

Labor Economics with STATA

Liyousew G. Borga



September 30, 2015

PART I: Getting Started

STATA manual

- [Menu > Help > PDF Documentation](#)
- Command: `help documentation`. Type “help documentation” for more information.
- Complete Stata Manuals are on-line at
https://iweb.cerge-ei.cz/internal/sw_documentation/stata13/ [▶ Link](#)

Books

- Hamilton L. C. (2013). Statistics with STATA: Version 12, 8th Edition. Cengage. University of New Hampshire
- Kohler, U. and Kreuter, F. (2005). Data Analysis Using Stata. Stata Press

On Screen Help

- The `help` command is particularly useful when used with a command name.
- Typing `help correlate`, for example, causes a description of that command to appear in a Viewer window.

- A direct way of searching for information in Stata's documentation or in the web-site's FAQs and other pages
[Menu](#) > [Help](#) > [Search](#) > [Search documentation and FAQs](#)
- Command: The [search](#) command can do similar things. Type [help search](#) for more about this command.
- Alternatively, we can search net resources including the Stata Journal. Search results in the Viewer window contain clickable hyperlinks leading to further information or original citations
- If we mistakenly type a Stata command, Stata responds with the error message and cryptic "return code" `r()`:

```
table  
varlist required  
r(100);
```
- Clicking on the return code `r()` in this error message brings up a more informative note.

- The structure of commands is some subset of the following elements
[prefix:] command [varlist] [= exp] [if exp] [in range] [weight] [using filename] [, options]
- Elements displayed in brackets are optional; elements without brackets are required
- Square brackets distinguish optional qualifiers and options from required ones
 - “command” is the name of the command that should describe what it does;
 - “varlist” is the list of variable names separated by spaces;
 - “prefix” examples: “by varlist, sort” prefix makes Stata sort the dataset according to the variable in the varlist and repeat a command for each subset of the data for which the values of the variables in varlist are equal.
 - “=exp” specifies the value to be assigned to a variable and is most often used with “generate” and “replace”.
 - “if exp” qualifier restricts the scope of a command to those observations for which the value of the expression is true.
 - “in range” qualifier means that the command is to use only the observations specified “list price in -10/1” says to list the last 10 observations.
 - “weight” indicates the weight to be attached to each observation.
 - “options” denotes a list of options.

- The best way to prepare for recording the session is by opening a log file at the start. Log files contain commands and results tables, but not graphs.
- To start a log: `log using filename`; To temporarily stop logging: `log off`; To resume: `log on`
- To stop logging and close the file: `log close`
- To create “filename.log” that you can load into your word processor: `translate filename.smcl filename.log`
- To create a PDF of filename.smcl on Windows or Mac: `translate filename.smcl filename.pdf`
- To append to an existing log: `log using filename, append`
- To replace an existing log: `log using filename, replace`

Creating and running dofiles

- You can execute do-files by typing `do` followed by the filename
- You can execute do-files by selecting **File > Do...**
- You can use the Do-file Editor to compose, save, and execute do-files
- To use the Do-file Editor, click on the “Dofile Editor” button, or type `doedit` in the Command window.

An important part of directly created do files are “comments”; these are initiated by:

- `*` at the beginning of a line (anything on that line after is comment); note that if anything precedes the star, the star will be considered as multiplication sign (changes its meaning)
- `//` anywhere on the line; anything on given line after `//` is considered a comment
- `/*` in beginning and `*/` at the end; this allows for commands spanning more lines. `/* */` can be put anywhere, even in the middle of a line
- line breaks are issued by “enter” in do files; or by the command `#delimit`. We can change the delimiter back to carriage return by typing `#delimit cr`
- If a command is too long, an alternative is to use `///`, which orders Stata to ignore the end of a line, and naturally continue reading on the following line. Alternatively, use `/*` at the end of the first line `//` and `*/` at the beginning of the second line, which both have to be recognized as one line.

- We usually start the do file by set of commands:

```
version 13.1  
clear all  
set more off
```

- version number (to help deal with changing command names over versions);
- clear all (to clear the data from memory);
- set more off (to deactivate partitioned display of output)

Local macros

- Macros are names (up to 31 characters) that can stand for strings, program-defined numerical results or user-defined values.
- A local macro exists only within the program that defines it, and cannot be referred to in another program.
- To create a local macro named “iterate”, standing for the number 0, type

```
local iterate 0
```

- To refer to the contents of a local macro (0 in this example), place the macro name within left and right single quotes.

```
display 'iterate'
```

- Thus, to increase the value of iterate by one, we write

```
local iterate = 'iterate' + 1  
display 'iterate'
```

Global macros

- Once defined, global macros remain in memory and can be used by other programs for the duration of your current Stata session.
- To refer to a global macro's contents, we preface the macro name with a dollar sign

```
global distance = 73  
display $distance * 2
```

