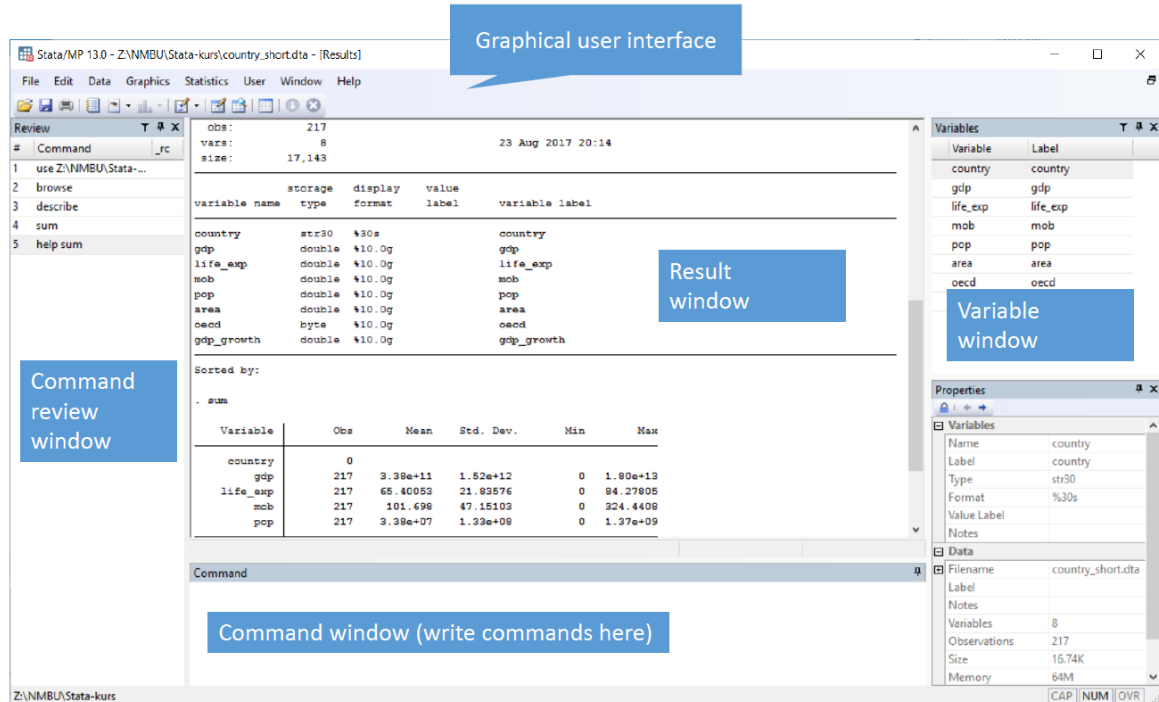


Getting started with Stata 2017: Cheat-sheet

4. september 2017

1 Get started



- Graphical user interface (GUI). Clickable. Simple.
- Commands. Allows for use of do-file. Easy to keep track.
 - Command window: Write commands here (or in do-file)
 - Command review window: Keeps track of what you execute in command window.
- Result window. Shows results
- Variable window. Show variables of the dataset in current memory.

The do-file

All the commands you write in the command line you can put into a do-file. The do-file allows you to save your commands! Using a do-file helps you keep track of everything you have done to your data. It allows you to re-run commands, analysis and make modifications.

1.0.1 Opening/creating a do-file

- GUI: Window → Do-file editor
- Command: `doedit` (creates new do-file)

