitkonvertierungsfunktionen* (teilweise Gruppe: Extraktion der Zeitdauer, teilweise Datumsextraktion).

YRMODA(jahr,monat,tag). *Numerisch.* Aus den als Integerwerte in drei Variablen gespeicherten Datumsangaben, Jahr, Monat und Tag berechnet man die Zahl der Tage seit den 15. Oktober 1582. (Dieser Tag wird in SPSS allgemein als Referenztag verwendet.) *Beispiel:* 18.4.1945 ist 132404 Tage vom Referenzzeitpunkt entfernt.

Die folgenden Konvertierungsfunktionen sind speziell für Zeitformate vorgesehen, funktionieren aber auch mit Datumsformaten. Der Referenzzeitpunkt variiert entsprechend dem benutzten Format.

CTIME.DAYS(zeit¹). *Numerisch.* Dazu muss eine Variable in einem SPSS-Datums- oder Zeitformat vorliegen. Dann wird die Zahl der Tage, einschließlich Bruchteilen von Tagen seit dem Referenzzeitpunkt ausgegeben. Der Referenzzeitpunkt ist je nach Art der Zeitvariablen unterschiedlich. Bei Datumsangaben ist das der 15. Oktober 1582, bei reinen Zeitangaben dagegen 0 Uhr Mitternacht usw. (⇒ Syntax Reference Guide „Date and Time in SPSS“).

CTIME.HOURS(zeit). *Numerisch.* Ebenso, jedoch wird der Abstand zum Referenzzeitpunkt in Stunden (einschließlich Bruchteilen von Stunden) ausgegeben.

CTIME.MINUTES(zeit). *Numerisch.* Ebenso, jedoch wird der Abstand zum Referenzzeitpunkt in Minuten (einschließlich Minutenbruchteilen) angegeben.

CTIME.SECONDS(zeit). *Numerisch.* Ebenso, jedoch wird der Zeitabstand zum Referenzzeitpunkt in Sekunden (einschließlich Sekundenbruchteilen) angegeben.

⑤ c) *Datums- und Zeit-Extraktionsfunktionen* (teilweise Gruppe: Extraktion der Zeitdauer, teilweise Datumsextraktion).

Diese Funktionen dienen dazu, aus einer im SPSS-Datums- bzw. Zeitformat vorliegenden Variablen eine Teilinformation zu extrahieren, z.B. aus einer Variablen, die Datum, Stunden und Sekunden enthält, ausschließlich das Datum. Allen so gewonnenen Variablen muss noch durch Umdefinieren ein geeignetes Variablenformat zugewiesen werden.

XDATE.DATE(datum²). *Numerisch im SPSS-Datumsformat.* Aus einer SPSS-Datumsvariablen werden alleine die Datumsinformationen extrahiert. *Beispiel:* 12.7.1945 22:30 wird 12.7.1945.

XDATE.HOUR(datum). *Numerisch.* Aus einer SPSS-Datumsvariablen werden alleine die Stundenangaben extrahiert. *Beispiel:* 12.7.1945 22:30 wird 22.

XDATE.JDAY(datum). *Numerisch.* Aus einer SPSS-Datumsvariablen wird ermittelt, um den wievielten Tag des Jahres es sich handelt (ergibt eine Zahl zwischen 1 und 366). *Beispiel:* Der 12.7.1945 ist der 193. Tag des Jahres.

XDATE.MDAY(datum). *Numerisch.* Es wird ermittelt, um den wievielten Tag eines Monats es sich handelt (ergibt eine ganze Zahl zwischen 1 und 31).

XDATE.MINUTE(datum). *Numerisch.* Extrahiert die Minutenangaben aus einer SPSS-Datumsvariablen (ergibt eine ganze Zahl zwischen 0 und 59).

XDATE.MONTH(datum). *Numerisch.* Extrahiert die Monatszahl aus einer SPSS-Datumsvariablen (gibt eine ganze Zahl zwischen 1 und 12).

XDATE.QUARTER(datum). *Numerisch.* Bestimmt aus einer SPSS-Datumsvariablen, um welches Quartal im Jahr es sich handelt und gibt den Wert (eine ganze Zahl zwischen 1 und 4) aus. Die Ausgangsvariable muss selbst in ihrem Format keine Quartalsangabe enthalten.

XDATE.SECOND(datum). *Numerisch.* Extrahiert die Sekunden aus einer SPSS-Datumsvariablen (eine Zahl zwischen 0 und 59).

¹ Das Argument „zeit“ verlangt immer die Eingabe von Werten im SPSS-Zeitformat.

² Das Argument „datum“ verlangt immer die Eingabe von Werten im SPSS-Datumsformat.

XDATE.TDAY(zeit). *Numerisch.* Rechnet eine Zeitangabe in ganze Tage um (ergibt eine ganze Zahl). Bei Anwendung auf Datumsangaben ergibt sich die Zahl der Tage seit 15. Okt. 1582.

XDATE.TIME(datum). *Numerisch.* Extrahiert die Tageszeit aus einer SPSS-Datumsvariablen und gibt sie als Sekunden seit Mitternacht aus. Eine so kreierte Variable muss erst in ein adäquates Datumsformat umdefiniert werden.

XDATE.WEEK(datum). *Numerisch.* Ermittelt aus einer SPSS-Datumsvariablen, um die wievielte Woche des Jahres es sich handelt (gibt eine ganze Zahl zwischen 1 und 53 aus).

XDATE.WKDAY(datum). *Numerisch.* Extrahiert aus einer SPSS-Datumsvariablen den Wochentag (gibt eine ganze Zahl zwischen 1 und 7 aus).

