

EINFÜHRUNG IN DIE ARBEIT MIT DEM
SOZIO-OEKONOMISCHEN PANEL (SOEP)
IN SPSS

1. SPSS für Windows 10.0 - Vorbemerkung.....	2
1.1. Der Daten-Editor.....	2
1.2. Der Syntax-Editor.....	4
1.3. Der Ausgabe-Navigator	5
2. Daten-Management des SOEP	5
2.1. SPSS-Befehle und Syntax.....	6
• aggregate	6
• compute	7
• do if - end if	7
• do repeat - end repeat.....	8
• execute.....	9
• get	9
• if	9
• match files.....	10
• missing values.....	11
• select if.....	11
• sort cases.....	11
• split file	12
• temporary	12
2.2 Verknüpfen von Haushaltsdaten mit Individuen.....	13
2.3 Verknüpfen von Individualdaten mit Haushalten.....	15
2.4 Verknüpfen von Daten über Individuen.....	16
Literatur:.....	18

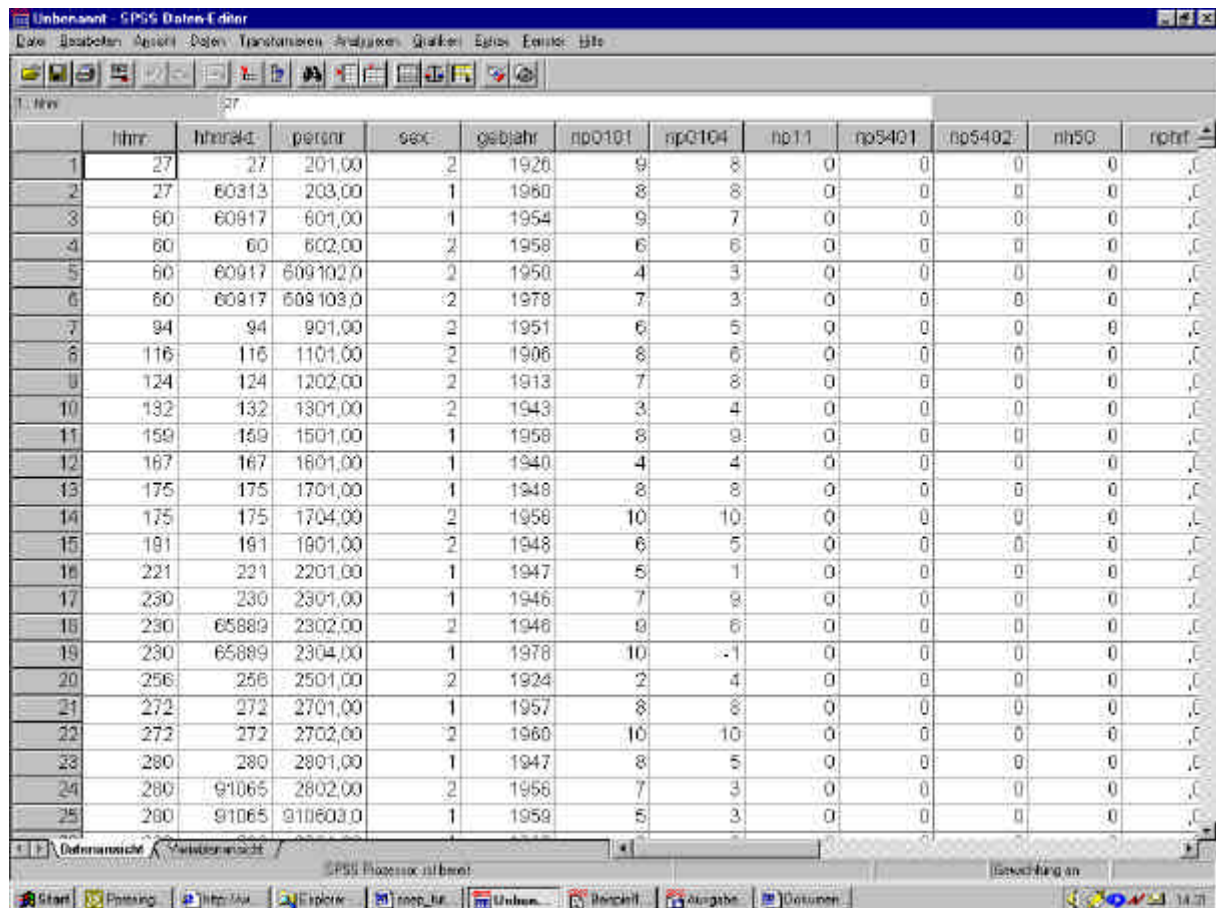
Guido Heineck
Otto-Friedrich-Universität Bamberg
Feldkirchenstr. 21 / Raum 217
96047 Bamberg
guido.heineck@sowi.uni-bamberg.de

1. SPSS für Windows 10.0 - Vorbemerkung¹


Dieser Leitfaden dient nicht als Ersatz für eine *echte* SPSS-Einführung.² Für Grundlagen sei auf die empfohlene Literatur hingewiesen. Das Arbeiten mit SPSS (hier in Version 10.0) wird allerdings durch die geläufige Windows-Oberfläche vereinfacht.

