

Eine Einführung in das Statistikpaket Stata

Vortrag am GESIS-ZA Zentralarchiv für Empirische Sozialforschung

Bernd Weiß

Forschungsinstitut für Soziologie
GESIS-ZA Zentralarchiv für empirische Sozialforschung
Universität zu Köln
`bernd.weiss@wiso.uni-koeln.de`

18. Juli 2008

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Warum Stata?

- ▶ Finanzielle Vorteile
- ▶ Lässt sich um Funktionen erweitern, die von Herstellerseite (zunächst) nicht zur Verfügung gestellt werden.
- ▶ Stata ist (teilweise) schneller als SPSS.
- ▶ Die Programmsyntax von Stata ist logisch und systematisch aufgebaut; ermöglicht kompakten Programmcode.
- ▶ Sehr viele statistische Routinen implementiert.
- ▶ Mächtiges Hilfesystem.
- ▶ Zwischenergebnisse lassen sich weiterverwenden („Container-Konzept“).

Welche Stata-Versionen gibt es?

- ▶ Stata/MP
- ▶ Stata/SE
- ▶ Stata/IC
- ▶ Small Stata

(Quelle: <http://www.stata.com/products/whichstata.html>)

Welche Stata-Versionen gibt es?

Package	Max. no. of variables	Max. no. of right-hand variables	Max. no. of observations	Max. matrix size	64-bit version available?	Fastest: designed for parallel processing?
Stata/MP	32,767	10,998	unlimited*	11,000	Yes	Yes
Stata/SE	32,767	10,998	unlimited*	11,000	Yes	No
Stata/IC	2,047	798	unlimited*	800	Yes	No
Small Stata	99	39	1,000	40	No	No

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Die Elemente der Benutzeroberfläche

- ▶ Eingabefenster (command)
- ▶ Ergebnisfenster (results)
- ▶ Protokollfenster (review)
- ▶ Variablenfenster (variables)
- ▶ Data-Editor
- ▶ Data-Browser
- ▶ Do-File Editor

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Stata aktualisieren

Stata-Syntax

```
update all
```

Beispiel

```
update all
```

Stata erweitern: The Power of Ados

Stata lässt sich um Funktionalitäten erweitern, die ursprünglich nicht von Stata Corp. vorgesehen waren.

- ▶ `net install packagename`
- ▶ `ssc install packagename` (ssc = Statistical Software Components)

Um den *packagename* herauszubekommen, erfolgt zunächst eine stichwortbasierte Suche mit Hilfe von `-findit-` oder `-search-`.

Sinnvolle Voreinstellungen oder: die Standard-Präambel

```
version 10
set memory 100m
set more off, permanently
//delimiter ;
cd Verzeichnispfad
```

Wichtige shortcuts

Im Stata-Editor

- ▶ Do (mit Ausgabe): CTRL + d
- ▶ Run (ohne Ausgabe): CTRL + r

Log-Files

Siehe dazu das Skript von Pfeffer et al. 2004.

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Aufbau eines Stata-Kommandos

Stata-Syntax

prefix: `command varlist if exp in range weight , options`

Beispiel

by v2: `table v3 if (v3 < 50)`

(Quelle: <http://www.stata.com/help.cgi?language>)

Do-Files

- ▶ SPSS-Syntax-Files heißen in Stata Do-Files.
- ▶ Zwar lässt sich auch Stata teilweise über das GUI steuern, empfohlen wird jedoch die Nutzung von Do-Files.
- ▶ Do-Files werden in Stata mit dem Befehl `-do-` bzw. `-run-` und dem *filename* als Argument aufgerufen, also

```
do meinDoFile.do
```

Kommentare

in Stata lassen sich vier unterschiedliche Kommentartypen unterscheiden:

1. Die Zeile beginnt mit einem `*`.
2. Zwischen `/* ... */` eingeschlossener Text; kann mehrere Zeilen umfassen.
3. Der Kommentar beginnt mit `//`.
4. Der Kommentar beginnt mit `///`; kann dazu verwendet werden, eine lange Befehlszeile in mehrere Zeilen zu unterteilen.

