Recitation 1: Introduction to Stata

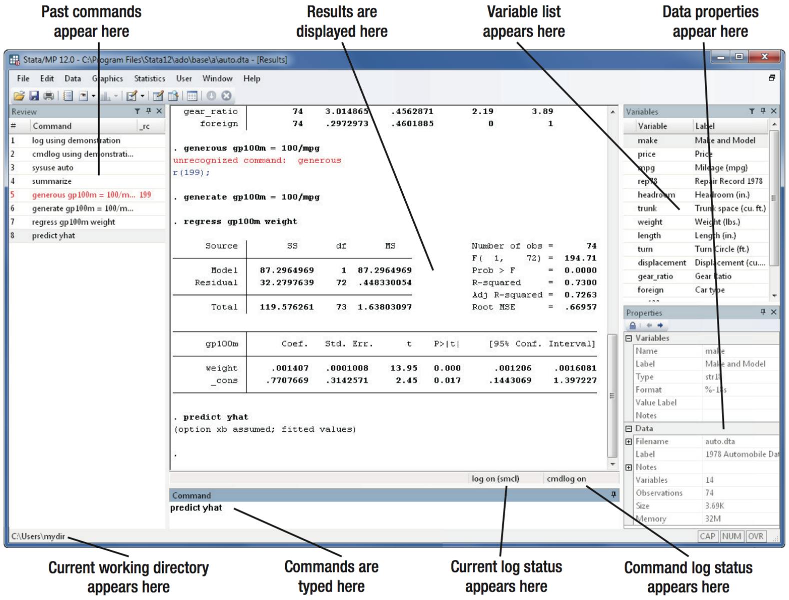
McCourt School of Public Policy, Georgetown University

Key Ideas:

- Variables vs. Observations
- String vs. Numeric Variables
- Missing Values
- File types: `.dta`, `.do`, `.log`
- Working in do-files / comments

Variables vs. Observations

- Open Stata by clicking icon
- Open example data set, auto.dta, using drop-down menus:
  - File > example data sets > auto.dta


Past commands appear here
Results are displayed here
Variable list appears here
Data properties appear here
Stata/MP 12.0 - C:\Program Files\Stata12\ado\base\a\auto.dta - [Results]
File
Edit
Data
Graphics
Statistics
User
Window
Help
Review
Variables
Properties
Command
log on (smcl)
cmdlog on
C:\Users\mydir
CAP
NUM
OVR
Current working directory appears here
Commands are typed here
Current log status appears here
Command log status appears here

#	Command	_rc
1	log using demonstration	
2	cmdlog using demonstrati...	
3	sysuse auto	
4	summarize	
5	generous gp100m = 100/m...	199
6	generate gp100m = 100/m...	
7	regress gp100m weight	
8	predict yhat	

```
  gear_ratio |        74    3.014865    .4562871       2.19       3.89
     foreign |        74    .2972973    .4601885          0          1

. generous gp100m = 100/mpg
unrecognized command:  generous
r(199);

. generate gp100m = 100/mpg

. regress gp100m weight

      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  1,    72) =  194.71
       Model |  87.2964969     1  87.2964969           Prob > F      =  0.0000
    Residual |  32.2797639    72  .448330054           R-squared     =  0.7300
-------------+------------------------------           Adj R-squared =  0.7263
       Total |  119.576261    73  1.63803097           Root MSE      =  .66957

------------------------------------------------------------------------------
      gp100m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |    .001407   .0001008    13.95   0.000      .001206    .0016081
       _cons |   .7707669   .3142571     2.45   0.017     .1443069    1.397227
------------------------------------------------------------------------------

. predict yhat
(option xb assumed; fitted values)

.
```

Command: predict yhat

Variable	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 1978
headroom	Headroom (in.)
trunk	Trunk space (cu. ft.)
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu....
gear_ratio	Gear Ratio
foreign	Car type

⊟ Variables	
Name	make
Label	Make and Model
Type	str18
Format	%-18s
Value Label	
Notes	
⊟ Data	
⊞ Filename	auto.dta
Label	1978 Automobile Dat
⊞ Notes	
Variables	14
Observations	74
Size	3.69K
Memory	32M

- Main Stata Window(s):
  - Results window: Shows commands entered after a `.` and the resulting output.

  - Variables window: Shows variable names and descriptions/labels. Click a variable name to enter it into the command line. Will be empty if no data is loaded.
  - Review window: Shows previous commands. Click a previous command to enter it into the command line.
  - Command line window: type in commands here and hit 'enter' to execute. More on this below.