1.1. Der Daten-Editor

Der Daten-Editor dient u.a. zur direkten Eingabe von Daten oder zum Einlesen bereits vorhandener Daten. Die hier verwendeten SOEP-Daten liegen bereits in SPSS-Format vor (*.sav), die Variablen sind in den Spalten abgelegt, die Fälle in den Zeilen.




	hhnr	hhnrakt	persnr	sex	gebjahr	np0101	np0104	np11	np5401	np5402	nh50	nprtr
1	27	27	201,00	2	1926	9	8	0	0	0	0	,C
2	27	60313	203,00	1	1960	8	8	0	0	0	0	,C
3	60	60817	601,00	1	1954	9	7	0	0	0	0	,C
4	60	60	602,00	2	1958	6	6	0	0	0	0	,C
5	60	60917	609102,0	2	1950	4	3	0	0	0	0	,C
6	60	60817	608103,0	2	1978	7	3	0	0	0	0	,C
7	94	94	901,00	2	1951	6	5	0	0	0	0	,C
8	116	116	1101,00	2	1906	8	6	0	0	0	0	,C
9	124	124	1202,00	2	1913	7	8	0	0	0	0	,C
10	132	132	1301,00	2	1943	3	4	0	0	0	0	,C
11	159	159	1501,00	1	1958	8	9	0	0	0	0	,C
12	167	167	1601,00	1	1940	4	4	0	0	0	0	,C
13	175	175	1701,00	1	1948	8	8	0	0	0	0	,C
14	175	175	1704,00	2	1958	10	10	0	0	0	0	,C
15	191	191	1901,00	2	1948	6	5	0	0	0	0	,C
16	221	221	2201,00	1	1947	5	1	0	0	0	0	,C
17	230	230	2301,00	1	1946	7	9	0	0	0	0	,C
18	230	65889	2302,00	2	1948	9	6	0	0	0	0	,C
19	230	65889	2304,00	1	1978	10	-1	0	0	0	0	,C
20	256	256	2501,00	2	1924	2	4	0	0	0	0	,C
21	272	272	2701,00	1	1957	8	8	0	0	0	0	,C
22	272	272	2702,00	2	1960	10	10	0	0	0	0	,C
23	280	280	2801,00	1	1947	8	5	0	0	0	0	,C
24	280	91065	2802,00	2	1956	7	3	0	0	0	0	,C
25	280	91065	910603,0	1	1958	5	3	0	0	0	0	,C

Über das Symbol  kann zwischen der Darstellung der Werte und der der Wertelabels gewechselt werden. (Für die Variable sex wird somit entweder '1' und '2' oder 'männlich' und 'weiblich' ausgegeben.)

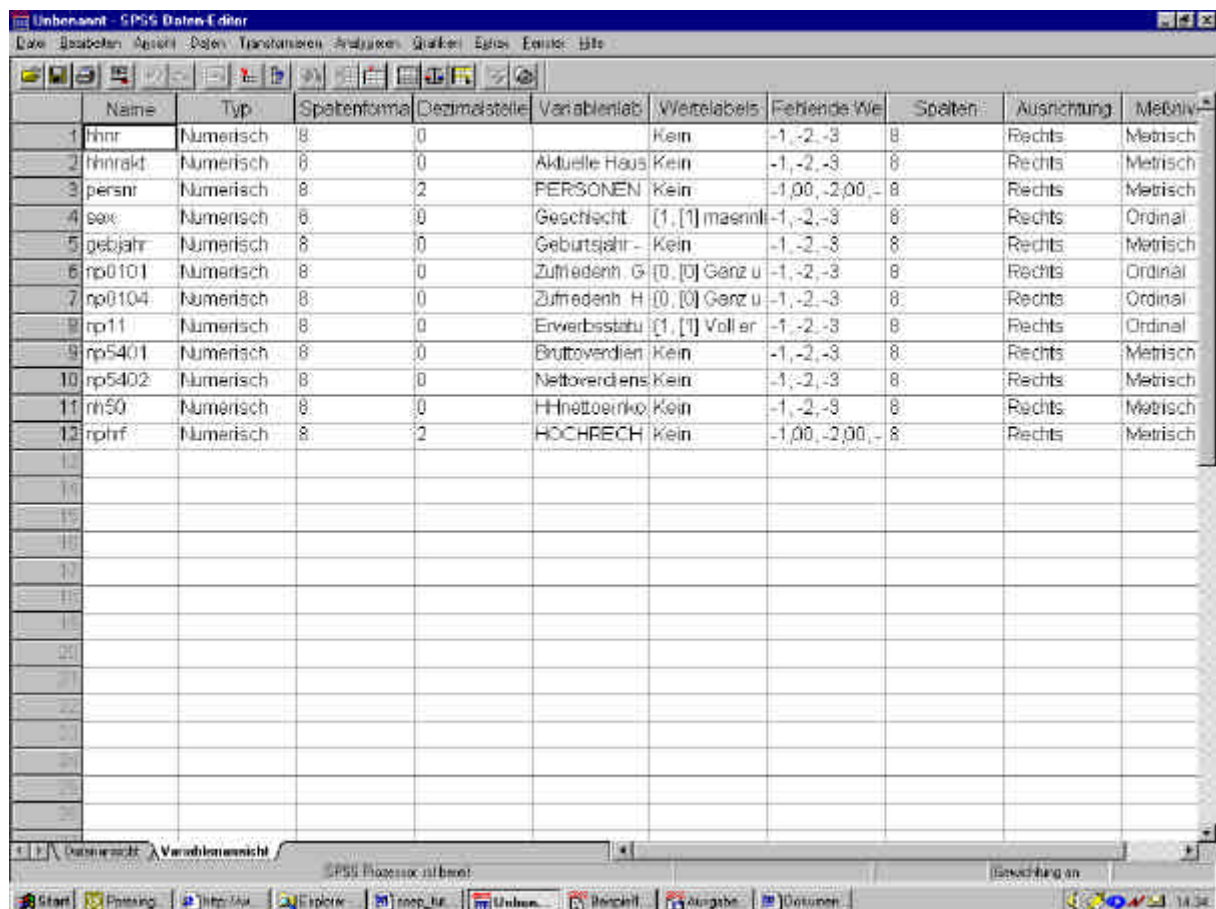
¹ Kapitel 1 und Kapitel 2.1 sind ein 'Konzentrat' aus: Angele, 'SPSS für Windows 7.5.2' und dem 'SPSS base syntax reference guide'.

