

# Effizientes Datenmanagement mit der *SPSS* Syntax: Eine Einführung

*IBM SPSS* gehört zu den populärsten Statistikprogrammen im wirtschaftswissenschaftlichen Studium und in der Praxis. Viele Anwender führen ihre Analysen ausschließlich mit Hilfe der grafischen Benutzeroberfläche durch, obwohl mit Hilfe der *SPSS* Syntax viele Analysen schneller realisiert sowie automatisiert werden könnten. Dieser Beitrag bietet eine Einführung in die Arbeit mit der *SPSS* Syntax.



**Prof. Dr. Marko Sarstedt**  
ist Professor für Marketing an der Otto-von-Guericke Universität Magdeburg.



**Prof. Dr. Tobias Schütz**  
ist Professor für Marketing und Customer Science an der ESB Business School Reutlingen.



**Prof. Dr. Sascha Raithel**  
ist Professor für Marketing an der Freien Universität Berlin.

**Stichwörter: Datenanalyse, Datenmanagement, Makros, *SPSS*, Syntax**

## 1. Vorteile der *SPSS* Syntax

Im Vergleich zur grafischen Benutzeroberfläche, die die Datenanalyse durch größtenteils selbsterklärende Menüs und

Symbole unterstützt, erscheint die Datenbeschreibung und Datenauswertung mit Hilfe der ***SPSS* Syntax** auf den ersten Blick umständlich. Jedoch bietet die Steuersprache in praktisch allen Bereichen wesentliche **Vorteile**. So können durch die Syntax viele Prozeduren schneller umgesetzt werden. Dies ist insbesondere in Hinblick auf komplexere **Datentransformationen** unabdingbar. Modifikationen einzelner Einstellungen beim Aufruf der Analyseverfahren lassen sich ebenfalls zeitsparend realisieren. Durch das Zusammenstellen mehrerer Befehle zu einem zusammenhängenden Programm lassen sich auch komplexe Aufgaben automatisieren, wobei die Übersichtlichkeit gewahrt bleibt. Anders als Dialogfeldeinstellungen können Syntaxbefehle zudem dauerhaft gespeichert werden, so dass nicht nur eine lückenlose Dokumentation des eigenen Vorgehens möglich ist, sondern die Analysen einfach replizierbar werden. Die Auswertungsschritte können zu einem späteren Zeitpunkt problemlos mit den gleichen oder neuen Daten wiederholt werden. Die Möglichkeit, Kommentare einzufügen erlaubt ferner eine bequeme Korrespondenz über vorgenommene Auswertungsschritte mit Dritten. Die Syntax ist auch deswegen so mächtig, weil durch sie über die Menüführung nicht zugängliche Funktionalitäten und Hilfsroutinen verfügbar gemacht werden.

Weitere Vorteile sind die

- (1) übersichtliche und schnellere Auswertung von Datensätzen mit besonders vielen Variablen,
- (2) Steigerung des Bewusstseins für die geplanten Auswertungen, da die einzelnen Schritte dezidiert formuliert werden müssen, und
- (3) Interoperabilität, d. h. Syntax-Befehle können unabhängig von Betriebssystemen genutzt werden (vgl. *Grotenhuis* und *Visscher* 2014).

Der vorliegende Beitrag bietet eine **Einführung** in die Arbeit mit der *SPSS* Syntax. Hierfür beschreibt der Beitrag zu-

nächst, wie Syntaxbefehle erzeugt und ausgeführt werden können. Im Anschluss werden einige Auswertungen und Datenmodifikationen vorgestellt, die mit Hilfe der SPSS Syntax effizienter und übersichtlicher erledigt werden können. Die Ausführungen basieren hierbei auf dem Buch „IBM SPSS Syntax – Eine anwendungsorientierte Einführung“ (Sarstedt et al., 2018) und greifen auf Daten der Germany Customer Satisfaction Index-Studie von J.D. Power & Associates Germany zurück. Der entsprechende Trainingsdatensatz kann kostenlos auf der Webseite <http://www.das-syntax-buch.de> heruntergeladen werden.