XDATE.YEAR(datum). *Numerisch.* Extrahiert die Jahreszahl aus einer SPSS-Datumsvariablen (gibt sie als vierstellige ganze Zahl aus).

© d) *Funktionen für Berechnungen mit Datums-, Zeitvariablen* (Gruppe: Datumsarithmetik)

DATEDIFF(zeit1,zeit2,"Einheit"). *Numerisch.* Berechnet die Differenz zwischen zwei Werten vom Typ Datum/Zeit. Zeit1 und 2 sind Datums- oder numerische Variablen, wobei zeit1 den späteren, zeit2 den früheren Zeitpunkt enthält. Das Ergebnis ist die Zeitspanne zwischen diesen beiden Zeitpunkten, ausgegeben in der durch den Parameter „Einheit“ festgelegten Zeiteinheit. Es werden nur ganze Zahlen ausgegeben, die Nachkommastellen werden gestrichen. Als "Einheit" kommen years, quarters, months, weeks, days, hours, minutes und seconds in Frage. Beachten Sie: Die Wörter für die Zeiteinheit müssen in Anführungszeichen gesetzt werden. *Beispiel:* In der Datei VZ.SAV ergibt der Ausdruck DATEDIFF(kontakt,beg_ueb,"months"), den Zeitraum zwischen Kontakt mit der Schuldnerberatung und Beginn der Überschuldung in Monaten.

DATESUM(zeit, wert, "Einheit", "Methode"). *Numerisch.* Bezeichnung: Datesum(4). Sie rechnet zu einer Datum-/Zeitangabe eine vorgegebene Anzahl von Einheiten hinzu. Dabei ist zeit die Variable in einem Datum-/Zeitformat (oder numerische Variable), zu der der Zeitraum hinzugerechnet wird. "Einheit" ist die Einheit, in der der zuzurechnende Zeitraum angegeben wird. Es sind dieselben wie bei Datediff und müssen ebenfalls in Anführungszeichen gesetzt werden. Als Methode kann "rollover" oder "closest" gewählt werden (wiederum Anführungszeichen nicht vergessen!!). Bei "rollover" werden überschüssige Tage in den nächsten Monat verschoben. Bei "closest" wird das nächstliegende gültige Datum innerhalb des Monats verwendet (Voreinstellung). Das Ergebnis ist eine Datum-/Zeitangabe. Um den Wert als Datum/Zeit darzustellen, müssen Sie der Ausgabevariablen ein entsprechendes Datum-/Zeitformat zuweisen. *Beispiel:* Ein Ausdruck DATESUM(kontakt,bearbeitungsdauer,"moths","rollover") zählt zum Datum kontakt (Erster Kontakt mit Schuldnerberatung) die Bearbeitungsdauer hinzu, die in Monaten angegeben ist und gibt eine neue Variable aus (z.B. Abschlussdatum). Dies ist eine numerische Variable, der Sie noch ein passendes Datumsformat zuweisen müssen.

DATESUM(zeit, wert, "Einheit", "Methode"). *Numerisch.* Bezeichnung: Datesum(3). Dasselbe, ohne Wahl der Methode.

© *Gruppe Verschiedene*

a) *Cross-Case Funktionen* (sind nur für Zeitreihen sinnvoll)

LAG(variable). *Numerisch oder String.* Bezeichnung Lag(1). Diese Funktion verschiebt die Werte für die Fälle einer Variablen. Der erste Fall bekommt einen System-Missing-Wert, jeder weitere Fall jeweils den Wert seines Vorgängers. Diese Funktion ist wichtig für die Analyse von Zeitreihen, z.B. zur Berechnung von Wachstumsraten. Kann auch auf Stringvariablen angewendet werden.

LAG(variable,n). *Numerisch oder String.* Bezeichnung Lag(2). Hat dieselbe Funktion. Die Anzahl der Fälle (n) bestimmt aber, wie weit die Werte verschoben werden. *Beispiel:* LAG(NR,2) bewirkt, dass die ersten beiden Fälle einen System-Missing-Wert erhalten, der dritte bekommt den Wert des ersten, der vierte des zweiten usw.

b) *Sonstiges*

\$CASENUM Zusätzlich wird in dieser Gruppe die Funktion \$CASENUM angezeigt. \$CASENUM ist eine von SPSS automatisch vergebene Fallzahl, beginnend mit 1 für den ersten Fall etc. Werden Fälle umsortiert, ändert sich deren Casenum, wenn sie danach in einer veränderten Reihenfolge stehen. Will man eine feste Zuordnung der ursprünglichen Casenum zu den Fällen erreichen, kann man diese \$CASENUM als Funktion bei „Variable berechnen“ verwenden und sie dadurch in eine neue Variable umwandeln.

VALUELABEL(variable). *Numerisch oder String..* Damit können Labels einer numerischen Variablen zu Werte einer String-Variablen umkodiert werden. Beachten Sie, dass Sie gegebenenfalls mit der Optionsschaltfläche „Typ & Label“ die gleichnamige Dialogbox öffnen und den Variablentyp „String“ einstellen müssen. *Beispiel:* VALUELABEL(geschl) wandelt die Labels der numerischen Variablen GESCHL in Werte einer Stringvariablen um. Hat ein Wert keinen Wertelabel, wird ein Systemmissing-Wert ausgegeben.