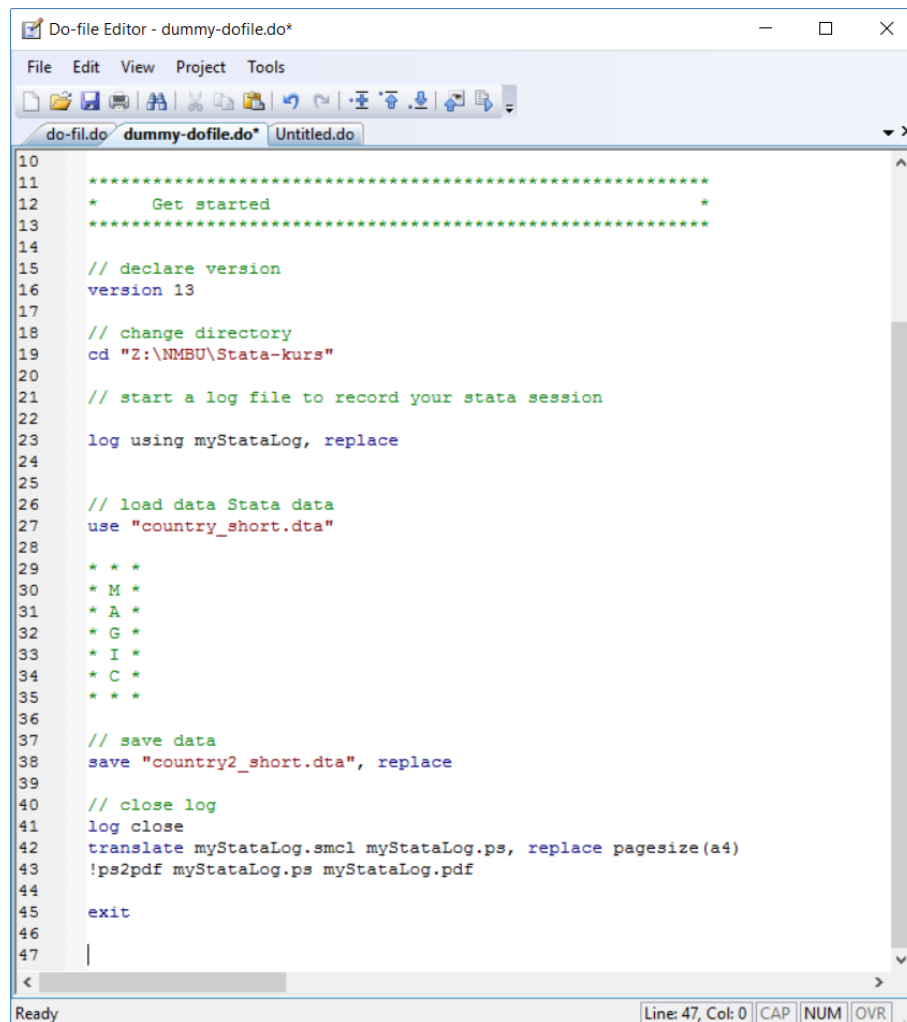
Running do-file

- GUI: Tools → Execute (do) (execute from do-file box)
- GUI: File → Do (if you do not have it open)
- Short-cut: CTRL D

Other

- In do-file Stata ignores lines that begin with *
- Place comments inside `/* */`, `/* comment here */`
- Place comments after `//`, `// comment here`
- Use `///` to split a long line. When you use `///`, the next line joins with the current line. `///` lets you split long lines across multiple lines in the do-file.

Do-file dummy



The screenshot shows the Stata Do-file Editor window titled "Do-file Editor - dummy-dofile.do*". The window has a menu bar (File, Edit, View, Project, Tools) and a toolbar. The main text area contains the following Stata code:

```
10
11 *****
12 *      Get started
13 *****
14
15 // declare version
16 version 13
17
18 // change directory
19 cd "Z:\NMBU\Stata-kurs"
20
21 // start a log file to record your stata session
22
23 log using myStataLog, replace
24
25
26 // load data Stata data
27 use "country_short.dta"
28
29 * * *
30 * M *
31 * A *
32 * G *
33 * I *
34 * C *
35 * * *
36
37 // save data
38 save "country2_short.dta", replace
39
40 // close log
41 log close
42 translate myStataLog.smcl myStataLog.ps, replace pagesize(a4)
43 !ps2pdf myStataLog.ps myStataLog.pdf
44
45 exit
46
47 |
```

The status bar at the bottom indicates "Ready" and "Line: 47, Col: 0" with buttons for CAP, NUM, and OVR.