## 2. Erzeugen und Ausführen der SPSS Syntax

Die Eingabe und das Ausführen von Befehlen erfolgen über den **Syntax-Editor**. Die Eingabe kann über drei Wege erfolgen:

1. manuell
2. mit Hilfe von Point-and-click-Operationen
3. durch das Kopieren und Einfügen des Log-Textes einer Analyse aus der Output Datei

Um eine Syntax manuell einzugeben, muss über das Menü ► *Datei* ► *Neu* ► *Syntax* ausgewählt werden. Dadurch öffnet sich der Editor und die Eingabe kann beginnen. Bei einer Point-and-click-Operation definiert der Nutzer die Analyse zunächst regulär über die Menüsteuerung, klickt dann aber statt dem gewohnten *OK* das Steuerfeld *Einfügen*. Dadurch werden die ausgewählten Befehle nicht ausgeführt, sondern als Text in den Syntax-Editor übertragen, der sich von selbst öffnet. In der Output Datei (Format: .spv) zeigt SPSS zu Beginn jeder Ausgabe einen sogenann-

ten Log-Text. Liegt eine Analyse bereits vor (z. B. da ein anderer Analyst diese erstellt hat), so kann dieser Log-Text durch kopieren/einfügen in einen geöffneten Syntaxeditor übertragen werden.

Die durch Point-and-click oder Kopieren/Einfügen erzeugte Syntax kann nun modifiziert und gespeichert werden. Auf diese Weise wird die Syntax auch Einsteigern schnell und einfach zugänglich gemacht.

Um dies zu illustrieren, soll auf Grundlage des Beispieldatensatzes *JDPA Germany CSI.sav* untersucht werden, ob ein Zusammenhang zwischen den Zufriedenheiten mit den einzelnen Aspekten des Fahrzeugdesigns besteht. Die Variablen *design\_frontansicht*, *design\_seitenansicht* und *design\_heckansicht* geben die Zufriedenheit des Befragten mit dem jeweiligen Aspekt des Außendesigns seines Fahrzeugs wider. Die Variable *design\_gesamteindruck* gibt Aufschluss über die Gesamtzufriedenheit mit dem Fahrzeugdesign. Es soll daher eine **Korrelationsanalyse** durchgeführt werden, welche durch den Syntax-Befehl **CORRELATIONS** umgesetzt werden kann. Durch die Menüsequenz ► *Analysieren* ► *Korrelation* ► *Bivariat...* wird zunächst das entsprechende Menü aufgerufen. Nach Auswahl der relevanten Variablen und des Pearson'schen Korrelationskoeffizienten (vgl. Sarstedt und Mooi, 2019) muss das Steuerfeld *Einfügen* betätigt werden (vgl. Abb. 1). SPSS zeigt daraufhin die in Tab. 1 dargestellte Programm-Syntax im (bereits geöffneten oder sich sodann öffnenden) Syntax-Editor.

Abb. 2 zeigt das entsprechende Vorgehen beim Kopieren und Einfügen des Log-Textes.

Damit der im Syntax-Fenster formulierte Befehl ausgeführt wird, muss er zur Verarbeitung an SPSS übergeben werden.