⑦ *Wahrscheinlichkeits- und Verteilungsfunktionen*

Im Prinzip lassen sich Wahrscheinlichkeitsverteilungen durch zwei auseinander ableitbaren Typen von Funktionen beschreiben³:

- *Wahrscheinlichkeitsfunktion, Wahrscheinlichkeitsdichte.* Diese Funktion gibt bei diskreten Verteilungen an, wie wahrscheinlich bei gegebener Verteilungsform mit gegebenen Parametern das Auftreten eines bestimmten diskreten Ergebnisses q (gebräuchlicher ist die Symbolisierung als x) ist. Bei kontinuierlichen Verteilungen lässt sich die Wahrscheinlichkeit p für das Auftreten eines konkreten Wertes nicht sinnvoll bestimmen. An dessen Stelle tritt die Wahrscheinlichkeitsdichte, das heißt der Grenzwert der Wahrscheinlichkeit eines Intervalls an dieser Stelle x mit Intervallbreite nahe Null.

³ Die Verteilungsfunktionen stellen Beziehungen zwischen konkreten Ergebnissen q und deren Wahrscheinlichkeit p beim Vorliegen einer bestimmten Verteilungsform mit gegebenen Spezifikationsparametern her. In SPSS werden die konkreten Ergebnisse z.T. als q (in den Auswahllisten), z.T. als x (in der Kontexthilfe) bezeichnet. Auch die Bezeichnung der Parameter variiert. Wir bezeichnen im Folgenden das Ergebnis mit q .

- *Verteilungsfunktion.* Diese Funktion gibt die kumulierte Wahrscheinlichkeit dafür an, dass ein Ergebnis $<$ einem bestimmten Wert q eintritt.

In beiden Fällen lässt sich die Betrachtung auch umkehren und für eine gegebene Wahrscheinlichkeit p der dazugehörige Wert q ermitteln.

Mit wenigen Ausnahmen (z.B. Bernoulli) bestimmt die Funktion die Grundform einer Schar von Verteilungen, deren genaue Form durch die variablen Parameter bestimmt wird. So haben z.B. alle Normalverteilungen die charakteristische Glockenform. Die Parameter μ und Sdvt. bestimmen aber, bei welchem Wert das Zentrum der Verteilung liegt und wie breit sie verläuft. Abb. 5.31.a stellt die Wahrscheinlichkeitsfunktion einer Normalverteilung mit $\mu = 2000$ und Stdv. = 500 dar, Abb. 5.31.b. deren Verteilungsfunktion.

SPSS bietet im Grunde *vier Funktionen* an, die allerdings mit bis zu 20 Verteilungen kombiniert werden können. Die Auswahlliste enthält jede dieser Kombinationen gesondert. Daher nehmen in ihr die Verteilungsfunktionen einen sehr breiten Raum ein. An dieser Stelle kann nicht jede Kombination, sondern nur das Aufbauprinzip erklärt werden. Sie können diese Beispiele anhand der Daten von ALLBUS90.SAV nachvollziehen.

Die ersten beiden beziehen sich auf die Wahrscheinlichkeits- bzw. Dichtefunktion.

- *RV-Funktionen.* (Gruppe: Zufallszahlen). Sie erzeugen für jeden Fall einen Zufallswert aus der angegebenen Verteilung. B.: RV.NORMAL(2096,1134) weist den einzelnen Fällen Zufallszahlen aus einer Normalverteilung mit dem Mittelwert 2096 und der Standardabweichung 1134 zu. Die Wahrscheinlichkeit, eines Wertes zugewiesen zu werden, hängt von der Wahrscheinlichkeitsdichte an der entsprechenden Stelle der Verteilung ab.
- *PDF-Funktion.* (Gruppe: Wahrscheinlichkeitsdichten). Gibt die Wahrscheinlichkeit bzw. die Wahrscheinlichkeitsdichte für einen bestimmten Wert q aus. B.: PDF.NORMAL(eink,2096,1134). Gibt bei jedem Fall für seinen konkreten Wert q in der Variablen EINK die Wahrscheinlichkeitsdichte aus, wenn man davon ausgeht, dass die Daten normalverteilt sind mit dem Mittelwert 2096 und der Standardabweichung 1134. Das wäre z.B. bei einem Fall, der ein Einkommen von 2096 DM aufweist 0,000352.

Die beiden anderen beziehen sich auf die Verteilungsfunktion.

- *CDF-Funktion.* (Gruppe Verteilungsfunktionen). Gibt die kumulierte Wahrscheinlichkeit dafür an, dass ein Ergebnis $<$ einem bestimmten Wert q eintritt. B.: CDF.NORMAL(eink,2096,1134). Gibt bei jedem Fall für seinen konkreten Wert q in der Variablen EINK die Wahrscheinlichkeitsdichte aus, wenn man davon ausgeht, dass die Daten normalverteilt sind mit dem Mittelwert 2096 und der Standardabweichung 1134. Das wäre z.B. für einen Fall, der das Einkommen 2096 hat, 0,5.
- *IDF-Funktion.* (Gruppe: Quantilfunktionen). Gibt umgekehrt für eine Wahrscheinlichkeit p den Wert q aus, unterhalb dessen die kumulierte Wahrscheinlichkeit bei gegebener Verteilung p beträgt. B.: IDF.NORMAL(einkcdf, 2096,1134). Gibt bei jedem Fall für seine konkret angegebene kumulierte Wahr-

scheinlichkeit p den Wert x , unter dem bei der konkreten Verteilung die Wahrscheinlichkeiten auf p kumulieren würden. Im Beispiel stammen die Wahrscheinlichkeiten aus einer Variablen EINKCDF, die mit der CDF-Funktion gebildet wurde. Die Rückrechnung führt wieder zu den Ausgangswerten, die in der Variablen EINK stehen. Ein p -Wert von 0,5 führt z.B. zu einem Wert $q = 2096$.