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Das Hilfesystem von Stata

Stata-Syntax

```
search word [ word... ] // lokale Suche
```

```
findit word [ word... ] // lokale Suche und Internet; nutze ich als  
Standard
```

```
help [ command_or_topic_name ]
```

Beispiel

- ▶ search logistic regression
- ▶ help logistic
- ▶ findit meta-analysis heterogeneity

Weitere Hilfequellen

- ▶ Kohler, Ulrich, und Frauke Kreuter, 2008: Datenanalyse mit Stata. München / Wien: Oldenbourg. (3. Auflage)
- ▶ Stata-Ressourcen: <<http://www.stata.com/support/>>
- ▶ Statalist: <<http://www.stata.com/statalist/>>
- ▶ Resources to help you learn and use Stata (UCLA): <<http://www.ats.ucla.edu/stat/stata/>>
- ▶ UCLA: Academic Technology Services, Statistical Consulting Group: How do I do this SPSS command in Stata?. from <http://www.ats.ucla.edu/stat/Stata/faq/spss_command_to_stata.htm> (accessed 16.7.2008).
- ▶ Juul, Svent, 2007: For users with SPSS experience. from <<http://www.folkesundhed.au.dk/uddannelse/stata/introduction/spssusers.pdf>> (accessed 17.7.2008).
- ▶ Pfeffer, Fabian; Lindner, Stephan, und Bernd Weiß, 2004: Erste Schritte mit Stata. <www.metaanalyse.de/material/master.pdf>.

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Stata-Datensatz laden

Stata-Syntax

```
use filename [, clear nolabel]
```

Beispiel

```
use ZA4501_AC06.dta, clear
```

ASCII-Daten laden

Stata-Syntax

```
insheet [varlist] using filename [, options]
```

Beispiel

```
insheet using dataAllbusNolab.csv, clear  
delimiter(“;”) names
```

- ▶ `delimiter(#)` beschreibt das Trennzeichen, hier „;“.
- ▶ `names` bzw. `nonames` gibt an, ob in der 1. Zeile der Datei Variablennamen stehen.

Stata-Datensätze abspeichern

Stata-Syntax

```
save [filename] [, save_options]
saveold filename [, saveold_options]
```

Beispiel

```
save dataAllbusNolab.dta, replace
```

Bei großen Datensätzen `-compress-` vor dem `-save-` ausführen („compress attempts to reduce the amount of memory used by your data“).

Stata-datensätze exportieren

Stata-Syntax

outfile

outsheet

Beispiel

```
outfile using ZA4501_AC06.dat, nolabel replace  
outsheet v1 v4 v27 v53 using "dataAllbusNolab.csv", ///  
nolabel replace nonames delimiter(";")
```

Große Datensätze

Siehe Kohler / Kreuter (2008: 342ff).

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Variablenlabels

Stata-Syntax

```
label variable varname ["label"]
```

Beispiel

```
label variable v1 "ID"
```

Wertelabels

Das Zuweisen von Wertelabels erfolgt in 2. Schritten.

1. Wertelabel(container) definieren.
2. Wertelabel(container) bestimmten Variablen zuweisen.

Stata-Syntax

```
label define lblname # "label" [# "label" ...] [, add  
modify nofix]
```

```
label values varlist [lblname|.] [, nofix]
```

Beispiel

```
label define erhebungsgebiet 1 "West" 2 "Ost"
```

```
label value v2 erhebungsgebiet
```

Fehlende Werte

- ▶ Stata kennt 27 Codes für fehlende Werte, . (system missing value), .a, .b, ..., .z (extended missing values).
- ▶ Bei Abfragen mit `if` gilt zu beachten, dass missing values intern mit dem Wert positiv Unendlich geführt werden.
- ▶ Mit `mvencode` lässt sich die missing values-Zuweisung wieder entfernen.

Stata-Syntax

```
mvdecode varlist [if] [in], mv(numlist | numlist=mv  
[ numlist=mv...])
```

Beispiel

```
mvdecode v3, mv(999 = .a)  
mvdecode v4, mv(0,9 = .a)  
replace v3 = . if v3 == 999
```

Neue Variablen erstellen: generate

SPSS

compute

Stata-Syntax

generate [type] newvarname [:lblname] =*expif in*

Beispiel

generate v4r = abs(v4 - 2)

Neue Variablen erstellen: egen

SPSS

compute

Stata-Syntax

```
egen [type] newvar = fcn(arguments) [if] [in] [, options]
```

Beispiel

```
egen myrowmean = rowmean(v2 v3 v4)
```

Recodieren

Stata-Syntax

```
recode varlist (rule) [(rule) ...] [, generate(newvar)]
```

Beispiel

- ▶ recode v3 (18/65 = 1)(66/93 = 2)(94 = 3), generate(v3r)
- ▶ recode v3 (18/35 = 1 Jungspund) ///
(36/45 = 2 Lebensblüte) ///
(46/55 = 3 Gesetzter) ///
(56/65 = 4 Frührenter) ///
(66/100 = 5 Renter), generate(v3r)

