

Teil 1: Übersicht über einige grundlegende Befehle

Allgemeine Befehlsstruktur:

[*prefix:*] *command varlist* [Bedingung wie *if* oder *in*] [, *option*]

Einige Befehle können abgekürzt werden. Es müssen dann nur die unterstrichenen Buchstaben eines Befehls eingegeben werden.

Hilfe:

help command

Zeigt eine ausführliche Hilfedatei zu dem Befehl *command* an. Es werden dort unter anderem alle Optionen, die mit diesem Befehl möglich sind, aufgeführt und erläutert.

search word

Durchsucht eine Schlüsselwortdatenbank und das Internet nach dem eingegebenen Begriff (*word*) und gibt unter anderem alle Befehle aus, deren Beschreibung diesen Begriff enthält.

1. Do-Files

doedit

Öffnet einen neuen Do-File. Ein Do-File besteht aus einer Liste von Befehlen.

do "filename"

Führt den Do-File *filename* aus, d.h. es werden alle Befehle des Do-Files nacheinander ausgeführt.

Alternativ kann ein Do-File mit diesem Icon  geöffnet werden.

Wird mit der rechten Maustaste ins Review-Fenster geklickt und *Save Review Contents* ausgewählt, werden alle Befehle der aktuellen Stata-Sitzung in chronologischer Reihenfolge in einem Do-File abgespeichert.

Leitet eine Kommentarzeile in einem Do-File ein. Diese werden beim ausführen des Do-Files zwar angezeigt, aber ansonsten ignoriert.

*/**

Leitet einen Kommentar innerhalb einer Befehlszeile ein.

**/*

Beendet einen Kommentar innerhalb einer Befehlszeile.

`version`

Legt die Version von Stata fest, mit der der Do-File ausgeführt werden soll.

`exit`

Zeigt das Ende des Do-Files an.

2. Nützliche Stata-Einstellungen

`set more off`

Sorgt dafür, dass mehr Zeilen angezeigt werden als auf dem Bildschirm gleichzeitig Platz haben.

`set memory Xm [, permanent]`

Setzt den Speicher auf x MB fest. `permanent` sorgt dafür, dass diese Einstellung dauerhaft ist.


3. Datensatz laden

`use "filename", clear`

Öffnet einen Datensatz. Die Option `clear` sorgt dafür, dass ein eventuell vorhandener Datensatz überschrieben wird.

Beispiel:

```
use "C:\Eigene Dateien\Zeitreihenanalyse\Übung  
01\winesales.dta", clear
```

Alternativ kann ein Datensatz über dieses Icon  geöffnet werden.


4. Log-File anlegen

`log using "filename", replace`

Öffnet einen Log-File, der alle Befehle und ihren Output speichert. Die Option `replace` sorgt dafür, dass eine eventuell vorhandene Datei überschrieben wird.

Beispiel:

```
log using "C:\Eigene Dateien\Zeitreihenanalyse\Übung  
01\Übung1.log", replace
```

Alternativ kann ein Log-File auch über dieses Icon  geöffnet werden.

`log close`

Schließt den Log-File.

`log off`

Unterbricht den Log-File.

`log on`

Schaltet die Aufzeichnung wieder an.

5. Datensatz kennen lernen

`describe varlist`

Beschreibt die Variablen in *varlist*. Wird keine *varlist* angegeben, werden alle Variablen des aktuellen Datensatzes beschrieben:

- Anzahl der Variablen
- Anzahl der Beobachtungen
- Speichertyp der Variablen
- Format, in dem die Variablen gespeichert sind
- Bezeichnungen (label) der Werte der Variablen
- Bezeichnung (label) der Variablen selbst

`list varlist`

Listet alle Beobachtungen der Variablen in *varlist* auf. Wird keine *varlist* angegeben, werden die Beobachtungen aller Variablen des aktuellen Datensatzes aufgelistet.

`q`

Unterbricht die Auflistung (bei `set more on`)

Alternativ kann auch dieses Icon  benutzt werden.

`browse`

Zeigt sämtliche Beobachtungen aller Variablen in einem Pop-up-Fenster an.

Alternativ kann auch dieses Icon  benutzt werden, um den Datenbrowser aufzurufen.