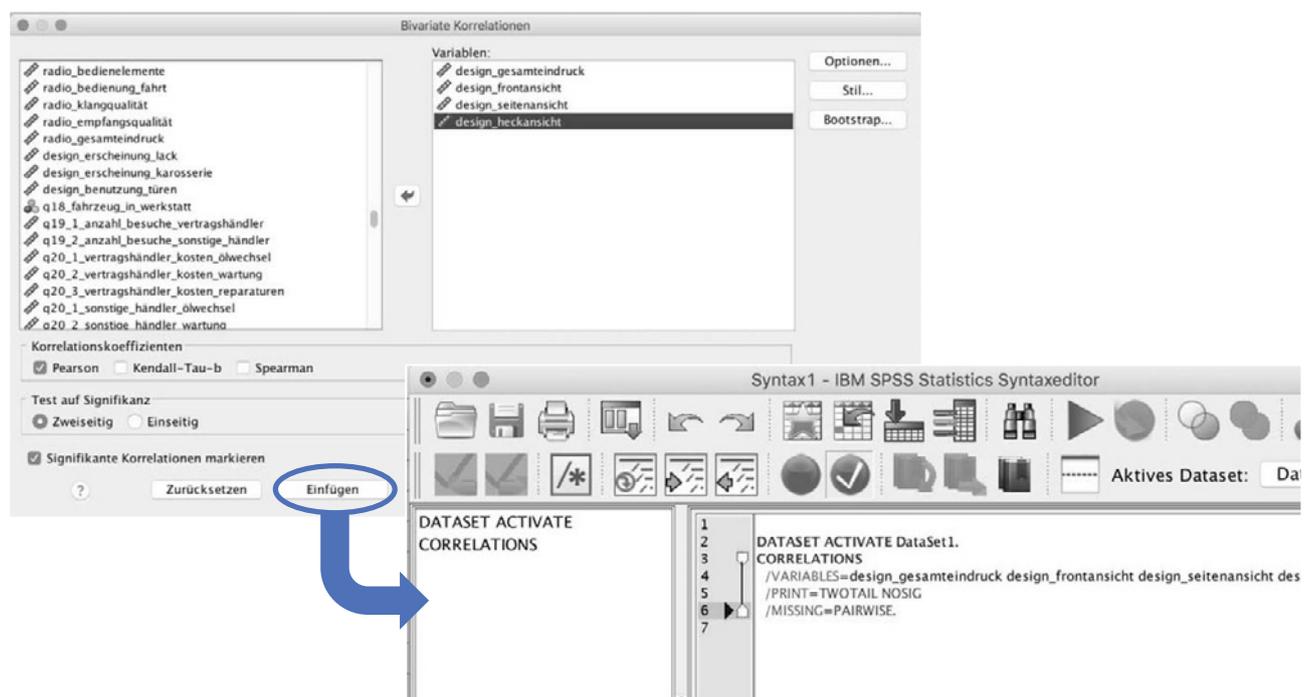


Abb. 1: Point-and-Click Operation

```

CORRELATIONS

/VARIABLES = design_gesamteindruck design_frontansicht
              design_seitenansicht design_heckansicht

/PRINT = TWOTAIL NOSIG

/MISSING =