- Some windows may not appear or are floating on Mac computers in earlier versions of Stata. To show/hide these, use the drop-down menu: Window> Command/Results/Review/Variables

- Open browse window using icon (looks like a mini-spreadsheet grid with a little magnifying glass, grid with pencil icon will open the browser in edit mode to change, add, or delete data) or using drop-down menus: * - file > data > data editor > data editor (browse)

- Variables are columns across the top and observations are the rows down the side.

- Browse window:

  - variable window
  - browse a subset of variables using check boxes
  - browse a subset of observations using filter icon

Basic Commands to Explore the Data Set

- `Describe`: from drop-down menu: data > describe data > describe data in memory

  - For specified variables (all variables if no specific variables specified in command), lists name, description, variable type (storage type–byte, int, and float are numeric; anything starting with `str` is a string variable), display format (how many digits before decimals are rounded), and label.
  - If run for all variables, also lists the number of observations, number of variables, and information about the data file.

- `Summarize`: from drop-down menu: data > describe data > summary statistics -For specified variables dislplays variable name, # of observations, mean, standard deviation, minimum value, and maximum value.

- `list`: from drop-down menu: data > describe data > list data

- Displays the value of specified variables (or all if none specifically selected) for each observation meeting conditions specified in `if` box.

- For `describe, summarize,` and `list` you can specify one or more variables to run the command on by listing their names in the variables line separated by a space ' ', rather than running the command for all variables by leaving the line blank. This is espeically useful for extremely large data sets where seeing all variables would be cumbersome

- `Display`: from drop-down menu: data > other utilities > hand calculator

- Performs basic athrimatic calcluations. Can use standard operators: `+`, `-`, `*` (multiplication), `/` (division), `^` (exponent/power). Expressions follow order of operations and can use parentheses `()` to emphasize specific order of operations.

- `Count`: from drop-down menu: data > data utilities > count observations satifying condition

- Counts the number of observations satisfying specified conditions in `if` box. (if no conditions specified, the command counts the total number of observations in your dataset).

Drop-down menus vs. command line

- So far, all commands have been issued through drop-down menus, but we also have the option of typing commands directly into the command line.

- Every command issued through the drop-down menu can also be issued as a typed command.

- Commands show up in the results window, in bold after a `.` and previous commands are recorded in the review window.

- To issue commands type its name or a shortened version/abbreviation into the command line window and hit enter. `-describe` or `des` `-summarize` or `sum` `-list` or `li` `-count` or `cou` `-display` or `di`

  - To issue the command for one or a subset of variables, list the name of each variable after the command separated by a space

- For example type `sum mpg price` and hit enter to summarize only the mpg and price variables.

- Note that:

  - commands and variable names are case sensitive
  - tab-completion of variable names
  - variables can be entered by clicking in variable window
  - Page-up cycles through old commands

Running commands on a subset of observations

- Commands with `if, by,` and/or `in` (the summarize, count, and list commands so far) can also be run on a SUBSET of OBSERVATIONS that meet the conditions in the `if` section or after `if` in a statement in the command line.
- Specify a subset of observations whose values meet conditions by using the relational operators below to specify the conditions that the value of variable(s) for each observation must meet to be included in calculation.

`>`	`greater than`
`<`	`less than`
`>=`	`> or equal`
`<=`	`< or equal`
`==`	`equal`
`!=`	`not equal`
`~=`	`not equal`

A double equal sign `==` is used for equality testing. A single equal sign will be used to assign a variable a value. We'll see more on this in future classes.

Example

- For example to summarize the price and weight variables for only observations with mpg greater than 20: Type `summarize price weight if mpg > 20` and hit enter. Notice that there are fewer observations included.

-To specify multiple conditions for a command, use the logical operators `&` for AND (must meet both condition) and `|` (shift+) for OR (must meet either one of the conditions).

- For example to summarize the price and weight variables for only observations with mpg greater than 20 AND that are foreign:

Type `summarize price weight if mpg > 20 & foreign == 1` and hit enter.

- Example 2, to summarize the price and weight variables for only observations that EITHER have mpg greater than 20 OR that are foreign:

Type `summarize price weight if mpg > 20 | foreign == 1` and hit enter.

Variables Types: String vs. Numeric

Strings are stored as text, and cannot be added, subtracted, etc.

