

Getting Started with Stata

Session Four: Automation and Programming

Yiming Cao

Department of Economics, Boston University
Office: SSW B02
Email: yiming@bu.edu
Website: <http://ymcao.weebly.com>

October 1, 2019

Previous Sessions

- Command syntax
- Data management
- Data visualization
 - Single-variable plots:
graph plotype varlist, options
 - Two-way plots:
graph twoway plotype varlist, options
 - Overlay multiple plots
 - Options

Today: Automation and Programming

- Working with dofiles
- Programming essentials
 - Macros
 - Loops
 - Conditional execution
- Recap and exploration
 - The command cheat sheet
 - User-written packages
 - General Advices

Why Dofile?

- Make your work replicable
- Facilitate the debugging process
- Save time and energy

The Dofile Editor

- To start the dofile editor:
 - Type in the command line: **doedit**
- To execute:
 - Click the **do** button in the tools menu
 - Hit **Ctrl+D** in Windows, or **Shift+Command+D** in MacOS
 - Type **do** in the command window or within another dofile
 - You can execute part of the code by selecting the relevant lines only

Comments

- Begin a line with a `/*`; Stata ignores such lines

```
1 /* I am a comment
```

- Place the comment within the `/* */` delimiters

```
1 /* I  
2     am  
3     another  
4     comment */
```

- Place the comment after two forward slashes, that is, `//`

```
1 // I am also a comment
```

- Examples

Long Lines

- Change the end-of-line delimiter (e.g., to ‘;’) by using ‘#delimit’

```
1 #delimit ;  
2 graph twoway (scatter y x)  
3             (lfit y x);  
4 #delimit cr
```

- Use the line-join indicator: ///
 - You can also add comments after the line indicator

```
1 graph twoway (scatter y x) ///  
2             (lfit y x)
```

- Examples

Debugging

- Stata will stop the execution of the dofile if it encounter an error and return an error message
- To ignore the error and continue the execution, put **capture** before the command (Be Cautious!)
- Hit the **break** button in the tool menu to stop the execution of a dofile

Macros

- Macros are the ‘variables’ in programming
 - Distinguish them from variables in the dataset
 - Think of a **macro** as simply a name associated with some text (to be called or evaluated later)
 - Two types of macros: **local** and **global**
- Local macros: valid only in the current program
 - To define: **local** macroname [=] macrocontent
 - To evaluate: **`macroname`**



- Local macros void after the dofile's execution finishes
- Global macros: available to all programs
 - To define: **global** macroname [=] macrocontent
 - To evaluate: **\$macroname**
 - Global macros remain valid after the dofile's execution finishes
- You can store the returned results (from commands) as macros:
 - **r()** stores results from general commands
 - **e()** stores results from estimation commands
 - Note the returned results may either be **scalars** or a **matrices**
- Examples

Loops

- Looping over sequences of numbers:

```
1 forvalues lcname=numlist {  
2 ... body of loop using 'lcname' ...  
3 }
```

- Looping over elements in a list:

```
1 foreach lcname in a_list_of_elements {  
2 ... body of loop using 'lcname' ...  
3 }
```

- Looping over specialized lists:

```
1 foreach lcname of varlist a_list_of_variables {  
2 ... body of loop using 'lcname' ...  
3 }
```

- Examples

Conditional Execution

- The **if** condition:

```
1 if some_expression {  
2 ... commands to be executed if the expression is true ...  
3 }  
4 else {  
5 ... optional block to be executed if the expression is false ...  
6 }
```

- Simplified version with one single command:

```
1 if some_expression the_command_to_execute_if_expression_is_true
```

- Distinguish the **if** here from the **if** qualifier in the command syntax
- Examples

For Reproduction

- Download the dofile that contains all the examples [HERE](#) for reproduction.