² Das Rechenzentrum der Universität Bamberg bietet entsprechende Kurse an.


Informationen zu den Variablen kann man entweder über das Menü 'Extras' → 'Variablen' bzw. das Symbol  erhalten. Angezeigt werden sodann die in der Arbeitsdatei vorhandenen Variablen und eine Variablenbeschreibung.



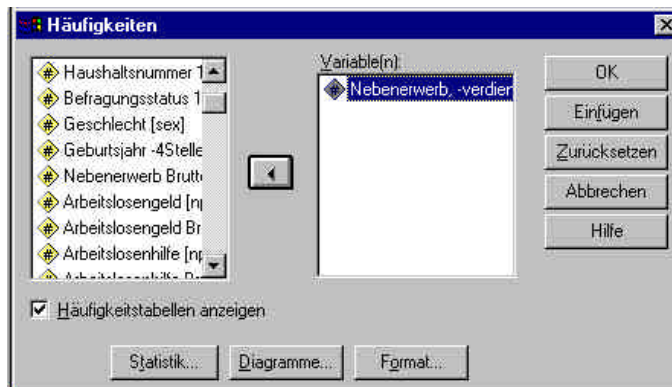
Über den Karteireiter ‚Variablenansicht‘ lässt sich im Daten-Editor zudem zu einer Mapper umschalten, in der ebenfalls Informationen zu den Variablen abgelegt sind. Hier lassen sich außerdem Variablen-Manipulationen vornehmen (z.B. das Anpassen von Dezimalstellen oder das Definieren geeigneter fehlender Werte).



1.2. Der Syntax-Editor


SPSS-Befehle können 'zu Fuß' oder über 'point-and-click' Operationen in den Syntax-Editor eingegeben werden. Letzteres ist zu empfehlen bei SPSS-Befehlen, mit denen man nicht vertraut ist. Direkte Eingaben in den Syntax-Editor helfen jedoch dabei, sich darüber klar zu werden, welche Operationen man auszuführen gedenkt. Auch können Programme damit 'häppchenweise' ausgeführt werden. Ferner gibt es SPSS-Befehle, wie z.B. TEMPORARY, die ausschließlich über die Syntax verfügbar sind. Hilfreich ist, dass in vielen Fällen die Möglichkeit besteht, über das Menü 'Einfügen' die zum jeweiligen Befehl gehörige Syntax im Syntax-Editor zu generieren, die dann dort über 'Ausführen' oder über das Symbol  aktiviert werden muss. Vorab müssen alle bzw. die interessierenden Befehle markiert werden.

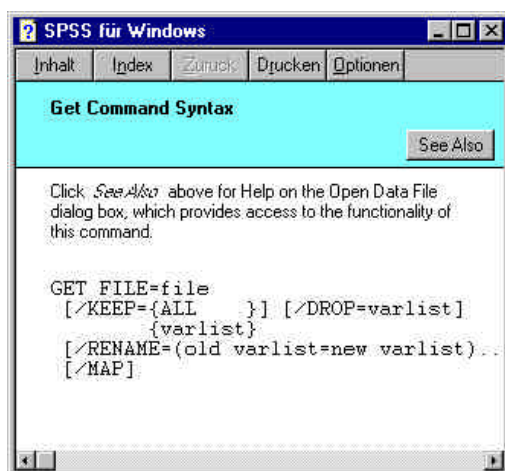
Bsp:



Durch Anklicken der Option 'Einfügen' erhält man die Syntax:

```
FREQUENCIES  
VARIABLES=np5801  
/ORDER ANALYSIS .
```

Über das Symbol  ist es möglich, ein Hilfe-Menü aufzurufen, welches Informationen zur Syntax-Struktur des betrachteten Befehls bereitstellt.



GET FILE ist vorab zu markieren, ohne Markierung wird ein Index für Syntaxdiagramme angezeigt.

Abgespeichert werden SPSS-Programme als Syntax-Dateien (*.sps).

1.3. Der Ausgabe-Navigator

Hier werden die Ergebnisse von Analysen angezeigt. Das Fenster wird bei Ausführung einer Prozedur automatisch geöffnet.

The screenshot shows the SPSS Viewer window with a pivot table titled 'Familienstand * Erwerbsstatus Kreuztabelle'. The table displays counts and percentages for different family statuses across various employment categories.

		Dünn		Fertig		Gesamt	
		N	Prozent	N	Prozent	N	Prozent
		13263	100,0%	0	,0%	13263	100,0%
		Familienstand * Erwerbsstatus Kreuztabelle					
						Er	
		Voll erwerbstätig		Teilzeit	Ausbildung, Lehre	Geringfügig beschäftigt	
Verheiratet zusammenlebend	Anzahl	3878	7,80	27	1,43		
	% der Gesamtzahl	29,2%	5,9%	,3%	1,4%		
Verheiratet getrenntlebend	Anzahl	113	,25	1	,05		
	% der Gesamtzahl	,9%	,2%	,0%	,0%		
Ledig	Anzahl	1473	11,1	205	1,43		
	% der Gesamtzahl	11,1%	,8%	80%	5%		
Geschieden	Anzahl	375	,71	5	,03		
	% der Gesamtzahl	2,8%	,5%	,0%	,1%		
Verwitwet	Anzahl	81	,19	1	,01		
	% der Gesamtzahl	,6%	,2%	,0%	,1%		
Gesamt		Anzahl	9918	1021	139	373	
		% der Gesamtzahl	44,6%	7,7%	3,3%	2,1%	

Im linken Fensterausschnitt befindet sich eine Gliederung der Objekte, die im rechten Ausschnitt zu sehen sind. Durch Anklicken in der Gliederung können Objekte gezielt ausgewählt werden, um sie z.B. zu entfernen oder um sie in eine andere Reihenfolge zu bringen. Zum Bearbeiten einzelner Objekte im rechten Fensterausschnitt müssen diese doppelt angeklickt werden (z.B. zur Veränderung von Texten oder zum Vertauschen von Spalten und Zeilen in Tabellen).

