

# Einige Stata-Befehle

Stata-Befehle werden normalerweise mit dem Ende der Zeile abgeschlossen (aus Darstellungsgründen sind einige der folgenden Befehle umgebrochen, dies wäre im Anwendungsfall zu ändern). Die meisten Befehle und Optionen können auf wenige Buchstaben (unterstrichen) abgekürzt werden. Achtung: Stata unterscheidet zwischen Gross- und Kleinschreibung! Stata kennt eine Vielzahl an Operatoren und Funktionen, die in Ausdrücken (*exp*) verwendet werden können. Darunter befinden sich: arithmetische Operatoren wie +, -, \*, /, ^; arithmetische Funktionen wie `ln(...)`, `exp(...)`; relationale Operatoren wie == (ja, für „gleich“ stehen zwei Gleichheitszeichen!), >, <, >=, <=, ~= (nicht gleich, auch: !=); logische Operatoren wie &, |; spezielle Funktionen wie z.B. `inrange(...)` oder `inlist(...)`, etc. (mehr Informationen dazu erhält man mit `help exp`, `help operators`, `help functions` oder auch `help egen`).

Anmerkungen zur Notation auf diesem Blatt: *var* steht für eine Variable, *varlist* für eine oder mehrere Variablen (z.B. „v1 v2 v3“, „v1-3“, „v\*“ [alle Variablen, die mit „v“ beginnen], „all“ [alle Variablen] ...; siehe `help varlist`); *num* für eine Zahl, *numlist* für eine oder mehrere Zahlen (z.B. „1 2 3“, „1/3“ ...; siehe `help numlist`); *exp* für einen mathematischen, logischen oder sonstigen Ausdruck (z.B. „3\*4“, „var~=.“, „inrange(var,1,8)“ ...)

Literaturhinweis: Kohler, U., F. Kreuter (2001). Datenanalyse mit Stata. München: Oldenbourg.

## Allgemeines

<code>help <i>Befehl</i></code>	Aufruf des Hilfe-Eintrags für einen Befehl.
<code>search <i>Text</i></code>	Suche in Stata's Hilfeeinträgen.
<code>net search <i>Text</i></code>	Suche in den Stata-Datenbanken im Internet.
<code>*<i>Text</i></code>	Bemerkungen zwischen Befehlen anbringen.
<code>... /* <i>Text</i> */ ...</code>	Bemerkungen innerhalb eines Befehls anbringen.
<code>... /* */ ...</code>	Befehlsumbruch auf eine neue Zeile.
<code>#delimit ;</code>	Umschaltung, so dass Befehle mit ; abgeschlossen werden (und nicht mit dem Zeilenende; Rückschaltung mit #d cr).
<code>cd <i>Laufwerk</i>:\Pfad</code>	Bestimmung des Arbeitsverzeichnisses
<code>set <u>memory</u> <i>Grösse</i></code>	Arbeitsspeicher für Stata reservieren: z.B. 1000k, 4m oder 100m (k steht für Kilobyte, m für Megabyte)
<code>clear</code>	Alles, was sich im Stata-Arbeitsspeicher befinden, wegwerfen.
<code><u>log</u> using <i>Datei</i>.log, replace</code>	Aufzeichnung der Befehle und der Ergebnisse in einer Textdatei (mit replace werden ältere Versionen überschrieben).
<code><u>capture</u> <u>log</u> <u>close</u></code>	Log-Datei schliessen.
<code>set <u>more</u> off</code>	Seitenweise Bildschirmausgabe auf kontinuierliche Ausgabe umschalten (nur sinnvoll, wenn der Output in einer Log-Datei protokolliert wird).

## Daten-Handling

*Befehl* if *exp*

for varlist *varlist*:  
*Befehl1* \ *Befehl2* ...

use *datei*, clear  
use *varlist* if *exp* using *datei*,  
clear

drop/keep *varlist*

order *varlist*

drop/keep if *exp*

sort *varlist*

compress

save *datei*, replace

insheet using *datei*, clear

outsheet *varlist* using *datei*,  
nolabel

generate *newvar*=*exp*

egen *newvar*=*exp*

replace *var*=*exp*

recode *var* *numlist*=*num* ...

