

Grundlagen der SPSS-Befehlssyntax

1. Einführung	2
2. Syntaxregeln	3
3. Ausführen von Syntaxbefehlen	5
3.1. Laden und Speichern der Syntaxdateien	5
3.2. Ausführen von Befehlen	5
3.3. Prinzipielle Abfolge von Befehlsgruppen	6
3.4. Abschluss von Transformationen	8
3.5. Fortran-Formate	9
4. Hilfestellung bei der Notation von Befehlen	11
4.1. Einfügen	11
4.2. SPSS-Journaldatei	11
4.3. Syntax-Guide	12
5. Wichtige Befehle	13
6. Weiterführende Literatur	14

*Dokument erstellt am: 21. Juni 2004
Letztmalig geändert am: 23. Juni 2004*

1. Einführung

Wie bereits genannt, wurde SPSS ursprünglich für die Nutzung auf Großrechnern entwickelt; eine Möglichkeit der Interaktion, wie sie für SPSS für Windows typisch geworden ist, gab es noch nicht. Alle Rechenschritte mussten dem ausführenden SPSS-Programm schriftlich mitgeteilt werden. Diese Form der Kommunikation nennen wir SPSS-Syntax. Genau genommen handelt es sich hier um eine Programmiersprache, sie enthält nicht nur Befehle zur Ausführung von Berechnungen, sondern auch Anweisungen zum Programmablauf wie bedingte Programmteile und Programm-Schleifen.

Die SPSS-Syntax ist weiterhin die Basis des Programms SPSS für Windows, auch wenn sie nicht mehr so offen ersichtlich ist.

Die SPSS-Syntax kann auch weiterhin als Werkzeug eingesetzt werden, auch wenn dies einen gewissen Lernaufwand voraussetzt. Die Vorteile liegen auf der Hand:

1. Immer wiederkehrende Berechnungen lassen sich deutlich zeitsparender ausführen. Auch weniger Erfahrene können mit vorbereiteten Syntax-Skripten hieraus Vorteile erzielen.
2. Modifikationen an einzelnen Befehlen im Zuge des Testens geeigneter Analyseverfahren lassen sich zeitsparender ausführen.
3. Die SPSS-Syntax ist der einzige Weg, um alle Syntax-Features ausnutzen zu können.

Im Rahmen des SPSS-Einführungskurses kann es natürlich nicht das Ziel sein, die SPSS-Syntax vollständig zu erlernen. Es ist vielmehr das Ziel, das Werkzeug kennen zu lernen und einfache Syntax-Skripten interpretieren zu können. Es wird in diesem Skript, von wenigen Ausnahmen abgesehen, auch keine Erläuterung der Möglichkeiten einzelner Befehle gegeben. Dies würde den Rahmen der Einführung sprengen. Wenn der Leser dies dennoch wünscht, sei auf die zum Programm SPSS gehörenden Syntax-Guides verwiesen.

Zu Übungszwecken stehen das Übungsblatt 3 zum Einführungskurs in SPSS (Umgang mit der SPSS-Syntax) und eine vorgefertigte Syntax-Datei zur Verfügung.

2. Syntaxregeln

Die einzelnen Befehle werden in einer einfachen Textdatei notiert, ihre Dateierweiterung ist üblicherweise SPS.

Ein Befehl kann aus mehreren Bestandteilen aufgebaut sein. Diese sind

- das *Befehlsschlüsselwort*, z.B. FREQUENCIES, das einen Befehl einleitet,
- ein oder mehrere *Unterbefehle*, dies sind weitere, den Befehl spezifizierende Zusatzbefehle,
- *Spezifikationen*, dies sind die Parameter von Befehlen bzw. Unterbefehlen. Hierzu gehören Variablennamen, Zahlen, arithmetische Operatoren und weitere, für die Verwendung in SPSS definierte Schlüsselörter,
- (die oben genannten) *Schlüsselwörter*, die für die Formulierung von Spezifikationen definiert worden sind. Hierzu gehören auch Befehle zur Programmablaufsteuerung, wie bedingte Anweisungen und Programm-Schleifen.