Ausgaben können als Ausgabe-Dateien (*.spo) abgespeichert werden.

2. Daten-Management des SOEP

Die hierarchische Struktur des SOEP (die Daten sind für jedes Jahr und für jedes Analyse-Niveau, wie Person, Haushalt etc. separat abgelegt) macht die 'Kunstfertigkeit' des richtigen Verknüpfens zum Kern des Daten-Managements. Nachfolgend werden die für diese Aufgabe häufig gebräuchlichen Befehle beschrieben (ohne Anspruch auf Vollständigkeit?!).

Grundsätzlich gilt, dass ein SPSS-Kommando mit einem Kommandoschlüsselwort zu beginnen hat und durch Subkommandos (und Spezifikationen, die hier nicht weiter von Interesse sind) erweitert werden kann. Abzuschließen sind SPSS-Kommandos mit einem Punkt.

Eine SPSS-Session ist logisch aufzubauen, da die Befehle sequentiell abgearbeitet werden. Die Befehle sollten also in der Reihenfolge stehen, in der ihre Ergebnisse benötigt werden. Allgemein gilt, dass nur eine definierte Variable modifiziert und/oder ausgewertet werden kann. Eine SPSS-Session hat demnach prinzipiell die folgende Struktur:

1. Befehle zur Datenbeschreibung (Datendefinition) [z.B. GET]
2. Befehle zur Datenmodifikation [z.B. COMPUTE]
3. Befehle zur Datenanalyse [z.B. FREQUENCIES]

Grundlegend sollte zudem eine sorgfältige Dokumentation der Arbeitsschritte vorgenommen werden – dies dient nicht nur zu Zwecken intersubjektiver Nachvollziehbarkeit, sondern um vielmehr den eigenen arbeitsorganisatorischen Unzulänglichkeiten (*wie bloß hab ich das damals noch mal gemacht???*) entgegen zu wirken. Hierzu bietet sich das Einfügen von Kommentaren an, die von SPSS als solche behandelt werden, d. h. nicht als auszuführende Befehle gewertet werden:

/ Alles, was zwischen den ‚Klammern‘ steht, interpretiert SPSS als Kommentar */*

2.1. SPSS-Befehle und Syntax

Die hier interessierenden Befehle im Einzelnen (alphabetisch geordnet):

- **aggregate**

aggregiert Gruppen von Fällen in der Arbeitsdatei zu einzelnen Fällen in einer neuen, aggregierten Datei. Die Werte einer oder mehrerer sogenannter 'break'-Variablen in der Arbeitsdatei definieren hierbei die Fallgruppen. Eine Menge von Fällen mit identischen Werten für eine 'break'-Variable wird 'break'-Gruppe genannt. In der neu erzeugten aggregierten Datei weisen die Variablen sodann einen Wert für jede 'break'-Gruppe auf. Wahlweise kann man eine neue Arbeitsdatei erzeugen, die die bisherige ersetzt und ohne vorheriges Abspeichern nicht für weitere Analysen zur Verfügung steht oder man erstellt über das Subkommando /OUTFILE eine neue

SPSS-Datei, die man eigens aufrufen muss, um damit arbeiten zu können. Neben dem Subkommando /OUTFILE und der Angabe der 'break'-Variablen ist wenigstens eine Aggregationsfunktion wie SUM (arguments) oder MEAN(arguments) und eine Quellvariable zu nennen.

Syntax-Struktur:

AGGREGATE

```
/OUTFILE = { * }           (Asterisk: ersetzt die Arbeitsdatei)
           { file }        (file-name: erzeugt neue SPSS-Datei)
/BREAK = varlist
/aggvar = function(arguments)
```

Bsp.: *AGGREGATE*

```
/OUTFILE = 'c:\temp\gek.sav'
/BREAK   = hhnrakt
/hheink  = sum(indeink)
```

- **compute**

erzeugt neue numerische Variablen oder modifiziert die Werte existierender numerischer Variablen (oder von string-Variablen). Durchgeführt werden können arithmetische Operationen sowie arithmetische und statistische Funktionen.

Syntax-Struktur:

```
COMPUTE Zielvariable = Ausdruck
```

Bsp.: *COMPUTE hheink = np5402 + np5802.*

- **do if - end if**

basierend auf einem oder mehreren logischen Ausdrücken führt DO IF - END IF Transformationen auf Teilmengen von Fällen aus. Mit dem optionalen Befehl ELSE IF können dann Transformationen durchgeführt werden, für die der logische Ausdruck unter DO IF nicht zutrifft.

Derlei logische Ausdrücke können u.a. die im folgenden beschriebenen relationalen und logischen Operatoren sein.

Die relationalen Operatoren sind:

EG	(equal to)	oder	=
NE	(not equal to)	oder	<>
LT	(less than)	oder	<
LE	(less than or equal to)	oder	<=
GT	(greater than)	oder	>
GE	(greater than or equal to)	oder	>=

Die logischen Operatoren sind:

AND	oder	&	beide Relationen haben wahr zu sein
OR	oder	x	eine der Relationen hat wahr zu sein
NOT			kehrt das Ergebnis eines Ausdrucks um

Syntax-Struktur:

```
DO IF [ logischer Ausdruck ]  
  Transformationsbefehl(e)  
  [ ELSE IF [ logischer Ausdruck ] ]  
  Transformationsbefehl(e)  
  $$  
  [ ELSE ]  
  Transformationsbefehl(e)  
END IF
```

Bsp.: *DO IF (vne GE 0 AND hheink GT 0).*
*COMPUTE vneant = vne/hheink*100.*
ELSE IF (vne EQ -1 OR hheink LT 0).
COMPUTE vneant = -9.
END IF.