`summarize varlist`

Zeigt zusammenfassende Maßzahlen der in *varlist* angegebenen Variablen an. Wird keine *varlist* angegeben, werden die Zusammenfassungen aller Variablen des aktuellen Datensatzes angezeigt:

- Anzahl der Beobachtungen
- arithmetisches Mittel
- Standardabweichung
- kleinster beobachteter Wert
- größter beobachteter Wert

6. Bedingungen `if` und `in`

`if var = exp`

Befehl wird nur für die Beobachtungen ausgeführt, die die Bedingung erfüllen.

Beispiel:

```
list time winesales if time <= 251
```

`in range`

Befehl wird nur für die in *range* angegebenen Beobachtungen ausgeführt.

Beispiel:

```
list time winesales in 1/12
```

Der Ausdruck / bedeutet "von ... bis".

Achtung: Der Befehl ist abhängig von der aktuellen Reihenfolge der Beobachtungen.

7. Erzeugen und Löschen von Variablen

`generate varname = exp`

Erzeugt eine neue Variable mit dem Namen *varname*.

Als *exp* kann z.B. `_n` verwendet werden. Damit wird eine Folge von natürlichen Zahlen (1, 2, 3, ...) erzeugt.

`_n` kann auch mit mathematischen Operationen benutzt werden.

`edit`

Neue Variablen und ihre Werte können auch direkt über den *Data Editor* eingegeben werden.

Alternativ kann auch dieses Icon  benutzt werden, um den Dateneditor aufzurufen.

`replace var = exp`

Ersetzt die Beobachtungen einer bestehenden Variablen *var* durch *exp*.

`drop varlist`

Löscht die in *varlist* angegebenen Variablen.

`drop _all`

Löscht alle Variablen und den sonstigen Speicherinhalt der aktuellen Stata-Sitzung.

```
keep varlist
```

Behält nur die in *varlist* angegebenen Variablen, d.h. Stata löscht alle Variablen, die nicht in *varlist* angegeben sind.

8. Bezeichnung von Variablen und Werten

```
label variable varname "label"
```

Fügt der Variablen *varname* eine Bezeichnung *label* an.

```
label define lblname # "label" [# "label" ...]
```

Erzeugt Platzhalter *lblname*, in dem Bezeichnungen (*label*) für bestimmte Werte (#) festgelegt werden.

```
label values varname lblname
```

Weist den Werten der Variablen *varname* die Bezeichnungen zu, die im Platzhalter *lblname* festgelegt wurden.

9. Local

```
local lclname = exp
```

Weist *lclname* einen bestimmten Ausdruck *exp* zu.

```
`lclname`
```

Ruft den Ausdruck auf.

10. Schleife

```
forvalues num = exp {  
    ... Befehle ...  
}
```

Erzeugt eine Schleife über die Zahlenfolge in *exp*, d.h. die Befehle innerhalb der Schleife werden jeweils für alle Zahlen der Zahlenfolge ausgeführt.

Innerhalb der Schleife wird die jeweilige Zahl des Durchlaufs mit ``num`` aufgerufen.

11. Graphische Darstellungen

`graph twoway scatter varlist`

Stellt die in *varlist* angegebenen Variablen in einem „Punkt“-Diagramm dar. Die *varlist* wird dabei wie folgt gelesen: *y_1* [*y_2* ...] *x*, d.h. die ersten Variablen werden auf der y-Achse abgetragen während die letzte Variable auf der x-Achse abgetragen wird.

||

Ermöglicht es, mehrere *twoway*-Graphiken in einer Graphik darzustellen.

`graph twoway line varlist`

Stellt die in *varlist* angegebenen Variablen in einem „Linien“-Diagramm dar.

12. Lineare Regression

`regress depvar indepvars`

Führt eine lineare Regression durch.

`lfit yvar xvar`

Erzeugt eine Regressionsgerade.

Beispiel:

`graph twoway line winesales time || lfit winesales time`

`predict varname, _residuals`

Erzeugt eine neue Variable mit dem Namen *varname*, die die Residuen der Regression enthält.

Nach einer Regression können die Regressionskoeffizienten wie folgt aufgerufen werden:

- `_b[_cons]` (Koeffizient der Konstanten)
- `_b[varname]` (Koeffizient der Variablen *varname*)

`predict varname, xb`

Erzeugt eine Variable mit dem Namen *varname*, die die „fitted values“ beinhaltet.

