TU Dresden - Institut für Soziologie 28.11.06
FS: Berufsverläufe und Berufserfolg von Hochschulabsolventen
Dozent: Mike Kühne

Referent: Matthias Ritter

Wiederholung SPSS und Einführung in die SPSS Syntax

Gliederung:

1. Einleitung
2. Grundlagen der Bedienung von SPSS
3. Deskriptive Statistik
4. Syntax

1. Einleitung

Ursprünglich: **S**tatistical **P**ackage for **S**ocial **S**ciences
Heute: **S**uperior **P**erforming **S**oftware **S**ystem

Warum SPSS?

Vorteile	Nachteile
• weltweit am häufigsten eingesetzt - vor allem in Sozial- bzw. Wirtschaftswissenschaften und in der Marktforschung verbreitet • Windowsbasierte Benutzeroberfläche → einfache, gewohnte Bedienung • wichtigste Tests vorhanden	• jährliche Lizenz • bestimmte Tests sind nicht vorhanden • Programmierbarkeit (v.a. eingeschränkter Grafikmodus)

2. Grundlagen in der Bedienung von SPSS

Dateitypen: .sav - Datendatei
.spo - Ausgabe- Navigator = Ergebnisse der SPSS Prozeduren
.sps – Kommandosprache

Typische Arbeitsschritte in SPSS

1. Variablen definieren
2. Daten auf Eingabefehler kontrollieren
3. Daten transformieren/ neue Variabeln erzeugen/ Daten aggregieren oder verschmelzen
4. Fälle/ Variable auswählen
5. Kennzahlen zur Vorbereitung statistischer Analysen berechnen
6. Daten zur Vorbereitung von statistischen Analysen grafisch darstellen (deskriptive Statistik)
7. Daten mit geeigneten Verfahren analysieren

Skalenniveau

Entscheidend für die richtige Wahl und richtige Anwendung statistischer Verfahren ist oftmals das Skalenniveau! Hier noch mal zur Erinnerung:

Skalenart	mögliche Aussagen	Beispiele
Nominalskala	Gleichheit oder Ungleichheit	Geschlecht, Religion
Ordinalskala	größer-kleiner Relationen	Schulnoten, soziale Schichtung
Intervallskala	Gleichheit von Differenzen	Temperatur in C, IQ-Skala
Verhältnisskala	Gleichheit von Verhältnissen	Gewicht, Umsatz

Quelle (bzw. in Anlehnung an): Bortz, J.: Statistik für Sozialwissenschaftler, Berlin,1999 S. 31.

3. Deskriptive Statistik

Die deskriptive Statistik ist Ausgangspunkt einer jeden Analyse.

Aufgabe der Voranalyse ist es, einen ersten Eindruck der Verteilung zu gewinnen, mittels:

- Grafischer Darstellung
- Tabellarischer Darstellung von Häufigkeiten
- Maßzahlen für die Verteilung
- Identifizierung von Ausreißern
- Untersuchung auf Normalverteilung

4. Syntax

Prozeduren können, wie den meisten bekannt, sehr komfortabel über die graphische Benutzeroberfläche (Menüs, Dialogfelder) erfolgen. Es gibt jedoch noch eine andere Möglichkeit, und zwar über den Syntaxeditor. Mittels einer Befehlssprache kann man SPSS ebenfalls anweisen, Prozeduren auszuführen.
Beide Varianten haben ihre Vor- und Nachteile und daher bietet es sich an, sie miteinander zu kombinieren. So kann man mit Hilfe der Dialogfenster Anweisungen formulieren und sie sich im Syntaxeditor mittels der Funktion -Einfügen- anzeigen lassen. So ist es schnell möglich Befehle zu wiederholen, leicht abzuwandeln und schlussendlich zu dokumentieren.

Nutzen der SPSS – Syntax

1) Ermöglicht das Speichern der einzelnen Arbeitsschritte. Wichtig für die Nachvollziehbarkeit von Arbeitsschritten.
2) Ermöglicht effizienteres Arbeiten gerade bei aufwendigen Arbeitsschritten. Die direkte Eingabe von Befehlen mittels Durchklicken ist (oft) nicht mehr nötig. Außerdem können wiederkehrende Arbeitsschritte schnell durch Kopieren und Einfügen erledigt werden
3) Es gibt einige Befehle, die sich nur über die Syntax aufrufen lassen.