- **do repeat - end repeat**

führt die selben Transformationen für die spezifizierte Variablenmenge durch und verringert somit die Anzahl der einzugebenden Kommandos, jedoch nicht die Anzahl der auszuführenden Kommandos. Benutzt wird eine sogenannte 'stand-in' Variable als Platzhalter für die Liste der Variablen, auf die die Transformationen angewendet

werden sollen. Zulässige Transformationen sind u.a. COMPUTE, IF, SELECT IF, DO IF.

Syntax-Struktur:

```
DO REPEAT stand-in var = {varlist}
  Transformationskommandos
END REPEAT
```

Bsp.: *DO REPEAT dummy1 = wgeld kgeld sozh pfvers*
 /dummy2 = nh4902 nh4905 nh4907 nh4909.
 IF (dummy2 GE 0) dummy1 = dummy2.
 END REPEAT.

- **execute**

werden Transformations-Befehle wie ADD FILES oder MATCH FILES manuell in den Syntax-Editor eingegeben, so führt EXECUTE lediglich dazu, dass die Anweisungen gelesen werden. Ein weiterer Einfluss auf die SPSS-Session besteht nicht. Alternativ kann man im Daten-Editor über das Menü 'Transformieren' ⇨ 'Offene Transformationen ausführen' das Datenlesen in Gang setzen.

Syntax-Struktur:

```
EXECUTE
```

- **get**

liest vorhandene SPSS-Daten in den Dateneditor ein.

Syntax-Struktur:

```
GET FILE = file
      [/KEEP = {varlist}]
```

Bsp: *GET FILE = 'c:\soep\ppfad.sav'*
 /KEEP = persnr.

- **if**

führt einzelne Transformationsbefehle aus, die auf logischen Bedingungen (vgl. die Erläuterungen zu logischen Ausdrücken im Abschnitt zu 'do if - end if') basieren. Es

können damit neue Variablen erzeugt oder die Werte bestehender Variablen modifiziert werden.

Syntax-Struktur:

IF [logischer Ausdruck] Zielvariable = Ausdruck

Bsp.: *IF (alter > 20 AND sex EQ 1) gruppe = 2.*

- **match files**

verknüpft die Arbeitsdatei mit einer oder mehreren Datendateien, um Fälle und/oder (in erster Linie) Variablen hinzuzufügen. Bei sogenannten parallelen Verknüpfungen werden die Fälle gemäß ihrer Reihenfolge in den jeweiligen Dateien miteinander verknüpft. Treten Variablen in allen zu verknüpfenden Dateien auf, so wird die Variable von der zuerst spezifizierten Datei ausgewählt. Haben Dateien eine ungleiche Anzahl von Fällen, so ist die längste Datei maßgeblich. Fälle, die nicht in den kürzeren Dateien existieren, werden als 'system-missing' für solche Variablen geführt, die nur in den kürzeren Dateien erfasst sind. Parallele Verknüpfungen sind mit dem Subkommando /FILE auszuführen.

Für die Arbeit mit dem SOEP sind indes nicht-parallele Verknüpfungen gebräuchlicher. Hierbei werden die Dateien entsprechend den Werten einer oder mehrerer Schlüsselvariablen, die zum Identifizieren und richtigen Zuordnen der Fälle dienen, miteinander verknüpft. Dies bedingt, dass die zu verknüpfenden Dateien aufsteigend nach diesen Schlüsselvariablen sortiert sein müssen. Die Schlüsselvariablen werden durch das Subkommando /BY, welches dem Subkommando /FILE bzw. /TABLE zu folgen hat, festgelegt. Das Subkommando /TABLE legt eine sogenannte 'table lookup'-Datei fest, die keine weiteren Fälle, sondern ausschließlich Variablen zur neuen Arbeitsdatei beiträgt. Stimmen die Werte für die Schlüsselvariablen überein, so werden die Variablen der 'table-lookup'-Datei den Fällen der anderen Dateien hinzugefügt. Das Subkommando /FILE legt hierbei die Datei(en) fest, die die Fälle bereitstellen.

Syntax-Struktur:

MATCH FILES {FILE} = file

[/{FILE} = ...] oder [/{TABLE} = ...]

[/KEEP {varlist}]

[/BY varlist]

Bsp: *MATCH FILES FILE = ** (ein Asterisk nimmt Bezug auf die Arbeitsdatei)

/TABLE = 'c:\soep\np.sav'

/KEEP = persnr np5401

/BY persnr.

- **missing values**

belegt Variablen mit sogenannten 'user-missing'-Werten. Fälle, die solche fehlende Werte aufweisen, werden bei 'normalen' Analysen ausgeschlossen, können aber auch, je nach Untersuchungsgegenstand, gezielten Analysen unterzogen werden. Im SOEP erscheinen als Standard folgende 'missings' :

-1: keine Angabe; -2: trifft nicht zu; -3: unplausible Angabe.

Syntax-Struktur:

```
MISSING VALUES {varlist} {value list}
                {ALL}
```

Bsp.: *MISSING VALUES all (-9).*

- **select if**

wählt permanent Fälle aus, die den spezifizierten logischen Bedingungen unterliegen. Derlei logische Ausdrücke können die unter 'DO IF - END IF' beschriebenen relationalen und logischen Operatoren sein sowie arithmetische Operationen und arithmetische und statistische Funktionen.