1. Your name. Date. Describe what the file does.
2. Declare the version you used when you wrote the do-file. `version year`
3. Change directory. `cd «adress»`
4. Begin log file. `log using myStatalog, replace`

5. Call up data : use `«dataname.dta», clear`
6. The main part. Data manipulation, statistics etc.
7. Save data under new name. `save «dataname.dta», replace`
8. Close log. `log close`
9. Print log to pdf. `translate myStataLog.smcl myStataLog.ps, replace pagesize(a4)`
`!ps2pdf myStataLog.ps myStataLog.pdf`

2 Basic commands

Syntax structure `command varlist, options.` Syntax is case sensitive.

Interrupt Stata Ctrl-Break or the Stop-button.

help type `help name_command` if you need help. E.g. `help regression`

search Use `search topic` to search the Stata command for analyses w.r.t. topic. For instance, `search regression` gives a compact survey of the commands relevant for regression analysis.

clear The command `clear`, clears all data in the memory

Common operators and commands

Arithmetic

+	addition
-	subtraction
*	multiplication
/	division
^	power
()	brackets

Logical

&	and
	or
!	not
~	not

Numeric and string

>	greater than
<	less than
>=	greater than or equal
<=	less than or equal
==	equal
!=	not equal
~=	not equal

Useful mathematical functions

<code>abs()</code>	the absolute value
<code>exp()</code>	exponential function
<code>int()</code>	integer obtained by truncation towards zero. <code>int(1.2)=1</code>
<code>log()</code>	the natural logarithm
<code>sqrt()</code>	square root

Directory

<code>cd</code>	change directory. E.g. <code>cd z:\NMBU\Stata</code>
<code>dir</code>	displays file names in current working directory
<code>pwd</code>	display path to your current working directory

3 Data

Open Stata dataset

- GUI: File → Open
- Command: use *name_of_dataset.dta*, clear

Import data from other sources

- GUI: File → Import → (choose your type of file)
 - Make sure to check correct options
- Command: import
 - `import excel «filename.xls», sheet("sheetname") firstrow clear`
 - Imports the sheet «Sheetname» in the Excel file «Filename» and treats the first row of Excel data as variable names.
 - Write `help import` to see the many other options!

Saving data

- GUI: File → Save
- Command: save *name_dataset*, replace

Saves your dataset as *.dta* (Stata dataset). Saves it in your chosen Stata directory (`cd`).

Inspect data

<code>browse</code>	browse data
<code>edit</code>	edit data
<code>list</code>	list all variables
<code>describe</code>	describe data
<code>sum</code>	summarize data
<code>inspect</code>	display simple summary of data's attributes
<code>duplicates list</code>	list all duplicates
<code>dotplot var1</code>	comperative scatterplot of var1

Data management

Appending data in Stata

- GUI: Data → Combine datasets → Append datasets
- Command: `append` using *filename*

Merge data (with your main dataset loaded in Stata)

- GUI: Data → Combine datasets → Merge datasets.
 - Most common option: One-to-one merge on specified key variables.
- Command: `merge 1:1 varlist` using *filename*

Making variables understandable

<code>label data <i>dataname</i></code>	label dataset
<code>label variable <i>varname</i></code>	label variable
<code>rename <i>oldname newname</i></code>	rename variables

