Introduction to Stata

Lecture I

Tomas R. Martinez

UC3M

September, 2019

Objectives

- **Goal:** Give you your first introduction on Stata - No previous knowledge required!
- If you are familiar with the software it can be a bit boring in the beginning, still I believe you will get something new by the end
- I will assume no knowledge of Econometrics, but some basic grasp of statistics might be helpful

What we will cover?

- Introduction: Help, do-files, log file
- Importing data
- Data manipulation
- Summarize our data
- Graphs
- Regressions: linear regression, time series, panel data
- Post estimation: exporting, residuals, inference
- Advanced: local and global variables, loops, if clauses, organizing your do-file

What is Stata?

- What is Stata?
  - Statistical software designed mainly for econometrics, biostatistics, and social scientists
- What are the other options out there?
  - “Easy” to use: Eviews, SPSS
  - “Bit harder” to use: Python, Matlab, R, Gauss, Julia
  - “Harder” to use: Fortran, C, C++

Why are we using Stata?

- Why are we using Stata?

Why are we using Stata?

- Why are we using Stata?
  - BECAUSE THEY TOLD US SO.....

Why are we using Stata?

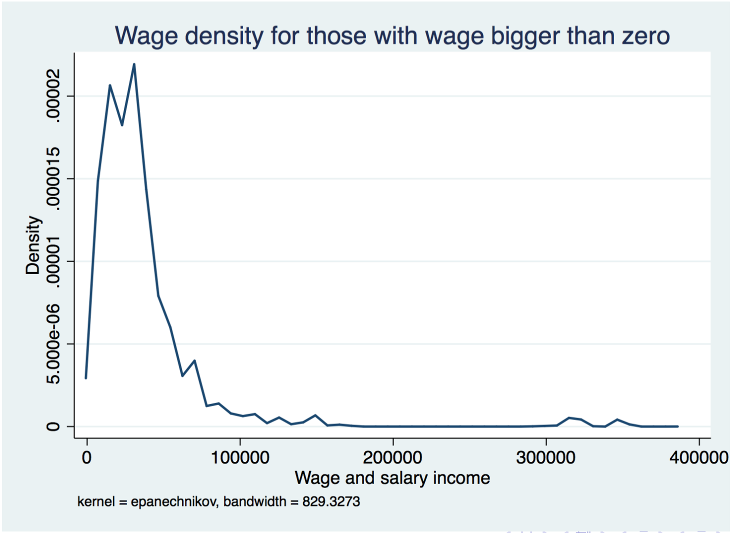
- Why are we using Stata?
  - BECAUSE THEY TOLD US SO.....
- **Good:**
  - Simple to use: spreadsheet-like but with in-line execution interface
  - Widely used in the econometrics community: lots of built in models and people writing commands for it!
  - Good graphing features, relatively fast even with large data
  - Combines graphical user interface with command lines and scripts
- **Bad:**
  - You have to pay for it
  - To do serious programming on it sometimes is very cumbersome
  - Only allows you to work with one dataset at a time
  - Outside of econometrics is not as powerful (e.g. GIS data or Machine Learning)

Small demo on the features of Stata

- United States Census (5%)
  - IPUMS web page
  - Data 2000
  - People older than 25, with complete information on past 12 months wage, age and gender
  - MORE THAN 9 MILLION OBSERVATIONS!