Syntax-Struktur:

```
SELECT IF [ (logischer Ausdruck) ]
```

Bsp.: *GET FILE = 'c:\soep\ppfad.sav'.*

SELECT IF (sex EQ 1 AND nnetto EQ 1).

- **sort cases**

ordnet die Reihenfolge der Fälle der Arbeitsdatei basierend auf den Werten einer oder mehrerer Variablen. Die Standardeinstellung bewirkt eine Sortierung in aufsteigender Reihenfolge, wahlweise kann aber auch absteigend sortiert werden. Wird

nach mehreren Variablen sortiert, so wird mit der zuerst genannten begonnen, die Fälle aller nachfolgenden Variablen werden in den Kategorien der jeweils zuvor genannten Variablen geordnet.

Syntax-Struktur:

<code>SORT CASES [BY] varlist [{A}]</code>	A: aufsteigend (Standard)
<code>{D}</code>	D: absteigend

Bsp.: *SORT CASES BY hhnr hhnrakt persnr.*

- **split file**

teilt die Arbeitsdatei in Untergruppen auf, die dann separat analysiert werden können und bietet sich in gemeinsamer Verwendung mit dem Befehl 'TEMPORARY' an. Die Fälle, die den gleichen Wert für die 'split'-Variable ausweisen, bilden eine Untergruppe, jeder Wert jeder 'split'-Variablen stellt eine 'break'-Gruppe dar. Zumeist ist vor der Anwendung des 'SPLIT FILE'-Befehls die Arbeitsdatei nach der 'split'-Variablen zu sortieren, da die Fälle nach 'break'-Gruppen geordnet sein müssen.

Syntax-Struktur:

`SPLIT FILE BY {varlist}`

Bsp.: *SORT CASES BY sex.*

SPLIT FILE BY sex.

FREQUENCIES vars = income /statistics = median.

- **temporary**

zeigt den Beginn temporärer Transformationen an, die lediglich für die auf diesen Befehl folgende Prozedur in Kraft sind. Werden nachstehend neue Variablen kreiert oder bereits existierende Variablen modifiziert, so hat dies ebenfalls einen nur temporären Effekt. Zulässige Kommandos sind unter anderem: COMPUTE, IF, DO REPEAT, DO IF, SELECT IF, SPLIT FILE.

Syntax-Struktur:

`TEMPORARY`

Bsp.: *SORT CASES BY sex.*

TEMPORARY.

SPLIT FILE BY sex.

FREQUENCIES vars = income /statistics = median.

2.2 Verknüpfen von Haushaltsdaten mit Individuen³

Um Haushaltsdaten mit Individuen zu verknüpfen, bietet sich die Prozedur 'MATCH FILES ... /TABLE = ...' an.

```
Bsp.: /*      Verknüpfen von Individual- mit Haushaltsdaten          */
      /*      Zunächst: Individualdaten aus NP.SAV                */
      GET FILE = 'c:\soep\np.sav'
          /KEEP = hhnr hhnrakt persnr np0104 np11 np5401 np5402

      /*      Verknüpfen mit NH.SAV: Haushaltseinkommen          */
      MATCH FILES /FILE = *
          /TABLE = 'c:\soep\nh.sav'
          /KEEP = hhnr hhnrakt persnr np0104 np11 np5401 np5402 nh50
          /BY hhnr hhnrakt.
```

Im Beispiel werden aus der 'NP.SAV'-Datei zunächst neben den Identifikatoren (hhnr, hhnrakt und persnr) auch Variablen in die Arbeitsdatei geladen, die Auskunft über den monatlichen Verdienst (np5401, np5402; brutto, netto) sowie über den Erwerbsstatus (np11) und die Zufriedenheit mit dem Haushaltseinkommen (np0104) geben.

Die Verknüpfung mit der 'NH.SAV'-Datei mittels des /TABLE-Subkommandos und den Schlüsselvariablen hhnr und hhnrakt ermöglicht es, dass mehrere Individuen dem selben Haushalt zugeordnet werden können. Die Arbeitsdatei enthält nun neben den Variablen über individuelle Verdienste sodann auch die Variable über das Haushaltseinkommen (nh50), wobei jedem Haushaltsmitglied der gleiche Wert für das Haushaltseinkommen zugewiesen wird. (Die Arbeitsdatei enthält somit Fälle, in denen z.B. bei Nicht-Erwerbstätigkeit zwar kein individueller Verdienst, aber dennoch ein (Haushalts-)Einkommen vorhanden sein kann.) Während es zwar möglich ist, mehrere /FILE-Subkommandos in eine MATCH FILE-Prozedur einzustellen, so darf lediglich eine Datei als 'table-lookup'-Datei Verwendung finden, was bedeutet, dass das /TABLE-Subkommando nur einmal herangezogen werden darf.

³ Kapitel 2.2 bis Kapitel 2.5 basieren auf Witte, James (1996).