```

Tab. 1: CORRELATIONS-Syntax

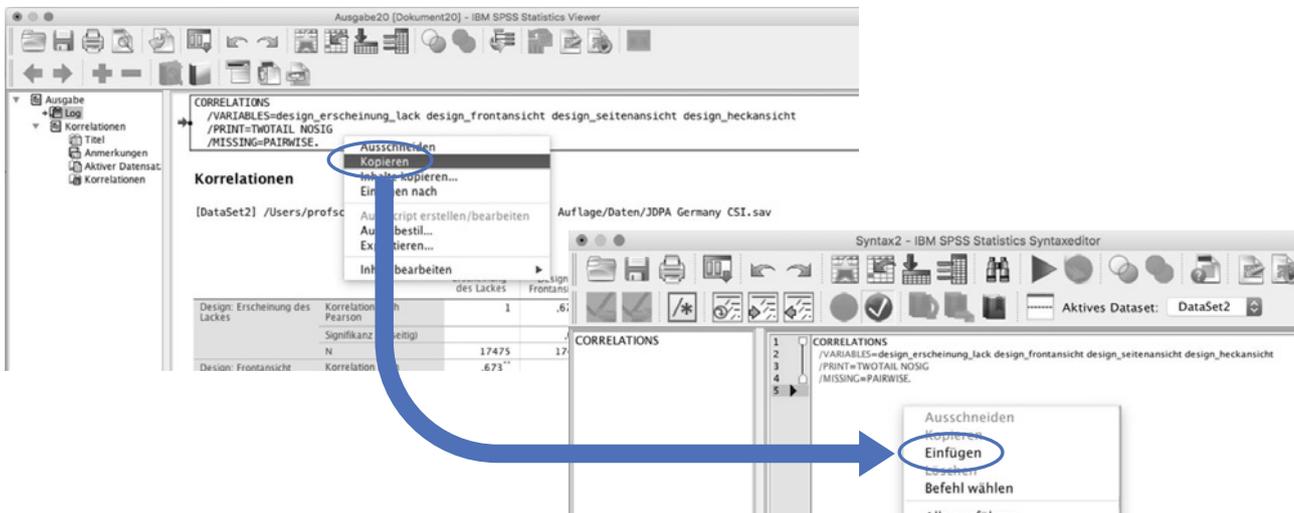


Abb. 2: Kopieren/Einfügen des Log-Textes aus der Ausgabedatei

Hierfür kann der Cursor an einer beliebigen Stelle innerhalb des Befehls positioniert und die Menüfolge ► *Ausführen* ► *Auswahl* beziehungsweise die Schaltfläche mit dem grünen Pfeil in der Menüleiste betätigt werden.

Falls der Nutzer nicht an allen Korrelationen interessiert ist, sondern sich primär mit der Variable *design\_gesamteindruck* beschäftigen möchte, so ist das mit der Syntax möglich. Die Variable *design\_gesamteindruck* umfasst die Zufriedenheit des Befragten mit allen Aspekten des Fahrzeugdesigns und sollte somit mit den Teilaspekten des Außen-Designs korrelieren. Wenn diese Annahme überprüft werden soll, könnte man nun umständlich drei Mal in Folge die Korrelation zwischen *design\_gesamteindruck* und jeder der anderen Variablen über die SPSS-Menüführung berechnen lassen oder eine bei vielen Variablen eher unübersichtliche Korrelationsmatrix aller bivariaten Beziehungen anfordern. Eine leichte Modifikation der Syntax durch Einfügen des Schlüsselwortes **WITH** löst das Problem deutlich eleganter (vgl. Tab. 2).

Die so modifizierte Syntax kann zudem mit einem **Kommentar** versehen werden, um das spätere Nachvollziehen der Analyse zu erleichtern. Kommentare werden im Syntax-Editor mit einem Stern (\*) eingeleitet, mit einem Punkt (.) abgeschlossen und können sich über mehrere Zeilen erstrecken. Im Beispiel ist ein einleitender Kommentar eingefügt. Dieser Kommentar beschreibt die nachfolgende Analyse (vgl. Tab. 2).

Das Ausführen dieser Syntax liefert eine deutlich kompaktere **Korrelationsmatrix** (vgl. Tab. 3).

Die neu generierte Syntax kann im Syntax-Editor durch die Menüsequenz ► *Datei* ► *Speichern* abgespeichert und zu einem späteren Zeitpunkt wieder genutzt werden. Die dafür notwendige Menüsequenz im Syntax-Fenster ist ► *Datei* ► *Öffnen*.

Anfänger werden ein Syntax-Programm in der Regel nicht von Grund auf in der Syntax-Sprache schreiben, sondern eher die oben beschriebenen Wege gehen und bestehende Syntax-Programme modifizieren.

```

* Analyse des Zusammenhangs zwischen den Zufriedenheiten mit
den einzelnen Aspekten des Fahrzeugdesigns.

CORRELATIONS

/VARIABLES = design_gesamteindruck WITH design_frontansicht
              design_seitenansicht design_heckansicht

/PRINT = TWOTAIL NOSIG

/MISSING = PAIRWISE.

```

Tab. 2: CORRELATIONS-Syntax mit dem Schlüsselwort WITH

		Korrelationen		
		design_frontansicht	design_seitenansicht	design_heckansicht
design_gesamteindruck	Korrelation nach Pearson	,796**	,821**	,778**
	Signifikanz (2-seitig)	,000	,000	,000
	N	17387	17365	17367

\*\* . Die Korrelation ist auf dem Niveau von 0,01 (2-seitig) signifikant.

