

Lab 2 - Introduction to Stata, Descriptive Statistics, Do-Files

Spring 2017

Contents

1	Do-files	2
2	Save and open your own dataset	4
3	Stata help	4
4	Log-files	4
5	Data Structures	9

1 Do-files

- A Do-File is a set of commands that will be executed by Stata when the file is opened. So, instead of typing commands one-by-one into the command window, you can type them all in one go within a do-File and simply run the do-File once.
- In itself, a Do-File is a text file saved with a `.do` extension. It can be edited using any text editor.
- A do-file allows you to have an exact record of what commands you have done and to replicate your analysis at a later time.
- To create a do-File, go to Windows \downarrow Do-file Editor \downarrow New do-file editor. You can then enter your commands in this Do File.

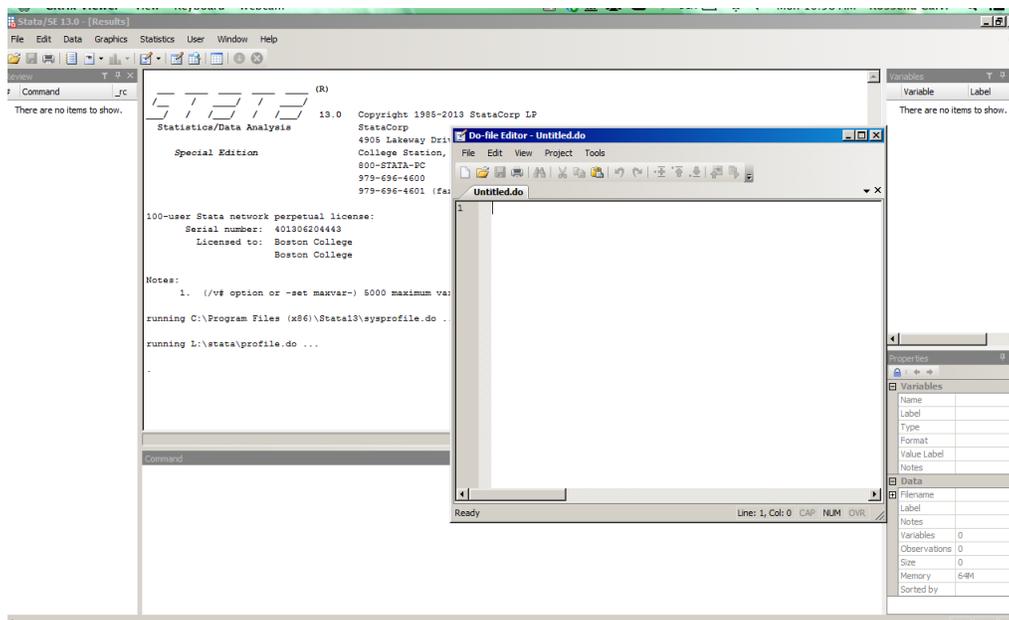


Figure 1: Open a new Do-File

Note: When you type the commands into a do file, Stata does not run them immediately! (However you can run the commands by using the appropriate button on the menu bar.) On the other hand, if you type commands in the command window, Stata will run them immediately. Therefore, a common strategy for completing problem sets is to type the commands in the command window and copy them into the do file when they work successfully.

- Example: try the following program below:

```
cd "L:\EC227"  
clear all  
bcuse affairs  
describe  
correlate naffairs ratemarr
```

Click the Execute button on the top row, and all the commands will be run at once.

- **Commenting a Do-File:** It's a good idea to put comments in your do-file so that you or other people know exactly what you're doing. Anything you write in a line with an `*` at the beginning will not be executed. Example:

```
* My first do-file

*make folder "EC227" my working directory
cd "L:\EC227"
*clear everything in memory
clear all
*load Affairs Data
bcuse affairs
*obtain a description of variables
describe
*calculate the correlation between happiness of marriage and number of affairs
correlate naffairs ratemarr
```

- You may also want to put this line at the beginning of your do-file:

```
set more off
```

This will turn off the option that causes Stata to display, by default, `-- more --` and pause until any key is pressed. `set more on` will turn it back on

2 Save and open your own dataset

- At some point you may modify an existent dataset or create your own. You can save your dataset in your current directory using the command

```
save dataset_name
```

- Always make sure your current directory is set correctly before saving in order to be able to locate the file afterwards!
- If you keep working on the same dataset and you want to overwrite an existing file with the dataset you are currently using, type instead

```
save dataset_name, replace
```

- Finally, when you want to load a dataset from your hard drive (or from the L drive), first set your current directory to the directory where your dataset is saved and then type

```
use dataset_name, clear
```

- The option `clear` clears Stata's memory before loading your dataset.

3 Stata help

- To get help on using Stata commands type `help` (command) e.g. `help describe`. **Note:** use `help` only if you know the name of the Stata command, otherwise you will get nothing. If you don't know the name of the command you need you can search for it. Stata has a `search` command with a few options, type `help search` to learn more; but I prefer `findit`, which searches the Internet as well as your local machine and shows results in the *Viewer*.
- Stata also has an online help manual. So, if you Google any of these commands, you should get a link to the relevant online Stata page. For more help, you can consult StataCorp's "Frequently Asked Questions" <http://www.stata.com/support/faqs/>
- If you are stuck with a programming problem, the Statalist archives may be useful. <http://www.stata.com/statalist/archive/>
- Finally, UCLA has an excellent website for those trying to learn to use STATA: <http://www.ats.ucla.edu/stat/stata/sk/default.htm>

– You may find it useful to start a google search with "stata ucla ..."