*Befehl* wird nur für die Fälle ausgeführt, für die *exp* zutrifft. Die if-Konstruktion kann bei den meisten Befehlen angewendet werden.

*Befehl1* etc. wird für jede Variable aus *varlist* wiederholt, wobei in den Befehlen X als Platzhalter verwendet wird. Der for-Befehl kann auch andere Arten von Listen verarbeiten (newlist, numlist, anylist). Ähnliche Strukturen werden durch `foreach`, forvalues und `while` bereitgestellt.

Lesen von Stata-Daten (clear erlaubt die Ersetzung von bereits geladenen Daten); in der zweiten Form werden nur die Variablen aus *varlist* und Fälle, die *exp* erfüllen, geladen.

Variablen löschen/behalten.

Variablen gemäss *varlist* anordnen.

Fälle, für die *exp* wahr ist, löschen/behalten.

Aufsteigende Sortierung der Daten nach den Werten der Variablen in *varlist*. Für komplexe Sortierungen gibt's gsort.

Komprimierung der Daten.

Stata-Daten speichern (replace erlaubt die Ersetzung von Dateien mit gleichem Namen; die Standardendung von Stata-Daten ist .dta).

Einlesen von Daten im Textformat (tab- oder comma-delimited).

Datenexport in eine tab-delimited Textdatei.

Bildung einer neuen Variablen.

Erweiterter Befehl zur Bildung einer neuen Variablen.

Ersetzung einer bestehenden Variablen. Typischerweise verwendet in Kombination mit if, also: `replace var=exp1 if exp2`.

Umcodierung, z.B.: `recode var 1 3=0 2=1 4/9=2` (4/9 bedeutet „4 bis und mit 9“).

<code>mvdecode varlist, mv(numlist)</code>	Missing-Values definieren, z.B.: <code>mvdecode _all, mv(-9/-1)</code> (alle Vorkommnisse eines Wertes von -9 bis -1 werden auf „missing“ gesetzt). Der Code für „missing“ ist ein Punkt und kann entsprechend angesteuert werden (z.B. würden mit <code>drop if v1==.</code> alle Fälle mit einem „missing“ auf Variable <code>v1</code> gelöscht). Achtung: Der Umgang von Stata mit Missing-Values ist manchmal etwas undurchsichtig. Bei Transformationsbefehlen ( <code>generate</code> und ähnlich) sollte man sich daher angewöhnen, Missings per Definition auszuschliessen ( <i>Befehl</i> <code>if var~=.</code> ).
<code>label variable var "label"</code>	Variable beschriften.
<code>label define lblname num1 "label1" num2 "label2" ...</code>	Werte-Labels definieren.
<code>label values varname lblname</code>	Einer Variable eine Werte-Label-Definition zuweisen.
<code>describe</code>	Gibt eine Übersicht über die Variablen im geladenen Datensatz.
<code>codebook varlist</code>	Erstellt ein Codebuch für <i>varlist</i> bzw. die Variablen im geladenen Datensatz, wenn <i>varlist</i> weggelassen wird.

## Gewichte, By-Konstruktion

<i>Befehl</i> <code>[weighttype=var]</code>	Gewichtung der Fälle mit den Werten von <i>var</i> . Für <i>weighttype</i> gibt es verschiedene Möglichkeiten, z.B. <i>fweight</i> als reines Häufigkeitsgewicht, <i>aweight</i> als sog. „analytic weight“ (im Prinzip Häufigkeitsgewichte mit Normierung auf die Stichprobengröße) oder <i>pweight</i> als sog. „sampling weights“ (Inverse der Auswahlwahrscheinlichkeit). Lediglich die „pweights“ führen im Falle einer Stichprobe mit ungleichen Auswahlwahrscheinlichkeiten zu korrekten Schätzwerten für die Standardfehler. Mehr dazu unter <code>help weights</code> .
<code>by varlist: Befehl</code> <code>bysort varlist: Befehl</code>	<i>Befehl</i> wird für jede durch <i>varlist</i> definierte Gruppe von Fällen wiederholt (z.B. für getrennte Auswertungen über verschiedenen Gruppen). In der zweiten Form kann auf einen vorgängigen <code>sort</code> -Befehl verzichtet werden.