Tab. 3: Reduzierte Korrelationsmatrix

### 3. Definition fehlender Werte

Ein wesentliches Element des Datenmanagements mit SPSS ist die Spezifikation benutzerdefiniert fehlender Werte. Diese erfolgt üblicherweise manuell über die Schaltfläche *Fehlend* im Dateneditor, lässt sich aber durch den Syntaxbefehl **MISSING VALUES** deutlich effizienter gestalten. Um die Verwendung dieses Befehls zu verdeutlichen, sollen für die Variablen *problem\_1* bis *problem\_90*, welche das Auftreten eines jeweils bestimmten Fahrzeugproblems erfassen, sämtliche Werte ab 2 bis zum numerisch größten Wert als fehlend gekennzeichnet werden. Dies ist in diesem Fall durchaus sinnvoll, da die Variablen logisch nur binär codiert sein können. Es wurde erfasst, ob ein bestimmtes Problem auftrat oder nicht, aber nicht, wie häufig ein Problem auftrat. Sollte eine der Variablen also einen Wert größer 1 annehmen, so handelt es sich offensichtlich um einen Eingabefehler. Die Syntax in Tab. 4 zeigt die entsprechende Syntax. Hierbei definiert das Schlüsselwort **TO** die Auswahl der Variablen und der Klammerterm **2 THRU HIGHEST** den als fehlend zu kodierenden Wertebereich. In vielen Situationen ist es nötig, einen bestimmten Wert für alle Variablen als fehlend zu definieren. Mit der in Tab. 5 dargestellten Syntax wird beispielsweise der Wert 99 für alle Variablen als fehlender Wert gesetzt.

### 4. Umkodieren von Variablen

Ein weiterer wichtiger Bereich des Datenmanagements, bei dem die Vorteile der Syntax besonders verdeutlicht werden können, ist das **Umkodieren** von Variablen. Besonders häufige Anwendungsfälle sind hierbei die Transformation einer intervallskalierten Variable in eine kategoriale Variable oder das Zusammenfassen mehrerer Kodierungen einer kategorialen Variablen. Anstatt die Anweisungen umständlich über die Menüfolge **Transformieren** **Umkodieren in andere Variablen...** einzugeben, ermöglicht der Befehl **RECODE** eine effizientere Handhabung.

Die Variable *q20\_2\_vertragshändler\_kosten\_wartung* erfasst z. B. die Kosten der Wartungsarbeiten beim Vertragshändler in den letzten zwölf Monaten. Mit Hilfe der in Tab. 6 dargestellten **RECODE**-Anweisung kann diese intervallskalierte Variable in eine ordinal skalierte Variable transformiert werden. Hierbei geben die Klammerterme an wie die ursprünglichen Werte umkodiert werden sollen. Beispielsweise bedeutet **LO THRU 100 = 1**, dass alle Werte vom numerisch kleinsten Wert (**LO**) bis einschließlich 100 in den Wert 1 umkodiert werden sollen. Dem Schlüsselwort **INTO** folgt der Name der neuen Variablen (*q20\_2\_kostenkategorien*), welche die neu kodierten Werte aufnimmt. Da es sich um eine sog. Datentransformation handelt, muss der Befehl mit **EXECUTE** abgeschlossen werden.

```
MISSING VALUES problem_1 TO problem_90 (2 THRU HIGHEST) .
```

Tab. 4: MISSING VALUES-Syntax (I)

```
MISSING VALUES ALL (99) .
```

Tab. 5: MISSING VALUES-Syntax (II)

```
RECODE q20_2_vertragshändler_kosten_wartung
  (LO THRU 100 = 1)
  (100 THRU 200 = 2)
  (200 THRU 300 = 3)
  (300 THRU 400 = 4)
  (400 THRU 500 = 5)
  (500 THRU HI = 6) INTO q20_2_kostenkategorien.
EXECUTE .
```

Tab. 6: RECODE-Syntax

## 5. Bedingte Datenmodifikationen

Oftmals soll die Ausführung eines Befehls oder mehrerer Befehle an die Erfüllung einer oder mehrerer Bedingungen geknüpft werden. So eine bedingte Datenmodifikation ist über die Menüfolge schwierig bzw. in komplexeren Fällen überhaupt nicht mehr abbildbar. Der **DO IF**-Befehl ermöglicht es hingegen, problemlos mehrere Variablenberechnungen unter eine oder mehrere Bedingungen zu stellen. Ein kurzes Beispiel soll die grundsätzliche Funktionsweise des Befehls verdeutlichen. Im vorliegenden Fall soll ein zuvor gebildeter Index der Gesamtzufriedenheit und wahrgenommenen Fahrzeugattraktivität bei Erstbesitzern in Abhängigkeit der Anzahl der aufgetretenen Probleme modifiziert werden. Dafür werden zunächst (mit Hilfe des **COMPUTE**-Befehls) zwei neue Indexvariablen *zufriedenheit\_4* (gleich der Variablen *customer\_satisfaction\_index\_gesamt*) und *attraktivität\_2* (gleich der Variablen *fahrzeugattraktivität\_index*) berechnet. Diese beiden Variablen sollen jeweils um den Betrag 10 reduziert werden, falls der Befragte mehr als fünf Probleme hatte und Erstbesitzer ist. Der **DO IF**-Befehl in *Tab. 7* löst das Problem.