SPSS unterscheidet nicht zwischen Groß- und Kleinschreibung; man kann aber die Groß- und Kleinschreibung dazu benutzen, um Befehle und Unterbefehle von den Spezifikationen besser unterscheiden zu können.

Folgende Syntaxregeln müssen eingehalten werden.

- Jeder Befehl beginnt auf einer neuen Zeile dergestalt, dass er unmittelbar am Zeilenanfang ohne Einfügung von Leerräumen notiert wird.
- Jeder Befehl sollte mit einem Punkt abgeschlossen werden.¹
- Eine Befehlszeile darf maximal 80 Zeichen (einschließlich der Leerräume) enthalten.
- Ein Befehl darf auf beliebig vielen Zeilen fortgesetzt werden, jede Folgezeile muss aber mit mindestens einem Leerraum eingerückt sein. Es bietet sich aus optischen Gründen an, bei Unterbefehlen eine neue Zeile zu beginnen.
- An Stellen, an denen ein Leerraum möglich ist, dürfen beliebig viele Leerräume bzw. ein Zeilenumbruch eingefügt werden.
- Für die meisten Befehlsschlüsselwörter genügt es, die ersten drei Buchstaben zu schreiben. Dennoch empfiehlt es sich der besseren Lesbarkeit wegen, diese Befehlsschlüsselwörter vollständig auszuschreiben.

¹ In der SPSS-Hilfdatei ist zwar angegeben, dass der Punkt optional sei. Es gibt aber immer wieder Fälle, in denen der Punkt als Endezeichen gefordert ist, anderenfalls wird eine Fehlermeldung ausgegeben.

- Unterbefehle werden in der Regel mit einem Schrägstrich („/“) unmittelbar vor dem Unterbefehl notiert; die Angabe des Schrägstriches vor dem ersten Unterbefehl ist optional.
- Variablennamen dürfen nicht abgekürzt werden. Variablennamen dürfen nicht identisch zu Befehlsschlüsselwörtern, Unterbefehlen oder sonstigen Schlüsselwörtern sein. In jedem Fall verbietet sich ein Punkt am Ende des Variablennamens, da er das Befehlsende vortäuscht.
- Zahlenangaben in den Spezifikationen dürfen nur den Punkt als Dezimaltrennzeichen benutzen, unabhängig von den Windows-Voreinstellungen.
- Zeichenketten, die in Hochkommata (Apostrophen, „‘“) bzw. Anführungszeichen notiert werden können, müssen auf der Zeile enden, auf der sie begonnen wurden, d.h., sie dürfen nicht auf eine neue Zeile umgebrochen werden.

Beispiele für zulässige Befehlsschreibweisen:

```
FREQUENCIES  
VARIABLES=job gender  
/PERCENTILES=25 50 75  
/BARChart.
```

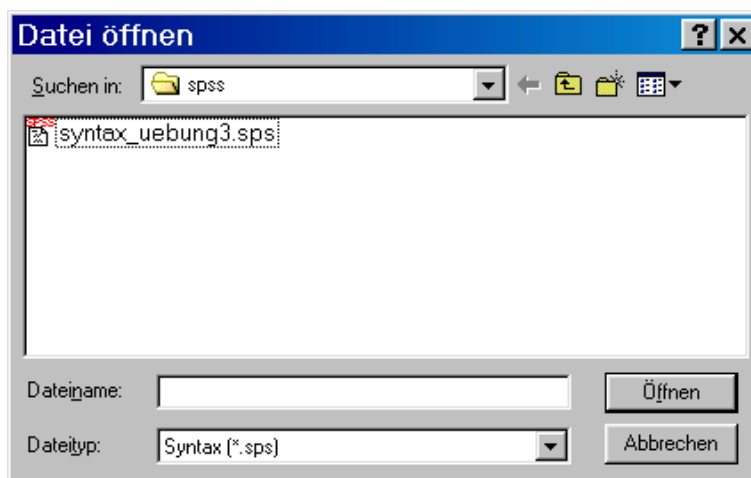
```
FREQ VARIABLES=job gender /PERCENTILES=25 50 75 /BARChart.
```

3. Ausführen von Syntaxbefehlen

3.1. Laden und Speichern der Syntaxdateien

Das Erstellen einer neuen Syntax-Datei geschieht mittels **Datei | Neu > Syntax**.