- *Signifikanz-Funktion.* (Gruppe: Signifikanz). Ab Version 13 stehen diese Funktionen, allerdings nur für Die Chi-Quadrat und F-Verteilung zur Verfügung. Sie ist komplementär zur CDF-Gruppe, gibt also umgekehrt die kumulierte Wahrscheinlichkeit p oberhalb eines bestimmten Wertes q an. Beispiel: SIG.CHISQ(1,2) ergibt 0,61, das heißt bei einer Chi-Quadrat-Verteilung mit 1 Freiheitsgrad liegen 61% der Fläche oberhalb von $q = 2$, Dagegen ergibt CDF.CHISQ(1,2) den Wert 0,39, d.h. 39% der Fläche liegen Unterhalb $q = 2$.

Darüber hinaus gibt es für einige Funktionen (BETA, CHISQ, F und T, Varianten für nicht zentrale Verteilungen).

- *NPDF-Funktion.* (Gruppe: Wahrscheinlichkeitsdichten). Gibt wie eine PDF-Funktion die Wahrscheinlichkeit bzw. die Wahrscheinlichkeitsdichte für einen bestimmten Wert q aus. Zu den Parametern dieser Verteilungen aus den PDF-Funktionen kommt jeweils ein Parameter nz für die Nichtzentralität hinzu. Die Nichtzentralität bezieht sich auf die Stelle q . Ein nz von 0 ergibt eine zentrale Verteilung. Mit steigendem nz verschiebt sich die Mitte der Verteilung nach rechts. *Beispiel:* Einem Chi-Quadrat Wert von 3,84 entspricht bei $df = 1$ in einer zentralen Verteilung (PDF.CHISQ(3.84,1)) eine Wahrscheinlichkeitsdichte von 0,298. In einer nicht zentralen Verteilung mit $nz = 1$ (NPDF.CHISQ(3.84,1,1)) einer Wahrscheinlichkeitsdichte von 0,665.
- *NCDF-Funktion.* (Gruppe: Verteilungsfunktionen). Gibt wie eine CDF-Funktion die kumulierte Wahrscheinlichkeit dafür an, dass ein Ergebnis $<$ einem bestimmten Wert q eintritt. Zu den Parametern dieser Verteilungen aus den CDF-Funktionen kommt jeweils ein Parameter nz für die Dezentralität hinzu. Dieser muss jeweils größer gleich 0 und kleiner sein als q . *Beispiel:* Bei einer zentralen Chi-Quadrat Verteilung mit $df = 1$ (CDF.CHISQ(3.84,1)) beträgt die kumulierte Wahrscheinlichkeit aller Wert $> 3,84$ 0,95. Bei einer dezentralen mit $nz = 1$ (NCDF.CHISQ(3.84,1,1)) dagegen 0,83.

In Abbildung 5.4.a. entspricht die Höhe der Säule über einem minimalen Bereich bei q der Wahrscheinlichkeit p dieses Wertes (genauer Wahrscheinlichkeitsdichte eines Minimalintervalls um diesen Wert) (Ergebnis von PDF). Die (grau eingefärbte) Fläche zwischen dem Minimalwert $-\infty$ und q gibt die kumulierte Wahrscheinlichkeit aller Werte $\leq q$ (CDF) bzw. der Fußpunkt der Säule den Punkt q , bis zu dem sich die Fläche p erstreckt. Umgekehrt bezeichnet die (weiße) Fläche von dort bis $+\infty$ die kumulierte Wahrscheinlichkeit aller Werte $> q$ (SIG), wobei SIG die Fläche bei gegebenen q , INV dagegen q bei gegebener Fläche ausgibt. In der Verteilungsfunktion entspricht die linke (grau eingefärbte) Fläche zwischen dem Minimalwert $-\infty$ und q der kumulierten Wahrscheinlichkeit aller Werte $\leq q$ (CDF), die (weiße) Fläche von dort bis $+\infty$ die kumulierte Wahrscheinlichkeit aller Werte $> q$ (ICDF).

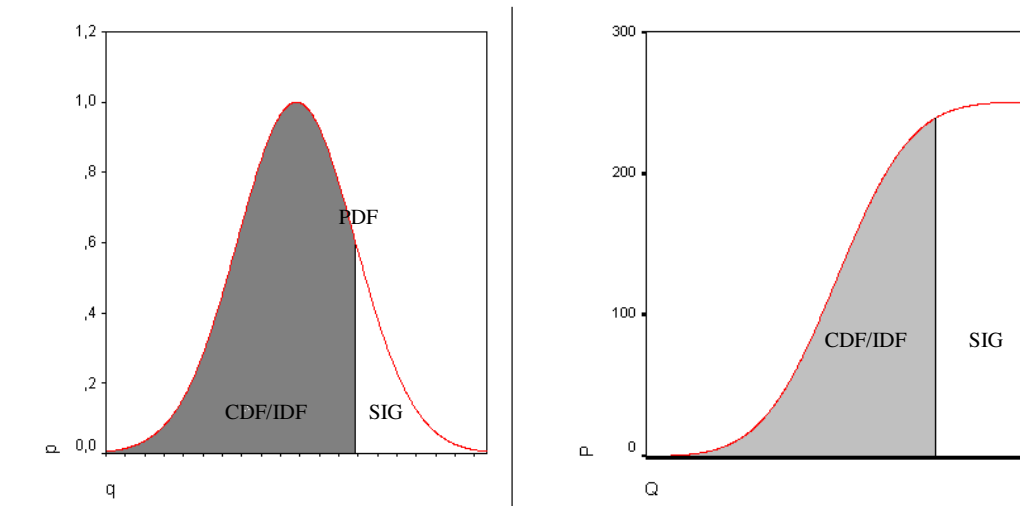


Abb. 5.31.a und b. Wahrscheinlichkeitsdichte- und Verteilungsfunktion einer Normalverteilung

