Development Data Boot Camp
Intro to Stata: Do-files

Ge Sun

University of Notre Dame

May 17, 2023

Outline

Outline

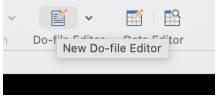
What is do-file?

- A do-file is a text file that contains commands instructing Stata to execute specific tasks.
- Instead of manually typing commands, writing them in a do-file allows for automated and reproducible data analysis.
- Stata provides the Do-file Editor, an integrated text editor that facilitates writing do-files. According to the Stata website, the Do-file Editor offers advanced features and is highly convenient to use.
- Additionally, there are other advanced text editors available, such as VSCode with the Stata plug-in.
- In this session, we will introduce how to utilize the Do-file Editor to write and execute do-files.

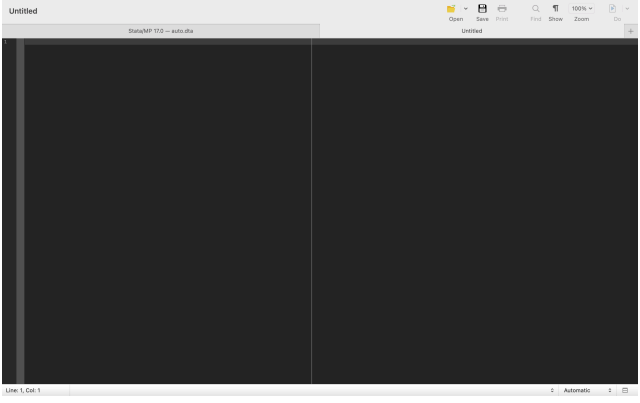
Outline

Do-file Editor

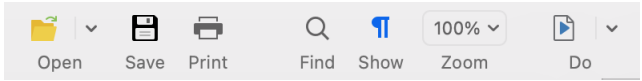
- Click on the Do-file Editor button



to open Do-file Editor.



Do-file Editor



Toolbars includes:

- New, Open, Save.
- find: for finding text, command+F (MacOS).
- Undo: undo the last change, command+Z (MacOS).
- Execute (do): run the commands in the do-file, showing all commands and their output. If text is highlighted, the button becomes the Execute Selection (do) button and will run **only the selected lines** and show the corresponding output.

Digressions: How to write comments?

- Comments are texts not executed in Stata, which usually displayed in a different color.
- Why are comments important?
  - We use comments to annotate our codes, reminding ourselves of why we did one thing and helping our collaborators understand what we did.
  - We use comments to divide long scripts. Comments can divide our codes into different parts to make long scripts more readable.
  - We use comments to remove certain parts of code temporarily. If we do not want to run certain parts of code but do not want to delete them (which usually happens when debugging), we can make them comments.

- How to write a comment in the Do-file:
  - For simgle-line comments, begin the line with an asterisk(*).
  - For a comment on the same line as a Stata command, use two slass(//) after the Stata command.
  - For multiple-line comments, /*...*/

```
************************** An Example ***************************************
*****************************************************************************


// open a new log file
log using introlog.log, replace     /* this opens a new log file */


// open one dataset
use baseline_census_cleaned.dta, clear
```

Outline

1. Comments at the beginning

The first thing when starting a new do-file, is writing a small paragraph at the beginning of the do-file that explicitly describes what the program does, which usually consists of:

- Title of the Do-file (what do you want to do in this Do-file).
- Date of then last time writing this code.
- The database used in this Do-file.
- The name of the output (*.dta) of this Do-file (optional).
- Version of the Stata (optional).

2. Basic settings

Several commands are used in this section:

1. *set memory 1000 M*: this command sets memory size allocated to Stata (if you want to allocate more memory to Stata). You can replace *1000* with other numbers as long as it does not exceed the limit of memory (which depends on your Stata implementation and the size of RAM of your laptop).
2. *set matsize 300*: this command sets the size of matrix (the number of variables in the dataset) in Stata.

Memory and matrix sizes are no longer needs to be set in modern Statas. Memory adjustments are performed on the fly automatically. Type *help set memory* for more details.

2. Basic settings

The other three commands are:

3. *clear*: this command clears any dataset that is already open in Stata.
4. *clear all*: not only closes the datasets, but also clears scalar and matrix.
4. *set more off*: "this command tells Stata to run the commands continuously without worrying about the capacity of the Results window to display the results. Otherwise Stata will pause each time the screen is full, unless we keep hitting –more– at the bottom of the Results window."

2. Basic settings

The last command is:

6. *capture log close*: this command closes any pre-existing open log files.

- If log file already exists, the command *log close* will close the pre-existing log files and then move to the next line.
- However, if there is no opened log files, the command *log close* will encounter an error message: no log file open. The prefix *capture* will ignore this error message and help Stata move to the next command.

Digressions: log files

- A log file is a file that contains your commands and the corresponding Stata's output. Sometimes, it is wise to retain the log file of the work you've done on a given project, to refer to while you are writing up your findings, or later on when you are revising a paper.
- You can use log files by command:

```
log using file_name, replace
```

- You can close the log file by command:

```
log close
```

- Type *help log* for more details.

Prefix: *capture*

- How to understand prefix *capture*? *capture* executes following command and suppresses all its output (including error messages, if any), which allows program to run even if the command encounters an error.
- In this case, adding prefix capture ensures that even if we can not successfully close the log file, we can still it skip and go to the next command.

3. Directory

- Directory is where you save datasets, do-files and outputs.
- Directory can be represented by absolute path or relative path:
  - An absolute path starts at the root and lists all the folders and subfolders needed to get from there to the location of the file or folder, for example,

    "/Users/gesun/Documents/Bootcamp/"

  - A relative path is the hierarchical path that locates a file or folder on a file system starting from the current directory. For example, if we are currently in the directory "/Users/gesun/Documents/Bootcamp/dofiles", then the relative path "/dofiles" is equivalent to

    "/Users/gesun/Documents/Bootcamp/dofiles"

3. Define directory

- Four useful commands relative to directory in Stata are:
  - *cd*: short for "change directory". We usually change it to our work directory, which includes datasets and Do-files.
  - *pwd*: short for "print work directory", Which tells us the directory of current folder.
  - *sysdir*: short for "system directory", which tells where STATA's default directory is.
  - *dir*: this command displays all the filenames in this current folder.

4. Write our first Do-file!

- Open log files:

```
log using log_name, replace
```

- Import data (usually *.dta):

```
use data_name.dta, clear
```

- Data processing.
- Save data or import output.