Teil 2: Übersicht über einige zeitreihenspezifische Befehle

13. Erzeugen einer Zeitreihenvariablen

```
format varname %t
```

Überführt die Variable *varname* in ein Datumsformat.

Format	Beschreibung	Kodierung
%td	täglich	0 = 01jan1960, 1 = 02jan1960
%tw	wöchentlich	0 = 1960w1, 1 = 1960w2
%tm	monatlich	0 = 1960m1, 1 = 1960m2
%tq	quartalsweise	0 = 1960q1, 1 = 1960q2
%th	halbjährlich	0 = 1960h1, 1 = 1960h2
%ty	jährlich	1960 = 1960, 1961 = 1961

Übersicht mit `help tdates`.

Beispiel:

```
generate time = _n + 239
```

```
format time %tm
```

Erzeugt eine Variable *time*, die monatliche Daten beginnend mit 1980m1 beinhaltet.

Operator	Beschreibung
c C	Jahrhundert ohne oder mit null am Anfang
y Y	Zweistellige Jahreszahl ohne oder mit null am Anfang
m M	Monat abgekürzt oder ausgeschrieben
d D	Tag des Monats ohne oder mit null am Anfang
j J	Tag des Jahres ohne oder mit null am Anfang
h	Halbjahr (1 oder 2)
q	Quartal (1, 2, 3 oder 4)
w W	Woche des Jahres (1, 2, ..., 52) ohne oder mit null am Anfang
—	Leerzeichen
.	Punkt
,	Komma
:	Doppelpunkt
—	Bindestrich
/	Slash

Übersicht mit `help dfmt`.

Beispiel:

```
format time %tmD/N/CY
```

Formatiert die Variable `time`, sodass sie monatliche Daten beginnend mit 01/02/1980 beinhaltet.

```
format time %tmM_CY
```

Formatiert die Variable `time`, sodass sie monatliche Daten beginnend mit January 1980 beinhaltet.

```
format time %tmmY
```

Formatiert die Variable `time`, sodass sie monatliche Daten beginnend mit Jan80 beinhaltet.

```
tsset varname
```

Kennzeichnet die Variable `varname` als Zeitreihenvariable.

```
tsline varname
```

Stellt die Zeitreihe `varname` graphisch dar.

14. Erzeugen von Lags, Leads und Differenzen

```
L.var
```

Beinhaltet die um eine Periode zeitliche verzögerte Variable `var`. Vor der Anwendung muss die Zeitreihenvariable mit `tsset` bestimmt werden.

Operator	Beschreibung
L.	zeitliche Verzögerung (Lag) um eine Periode x_{t-1}
L2.	zeitliche Verzögerung (Lag) um zwei Perioden x_{t-2}
...	...
F.	zeitlich vorgelagert (Lead) um eine Periode x_{t+1}
F2.	zeitlich vorgelagert (Lead) um zwei Perioden x_{t+2}
...	...
D.	zeitliche Differenz um eine Periode $x_t - x_{t-1}$
D2.	Differenz der Differenz $x_t - x_{t-1} - (x_{t-1} - x_{t-2}) = x_t - 2x_{t-1} + x_{t-2}$
...	...
S.	„saisonale“ Differenz um eine Periode $x_t - x_{t-1}$
S2.	„saisonale“ Differenz um zwei Perioden $x_t - x_{t-2}$
...	...

Übersicht mit `help tsvarlist.D. = S.;` aber `D2. ≠ S2.` usw.

`L.var`

Die Operatoren werden direkt der Variablen `var` vorangestellt:

`L3.var`

Die Operatoren können wiederholt werden: `L3.` ist gleichbedeutend mit `LLL.`

Die Operatoren können kombiniert werden: `L3.` ist gleichbedeutend mit `LL2.` und `L2L.`

`LF.var`

Verändert die Variable `var` nicht.

`L(1/3).var`

Die Operatoren können statt mit Zahlen auch mit „Zahlausdrücken“ benutzt werden: `L(1/3).` ist gleichbedeutend mit dem Eingeben von `L.`, `L2.` und `L3.`