Verfügbare Verteilungen. Die verfügbaren Verteilungen sind in Tabelle 5.1 zusammengestellt⁴. In ihr sind die Parameter (Kennwerte) für alle drei Verteilungsarten angegeben. Sie müssen daraus die für die jeweilige Funktion geltenden zusammenstellen (\Rightarrow Anmerkungen zur Tabelle). Es sind auch die Bereichsgrenzen für die jeweiligen Spezifikationsparameter angegeben. *Beispiel:* Aus der Tabelle entnehmen Sie für die Chi-Quadrat-Verteilung die Spezifikationen q , p und df ; dabei ist df = Zahl der Freiheitsgrade der für die Chi-Quadrat-Funktion charakteristische einzige Parameter; q dagegen steht für das jeweilige konkrete Ergebnis q und p für dessen Wahrscheinlichkeit bzw. die kumulierte Wahrscheinlichkeit aller Werte kleiner q (diese Spezifikationen kommen bei allen Verteilungen in der entsprechenden Form vor). Für die RV-Funktion ist ausschließlich der charakteristische Parameter der Chi-Quadrat-Verteilung, die Zahl der Freiheitsgrade df relevant, denn es sollen Zufallszahlen aus der so charakterisierten Verteilung generiert werden. Die charakteristischen Verteilungsparameter sind selbstverständlich bei allen Funktionstypen relevant. Die anderen Spezifikationsparameter aber wechseln. Für eine PDF-Funktion ist neben df auch q relevant, denn für bestimmte Werte q soll die Wahrscheinlichkeitsdichte p ermittelt werden. Dasselbe gilt für die CDF-Funktion. Hier soll die kumulierte Wahrscheinlichkeit p für alle Werte kleiner q ermittelt werden. Die IDF-Funktion dagegen verlangt die Angabe von p , der kumulierten Wahrscheinlichkeit, für die dann der Wert q ermittelt werden soll unter dem die Werte liegen, deren Wahrscheinlichkeit zusammen p ergibt.

Beachten Sie weiter: Alle Funktionen geben *numerische Werte* aus. RV-Verteilungen führen zur Zuweisung von Pseudozufallszahlen. Wie bei der Verwendung des Zufallszahlengenerators generell, sollte man, wenn man die neu gebildete Verteilung reproduzieren möchte, zunächst immer unter „Transformieren“, „Startwert für Zufallszahlen...“ den Startwert setzen, von der die Auswahl ausgehen soll.

⁴ Mit einigen Ausnahmen, insbesondere den Verteilungen der Gruppe Signifikanz, die anschließend erläutert werden.

Die Funktion CDFNORM(zWert), die unter anderen Funktionen aufgeführt ist, erfüllt für die Standardnormalverteilung dieselbe Aufgabe wie die CDF.Normal, setzt aber voraus, dass z-transformierte Daten als q-Werte eingesetzt werden.

Tabelle 5.3. Verfügbare Verteilungen der Verteilungsfunktionen

Verteilung	Spezifikationen, Spezifikationsbereiche
BERNOULLI ²⁾	(q, p). $0 \leq p \leq 1$. p ist die Wahrscheinlichkeit für einen Erfolg.
BETA ¹⁾³⁾	(q, p, form1, form2). $0 \leq q \leq 1$, $0 \leq p \leq 1$, form > 0 .
BINOM ²⁾	(q, n, p). n ist Zahl der Versuche, p die Wahrscheinlichkeit, bei einem Versuch Erfolg zu haben. q muss eine positive Integerzahl sein. $0 \leq p \leq 1$.
CAUCHY ¹⁾	(q, p, lage, Skala). $0 < p < 1$, skala > 0 .
CHISQ (Chi-Quadrat) ^{1) 3)}	(q, p, df). $q \geq 0$, $0 \leq p < 1$, df ≥ 0 .
EXP (Exponential) ¹⁾	(q, p, form). $q \geq 0$, $0 \leq p \leq 1$, form > 0 .
F ^{1) 3)}	(q, p, df1, df2). $q \geq 0$, $0 \leq p < 1$, df1 und df2 > 0
GAMMA ¹⁾	(q, p, form, skala). $q \geq 0$, $0 \leq p < 1$, form und skala > 0 .
GEOM (Geometrische) ²⁾	(q, p). $0 < p < 1$. q ist die Zahl der Versuche (inklusive dem letzten m), die benötigt werden, bevor ein Erfolg erzielt wird, p die Wahrscheinlichkeit für einen Erfolg in einem einzigen Versuch.
HALFNRM (Halbnormal) ¹⁾	(q, p, schwelle, skala), $0 < p < 1$, schwelle > 0 .
HYPER (Hypergeometrische) ²⁾	(q, gesamt, stichpr, treffer). gesamt = Zahl der Objekte in der Grundgesamtheit, stichprobe = Größe einer Zufallsstichprobe, gezogen ohne Zurücklegen, treffer = Zahl der Objekte mit der festgelegten Eigenschaft in der Grundgesamtheit (alle drei müssen positive ganze Zahlen sein), q = die Zahl der Objekte mit dieser Eigenschaft in der Stichprobe.
IGAUSS (inverse Normalverteilung) ¹⁾	(q, p, mittel, skala) $0 < p < 1$, mittel > 0 , skala > 0 .
LAPLACE (Doppelexponentialverteilung) ¹⁾	(q, p, mittel, skala). $0 < p < 1$, skala > 0 .
LOGISTIC (Logistische) ¹⁾	(q, p, mittel, skala). $0 < p < 1$, skala > 0 .
LNORMAL (Lognormal) ¹⁾	(q, p, a, b). $q \geq 0$, $0 \leq p < 1$, a, b > 0 .
NEGBIN (negative Binomial) ²⁾	(q, p, schwelle). $0 < p \leq 1$. a ist die Erfolgswahrscheinlichkeit bei einem Versuch. Schwelle ist die Zahl der Erfolge, muss eine ganze Zahl sein. q ist die Zahl der Versuche (inklusive dem letzten), bevor die durch Schwelle angegebene Zahl von Erfolgen beobachtet werden. (Beim Schwellenwert 1 identisch mit der geometrischen Verteilung.)
NORMAL ¹⁾	(q, p, mittel, stdAbw). $0 < p < 1$, stdAbw > 0 .