Ähnliche Syntax-Prozeduren wie die angeführte bieten sich jedesmal dann an, wenn Informationen eines aggregierten Niveaus auf untergeordnete Einheiten zurückgeführt werden sollen. (Haushalt ⇔ Individuen; Individuum ⇔ z.B. Dauer von Arbeitslosigkeitsperioden)

Will man Individual- und Haushaltsdaten über mehrere Wellen verknüpfen, so kann es zum Erstellen einer 'balanced panel'-Datei hilfreich sein, aus PPFAD zunächst Informationen über die Teilnahme in den betrachteten Wellen heranzuziehen. Ist die hier betrachtete Teilnahme in beiden Wellen sichergestellt ($mnetto = nnetto = 1$), so besteht das Programm daraus, MP.SAV, MH.SAV, NP.SAV und NH.SAV mit der aktiven Arbeitsdatei zu verknüpfen. Das Subkommando /BY, das verwendet werden muss, wenn die Verknüpfung über das Subkommando /TABLE durchgeführt wird, ändert sich in Abhängigkeit davon, ob die zu verknüpfende Datei eine Individual-Datei (/BY = HHNR HHNRAKT PERSNR) oder eine Haushaltsdatei (/BY = HHNR HHNRAKT) ist. Zudem muss HHNRAKT vor der Verknüpfung mit Variablen einer anderen Welle umbenannt werden, die Arbeitsdatei ist zu sortieren.

```
Bsp.: /*      Verknüpfen von Individual- mit Haushaltsdaten über zwei Wellen      */
      /*      Bsp: Lebenszufriedenheit und Haushaltseinkommen in '96 und '97 */
      GET FILE = 'c:\soep\ppfad.sav'
          /KEEP = persnr hhnr mhhnr nhnr mnetto nnetto.

      /*      Auswahl der erfolgreich verlaufenen Interviews in '96 und '97      */
      SELECT IF (mnetto EQ 1 AND nnetto EQ 1).
      /*      Aktuelle Haushaltsnummer zuordnen und Sortieren                    */
      COMPUTE hhnrakt = mhhnr.
      SORT CASES BY hhnr hhnrakt persnr.

      /*      Mit MP.SAV verknüpfen zum Erfassen der Lebenszufriedenheit        */
      MATCH FILES FILE = *
          /TABLE = 'c:\soep\mp.sav'
          /KEEP = hhnr hhnrakt persnr nhnr mp11001
          /BY = hhnr hhnrakt persnr.

      /*      Mit MH.SAV verknüpfen zum Erfassen des Haushaltseinkommens        */
      MATCH FILES FILE = *
          /TABLE = 'c:\soep\mh.sav'
          /KEEP = hhnr hhnrakt persnr nhnr mp11001 mh50
          /BY = hhnr hhnrakt.

      /*      Umbenennen der Haushaltsnummer für 1997 und Sortieren            */
      COMPUTE hhnrakt = nhnr.
      SORT CASES BY hhnr hhnrakt persnr.
```

```

/*      Mit NP.SAV verknüpfen zur Erfassung der Lebenszufriedenheit      */
MATCH FILES FILE = *
      /TABLE = 'c:\soep\np.sav'
      /KEEP = hhnr hhnrakt persnr mp11001 mh50 np11701
      /BY hhnr hhnrakt persnr.

/*      Mit NH.SAV verknüpfen zum Erfassen des Haushaltseinkommens      */
MATCH FILES FILE = *
      /TABLE = 'c:\soep\nh.sav'
      /KEEP = hhnr hhnrakt persnr mp11001 mh50 np11701 nh50
      /BY = hhnr hhnrakt.

```

2.3 Verknüpfen von Individualdaten mit Haushalten

Will man bei der Analyse eines Untersuchungsgegenstandes auf der übergeordneten Ebene auf Daten untergeordneter Elemente zurückgreifen, so ist das Kommando /AGGREGATE anzuwenden. Im folgenden Beispiel werden individuelle Angaben, ob eine Nebenerwerbstätigkeit vorliegt, auf die Haushaltsebene aggregiert, so dass man Information darüber erhält, wieviel Personen in einem Haushalt nebenerwerbstätig sind.

```

Bsp.: /*      Verknüpfen von Individualdaten mit Haushalten      */
      /*      zunächst werden aus NP.SAV die Identifikatoren und      */
      /*      die Variable zur Nebenerwerbstätigkeit herangezogen      */
      GET FILE = 'c:\soep\np.sav'
      /KEEP = hhnr hhnrakt np5504.

      /*      nun wird eine Variable zur Nebenerwerbstätigkeit erzeugt      */
      /*      (np5504 fragt, ob kein Nebenerwerb vorliegt)      */
      COMPUTE nt = 0.
      FORMATS nt (F2).
      VARIABLE LABELS nt 'Nebenerwerbstätigkeit'.
      VALUE LABELS nt 1 'nebenerwerbstätig' 0 'nicht nebenerwerbstätig'.

      IF (np5504 EQ -2) nt = 1.

      /*      vor dem Aggregieren ist die Datei, die auf dem Individual-Level      */
      /*      vorliegt, nach der 'break-Variablen' zu sortieren      */
      SORT CASES BY hhnr hhnrakt.

      /*      mit 'AGGREGATE' zur Haushaltsebene aggregieren      */
      AGGREGATE
      /OUTFILE = * /*c:\test\hh_nt.sav'*/
      /BREAK = hhnr hhnrakt
      /hhnt 'nebetätige Personen im Hh' = sum(nt).

      /*      Verknüpfung mit bspw. HHGEN.SAV, um die Haushaltstypologie      */
      /*      erfassen zu können und danach mit NH.SAV zur Erfassung      */
      /*      des Haushaltseinkommens      */

```

```

MATCH FILES FILE = *
  /TABLE = 'c:\soep\hhgen.sav'
  /KEEP = hhnr hhnrakt hhnt htyphh1 htyphh2
  /BY = hhnr hhnrakt.

```

```

MATCH FILES FILE = *
  /TABLE = 'c:\soep\hh.sav'
  /KEEP = hhnr hhnrakt hhnt htyphh1 htyphh2 nh50
  /BY = hhnr hhnrakt.