Das Öffnen einer bereits bestehenden Syntaxdatei erfolgt mittels **Datei | Öffnen > Syntax...** Es wird das bekannte Dateidialogfenster geöffnet.

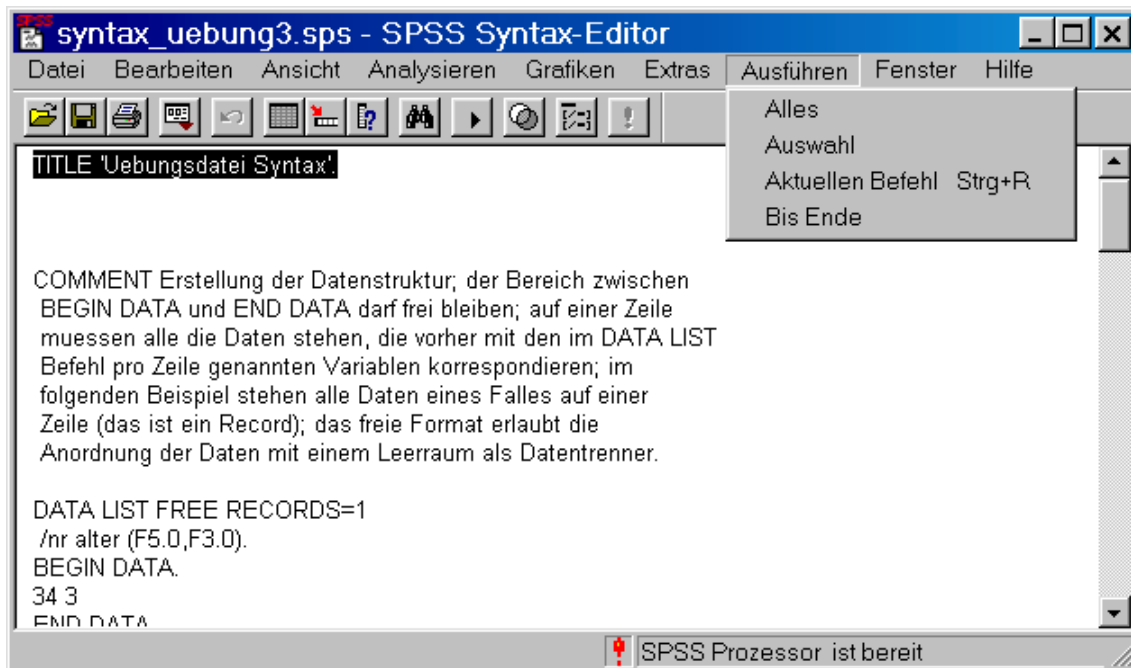



Die Syntax-Datei wird in einem neuen Fenster, dem Syntax-Fenster geöffnet. Es dürfen mehrere Syntax-Fenster gleichzeitig geöffnet bzw. verwendet werden.

Das Abspeichern der Syntax-Dateien erfolgt über **Datei | Speichern** oder **Datei | Speichern unter...**

3.2. Ausführen von Befehlen

Es ist sowohl möglich, die gesamte Syntax-Datei mit einem Male, den einzelnen, durch den Cursor markierten Befehl oder ausgewählte Blöcke, das sind markierte Teile der Datei, ausführen zu lassen.



Zum Ausführen nutzt man entweder die Befehle des Menüs **Ausführen** oder das -Symbol in der Symbolleiste. Letzteres Symbol dient dem Ausführen einer einzelnen Befehlszeile (keine Markierung, Cursorposition) oder eines markierten Befehlsblockes.