4 Log-files

- A **log-file** keeps a record of all the commands you put into Stata AND the resulting Stata output during the period that the log-file is open. Note that this does not happen when you simply run a do-file! When you push the "execute" button on the top row of a do-file, you will get the results in the Results window directly. In contrast, when you open a log-file before running your commands, your results will

- appear in the Results window, and
 - be saved in a separate document (for example, `.smcl`)
- Also, the Results window holds a limited amount of information (output). When the limit is reached, older output at the top of the buffer is replaced by new output when it appears at the bottom of the Results window. So, in order to see all your results, you should use a log-file for your work.
 - Stata will automatically save log-files with the extension `.smcl`, but you can choose an alternative extension, such as `.txt`. If you have a lot of output, it can be easier to open the log file in a text editor and read it there, instead of in the Results window.
 - By default, Stata will save your log-files in the current working directory. So it is important to set to it before you start your work.
Note that you do not need to manually save your log-file, since it is automatically saved.

- **How to create a log-file**

1. Open the do-file editor (go to Windows *i* Do-file Editor *i* New do-file editor).
2. Write a do-file with the following commands:

```
cd "L:\EC227"
clear all
capture log close
log using lab1_yourname, text replace
webuse auto
summ price
tab foreign
corr price mpg
log close
exit
```

3. run it

- **What have we generated?**

- We specified the Current Working Directory, “EC227”, with the `cd` command.
- We cleared everything in memory with the `clear` command
- We created a log-file with the line:

```
log using lab1_yourname, text replace
```

- * `log using` actually creates the log-file and it has to be followed by the file name you want (it can be anything),
 - * the commands after the comma are the options,
 - * if the log-file already exists, the `replace` option allows it to be overwritten,
 - * the `text` option saves the log-file in a format that can be opened by most editors (say Notepad for Windows or TextEdit for Mac).
- We put the list of commands we wanted: `bcuse, summ, tab, corr`
 - We closed the log-file by typing `log close`.

- It is also a good idea to put `capture log close` at the beginning of your do-file, before the `log using` command, so to close all log-files that may still be open.

IMPORTANT: Locate the log-file in the “EC227” folder!

SUGGESTION FOR PROBLEM SETS (not only): Use the example do-file above as a template for all Problem Sets:

- begin your do-files with:

```
cd "L:\EC228"  
clear all  
capture log close
```

- create the log-file with `log using` and just change the filename
- insert the new list of commands for each Problem Sets
- `log close` at the end

- to **view your log-file in Stata**, you can type

```
view log1_yourname.log
```

in the command window (make sure you are specifying the correct extension, either `.smcl` or `.log`, depending on the options you used when opening the log-file)

- to **print your log-file from Stata**, you got to `FILE-;PRINT-;VIEWER`
- to **open your log-file in a text editor**, you can go to your L: drive (from Finder or your desktop) and open it; you can also print your log-file from the text editor
- in the next two pages you can find how a log-file looks like: as you can see, you have both the commands and the results

```

-----
name: <unnamed>
log: L:\EC228\log1_yourname.log
log type: text
opened on: 15 Sep 2013, 14:19:27

```

```

. *load Automobile Data
. webuse auto
(1978 Automobile Data)

```

```

. *obtain a description of variables
. describe

```

```

Contains data from http://www.stata-press.com/data/r13/auto.dta
obs:          74          1978 Automobile Data
vars:         12          13 Apr 2013 17:45
size:        3,182        (_dta has notes)

```

```

-----
variable name    storage   display   value     variable label
                  type     format    label
-----
make             str18    %-18s          Make and Model
price            int      %8.0gc        Price
mpg              int      %8.0g         Mileage (mpg)
rep78            int      %8.0g         Repair Record 1978
headroom         float    %6.1f         Headroom (in.)
trunk            int      %8.0g         Trunk space (cu. ft.)
weight           int      %8.0gc        Weight (lbs.)
length           int      %8.0g         Length (in.)
turn             int      %8.0g         Turn Circle (ft.)
displacement     int      %8.0g         Displacement (cu. in.)
gear_ratio       float    %6.2f         Gear Ratio
foreign          byte     %8.0g         origin      Car type
-----

```

```
Sorted by: foreign
```

```

. *obtain detailed summary statistics for the variable price
. summarize price, detail

```

```

-----
Price
-----
Percentiles      Smallest
1%                3291          3291
5%                3748          3299
10%               3895          3667      Obs          74
25%               4195          3748      Sum of Wgt.  74

50%              5006.5
75%               6342          Largest
90%               11385         13466
95%               13466         13594
                  14500      Variance     8699526
                  Skewness     1.653434
-----

```

99% 15906 15906 Kurtosis 4.819188

```
. *obtain frequency of foreign  
. tab foreign
```

Car type	Freq.	Percent	Cum.
Domestic	52	70.27	70.27
Foreign	22	29.73	100.00
Total	74	100.00	

```
. *obtain correlation and covariance between price and mpg  
. corr price mpg  
(obs=74)
```

	price	mpg
price	1.0000	
mpg	-0.4686	1.0000

```
. corr price mpg, cov  
(obs=74)
```

	price	mpg
price	8.7e+06	
mpg	-7996.28	33.472

```
.  
. *close log-file  
. log close  
  name: <unnamed>  
  log: L:\EC228\log1_yourname.log  
  log type: text  
  closed on: 15 Sep 2013, 14:19:28
```

5 Data Structures

1. Cross Sectional Data

- Each observation is a new individual, firm, etc. with information at a point in time

```
webuse auto, clear
```

2. Time Series Data

- Time series data has a separate observation for each time period e.g. stock prices

```
bcuse conump, clear
```

3. Panel Data

- Look at multiple individuals over time

```
bcuse cornwell, clear
```