Mit Teildatensätzen arbeiten

Stata kennt (mindestens) zwei Möglichkeiten, um bei den Analysen nur Teildatensätzen zu berücksichtigen:

- ▶ Dauerhaft: drop oder keep
- ▶ Temporär: if innerhalb der meisten Befehle, etwa table

Stata-Syntax

```
drop if exp
```

```
keep if exp
```

```
table varname [if]
```

Beispiel

```
drop if v2 == 1
```

```
table v2 if ((v3 > 50) & (v4 == 1))
```

Aggregieren

Stata-Syntax

```
collapse [(stat)] target_var=varname [if] [in] [weight] [, options]
```

Beispiel

```
collapse (mean) v3agg=v3 , by(v2)
```

Datensätze zusammenspielen (merge)

- ▶ Das Zusammenspielen von Datensätzen via merge setzt voraus, dass die Datensätze nach der/den Schlüsselvariablen sortiert sind.
- ▶

Stata-Syntax

```
merge [varlist] using filename [filename ...] [, options]
```

Beispiel

```
sort v2  
merge v2 using dataAllbusNolab.dta
```

Daten sortieren

Stata-Syntax

```
sort varlist [in] [, stable]
```

Beispiel

```
sort v1
```

Variablen löschen, umbenennen, umstellen

- ▶ drop
- ▶ keep
- ▶ rename
- ▶ order

Gliederung

Gründe für Stata und Versionen

Die Benutzeroberfläche von Stata

Administration von Stata

Die Programmsyntax

Das Hilfesystem und weitere Hilfequellen

Ein- und Ausgabe von Daten

Datenaufbereitung

Datenbeschreibung

Ein Codebook ausgeben

Stata-Syntax

```
codebook [varlist] [if] [in] [, options]
```

Beispiel

```
codebook
```

Variableninhalte auflisten

Stata-Syntax

```
list [varlist] [if] [in] [, options]
```

Beispiel

```
list v2 v3 in 1/10
```

```
list v1 v2 v3 v4 if v3 > 92 // Missings beachten!
```

Struktur des Datensatzes beschreiben

Stata-Syntax

```
describe [varlist] [, memory_options]
```

Beispiel

```
describe
```

Struktur von Variablen beschreiben

Stata-Syntax

```
inspect [varlist] [if] [in]
```

Beispiel

```
inspect
```

```
inspect v1 – v4
```

Absolute und relative Häufigkeiten

- ▶ `tabulate`: 1- oder 2-dimensionale Tabellen
- ▶ `tab1`: nur 1-dimensionale Tabellen für Variablenliste
- ▶ `tab2`: alle Kombinationen 2-dimensionaler Tabellen

Stata-Syntax

```
tabulate varname [if] [in] [weight] [, nolabel missing  
generate(varname) ]
```

Beispiel

```
tabulate v2 // tabulate oneway
```

```
tab1 v2 v4
```

```
tabulate v4 v2, column // tabulate twoway
```

```
tabulate v4 v2, column row chi2
```

```
tab2 v4 v3r v2, column row chi2 nofreq
```

Deskriptive Statistiken mit -summarize-

Stata-Syntax

```
summarize [varlist] [if] [in] [weight] [, options]
```

Beispiel

```
summarize v1 – v4
```

Deskriptive Statistiken mit -tabstat-

Stata-Syntax

```
tabstat varlist [if] [in] [weight] [, options]
```

Beispiel

```
tabstat v3 // Mittelwert als default
```

```
tabstat v3, statistics (mean sd min max range sum)
```

```
tabstat v3, by(v2) statistics (mean sd min max range sum)
```

Das Präfix -by- und -bysort-

Um Analysen getrennt nach den Ausprägungen einer (mehrerer) Variablen durchzuführen, gibt es das Präfix -by- bzw. -bysort-. Es entspricht in SPSS dem SPLIT FILE BY.

Stata-Syntax

```
by varlist: stata_cmd // vorher nach varlist sortieren
```

```
bysort varlist: stata_cmd
```

Beispiel

```
bysort v2: summarize v3 v4
```

Einfache Grafiken

Eine gute Übersicht bietet `<http://www.ats.ucla.edu/stat/stata/modules/graph8/intro/graph8.htm>`.

- ▶ Streudiagramme mit `-scatter-`
- ▶ Histogramme mit `-histogram-`
- ▶ Box-and-Whisker-Plots mit `-graph box-`