PARETO ¹⁾	(q, p, Schwelle, Form). $q \geq a$, $0 \leq p < 1$, Schwelle, Form > 0 .
POISSON ²⁾	(q, Mittel). Mittel > 0 . Mittel ist die Zahl der Ereignisse eines bestimmten Typs, die im Durchschnitt in einer festgelegten Zeitperiode eintreten. q muss eine positive Integerzahl sein.
SMOD (studentisiertes Maximalmodul) ¹⁾⁴⁾	(q, p, Größe, df). a und $df \geq 1$.
SRANGE (studentisierte Spannweite) ¹⁾⁴⁾	(q, p, Größe, df). Größe und $df \geq 1$. q = studentisierte Spannweite, Größe = Zahl der Fälle (verglichenen Stichproben).
T ¹⁾³⁾	(q, p, df). $0 < p < 1$. $df > 0$.
UNIFORM ¹⁾	(q, p, min, max). $0 \leq q \leq 1$, $0 \leq p \leq 1$.
WEIBULL ¹⁾	(q, p, a, b). $q \geq 0$, $0 \leq p < 1$, a und $b > 0$.

1 Kontinuierliche Funktionen. Bei Verwendung von PDF, CDF, NPfD und NCDF entfällt der Parameter p, bei IDF der Parameter q, bei Verwendung von RV entfallen jeweils p und q (q = quantity, Wert, für den die Wahrscheinlichkeit gesucht wird; p = probability, Wahrscheinlichkeit, für die der Wert gesucht wird).

2 Diskrete Funktionen. Es existieren nur CDF und RV-Funktionen (a ist jeweils ein Wahrscheinlichkeitsparameter; q entfällt bei Verwendung von RV).

3 Auch als nicht-zentrale Verteilung (NCDF) verfügbar. Für diese gelten dieselben Spezifikationsparameter wie für CDF-Funktionen, jedoch ergänzt durch den Nichtzentralitätsparameter nc.

4 Nur CDF und IDF-Funktion.

Dabei bedeutet gewöhnlich: Lage = Mitte der Verteilung (arithmetisches Mittel), Skala = Streuung (Standardabweichung oder einen Skalenparameter λ), Schwelle = Wert, ab dem die Verteilung beginnt.

⊗ Andere Verteilungsfunktionen

a) Gruppe: Verteilungsfunktionen und Wahrscheinlichkeitsdichten

CDFNORM(zWert). Numerisch. Gibt die Wahrscheinlichkeit dafür an, dass eine normalverteilte Zufallsvariable mit einem arithmetischen Mittel von 0 und einer Standardabweichung von 1 (= Standardnormalverteilung) unter dem vom Benutzer definierten z-Wert liegt. Der z-Wert kann zwischen 0 und etwa 3 variieren. (Dezimalstellen müssen mit Punkt eingegeben werden.) Diese Funktion muss auf bereits z-transformierte Variable angewendet werden. *Beispiel:* Die Variable ZEINK enthält z-transformierte Einkommenswerte. CDFNORM(zeink) gibt dann für jeden Fall die Wahrscheinlichkeit aus, mit der ein Fall einen geringeren Wert als der Fall selbst erreicht oder anders ausgedrückt, welcher Anteil der Fälle ein geringeres Einkommen besitzt. (*Voraussetzung:* Die Normalverteilungsannahme ist zutreffend.) *Beispiel:* In der Datei ALLBUS90 hat der 3. Fall ein Einkommen von 1450 DM, daraus – sowie aus dem Mittelwert und der Streuung der Verteilung – errechnet sich ein z-Wert $-0,57046$ und dieser ergibt wiederum die Wahrscheinlichkeit von 0,28 dafür, dass ein anderer Fall ein geringeres Einkommen besitzt.

CDF.BVNOR(q1, q2, Korr). Numerisch. Diese Funktion gibt die kumulative Wahrscheinlichkeit zurück, mit welcher ein Wert aus der bivariaten Standardnor-

malvorteilung mit dem angegebenen Parameter r = Korrelation kleiner als q_1 und q_2 ist.

PDF.BVNOR(q_1, q_2, Korr) Numerisch. Diese Funktion gibt die Wahrscheinlichkeitsdichte zurück für eine Wertekombination q_1/q_2 in einer bivariaten Standardnormalverteilung mit dem angegebenen Parameter r für die Korrelation zwischen den beiden Variablen q_1 und q_2 .

b) Gruppe: Signifikanz

SIG.CHISQ(df, q). Die Wahrscheinlichkeit, dass ein Wert aus einer Chi-Quadrat-Verteilung mit df Freiheitsgraden größer ist als q .

SIG.F(df_1, df_2, q). Die Wahrscheinlichkeit, dass ein Wert aus einer F-Verteilung mit df_1 und df_2 Freiheitsgraden größer ist als q .