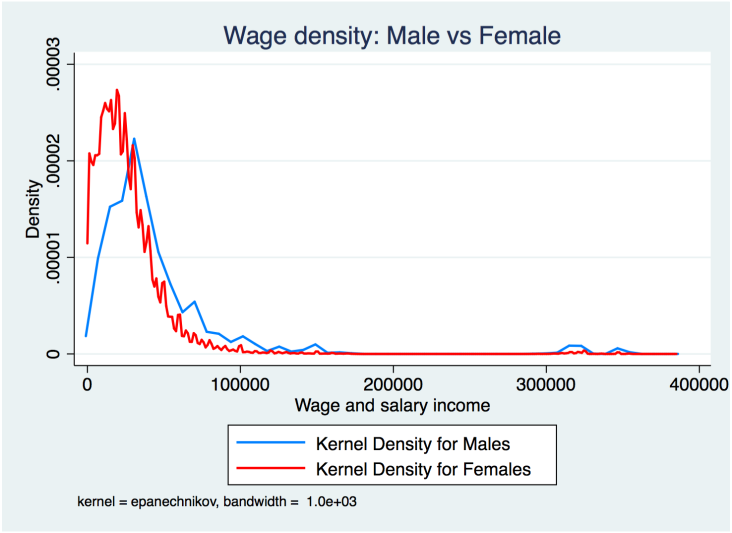
Small demo on the features of Stata

- What is the distribution of Wages (for those who have one)?



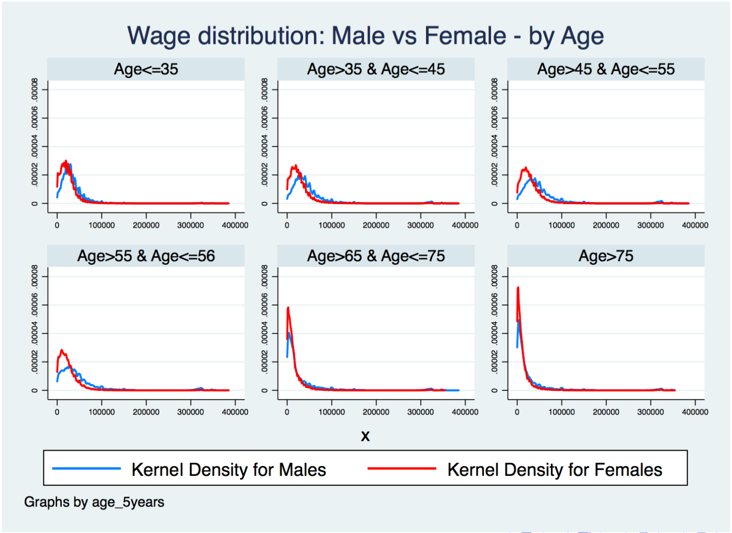
Small demo on the features of Stata

- Is the distribution different for men and women?



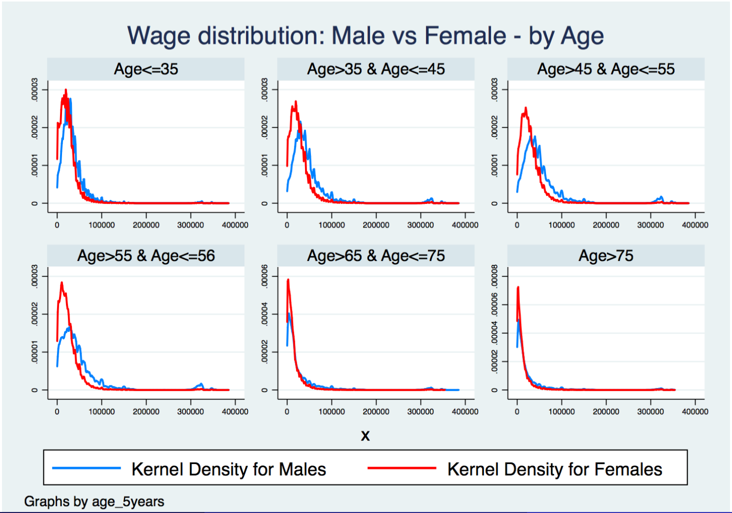
Small demo on the features of Stata

- Is the distribution different for men and women, for all age profiles?



Small demo on the features of Stata

- Is the distribution different for men and women, for all age profiles (CHANGING THE Y-AXIS)?