Managing variables

<code>sort</code>	sort data in ascending order
<code>gen</code>	create contents of variable. E.g. <code>gen var2=var1*2</code>
<code>egen <i>mean_var1</i>=mean(<i>var1</i>)</code>	creates a constant with the mean of <i>var1</i> . Other options include <code>max()</code> (maximum value), <code>median()</code> , <code>min()</code> (minimum value) and more.
<code>replace</code>	replace contents of existing variable. E.g. <code>replace var1=var2 if var1==0</code>
<code>sum var1</code>	replace zero values of variable <i>var1</i> with the mean value of variable <i>var1</i>
<code>return list</code>	
<code>replace var1=r(mean) if var1==0</code>	
<code>drop <i>varlist</i></code>	drop variables
<code>drop <i>var1</i> if <i>var1</i>==0</code>	drop observations, here drop <i>var1</i> if it is equal to zero
<code>drop in <i>range</i></code>	drop a range of observations
<code>keep <i>varlist</i></code>	keep variables
<code>keep if <i>var1</i>>0</code>	keep observations, here keep <i>var1</i> if it is larger than zero
<code>keep in <i>range</i></code>	keep a range of observations

4 Descriptive statistics

<code>sum</code>	summary of data
<code>tabstat <i>var1 var1</i>, stat(<i>option</i>) by(<i>varname</i>)</code>	Compact table of summary statistics for x_1 and x_2 . Group them by variable <code>by(varname)</code> . Report the specific statistics <code>stat(options)</code> . Options include (but are not limited to) mean, max, min and standard deviation.

Correlation

<code>corr <i>varlist</i></code>	correlation matrix for variable list <i>varlist</i>
<code>pwcorr <i>var1 var2</i></code>	pairwise correlation for variables <i>var1</i> and <i>var2</i>

Graphs

- GUI: Graphics → (take your pick)

<code>twoway</code>	twoway graphs
<code>twoway(scatter <i>var1 var2</i>)</code>	scatterplot of variables <i>var1</i> and <i>var2</i>
<code>twoway (line <i>var1 var2</i>)</code>	line plot of variables <i>var1</i> and <i>var2</i>
<code>histogram <i>var1</i></code>	histogram for variable <i>var1</i>
<code>kdensity <i>var1</i>, normal</code>	Kernal density plot of variable <i>var1</i> with the normal density overlaid for comparison
<code>graph save <i>graphname filename</i></code>	Save your graph. E.g. <code>graph save <i>graph1</i> eps</code> or <code>graph save <i>graph2</i> png</code> . Default is <code>.gph</code>
<code>graph combine <i>graph1.filename graph2.filename</i></code>	Combine two graphs. E.g. <code>graph combine <i>graph1.gph graph2.gph</i></code>

Simple testing

- GUI: Statistics → Summaries, tables and tests → Classic test of hypotheses

<code>ttest <i>var1==var2</i></code>	t-test (mean-comparison tests)
<code>ttest <i>var1==0</i></code>	t-test (mean equal to zero)

5 Regression

Estimation

- GUI: Statistics → Linear models and related → Linear regression

<code>reg <i>depvar var1 var2 var3</i></code>	linear regression
<code>reg <i>depvar var1 var2 var3</i>, noconstant</code>	linear regression, suppress constant term
<code>reg <i>depvar var1 var2 var3</i>, vce(<i>vcetype</i>)</code>	Use other standard errors. <i>vcetype</i> may be <code>ols</code> , <code>robust</code> , <code>cluster</code> ect.

Postestimation

- GUI: Statistics → Postestimation

<code>estimates store <i>name</i></code>	store estimates in memory
<code>estimates restore <i>name</i></code>	restore estimates in memory
<code>predict xb</code>	linear prediction (predict $\hat{y}$)
<code>predict residuals</code>	predict residuals
<code>predict rstandard</code>	predict standardized residuals
<code>predict rstudent</code>	Studentized residuals
<code>predict cooks</code>	Cook's distance
<code>leverage</code>	leverage
<code>test <i>var1</i></code>	test linear hypotheses after estimation. test if beta for <i>var1</i> is equal to zero
<code>test (<i>var1=1</i>)</code>	test if beta for <i>var1</i> is equal to one
<code>estimates table <i>model1 model2</i>, star(.05 .01 .001) style(online)</code>	Creates a table with regression results (coefficients) from <i>model1</i> and <i>model2</i> with stars indicating significance.
<code>rvfplot</code>	residual-versus-fitted plot