3.3. Prinzipielle Abfolge von Befehlsgruppen

Variablendefinitionen, Transformationen, Analysen usw. können erst dann ausgeführt werden, wenn bereits Daten zur Verfügung stehen bzw. Variablendefinitionen vorgenommen wurden. Anderenfalls, d.h. im Falle einer jungfräulichen Datenmatrix, erhält man eine Fehlermeldung ähnlich der nachfolgenden:

```
>Error # 100. Command name: NUMERIC
>This command is not permitted before the beginning of file definition
>commands.
>This command not executed.
```

Somit ergibt sich folgender Ablauf:

1. Datensatz einlesen bzw. Daten bereitstellen,
2. Ausführung von Berechnungen,
3. Abspeichern der Datenmatrix und der Ergebnisse.

Das Bereitstellen der Daten kann auf zweierlei Weise erfolgen:

1. Mit dem DATA-LIST-Befehl

```
DATA LIST FREE RECORDS=1
  /nr alter (F5.0,F3.0).
BEGIN DATA.
34 3
END DATA.
EXECUTE.
```

Der Befehl besteht streng genommen aus den vier Befehlen DATA LIST², BEGIN DATA, END DATA und EXECUTE.

Der Befehl DATA LIST stammt eigentlich noch aus Zeiten der Datenbereitstellung über Lochkarten (in unserem Falle auch einzelne Zeilen der Syntax-Datei), dabei konnten die Daten im festen Fortran-Datenformat vorliegen (FIXED) oder in freier Abfolge mit Leerzeichen als Trenner (FREE) notiert sein. Es folgt die Angabe der Lochkarten je Datensatz (RECORDS=Anzahl Lochkarten), auch hier gilt die maximale Zeichenzahl von 80 Zeichen pro Lochkarte. Mit den folgenden Unterbefehlen, eingeleitet mit „/“ oder „/Nummer“ der Lochkarte (z.B. „/1“, „/2“) werden die auf den einzelnen Lochkarten notierten Variablenwerte samt ihrem Format spezifiziert.

Der Block BEGIN DATA und END DATA umschließt die einzulesenden Daten. Dieser Block muss unbedingt vorhanden, kann aber leer, d.h. ohne Daten, sein.

² Wir haben hier bewusst eine einfache Form gewählt, das freie Format ermöglicht eine wahlfreie Angabe der Daten, und dennoch lassen sich die Variablentypen gleich in einem Schritt mitdefinieren. Etwas aufwändiger gestaltet sich das feste Format, wie es insbesondere auf Lochkarten verwendet werden musste. Hier müssen die vorgegebenen Formate unbedingt eingehalten werden, wie folgendes Beispiel zeigt:

```
DATA LIST FIXED RECORDS=1
  /nr alter (F5.0,F3.0).
BEGIN DATA.
  34 3
 1234 24
 546103
END DATA.
EXECUTE.
```

Das Aneinanderstoßen zweier aufeinander folgender Daten lässt sich durch einen einzugebenden Leerraum beseitigen, die Formatangabe wäre dann wie folgt zu ändern:

```
/nr alter (F5.0,1X,F3.0).
```

Für die Leerraumangabe wird keine Variable definiert.

Dieser Befehlsblock nimmt dann auch eine erste Definition von Variablen vor. Weitere Variablenspezifikationen und -definitionen können beliebig folgen.

2. Mit dem GET-Befehl

```
GET FILE='D:\Eigene Dateien\uebung_syntax.sav'.  
EXECUTE.
```

Der GET-Befehl dient zum Einlesen einer bereits bestehenden Datendatei (Datenmatrix, *.SAV).