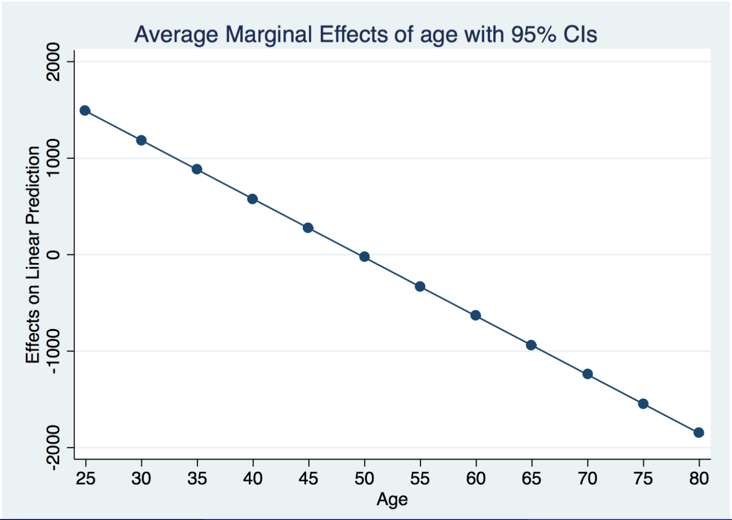


Small demo on the features of Stata

- What is the marginal effect of age, on expected wage, for a person, no matter if it is man or woman ?
- $Wage_i = \alpha + \beta_1 age_i + \beta_2 age_i^2 + \beta_3 Sex_i + \varepsilon_i$
- We can estimate all these parameters, and its standard errors, using Stata
- We are interested in the marginal effect: $\frac{dY}{dx} = \beta_1 + 2\beta_2 age_i$
- The marginal effect depends on age itself.
- We can plot this (average) marginal effects for different ages

Small demo on the features of Stata

- What is the marginal effect of age, on expected wage, for a person, no matter if it is man or woman ?

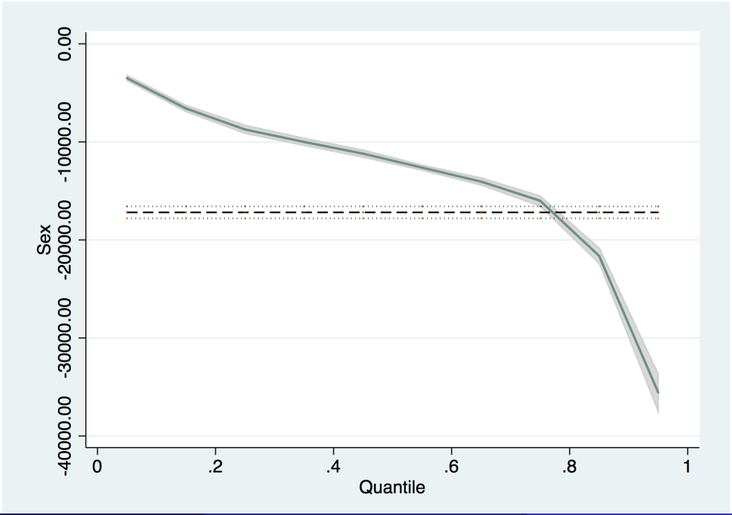


Small demo on the features of Stata

- What about the effect of being a woman?
- You might not be willing to look at the averages
- The effect of being a woman, if your wage is low, might be different of the effect of being a woman, if your wage is high
- We can use "Quantile regression" and plot these effects also.