Die Anweisungsfolge wird mit der Bezeichnung **DO IF** eingeleitet und mit der Bezeichnung **END IF** abgeschlossen. Die innerhalb der Struktur aufgeführten Transformationen werden nur dann ausgeführt, wenn die neben **DO IF** in

Klammern gesetzte Bedingung erfüllt ist. Hierbei stehen die Schlüsselwörter **EQ** für „equal to“ (gleich) und **GT** für „greater than“ (größer als).

In einer Erweiterung sollen sich nun die Werte für Zufriedenheit und Fahrzeugattraktivität nicht nur bei mehr als fünf Problemen verschlechtern, sondern auch bei fünf oder weniger Problemen verbessern. Diese Aufgabe kann durch die ergänzende Nutzung des Befehls **ELSE** gelöst werden. Hierbei werden die nach **ELSE** aufgeführten Transformationen nur dann ausgeführt, wenn die im **DO IF**-Befehl spezifizierte Bedingung nicht erfüllt ist. *Tab. 8* zeigt die entsprechende Syntax.

Der Übergang bei der Veränderung von Zufriedenheit und Fahrzeugattraktivität ist bei dem eben genannten Beispiel sehr abrupt. Hat ein Befragter bis zu fünf Probleme wird bei beiden Variablen jeweils der Betrag von zehn addiert, ansonsten abgezogen. Besser wäre es, wenn die Ausführung bestimmter Transformationen durch weitere Bedingungen variiert werden könnte. Dies gelingt durch Angabe des **ELSE IF**-Befehls. Hierbei dürfen beliebig viele **ELSE IF**-Befehle innerhalb der zwischen **DO IF** und **END IF** spezifizierten Anweisungsfolge angegeben werden, wobei die jeweils angeführten Bedingungen der Reihe nach überprüft werden. Ist eine Bedingung erfüllt, so werden genau die Transformationen durchgeführt, deren zugrundeliegenden Anweisungen unmittelbar hinter der zugehörigen **DO IF**- beziehungsweise **ELSE IF**-Anweisung folgen.

```
COMPUTE zufriedenheit_4 = customer_satisfaction_index_gesamt.
COMPUTE attraktivität_2 = fahrzeugattraktivität_index.
DO IF (q3_erstbesitzer EQ 1 AND anzahl_probleme_fahrzeug GT 5).
  COMPUTE zufriedenheit_4 = customer_satisfaction_index_gesamt
    - 10.
  COMPUTE attraktivität_2 = fahrzeugattraktivität_index - 10.
END IF.
EXECUTE.
```

Tab. 7: DO IF-Syntax

```
COMPUTE zufriedenheit_5 = customer_satisfaction_index_gesamt.
COMPUTE attraktivität_3 = fahrzeugattraktivität_index.
DO IF (q3_erstbesitzer EQ 1).
  DO IF (anzahl_probleme_fahrzeug GT 5).
    COMPUTE zufriedenheit_5 =
      customer_satisfaction_index_gesamt - 10.
    COMPUTE attraktivität_3 = fahrzeugattraktivität_index - 10.
  ELSE.
    COMPUTE zufriedenheit_5 =
      customer_satisfaction_index_gesamt + 10.
    COMPUTE attraktivität_3 = fahrzeugattraktivität_index + 10.
  END IF.
END IF.
EXECUTE.
```

Tab. 8: DO IF/ELSE-Syntax

# Digitalisierung verändert ganze Lebenswelten.



Portofreie Lieferung ≡ vahlen.de/25843807

**Müller-Seitz/Weiss**  
**Strategien zum Umgang mit der digitalen Transformation**

2019. 139 Seiten. Kartoniert € 24,90  
ISBN 978-3-8006-5923-4

## Das Buch

beleuchtet das Thema der digitalen Transformation aus Sicht der Managementforschung und -praxis. Dadurch wird das facettenreiche Phänomen der digitalen Transformation ganzheitlich und fundiert behandelt.