- Define path to files used during the seminar

```
global path_o="D:\Study\CERGE-EI\TA\Stata_seminar\Datasets"  
global path_w="D:\Study\CERGE-EI\TA\Stata_seminar\Datasets_working"
```

- If you enter filename without path, Stata automatically looks in the current directory
- There are several general commands related to working directory:

`pwd`- displays present working directory

`dir`- shows list of files that are stored in current folder

`cd`- changes directory, from which is stata working and where it is saving files

- Instead of setting the path, we could change current directory:

```
cd "D:\Study\CERGE-EI\TA\Stata_seminar\Datasets"
```

Hamilton L. C. (2013). Statistics with STATA: Version 12, 8th Edition. Cengage. University of New Hampshire, Ch. 1

- Given small time series, which covers satellite-era (1979 to 2011) observations of ice on the Arctic Ocean in September
- Variables:
 - "extent" is a satellite-based measure of the Northern Hemisphere sea area with at least 15% ice concentration each September
 - "area" numbers represent the area of sea ice itself
 - "tempN" describes mean annual surface air temperature above 64°N latitude. Temperatures are expressed as anomalies, which are deviations from the 1951-1980 average, in degrees Celsius.
- 33 observations (years) and 8 variables.

Opening a log file

```
log using $path_o\session1.smcl, replace
```

Alternative ways:

- **File > Log > Begin ...** from the top menu bar, and specify a name and folder for the resulting log file; Select which type of log file you want, click **Save**.
- Use Log button
- Log files can be created either in a special Stata format (.smcl), or in ordinary text or ASCII format (.log).
- A .smcl (Stata markup and control language) file will be nicely formatted for viewing or printing within Stata.
- It could also contain hyperlinks that help to understand commands or error messages.
- .log (text) files lack such formatting, but are simpler to use if you plan later to insert or edit the output in a word processor.

- Opening an existing Stata-format dataset named “Arctic9.dta”

```
use $path_o\Arctic9.dta, clear
```

Alternative ways:

- File > Open > Arctic9.dta using the top menu bar
- Use Log button
- A brief description of the dataset now in memory

```
describe
```

Alternative ways:

- Many Stata commands can be abbreviated to their first few letters.
- We could shorten ‘describe’ to just the letter ‘d’.
- Data > Describe data > Describe data in memory > (OK).
- List the first 10 observations (years)

```
list in 1/10
```

Alternative ways:

- Shorten to the letter “l” as in “l in 1/10”
- Data > Describe data > List data > (OK).

- Summary statistics

```
summarize
```

Alternative ways:

- Shorten to the letters ‘‘su’’
- `Statistics > Summaries, tables, and tests > Summary and descriptive statistics > Summary statistics > (OK)`

- Correlations among variables

```
correlate extent area volume tempN
```

Alternative ways:

- `Statistics > Summaries, tables, and tests > Summary and descriptive statistics > Correlation and covariance` Then choose the variables to be correlated.
- September sea ice extent, area and volume all have strong positive correlations.
- Their correlation with annual air temperature is negative: the warmer the air, the less ice (or vice versa).

- Plot “extent” against “year”

```
graph twoway connect extent year
```

- The first-named variable in this command, extent, defines the vertical or y axis; the last-named variable, year, defines the horizontal or x axis.
- We see an uneven but steepening downward pattern, as September sea ice extent declined by more than a third over this period.
- To print results **File > Print > Results**. Copy results to your word processor. Change the pasted text to a fixed-width font such as Courier.
- To print graph, go to the Graph window and click its “print icon” or **File > Print**.
- To copy the graph into a word processor, right-click on the graph, and select **Copy Graph**.
- To save the graph either right-click and Save Graph, click “save icon” in the Graph window,
- or select **File > Save As** from the Graph window’s top menu bar.

An Example Stata Session

- View log file session1.smcl
- File > Log > View > OK
- Print the log file by clicking the icon on the top bar of the log file's Viewer window.
- Log files close automatically at the end of a Stata session, or earlier if instructed by
- Close log file, typing the command

```
log close
```

or by choosing File > Log > Close

- To create an output file that can be opened easily by your word processor, either translate the log file from .smcl (a Stata format) to .log (standard ASCII text format) by typing:

```
translate $path_o\session1.smcl $path_o\session1.log, replace
```

or start out by creating the file in .log instead of .smcl format either by specifying that you want the log in text format:

```
log using session1, text
```

or by specifying that the file to be created is a .log file:

```
log using session1.log
```


Creating New Datasets

You can create a new dataset by:

- Typing in data
- Importing from other programs

Typing in data

- Selecting `Window > Data Editor` from the menu bar
- Typing the command `edit`
- Renaming and labeling variables can also be done through the `rename` and `label` variable commands:

```
rename (var1 var2 var3 var4 var5) (place pop unemp mlife flife)
label data "Canadian dataset 1"
label variable place "Place name"
label variable pop "Population in 1000s, 1995"
```

Importing from other programs

File > Import > Excel spreadsheet (.xls; .xlsx)

Or via the command line:

```
import excel using $path_o\snowfall.xls, sheet("Berlin") cellrange(a4:o56)
describe
```

- Spreadsheets often contain titles, notes, subtables, multiple sheets, graphs or other features
- To restrict the import operation to a particular range of cells, use a "cellrange()" option
- The "sheet()" option can specify what sheet within the spreadsheet to import
- A "firstrow" option tells Stata that the first row of these cells contain variables names.
- To meet Stata criteria, start variable names with a letter or underscore, and have no embedded blanks
- Missing values should be replaced with blanks or numerical codes

Importing from other programs

- We can read these data into Stata using the `insheet` command, with options to specify that values are tab-separated, and the first row contains variable names

```
insheet using $path_o\MEI.raw, tab name clear  
save $path_w\MEI0.dta, replace  
describe
```

- With a comma option instead of tab, `insheet` could read a text file of comma-separated values
- Type `help import` to learn more about different methods:

```
import excel, insheet, odbc, infile (free format), infix (fixed  
format), infile (fixed format), import sasxport, haver, xmluse
```

Importing from other programs

- We can read these data into Stata using the `insheet` command, with options to specify that values are tab-separated, and the first row contains variable names

```
insheet using $path_o\MEI.raw, tab name clear
save $path_w\MEI0.dta, replace
describe
```

- With a comma option instead of tab, `insheet` could read a text file of comma-separated values
- Type `help import` to learn more about different methods:
`import excel, insheet, odbc, infile (free format), infix (fixed format), infile (fixed format), import sasxport, haver, xmluse`

Note: To export data from Stata to other programs you may use:

- `File > Export > Excel spreadsheet (.xls; .xlsx)`
- Copy your data from Stata's Data Editor or Data Browser and paste this directly into a spreadsheet such as Excel
- The `outsheet` and `outfile` commands can write text files in several different formats

Combining Two or More Stata Files

Combine Stata datasets in two general ways:

- `append` a second dataset that contains additional observations
- `merge` with other datasets that contain new variables or values

Combining Two or More Stata Files

Append example

The File `lakewin1.dta` contains the ice-out dates for New Hampshire's largest lake recorded from 1887 through 2007

```
use $path_o\lakewin1.dta, clear
describe
list in -4/1
```

The File `lakewin2.dta` contains newer data from 2008 through 2012. It has the same variables and format, so we can combine the update in `lakewin2.dta` with the older information in `lakewin1.dta`, using the "append" command.

```
use $path_o\lakewin2.dta, clear
describe
list

append using $path_o\lakewin1.dta

sort year
label data "Lake Winnepesaukee ice out 1887-2012"
save $path_w\lakewin3, replace
list in -7/1
```

Combining Two or More Stata Files

Merge example

- File `lakesun.dta` contains ice out dates for New Hampshire's lake over the years 1869 through 2012.
- Lake Sunapee (`lakesun.dta`) and Lake Winnepesaukee (`lakewin3.dta`) both form yearly series that could easily be combined into one dataset.

```
use $path_o\lakesun.dta, clear
sort year
describe
merge 1:1 year using $path_w\lakewin3.dta

label data "Sunapee & Winnepesaukee ice out 1869-2012"
describe
save $path_w\lakesunwin.dta, replace
```

Combining Two or More Stata Files

Merge example

- The merge results tell us that 126 years were present in both the “master” dataset (the data currently in memory, `lakesun.dta` in this example) and the using dataset (`lakewin3.dta`).
- A further 18 years (1869 to 1886) existed only in `lakesun.dta`, so the Lake Winnepesaukee variables will have missing values in those years
- `merge` commands create a variable named `_merge` that records whether the observation came from:

```
the master data only (_merge = 1)
the using data only (_merge = 2)
both (_merge = 3)
```

- By default, whenever the same variables are found in both datasets, those of the master data are retained and those of the using data ignored

Combining Two or More Stata Files

Merge Options

```
merge 1:1 year using newdata.dta, update
```

- A command would allow any missing values in the master data to be replaced by corresponding nonmissing values found in the using data (here, newdata.dta)

```
merge 1:1 year using newdata, update replace
```

- A command causes any values from the master data to be replaced by nonmissing values from the using data, if the latter are different

Type `help merge` for other merge options:

- Also possible are 1-to-many (1:m), many-to-1 (m:1) or many-to-many (m:m) operations.
- Merging and appending data can be accomplished through **Data > Combine** datasets menus as well.