Small demo on the features of Stata

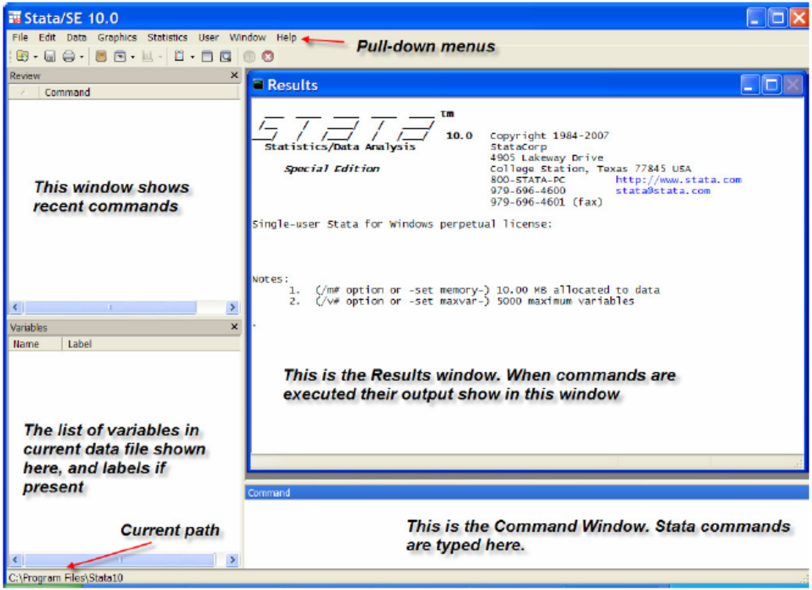
- Effect of being a woman, holding age constant, on different quantiles of the wage



Small demo on the features of Stata

- We can summarize everything we have done in a $do - file$.
- Show lecture1.do

What Stata looks like?


Stata/SE 10.0
File Edit Data Graphics Statistics User Window Help
Pull-down menus
Review
Command
This window shows recent commands
Variables
Name Label
The list of variables in current data file shown here, and labels if present
Current path
C:\Program Files\Stata10
Results
This is the Results window. When commands are executed their output show in this window
Command
This is the Command Window. Stata commands are typed here.

```
                                          tm
  ___  ____  ____  ____  ____
 /__    /   ____/   /   ____/           10.0   Copyright 1984-2007
___/   /   /___/   /   /___/                   StataCorp
  Statistics/Data Analysis                     4905 Lakeway Drive
                                               College Station, Texas 77845 USA
     Special Edition                           800-STATA-PC        http://www.stata.com
                                               979-696-4600        stata@stata.com
                                               979-696-4601 (fax)

Single-user Stata for Windows perpetual license:


Notes:
      1.  (/m# option or -set memory-) 10.00 MB allocated to data
      2.  (/v# option or -set maxvar-) 5000 maximum variables

.
```

How to make Stata work?

- You can enter your commands in three different ways:

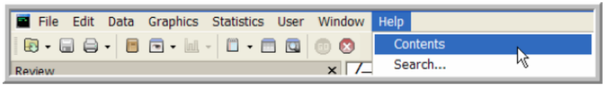
  1. Interactively: you just go throw the menu on the top of the screen
  2. Manually: you type the first command in the command window and execute it, then the next, and so on
  3. Do-file: type up a list of commands in a "*do-file*", essentially a computer programme, and execute it

Getting help

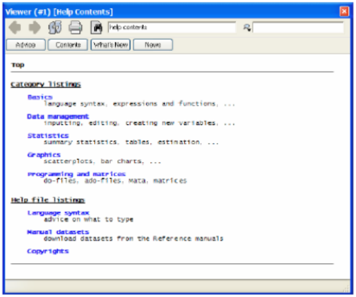
- Stata is command driven: more than 500 different commands
- I will provide the do files at the end of every class
    - It might not be enough
- **You need to practice!!!**
- Where to find help
    - *help* function - I will guide you on this
    - Google it:
        - FAQ: http://stata.com/support/faqs/
        - STATALIST: http://stata.com/statalist/
    - Ask your colleagues
- Like any other programming language / software the best way to learn is by using it

Using the help function of Stata

- One of the reasons we use Stata instead of other softwares is the richness of its help function



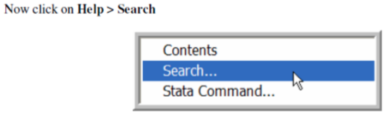
Select **Contents.**

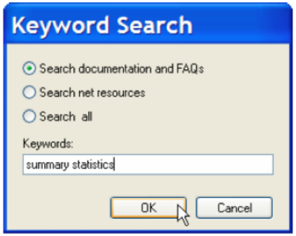


```
Top

Category listings
    Basics
        language syntax, expressions and functions, ...
    Data management
        inputting, editing, creating new variables, ...
    Statistics
        summary statistics, tables, estimation, ...
    Graphics
        scatterplots, bar charts, ...
    Programming and matrices
        do-files, ado-files, Mata, matrices

Help file listings
    Language syntax
        advice on what to type
    Manual datasets
        download datasets from the Reference manuals
    Copyrights
```

