

Introduction to STATA and Do-file

Prof. Tzu-Ting Yang

楊子霆

Institute of Economics, Academia Sinica

中央研究院經濟研究所

- Stata is a full-featured statistical programming language for Windows, Mac OS X, Unix and Linux
- Stata is available in several versions: Stata/IC (the standard version), Stata/SE (an extended version) and Stata/MP (for multiprocessing)
- The major difference between the versions is the number of variables allowed in memory, which is limited to 2,047 in standard Stata/IC, but can be much larger in Stata/SE or Stata/MP

Stata's user interface

Stata's user interface

Stata 12/13+ screen

Variables in dataset here

The screenshot shows the Stata 13.1 interface with several windows and annotations:

- Review Command window:** Shows the command history. An arrow points to it with the text "History of commands, this window".
- Command window:** At the bottom, where commands are entered. An arrow points to it with the text "Write commands here".
- Output window:** The central area showing the results of commands. An arrow points to it with the text "Output here". It contains the following text:

```
Notes:
      1.  (/v# option or -set maxvar-) 5000 maximum variables

. cd H:
H:\

. log using mywork.log

name: <unnamed>
log: H:\mywork.log
log type: text
opened on: 14 Apr 2014, 15:28:47

. import excel "http://dss.princeton.edu/training/mydata.xls", sheet("Sheet1") firstrow clear

. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
Year	0				
CountryName	0				
GDPpercapa-200	4542	9482.967	11285.24	101.5976	76319.47
Unemploye-m	4521	.0478866	.0724682	0	.686
Unemploye-b	4521	.0366029	.0544155	0	.546
Unemploye-l	4521	.0425112	.0601523	0	.595
Exportsofg-o	3661	6.49e+10	1.64e+11	4.50e+07	1.78e+12
Importsf-g-o	3661	6.43e+10	1.74e+11	9.42e+07	2.20e+12
polityorig-l	4542	-.2573756	16.28321	-88	10
polity2adj-d	4498	2.409738	7.03114	-10	10

An arrow points to the "Year" row in the table with the text "?????".
- Variables window:** On the right, showing a list of variables in the dataset. An arrow points to it with the text "Variables in dataset here".
- Properties window:** On the right, showing the properties of the selected variable (Year). An arrow points to it with the text "Property of each variable here".

Files will be saved here

Stata's user interface

- The Toolbar contains icons that allow you to Open and Save files, Print results, control Logs, and manipulate windows
- Some very important tools allow you to open the Do-File Editor, the Data Editor and the Data Browser
- The **Data Editor** and **Data Browser** present you with a spreadsheet-like view of the data, no matter how large your dataset may be
 - Type **Ctrl+8** to call **Data Editor**
- The **Do-File editor** allows you to construct a file of Stata commands
 - Type **Ctrl+9** to call **Do-File editor**

Stata's user interface

- There are four windows in the default interface:
- **Command Review window:**
 - When a command is executed-with or without error-it appears in the Review window
 - You may click on any command in the Review window and it will reappear in the Command window
- **Variables window:**
 - Once you have loaded data into the program, the Variables window will be populated with information on each variable
 - That information includes the variable name, its label (if any), its type and its format

- **Results window:** The results of the command (or an error message) appears in the Results window
- **Command window:** You may only enter one command in that window, so you should not try pasting a list of several commands

How to Type Commands: Use do files

Type Commands

- You can enter commands in either of three ways:
 - 1 **Interactively:** you click through the menu on top of the screen
 - 2 **Manually:** you type the first command in the command window and execute it, then the next, and so on
 - 3 **Do-file:** type up a list of commands in a “do-file”, essentially a computer programme, and execute the do-file

- The vast majority of your work **should use do-files**
- If you have a long list of commands, executing a do-file once is a lot quicker than executing several commands one after another
- Furthermore, the **do-file is a permanent record of all your commands** and the order in which you ran them

Typical Format of a Do-file

```
1 clear
2
3 set more off
4
5 [STATA commands]
```

STATA command: clear

1 clear

- **clear**: clean up any previous data, otherwise you can not load new dataset

STATA command: set more off

1 `set more off`

- When there are a lot of results in the results window, Stata pauses the do-file to give you a chance to review each page on-screen and you have to press a key to get more
- **set more off** tells Stata to run the entire do-file without pausing.
- You can then review the results in the log file

How to Put Your Notes in Do-file

Put Your Notes in Do-file

- It is good practice to keep extensive notes within your do-file
- Thus, when you look back over it you know what you were trying to achieve with each command or set of commands
- You can insert comments in several different ways

Method 1: Put Your Notes in Do-file

- Stata will ignore a line if it starts with **two consecutive slashes //**, so you can type whatever you like on that line.

```
1 use "$rawdata\acs_2015.dta" // opens 2015 acs  
   data
```

- Note, comments are also useful for getting Stata to temporarily ignore commands
- If you decide later to re-insert the command into your do-file, just delete the slashes or the asterisk

Method 2: Put Your Notes in Do-file

- You can place notes after a command by inserting it inside these **pseudo-parentheses** `/* */`, for example:

```
1 use "$rawdata\acs_2015.dta" /* opens 2015 acs data */
```

- These **pseudo-parentheses** are also useful for temporarily blocking a whole set of commands:
 - Place `/*` at the beginning of the first command, `*/` at the end of the last, and Stata will just skip over all of them

Method 3: Put Your Notes in Do-file

- Lastly you can use **three consecutive slashes** `///` which will result in the rest of the line being ignored and the next line added at the end of the current line
- This comment is useful for splitting a very long line of code over several lines

Method 3: Put Your Notes in Do-file

```
1 graph twoway (scatter age inctot if year == 2015) ///  
    This combines two scatter plots  
2 (scatter age incwage if year == 2015)
```