## Die Vorteile

- Besonders geeignet für mittelständische Unternehmen
- Verbindung aus Theorie und Praxis
- Managementorientierte Perspektive
- Buch zeigt die Ansätze eines "hidden champions" zur digitalen Transformation



*Wer sich die Expertise teurer Berater sparen will oder muss, kann zu diesem Buch greifen.*

in: Karriere (Berliner Morgenpost) 2.2.2019

Erhältlich im Buchhandel oder bei:  
**vahlen.de** | Verlag Franz Vahlen GmbH · 80791 München  
kundenservice@beck.de | Preise inkl. MwSt. | 170344

# Vahlen

## 6. Zusammenfassung und Ausblick

SPSS gehört zu den populärsten Statistikprogrammen im wirtschaftswissenschaftlichen Studium und in der Praxis. Die meisten Anwender führen ihre Analysen ausschließlich mit Hilfe der grafischen Benutzeroberfläche durch. Ihnen ist hierbei nicht bewusst, dass sie durch die **ergänzende** Verwendung der Syntax viele Prozeduren effizienter realisieren können. Manche Analysen lassen sich sogar ausschließlich mit der Syntax durchführen.

Natürlich kann in dieser knappen Einführung nur ein Bruchteil der Anwendungsgebiete der Syntax dargestellt werden. So bietet die Syntax eine Reihe weiterer Vorteile im Bereich der (bedingten) Datenmodifikation, der Daten-selektion oder dem Arbeiten mit Matrizen. Gleichzeitig sollte jedoch deutlich geworden sein, dass Anwender nicht zwingend die Syntaxsprache im Detail erlernen müssen, um ihre Vorteile zu nutzen. Schon **Syntax-Grundkenntnisse** machen die Arbeit mit SPSS schneller und einfacher.

Der Umfang von Syntax-Programmen kann sehr umfangreich werden, so dass sich der Wunsch nach mehr Übersichtlichkeit und Vereinfachung aufdrängt. Gleichzeitig wäre es angenehm, wenn man auf einzelne sorgfältig programmierte und fehlerfrei laufende Routinen ohne große Anpassungen zurückgreifen könnten und diese nicht jeweils neu schreiben oder umfangreich anpassen müssten. Das zweitgenannte Problem ist mit kopieren/einfügen und einigen Anpassungen in der jeweiligen Syntax zwar annäherungsweise in den Griff zu bekommen – dieses Vorgehen ist jedoch sowohl fehleranfällig als auch unnötig zeitaufwändig. Um die beiden beschriebenen Punkte elegant zu lösen, können Nutzer auf **Makros** zurückgreifen. Makros ermöglichen beispielsweise die Automatisierung von sich regelmäßig wiederholenden Prozeduren und Kombination mehrerer SPSS-Prozeduren und/oder Auswertungsschritte in einem Kurzbefehl. Durch den geschickten Einsatz von Makros lässt sich die Länge des Syntax-Codes um bis zu 90 % reduzieren. *Sarstedt et al. (2018)* bieten eine ausführliche Einführung in die Makro-Programmierung im Rahmen der SPSS-Syntax.

## Literatur

*Grotenhuis, M., Visscher, C.,* How to Use SPSS Syntax: An Overview of Common Commands, Thousand Oaks, CA, 2014.?

*Sarstedt, M., Mooi, E.,* Concise Guide to Market Research. The Process, Data, and Methods Using IBM SPSS Statistics, 3. Aufl., Berlin, Heidelberg 2019.

*Sarstedt, M., Schütz, T., Raithe, S.,* SPSS Syntax. Eine anwendungsorientierte Einführung, 3. Aufl., München, 2018.