Using the help function of Stata

- One can also search for something more specific

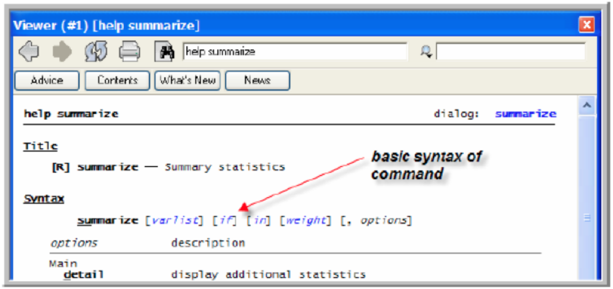

Now click on **Help > Search**



Search documentation and FAQs [✓]
Search net resources
Search all

Keywords:
summary statistics

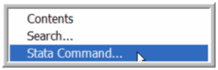
Help box



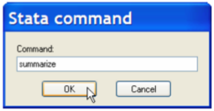
```
help summarize                                    dialog:  summarize

Title

     [R] summarize — Summary statistics

Syntax

        summarize [varlist] [if] [in] [weight] [, options]

    options              description
    Main
      detail             display additional statistics
```

Using the Help

- If you know the name of the command you want to use



In the resulting dialog box type in the name of the command and click **OK**.



- Syntax: help *command*

- Example: help *summarize*

You know the command, but not remember the details

- db command
- Example: db summarize

Stata Syntax

- Stata commands are structured like this

command [varlist] [if] [in] [weight] [, options]

- The terms in brackets [ ] are various optional command components that could be used.
- **[varlist]** is the list of variables for which the command is used
- **[if]** is a condition imposed on the command
- **[in]** specifies range of observations
- **[weight]** when some sample observations are to be weighted differently than others
- **[, options]** command options go here

How to import some commands to Stata?

- Sometimes what we want to do is not built in Stata
- But someone else have written this command
- **Example:** Test for normality Chen-Shapiro
- **help chens** or **findit chens**
- We can also install using **ssc install command**
- **Example:** count non-missing **ssc install nmissing**

The do-file

- In practice most of the researchers write all their codes in a Do-file
- It is quicker, records all your commands, easier to replicate, etc.
- **TRY TO BE AS ORGANIZED AS POSSIBLE!**
- Comment all your do-file:
  - It helps other people to understand what you did (including you 3 months later)
  - Write * and // before your commands
  - If is too long, writing it between /* comment here */ to commend across different lines

The do-file

- It is also nice to write a preamble saying what the code is suppose to do
- Also, try to organize your do-file in sections: generate variables, sample selection, regressions...
- One useful section is the housekeeping: it cleans everything before the actual data analysis
  - **cd "C:/blabla"**: set the working directory
  - **clear**: clear all your data set
  - **set more off** : prevents Stata to stop when there is a long output in the screen
  - **set memory 2000M** : allocates more memory if the data set is too large (if you use a new stata version this is unlikely to make a difference)

Keeping track of all your results

- We already know that the do file keeps track of all commands we are using
- But how to keep track of all the results we are getting?
- Log files!
- Use **log using logname.log** to start recording your session
- **log close** to stop

Exercise 1: Running our first do-file

1. Create a new folder and include the data set **microdata_lecture1.dta**
2. Open a new do-file and start comment in the beginning your name and any other relevant information, make sure the do-file is well commented
3. Start your do-file with the command **cd** to set the directory to the folder of point 1
4. Include any other relevant "housekeeping" command
5. Record a **log** of your do-file in *text*, use the command **help** to learn how to do it
6. Open the data set using the command **use** including all the relevant options (again use **help** if needed)
7. Write the command **describe** and close your log
8. Save your do-file in your directory and write **do dofilename.do** in the command window