

Introduction to Stata

Matthias Krapf
University of Basel

Basel - March 6, 2023

Agenda

- Introduction
 - Introduction
 - Where to find help?
 - Interface
 - What you need for a project
 - Data Management
 - Stata Commands
 - Data Manipulation
 - Descriptive Statistics
 - Regressions
 - Graphs
- Exercise

Introduction

Command-based Software Package for Econometric Analyses of

- Cross-sectional Data
- Panel Data
- (Time-Series Data)

Access

- In the PC-Labs at WWZ
- Buy your own license

Where to get help?

- Help Command: *help regress*
- Stata User Guide
- www.google.com
- Short Stata Guide by Professor Schmidheiny ([click here](#))
- If all that does not help: matthias.krapf@unibas.ch

Interface

The screenshot displays the Stata/SE 12.1 interface with the following components:

- Top Menu Bar:** Open, Save, Print, Log, Viewer, Graph, Do-File Editor, Data Editor, Data Browser.
- Results Window (Red Border):**
 - size: 798,952
 - Table with 5 columns: variable name, storage type, display format, value label, variable label.
 - Sorted by:
 - end of do-file
- Command Window (Green Border):** Command
- Variables Panel (Blue Border):**
 - Search: Enter filter text here
 - Table with 2 columns: Name, Label.
 - Variables listed: idcode, year, birthyr, race, msp, grade, collgrad, south, indcode, exper, tenure, hours, wage, unionfee.
- Properties Panel (Grey Border):**
 - Variables: Name, Label, Type, Format, Value Label, Notes.
 - Data: Filename (statadata.dta), Label, Notes, Variables (14), Observations (28,534), Size (780.23K), Memory (64M).

What you need for a project:

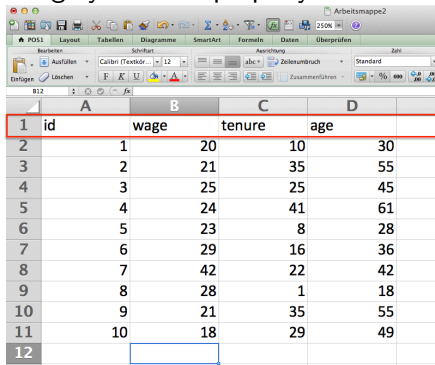
- Working directory - e.g.: "C:\documents\myproject"
- Data stored in directory in either format:
 - .dta (Stataformat)
 - .csv (Comma Spread Value)
 - .xlsx (Excel as of Stata 12)
- Do-File where you store your commands.
 - Open Do-File-Editor
 - save the new empty do-file (in your project directory)
 - easy to replicate or manipulate later
 - save all commands of a project in one file

Data Management I

- If you have data in stata format *.dta* you can directly open it with the command:
 use "mydata.dta"
- usually you have data as excel sheets or as csv-files, then: ...

Data Management II

- Arrange your data properly:



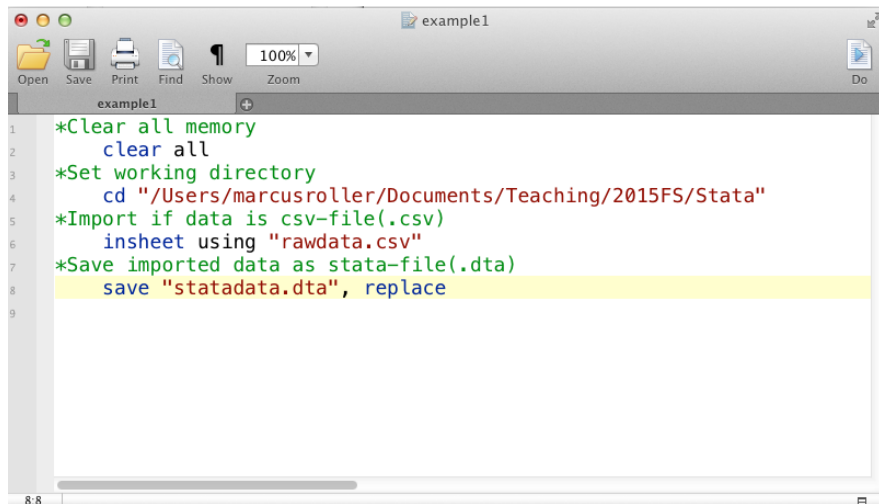
	A	B	C	D	E
1	id	wage	tenure	age	
2		1	20	10	30
3		2	21	35	55
4		3	25	25	45
5		4	24	41	61
6		5	23	8	28
7		6	29	16	36
8		7	42	22	42
9		8	28	1	18
10		9	21	35	55
11		10	18	29	49
12					