© *String-Funktionen* (Gruppe: String)

CONCAT($\text{strAusdr}, \text{strAusdr}, \dots$)⁵. *String*. Verbindet die String-Werte aller Argumente zu einem resultierenden String-Wert. *Beispiel*: In einer String-Variablen LAND stehen die Namen von Ländern, in einer anderen Stringvariablen ENTW wird deren Entwicklungsstand (z.B. „unterentwickelt“) festgehalten. Die Funktion CONCAT(Land,Entw) fasst beides in einem neuen String zusammen (etwa wird aus „Bangla Desh“ und „unterentwickelt“ „Bangla Desh unterentwickelt“). Diese Funktion benötigt mindestens zwei Argumente, die String-Werte sein müssen.

LENGTH(strAusdr). *Numerisch*. Ergibt die Länge des Stringausdrucks (strAusdr). Das Ergebnis ist die definierte, nicht die tatsächliche Länge. Für eine Stringvariable LAND mit 30 Zeichen Länge gibt die Funktion ohne Bezug auf den Stringwert eines Falles immer das Ergebnis 30 aus. Wenn Sie die tatsächliche Länge erhalten möchten, geben Sie den Ausdruck in folgender Form an: LENGTH(RTRIM(strAusdr)). *Beispiel*: LENGTH(RTRIM(Land)) gibt bei dem genannten Beispiel für den Stringwert „Deutschland“ das Ergebnis 11, für „England“ das Ergebnis 7 aus.

LOWER(strAusdr). *String*. Wandelt alle Großbuchstaben in „ strAusdr “ in Kleinbuchstaben um. Alle anderen Zeichen werden nicht verändert.

LPAD($\text{strAusdr}, \text{länge}$) (Bezeichnung: Char.Lpad(2)). *String*. Ergibt einen String, in dem der Ausdruck, der in „ strAusdr “ enthalten ist (kann eine Variable sein) von links mit Leerzeichen aufgefüllt wird, bis die im Argument „ länge “ angegebene Gesamtlänge erreicht ist. „ länge “ muss eine positive ganze Zahl zwischen 1 und 255 sein. *Beispiel*: LPAD(land,50) macht aus der bisher 30stelligen Stringvariablen LAND eine 50stellige und speichert die Stringwerte nicht mehr linksbündig, sondern (allerdings links beginnend in der gleichen Spalte, orientiert am längsten Wert) am rechten Rand der Variablen.

LPAD($\text{strAusdr}, \text{länge}, \text{zeichen}$) (Bezeichnung: Char.Lpad(3)). *String*. Ergibt einen String, in dem der in „ strAusdr “ enthaltene String (kann auch eine Variable sein) von links mit dem im Argument „ zeichen “ angegebenen Zeichen aufgefüllt wird, bis die im Argument „ länge “ angegebene Gesamtlänge erreicht ist. Länge muss eine positive ganze Zahl zwischen 1 und 255 sein. „ Zeichen “ kann ein einzelnes Zeichen, eingeschlossen in Anführungszeichen, sein oder das Ergebnis einer

⁵ strAusdr bedeutet, dass der Name einer Stringvariablen einzugeben ist.

String-Funktion, die ein einzelnes Zeichen liefert. *Beispiel:* LPAD(land,50,"x"). Erbringt dasselbe Ergebnis wie oben. Aufgefüllt werden aber nicht Leerzeichen, sondern x-Zeichen.

LTRIM(strAusdr) (Bezeichnung: Ltrim(1). *String.* Entfernt vom im Argument „strAusdr“ enthaltenen String alle führenden Leerzeichen. *Beispiel:* ´ Deutschland´ ergibt ´Deutschland´.

LTRIM(strAusdr,zeichen) (Bezeichnung: Ltrim(2). *String.* Entfernt vom im Argument „strAusdr“ enthaltenen String die im Argument „zeichen“ angeführten führenden Zeichen. Zeichen kann ein einzelnes Zeichen, eingeschlossen in Anführungszeichen, sein oder das Ergebnis einer String-Funktion, die ein einzelnes Zeichen liefert. *Beispiel:* LTRIM(Land2,"x"). Aus ´xxxDeutschland´ wird ´Deutschland´.

Beachten Sie: Immer wenn in Stringvariablen ausgegeben wird, muss, bevor der Befehl abgeschickt wird, in der nach Anklicken der Schaltfläche „Typ und Label...“ geöffneten Dialogbox „Variable berechnen: Typ und Label“ in der Gruppe „Typ“ die Option „String“ gewählt und eine Stringbreite eingetragen werden.

REPLACE(strAusdr, strAlt, strNeu). *String.* (Bezeichnung: Replace(3)). *Beispiel:* REPLACE(Bank,"Deutsche Bank","Bank"). In einer neu berechneten String-Variablen werden die Werte „Deutsche Bank“ aus der Herkunftsstringvariablen durch die Werte „Bank“ ersetzt. Alle anderen Werte werden übernommen. Denken *Beachten Sie:* Immer wenn in Stringvariablen ausgegeben wird, muss, bevor der Befehl abgeschickt wird, in der nach Anklicken der Schaltfläche „Typ und Label...“ geöffneten Dialogbox „Variable berechnen: Typ und Label“ in der Gruppe „Typ“ die Option „String“ gewählt und eine Stringbreite eingetragen werden.