```

Nach dem Erzeugen der Nebentätigkeitsvariablen *nt* erfolgt mit 'AGGREGATE' das Aggregieren der Individualdaten zur Haushaltsebene. Über das Subkommando */OUTFILE = ** werden die Haushaltsdaten zur aktiven Arbeitsdatei, was bedeutet, dass die bislang aktive Arbeitsdatei auf Individualniveau verloren geht. Will man dies vermeiden, kann man mit */OUTFILE = 'verzeichnisstruktur/name.sav'* die Haushaltsdaten abspeichern und später darauf zurückgreifen. Die Verknüpfung der erzeugten Haushaltsdaten verläuft sodann mittels einer einfachen MATCH FILES-Prozedur. In gewähltem Beispiel ist das vorherige Sortieren nach den Identifikatoren *hhnr* und *hhnrakt* nicht mehr nötig, da diese bereits in der 'AGGREGATE'-Prozedur als break-Variablen genutzt werden.

2.4 Verknüpfen von Daten über Individuen

Liegt das Untersuchungsinteresse auf der Individualebene und hierbei insbesondere darauf, wie Haushaltsmitglieder zueinander stehen, müssen 'Umwege' gegangen werden, da es in SPSS nur bedingt möglich ist, Informationen über Fälle hinweg direkt miteinander zu kombinieren. Ist man bspw. daran interessiert, ob und wie das Arbeitsangebot verheirateter Frauen von der Arbeitsmarktpartizipation ihrer Ehemänner abhängt, so bietet sich folgende Syntax-Struktur an:

```

Bsp.: /*      Verknüpfen von Daten über Individuen                */
      /*      Zunächst Info über Arbeitsmarktbeteiligung der Männer */
      GET FILE = 'c:\soep\hh.sav'
          /KEEP = hhnr hhnrakt persnr np08 np11 np11201.
      SELECT IF (np11201 EQ 1).

      /*      Umbenennen der Variablen, um ein Überschreiben der Werte   */
      /*      für Frauen zu vermeiden                                     */
      RENAME VARS (np11201 np11 np08 persnr = pn11201 pn11 pn08 partnr97).

      /*      Abspeichern der Arbeitsdatei zur späteren Verfügbarkeit     */
      SAVE OUTFILE = 'c:\test\hhpart.sav'
          /KEEP = hhnr hhnrakt partnr97 pn08 pn11 pn11201.

```

```

/*      Nun Info über die Arbeitsmarktbelastung der Frauen      */
GET FILE = 'c:\soep\np.sav'
      /KEEP = hhnr hhnrakt persnr np08 np11 np11201.
SELECT IF (np11201 EQ 2).

/*      Verknüpfung mit NPGEN.SAV, um die Partner-ID zu erlangen      */
MATCH FILES FILE = *
      /TABLE = 'c:\soep\npgen.sav'
      /BY hhnr hhnrakt persnr
      /KEEP hhnr hhnrakt persnr np08 np11 np11201 partz97 partnr97.

/*      Sortieren nach der Partner-ID und Verknüpfen mit NPART.SAV      */
/*      zur Erfassung der Informationen über die männlichen Partner      */
SORT CASES BY hhnr hhnrakt partnr97.

MATCH FILES FILE = *
      /TABLE = 'c:\test\npart.sav'
      /KEEP = hhnr hhnrakt persnr np08 np11 np11201 partz97 partnr97
            pn08 pn11
      /BY hhnr hhnrakt partnr97.

```

Im Beispiel werden die Variablen, die Auskunft über die Erwerbsbeteiligung (np08: arbeitslos gemeldet; np11: Erwerbstatus) geben, für Männer selektiert (np11201 = 1). Das Umbenennen der Variablen (Tausch der ersten beiden Lettern) hat zu erfolgen, da ansonsten beim anschließenden Verknüpfen die Informationen aus der zuerst im MATCH-Kommando festgelegten Datei die Informationen aus der zweiten Datei überschreiben würden. Die entstehende aktive Arbeitsdatei würde also nicht Informationen in bezug auf beide Partner enthalten. Zunächst wird die Arbeitsdatei für Männer mittels /SAVE OUTFILE = 'verzeichnisstruktur\name.sav' abgespeichert, um die so erzeugte Datei für die Verknüpfung zur Verfügung zu haben. Danach wird die Selektion für die Variablen über die Erwerbsbeteiligung der Frauen vorgenommen. Anschließend wird diese Arbeitsdatei mit NPGEN.SAV verknüpft, um an die Variablen über den Partnerstatus (partz97: verheiratet, Single, Lebenspartner u.a.) und die Identifikationsnummer des Partners (partnr97: identisch mit der Personnummer des Mannes) zu gelangen. Die Verknüpfung mit NPART.SAV erfolgt nach Sortierung mit hhnr, hhnrakt und partnr97, der Personnummer des ggf. vorhandenen Partners. NPART.SAV als 'table'-Datei zu verwenden stellt sicher, dass bei der Verknüpfung keine weiteren Fälle, sondern ausschließlich die zwei interessierenden Variablen (pn08 und pn11) hinzugefügt werden. Die so erzeugte Arbeitsdatei schließlich enthält Informationen über die Erwerbsbeteiligung von Frauen sowie, wenn vorhanden, deren männliche Partner.

Literatur:

- **Angele, German:** *'SPSS für Windows 7.5.2'*, Schriftenreihe des Rechenzentrums der OFU Bamberg, 1998, Heft 008
- **Witte, James:** *'Data Management and Analysis of the German Socio-economic Panel Using SPSS for Windows'*, 1996 (zu finden auf der SOEP-CD)
- **Frick, Joachim:** *'General Introduction to the German Socio-Economic Panel Study (GSOEP)'*, Handout zum GSOEP Summer Workshop, 1998
- **Frick, Joachim / Haisken-De New, John:** *'Desktop Companion to the German Socio-Economic Panel Study (GSOEP)'*, 1998 (zu finden auf der SOEP-CD)
- **SPSS Inc.:** *'SPSS Base Syntax Reference Guide'*, online-Hilfe