Grammatik der SPSS – Syntax:

- Der Punkt schließt den Befehl ab.
- Groß- und Kleinschreibung sind der Syntax egal (Jedoch Verwendung zur Übersichtlichkeit: Häufig werden `BEFEHLS`- und `FUNKTIONSNAMEN` groß geschrieben, während `varibalennamen` und `labels` klein geschrieben werden)
- Befehlsnamen kann man bis auf die 3 Anfangsbuchstaben abkürzen `VARIABLE LABELS = VAR LAB`
- Leerzeichen sind beliebig oft möglich, jedoch muss mindestens eins zwischen 2 Wörtern stehen
- Text muss in Anführungszeichen (`"Text" / 'Text'`) stehen, und darf sich nicht über mehrere Zeilen erstrecken
- Zahlen stehen nicht zwischen Anführungszeichen, es sei denn sie stellen eine Textvariable dar.
- Dezimalzahlen werden mit Punkt geschrieben nicht mit Komma.
- In einer Liste von Zahlen- oder Textwerten, werden die einzelnen Werte mit Komma getrennt. `1,2,3,4.9,5.2, .... „ja", „nein", „vielleicht"`
- Gibt man mehrere Variablen an, können diese einfach durch ein Leerzeichen getrennt geschrieben werden `variable1 variable2`
- Kommentare werden durch `/* Kommentar */` von der Syntax getrennt

Wichtige Befehle

- Der Befehl COMPUTE generiert eine neue Variable.
- Der Befehl VARIABLE LABELS definiert für diese Variable das Variablenlabel.

Beispiel 1:

```
COMPUTE alter = 2006-fj2.                    * eine neue Variable alter wird mit der Rechenoperation
                                               2006 – fj2 (der Variable Geburtsjahr) erstellt.
VARIABLE LABEL alter 'Alter der Befragten'.  * der (neuen) Variablen alter wird das Variablenlabel ‚Alter
                                               der Befragten' zugewiesen.
EXECUTE.
```

- Der Befehl RECODE kodiert die Werte einer Variablen um und schreibt die Werte in eine neue Variable. (Diese Variable wird automatisch neu erstellt)
- mit dem Befehl VALUE LABLE werden die angeführten Wertelabels für die Variable definiert.
- Der Befehl FREQUENCIES erstellt eine Häufigkeitstabelle für die Variable.

Beispiel 2:

```
RECODE  alter
  (25 thru 32=1)  (33 thru 40=2)  (41 thru 48=3)  (49 thru 56=4)
 INTO  alter_kat .

VALUE LABELS alter_kat 1 '25 - 32'
                       2 '33 - 40'
                       3 '41 - 48'
                       4 '49 - 56'.

VARIABLE LABELS alter_kat 'Alterskategorisierung'.
EXECUTE .


FREQUENCIES
  VARIABLES=alter_kat.
  /ORDER=  ANALYSIS .
E>XECUTE.
```

- Ein weiterer nützlicher Befehl ist der IF Befehl, er dient zur Berechnung von Bedingungen. Abhängig davon ob die Bedingung wahr ist, wird die Variable in die Berechnung einbezogen.

Beispiel 3:

```
COMPUTE zielgr=0.                    * eine neue variable zielgr wird erstellt
                                       und alle Werte werden gleich 0 gesetzt

IF (alter>=35) zielgr = 1.           * Alle Befragten über 34 erhalten den Wert 1 in zielgr.

EXECUTE .
```

- Eine Erweiterung des IF- Befehls ist der DO IF Befehl. Er ermöglicht neben den Berechnungen von COMPUTE und IF auch andere Transformationen, wie RECODE oder COUNT Befehle in Abhängigkeit zu einer Bedingung auszuführen. Die Befehlsstruktur ermöglicht es mehrere Bedingungen zu formulieren.

```
DO IF          (Bedingung1).   Befehl für den Fall, dass Bedingung 1 wahr ist.
ELSE IF        (Bedingung2).   Befehl für den Fall, dass Bedingung 2 wahr ist.
ELSE           .          Befehl für den Fall, dass alle Bedingungen davor falsch sind.
END IF.
```

- Außerdem lassen sich mehrere DO IF .. END IF. Befehlsstrukturen ineinander verschachteln, so dass auch komplexe Bedingungsgeflechte mit mehreren Entscheidungsebenen entstehen können.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.