Command Cheat Sheet I

- Syntax: **[prefix:] command [varlist] [if] [in] [weight] [,options]**
- Set up
 - Open the dofile editor: **doedit**
 - Show the current working directory: **pwd**
 - Change directory: **cd**
 - Open a log file: **log using**
 - Close a log file (without errors): **capture log close**
 - Clear the -more- option in the results window: **set more off**
 - Unload the current data from memory: **clear**
 - Getting help: **help**
 - Exit Stata: **exit**
- Open and import data:
 - Open Stata formatted dataset (.dta): **use, webuse, sysuse**
 - Import dataset from Excel: **import excel using filename, [firstrow]**
 - Import dataset from delimited text (.csv, .txt, ...):
import delimited using filename
- Save and export data:
 - Save data in Stata formatted (.dta): **save**
 - Save data to Excel: **export excel**
 - Save data to Delimited text: **export delimited**

Command Cheat Sheet II

- Inspect dataset:
 - Open the data editor: **edit**
 - Open the data editor in read-only mode: **browse**
 - List the observations in the result window: **list**
 - Describe the structure of the dataset: **describe**
 - Explore the contents of the variables: **codebook**
 - View summary statistics: **summarize**
 - Find one or more variables: **lookfor**
 - Show a breakdown of values for one or multiple variables: **tabulate**
 - Count the number of observations that meet certain criteria: **count if**
 - Generate the correlation table: **correlate**
 - List all the duplicated observations: **duplicates list**
- Data management
 - Create new variables: **generate**
 - Update the value of existing variables: **replace**
 - Create new variables with advanced functions: **egen**
 - Change the name of a variable: **rename**
 - Create a text description of a variable: **label variable**
 - Create text descriptions for a set of numerical values: **label define**
 - Assign value labels (defined previously) to a variable: **label values**
 - Remove variables or observations if meeting certain criteria: **drop**
 - Remove variables or observations if not meeting certain criteria: **keep**

Command Cheat Sheet III

- Data management (cont.)
 - Change the order of the variables in the dataset: **order**
 - Re-order the dataset based on the value of one or more variables: **sort, gsort**
 - Recode a variable based on its values: **recode**
 - Convert numerical variables to string ones: **tostring**
 - Convert string variables to numerical ones: **destring**
 - Create numerical variables (with value labels) based on string categories: **encode**
 - Create string variables from value labels of the numerical variables: **decode**
 - Combine multiple datasets based on one or more key variables: **merge**
 - Append observations from another dataset to the current dataset: **append**
 - Change the structure of the dataset: **reshape**
 - Make duplicated observations: **expand**
 - Rectangularize the dataset: **fillin**
 - Make a dataset that contains the summary statistics of the original dataset: **collapse**
- Data analysis:
 - Conduct a t-test: **ttest**
 - Linear regression: **regress yvar xvars**
 - Linear regression with robust standard errors: **regress yvar xvars,robust**
 - Set up the panel identifiers: **xtset xvar tvar**
 - Panel data regression with fixed effects: **xtreg yvar xvars,fe**
 - Get fitted values after the regression: **predict varname**
 - Store estimation results: **estimates store**

Command Cheat Sheet IV

● Graphics

- Histogram plot: **histogram**
- Kernel density plot: **kdensity**
- Bar plot: **graph bar**
- Box plot: **graph box**
- Twoway scatter plot: **twoway scatter**
- Twoway line plot: **twoway line**
- Twoway fitted line plot: **twoway lfit**
- Combine multiple graphs (saved as .gph): **graph combine**
- Export graphs: **graph export**

● Programming

- Define local macros: **local**
- Define global macros: **global**
- Display content of a variable: **display**
- Looping over sequences of numbers: **forvalues lcname=numlist**
- Looping over elements in a list: **foreach lcname in list_of_elements**
- Looping over a list of variables: **foreach lcname of varlist list_of_variables**
- Preserve the current dataset and restore it after the termination: **preserve**
- Force a restore of the preserved dataset: **restore**
- Ignore the error and execute the remaining actions: **capture**

User-written Packages

- You can download user-written packages from the Statistical Software Component (SSC) Archive
 - Use the command: **ssc install packagename**
- Some useful packages:
 - To download data from worldbank database: **wbopendata**
 - To convert shapefile into Stata formatted files: **shp2dta**
 - To Plot maps: **spmap**
 - Linear regression with high-dimensional fixed effects: **reghdfe**
 - Advanced Instrumental variable regressions: **ivreg2**
 - To plot estimated coefficients and confidence intervals: **coefplot**
 - To make publication-standard tables: **estout**
- To view a list of the most popular packages: **ssc whatshot**

General Advices

- Learn by working on your own project
- Always work from the dofile, if possible
- Comment your dofile in detail
- Refer to the help documents

Enjoy!