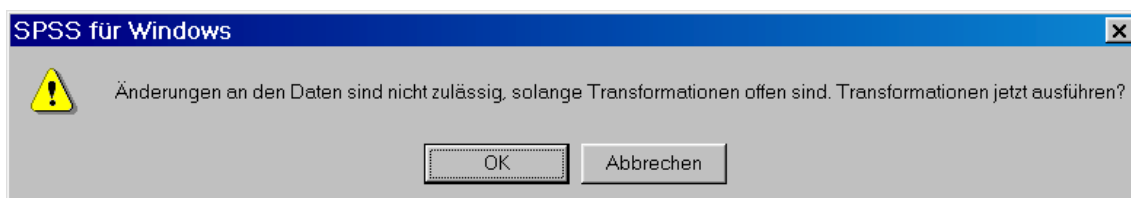
Die Variablendefinition in der Variablenansicht erspart die Bereitstellung von Daten bzw. Variablendefinition in der obigen Weise.

3.4. Abschluss von Transformationen

Eine Reihe von SPSS-Befehlen, z.B. das Einlesen von Daten, Datendefinitionen und Daten-Transformationsanweisungen wie Umkodieren, werden nicht sofort ausgeführt, sondern werden in der Schwebe gehalten, bis eine explizite Anweisung deren Ausführung erzwingt. Dieses Verfahren nennt man „transformation pending“. Dennoch wird in vielen Fällen deren (unmittelbare) Ausführung sinnvoll sein.

Der spezifikationslose Befehl EXECUTE dient hierzu, er bewirkt die Ausführung der zuvor aufgerufenen Befehle. Man stellt diesen Befehl üblicherweise an das Ende eines Blockes gemeinsam auszuführender Kommandos.

In jedem Fall ist es wichtig, offene, also nicht abgeschlossene Transformationen, zu beenden, wenn in der Datenmatrix Änderungen vorgenommen werden sollen. Man erhält eine entsprechende Fehlermeldung, wenn der Abschluss nicht ausgeführt wurde mit der Möglichkeit, dies zum jetzigen Zeitpunkt zu tun.



Analysen, auch statistische Prozeduren genannt, sind selbstaufführend. Wenn man sich nicht sicher ist, kann man auch an unnötigen Stellen die EXECUTE-Anweisung angeben. Sie schadet nicht, erspart aber diesbezügliche Fehlermeldungen.

3.5. Fortran-Formate

Die bei einigen Befehlen möglichen Datentypangaben, wie z.B. STRING und NUMERIC³, orientieren sich an der Schreibweise des Programms Fortran. Dem Buchstaben, der den eigentlichen Typ spezifiziert, folgt die Angabe der Breite der Variablen, gefolgt, wenn sinnvoll, von einem Punkt und der Anzahl der Nachkommastellen. Der Formatanweisung kann ein Wiederholungsfaktor vorangestellt sein.

Beispiele:

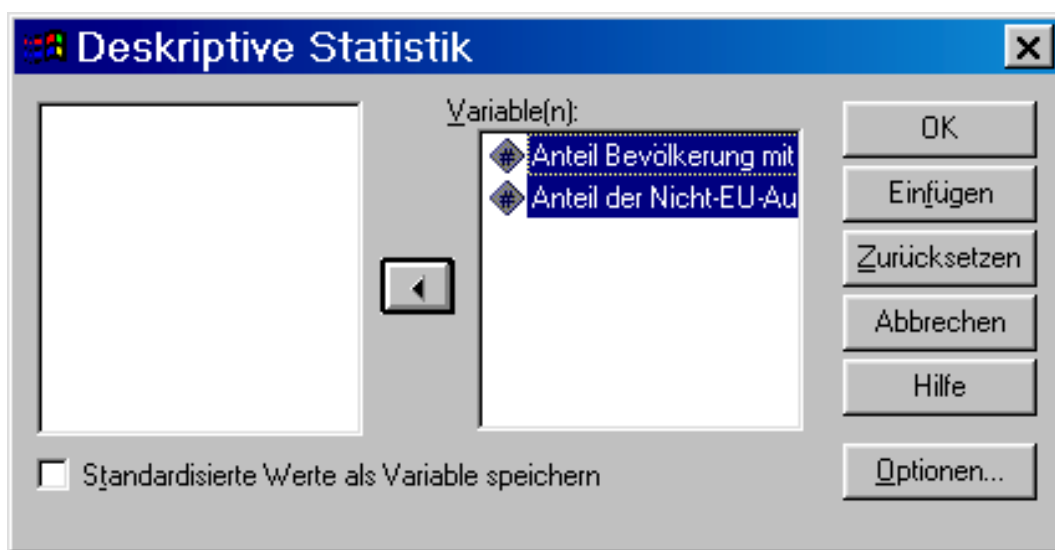
- A8 String (alphanumerische Zeichenkette) der Breite acht Zeichen.
- F8.2 Numerisches Format (reelle Zahlen) ohne Exponentendarstellung der Breite acht Zeichen, darin enthaltend zwei Nachkommastellen und das Komma selbst.
- 1X Format für eine Leerstelle.
- 5X Format für fünf aufeinander folgende Leerstellen.