REPLACE(strAusdr, strAlt, strNeu, wdhAnz). (Bezeichnung: Replace(4)). *Beispiel:* REPLACE(Banken,"Deutsche Bank","Bank",2) In einer neu berechneten String-Variablen werden die Werte „Deutsche Bank“ aus der Herkunftsstringvariablen durch die Werte „Bank“ ersetzt. Der Unterschied zur vorigen Variante kann das innerhalb desselben Strings (Wertes) wiederholt geschehen, im Beispiel zwei Mal. Dabei erfolgt die Zählung innerhalb des Strings vom Ende her. Alle anderen Werte werden übernommen. (Hinweis, s. vorige Variante.)

RPAD(strAusdr,länge). (Bezeichnung: Char.Rpad(2)). *String.* Ergibt einen String, in dem der im Ausdruck „strAusdr“ enthaltene String (kann eine Variable sein) von rechts mit Leerzeichen aufgefüllt wird, bis die im Argument „länge“ angegebene Gesamtlänge erreicht ist. Länge muss eine positive ganze Zahl zwischen 1 und 255 sein. Zeichen kann ein einzelnes Zeichen (eingeschlossen in Anführungszeichen) sein oder das Ergebnis einer String-Funktion, die ein einzelnes Zeichen liefert.

RPAD(strAusdr,länge,zeichen). (Bezeichnung: Char.Rpad(2)). *String.* Identisch mit der vorherigen Funktion, jedoch wird der String von rechts mit dem im Argument „zeichen“ angegebenen Zeichen aufgefüllt.

RTRIM(strAusdr). (Bezeichnung: Rtrim(1) *String.* Entfernt vom im Argument „strAusdr“ angegebenen String (kann auch eine Variable sein) alle nachstehenden Leerzeichen.

RTRIM(strAusdr,zeichen). (Bezeichnung: Rtrim(2)). *String*. Entfernt vom im Argument „strAusdr“ angegebenen String (kann auch eine Variable sein) jedes Vorkommen des im Argument definierten Zeichens als nachstehendes Zeichen.

SUBSTR(strAusdr,pos). (Bezeichnung: Char.Substr(2)). *String*. Liefert einen Teil des Ausdrucks im Argument „strAusdr“ ab der Stelle „pos“ bis zum Ende von „strAusdr“. *Beispiel:* SUBSTR(land,8) ergibt für den String „Deutschland“ in der Variablen LAND den neuen Wert „land“.

SUBSTR(strAusdr,Pos,länge). (Bezeichnung: Char.Substr(3)). *String*. Liefert den Teil des Ausdrucks „strAusdr“, der an der Stelle „pos“ beginnt und die im Argument „länge“ angegebene Länge hat. *Beispiel:* SUBSTR(land,8,3) ergäbe anstelle von „Deutschland“ „lan“.

UPCAS(strAusdr). *String*. Wandelt alle Kleinbuchstaben des im Argument „strAusdr“ enthaltenen Strings in Großbuchstaben um. Alle anderen Zeichen werden nicht verändert.

Weitere Funktionen (Index) sind in auch in der Gruppe „Suchen“ verfügbar, andere sind nur für die arabische Sprache oder das Codeformat der Zeichen interessant.

⑩ *Funktionen zur Umwandlung (numerisch – string)* (Gruppe: Umwandlung)

NUMBER(strAusdr,format). *Numerisch*. Diese Funktion benutzt man, um in einer Stringvariablen gespeicherte Zahlenangaben (!) in einen numerischen Ausdruck zu verwandeln. Das kann insbesondere bei Übernahme von Zahlen aus Fremdprogrammen interessant sein. Numerische Variablen sind in SPSS besser zu verarbeiten. Das Argument „format“ gibt das Einleseformat des numerischen Ausdrucks an. Die Formatangabe erfolgt entsprechend den dazu vorgesehenen Syntaxbefehlen. Festes Format der Breite 8 z.B. wird mit f8 eingegeben. Das bedeutet Folgendes: Wenn der Stringausdruck acht Zeichen lang ist, entspricht NUMBER(StrAusdr, f8) dem numerischen Einleseformat. Wenn der Stringausdruck nicht mit dem angegebenen Format eingelesen werden kann oder wenn er nicht interpretierbare Zeichen enthält, ist das Ergebnis ein System-Missing-Wert.

Wenn die angegebene Länge n des numerischen Formats kleiner als die Länge des Stringausdrucks ist, werden nur die ersten n Zeichen zur Umwandlung herangezogen. (*Beachten Sie:* Sie müssen der Ergebnisvariablen vorher ein numerisches Format geben. Es können so auch Zahlen mit Kommastellen eingelesen werden. Das Einleseformat gibt die Kommastellen nicht an, wenn diese explizit sind. Das Ergebnisformat bildet sie automatisch. Bei impliziten Kommastellen dagegen müssten Sie im Einleseformat angegeben werden. Evtl. muss die Ergebnisvariable noch in geeigneter Form umdefiniert werden.) *Beispiel.* Eine Stringvariable EINKSTR, Breite 15, enthält Einkommensdaten. Sie sollen in eine numerische Variable überführt werden. Ein Stringwert '3120,55' wird dann durch NUMBER(einkstr, f15) in einen numerischen Wert 3120,55 überführt, der wie alle numerischen Werte auch rechtsbündig gespeichert ist.

STRING(numAusdr,format). *String*. Wandelt einen numerischen Ausdruck in einen String um. Das Argument „format“ muss ein gültiges numerisches Darstellungsformat sein. *Beispiel:* STRING(eink,F10.2) ergibt für einen in der numerischen Variablen EINK gespeicherten Wert 1250,55 den Stringausdruck '1250,55'.

Aus einer letzten Gruppe „Aktuelles Datum/ aktuelle Uhrzeit“ können Datum und Uhrzeit in verschiedenen Formaten zur Berechnung neuer Variablen herangezogen werden.