- First Row: Variable Names
- Columns=Variables
- Rows=Observations

Data Management III

- save the data as comma spread value (*mydata.csv*)
- you can directly use Excel Files (NOT before Stata 12)
- read the data into stata using the following command:
insheet using “mydata.csv”
- save the data by:
save “mydata.dta”, replace

Data Management Example



The screenshot shows a Stata command window titled "example1". The window has a menu bar with "Open", "Save", "Print", "Find", "Show", and "Zoom" (set to 100%). Below the menu bar is a toolbar with icons for these functions. The main area of the window contains a list of Stata commands, each preceded by a line number from 1 to 9. The commands are: 1. *Clear all memory, 2. clear all, 3. *Set working directory, 4. cd "/Users/marcusroller/Documents/Teaching/2015FS/Stata", 5. *Import if data is csv-file(.csv), 6. insheet using "rawdata.csv", 7. *Save imported data as stata-file(.dta), 8. save "statadata.dta", replace. The command on line 8 is highlighted in yellow. The status bar at the bottom left shows "8:8".

```
1 *Clear all memory
2   clear all
3 *Set working directory
4   cd "/Users/marcusroller/Documents/Teaching/2015FS/Stata"
5 *Import if data is csv-file(.csv)
6   insheet using "rawdata.csv"
7 *Save imported data as stata-file(.dta)
8   save "statadata.dta", replace
9
```

Stata Commands

Stata commands usually take the following form:

regress *wage tenure age* *if gen==1* , *vce(robust)*

command variable list condition options

- Each command has its own options
- All options are listed in the help menu (e.g. *help regress*).
- Almost all commands on data can be accompanied by logical conditions.
- If you do not know the relevant command → see “Where to find help”

Operators

Mathematical Operators	Logical Operators
<code>==</code> equals	
<code>></code> greater	<code>&</code> and
<code><</code> less	<code> </code> or
<code>>=</code> greater or equal	<code>!</code> not
<code><=</code> less or equal	

Data Manipulation

- Generate new variables:
generate area=length*width
- Replace existing variables:
replace length=width
- Generate variables containing statistics:
egen meanlength=mean(length)
- Generate variables containing statistics for groups:
egen meanlength=mean(length), by(brand)
- Drop variables:
drop length
- Drop observations:
drop if length>10

Data Manipulation

- Generate dummies:
generate dummy=0 if length<=4
replace dummy=1 if length>4
- OR:
generate dummy=0 if !missing(race)
replace dummy=1 if race==2
- If reference variable contains missing values:
generate dummy=0 if length<=4
replace dummy=1 if length>4 & !missing(length)

Summary statistics

- Mean, Variance,...:
summarize length width race
- With percentiles:
summarize length width race, detail
- Correlations:
correlate length width
- Frequencies:
tabulate race
- Frequencies twoway:
tabulate race sex
- Test Difference of means of two groups:
ttest wage, by(gender)

- Label Variables
label variable gender "Gender of Individual"
- Label Values:
label define sex 0 "Male" 1 "Female"
label values gender sex

Regressions

- Simple OLS:
`regress length width height`
- With robust standard errors:
`regress length width height, vce(robust)`
- and many more regression commands...
- Wald-Test
`test width=height=0`

- Histogram:
histogram length
- Scatter Plot:
scatter length width
- and many more graph commands...

Log-files

- save all output of a project in one text file
- Create log file in the header:
capture log close
log using "mylog.log", text replace
- Close log file at the very end of your do-file:
log close
- View log file:
view mylog. log