³ Für die Variablentypen Punkt, Komma, wissenschaftliche Notation, Datum, Dollar, spezielle Währung gibt es keine Befehle. Diese Variablentypen müssen in der Variablenansicht verändert werden.

4. Hilfestellung bei der Notation von Befehlen

4.1. Einfügen

Die Vielzahl der Befehle erschwert es vielen Nutzern, dieses Werkzeug anzuwenden. Nachfolgend seien einige Hinweise gegeben, wie man recht effizient, die grafischen Bedienelemente nutzend, derartige Befehle erstellen kann.



Alle Dialogfenster der Menüs Transformieren, Analysieren und Grafiken enthalten neben dem Aktionsschalter „OK“ den Aktionsschalter „Einfügen“, wie obige Abbildung zeigt. Wird letzterer Aktionsschalter bedient, wird der Befehl nicht ausgeführt, aber der zugehörige Befehl in das Syntaxfenster geschrieben.

4.2. SPSS-Journaldatei

Alle Befehle der der Menüs Transformieren, Analysieren und Grafiken werden zusätzlich in der SPSS-Journaldatei SPSS.JNL in Befehlsform protokolliert. Dort lassen sich die benötigten Befehle herauskopieren. Die Lage und die Aufzeichnungsweise (über alle Sitzungen fortführend oder mit jeder neuen Sitzung die vorherigen überschreibend) finden Sie im Dialogfenster *Optionen* auf der Registerkarte *Allgemein*. Sie gelangen zu den Optionen über das **Bearbeiten**-Menü.

4.3. Syntax-Guide

Letztendlich stehen Ihnen im Menü **Hilfe | Syntax Guide** die vollständigen Beschreibungen aller Befehle zur Verfügung. Beachten Sie bitte bei der Installation, dass diese mit installiert werden, um einen schnelleren Zugriff zu diesen PDF-Dateien zu haben.

5. Wichtige Befehle

Gestaltung der Syntaxdatei und der Ausgabedatei

TITLE
COMMENT

Einlesen und Speichern von Daten

DATA LIST
GET
SAVE

Definition von Variablen

NUMERIC
STRING

VARIABLE LABELS
VARIABLE LEVEL
VARIABLE WIDTH
VALUE LABELS
MISSING VALUES

Transformationen

RECODE

Statistische Verfahren

DESCRIPTIVES
FREQUENCIES

Abschluss offener Transformationen

EXECUTE

Die Beschreibung der Befehle entnehmen Sie bitte dem Syntax-Guide.

6. Weiterführende Literatur

- Programm-Handbücher zu SPSS.
- Bühl, Achim; Zöfel, Peter: SPSS 12: Einführung in die moderne Datenanalyse unter Windows; [Analysen anhand von Fallstudien; mit dem Modul Tables; Neu: Multinominale logistische Regression]. München [u.a.]: Pearson Studium, 2004.
ISBN: 3-8273-7117-1
- Zöfel, Peter: SPSS-Syntax: Die ideale Ergänzung für effiziente Datenanalyse. – München [u.a.]: Pearson Studium, 2002.
ISBN: 3-8273-7052-3
- Syntax-Guides, zum Lieferumfang des Programms SPSS für Windows gehörend (Menü **Hilfe | Syntax-Guides**).