Example variable: make

- browse dataset
- Contrast string variable make with labeled numeric variable, foreign
- Strings are red, labeled numeric are blue
- For strings, the actual value is text.
- For labeled numeric, the actual value is a number

  describe

- Look at storage type column
- Anything that begins with str is a string
- Anything else is a number, including byte, int, float, long, double.

  summarize

- String variables are not included in summary statistics, because it is not

- possible to perform calculations on them.

- We will cover string variables in more depth.

- For now, just know what they are and how to identify them.

Missing Values

Sometimes not every observation has a value for every variable. In survey data, this could arise if people skip a question.

We will talk more about missing data in semester 2. For now, you should just know that missing data exists and what it looks like in Stata.

In Excel, missing data is represented simply as a blank cell. In Stata, missing data for numeric variables is represented by a period or dot . In Stata, missing data for string variables is represented by an empty string ‘

Missing data is not included in calculations for summary statistics

- Example: `rep78`

- There are 74 observations, or rows, in this data set -Enter `describe` or `count`

- But rep78 only has 69 observations in summary statistics -Enter `summarize rep78`

- The difference is because of observations with missing data are not included in the calculation. `-display 74-69`

- 5 observations with missing data

- `browse`

- Observations with a variable missing data have a `.` representing this when viewing them in the browser.

- We can see this with the count command `count if rep78==.`

In-Class Activity 1

Open the auto.dta data set and issue the `describe` and `summarize` commands from above.

Try to answer the questions below.

Questions using the auto.dta example dataset: 1. describe all variables 2. summarize price mpg weight and length
3. What is the difference between the highest and lowest mpg? 4. Summarize price, mpg, weight, and length for cars costing less than $4,000 5. List the make and price of the most expensive car

- Result should be something like this:

sysuse auto.dta

- 1. describe all variables > describe

- 2. summarize price mpg weight and length > summarize price mpg weight length

- 3. What is the difference between the highest and lowest mpg? > display 41-12

- Difference between highest and lowest mpg is 29.

- 4. Summarize price, mpg, weight, and length for cars costing less than $4,000 > summarize price mpg weight length if price < 4000

- 5. List the make and price of the most expensive car > list make price if price == 15906

Stata File Types

- `.DTA` You've already encountered the Stata dataset, .dta file type.
- `.DO` Next is the do-file: holds a list of commands that are executed as if you typed them into the command line one-by-one, .do file type.

Open a do-file using the drop down menu (`Window > Do File Editor > New Do-File Editor`):

Input commands from above

Various methods to execute code from a do-file: - entire file, just click execute (do) or highlighted portions, highlight line(s) and click execute (do) note: partially highlighted lines are executed as entire lines - Execute entire file with intentional error e.g. `stop` typed on a line for debuggind purposes - do button or ctrl-d

Always work in a do-file because: - Professors will require it - Helps you stay organized - Helps you recover from mistakes - Is a way of saving your work - Saves you typing

Comments can help you stay organized - Comments are text in your do-file that are not executed by Stata. - Comment lines begin with `*` - Commen sections (multiple lines) begin with `/*` and end with `*/`

- `.LOG` Log files copy everything that is printed in the results window. There are two types of log files: `.log` is a text file that can be opened with many programs, Word, Notepad, etc. `.smcl` is a special Stata format that can only be opened with Stata. `.smcl` formats output just as it appears in results window, while `.log` is plain text.

Logs begin recording when they are opened/started, and stop recording when they are closed. Start recording a log file using the drop-down menu (File > Log > Begin…). Stop recording a log file using the drop-down menu (File > Log > Close).

After-Class Exercises using the `lifeexp.dta` example dataset:

Create a do-file to open the lifeexp.dta example data set (File>Example Data Sets> Example datasets installed with Stata Then Click 'use' to the right of lifeexp.dta Be sure to clear any data you have in memory first by typing `clear` in the command line and hitting enter)

Try to answer the questions below. Include at least one comment documenting the question you are answering and your answer to the question, if necessary

Practice various methods for running the do-file.

When the do-file is complete, start a log using the drop-down menu. Run the do-file once to record output. Close the log. For problem sets, students should submit do-file and log.

1. How many variables are there? How many variables are string vs. numeric variables?
2. How many observations are there?
3. Are any variables missing data for observations? If so, what are they and how many observations have missing data for each?
4. What is the difference between the country with the greatest and least life expectancy?

We can go over the answers at the beginning of our session next week if there is interest

Getting Started Guides

- For anyone who wants to get ahead or as a resource in the future
- Dropdown menu Help > PDF Documentation Show Getting Started Guide for Windows and Mac