## Univariate und bivariate Statistik

<code>tabulate var</code> <code>tab1 varlist</code> <code>summarize varlist</code>	Eindimensionale Häufigkeitstabelle(n) für <i>var</i> bzw. für die Variablen in <i>varlist</i> .
<code>means varlist</code>	Univariate Statistiken (arith. Mittel, Standardabweichung, Min., Max., Anzahl Fälle).
<code>ci varlist, binomial</code>	Arithmetisches, geometrisches und harmonisches Mittel.
	Konfidenzintervalle für Mittelwerte bzw. für Anteilswerte bei Spezifikation von <code>binomial</code> .

<code>graph varlist, histogram</code>  <code>tabulate var1 var2, column row cell</code> <code>chi2 lrchi2 gamma taub V</code>  <code>correlate varlist</code> <code>pwcorr varlist, sig</code> <code>spearman var1 var2</code>  <code>ktau var1 var2</code>  <code>ttest var1, by(var2) unequal</code>  <code>median var1, by(var2)</code>  <code>ranksum var1, by(var2)</code>	Histogramm.  Kontingenztafel mit absoluten, relativen und bedingten Häufigkeiten, $\chi^2$ -Test, Likelihood-Ratio-Test, Cramér's $V$ , Kendall's $\tau_b$ und Goodman und Kruskal's $\gamma$ .  Korrelationsmatrix. In der zweiten Form mit Signifikanz.  Spearman's Rangkorrelationskoeffizient inkl. Signifikanz  Kendall's $\tau_a$ und $\tau_b$ inkl. Signifikanten  T-Test auf Mittelwertsdifferenzen für <i>var1</i> nach <i>var2</i> (unter Annahme ungleicher Varianzen).  Mediantest für <i>var1</i> nach <i>var2</i> .  Rangsummentest für <i>var1</i> nach <i>var2</i> .
--	--

## Regressionsverfahren

<code>regress depvar varlist</code>  <code>regress depvar var</code> <code>predict yhat</code> <code>graph depvar yhat var, symbol(o.)</code> <code>connect(.l) sort</code>  <code>logit depvar varlist</code>  <code>probit depvar varlist</code>	Schätzung eines linearen Regressionsmodells (OLS).  Streudiagramm mit Regressionsgerade.    Schätzung eines logistischen Regressionsmodells.  Schätzung eines Probit-Modells.
---	---

## Komplexe Stichproben

<code>help svy</code>  <code>svyset thingto set var</code>          <code>svy mean varlist</code>  <code>svyprop varlist</code>  <code>svytab var1 var2</code>  <code>svyreg depvar varlist</code>  <code>svylogit depvar varlist</code>  <code>svyprobit depvar varlist</code>  <code>svylc ...</code>	Übersicht.  Definition der Samplestruktur, wobei für <i>thingto set</i> <i>strata</i> (Schichten), <i>psu</i> (Primary Sampling Unit für Klumpenstichproben), <i>fpc</i> (Finite Population Correction) und <i>pweight</i> (Sampling Gewichte) zur Verfügung stehen.  Mittelwerte und Konfidenzintervalle.  Anteilswerte und Standardfehler.  Kontingenztafel.  Lineare Regression.  Logistische Regression.  Probit.  Test von Linearkombinationen von Parametern (z.B. für Mittelwertvergleich nach <code>svy mean</code> ).
---	--