Getting Started with Stata

Session Two: Data Management

Yiming Cao

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

September 17, 2019

Last Week

- Basic features (view, manage, visualize and analyze data)
- The basic command syntax
- A sample session
- Getting help

Today: Data Management

- More on the command syntax
  - Illustrated with the **list** command
- Basic data management
  - Open and save data
  - Change variable names and labels
  - Create new variables
  - Delete variables/observations
- Advanced data management
  - A list of the most useful commands that you should explore

Revisiting the first steps

- Set your working directory
  - Choose your own directory (where you would like to open and save files)
  - use "" if there are spaces in the path:

    cd "D:/My Workspace"
- Set log files
  - Add the *text* option if you would like your log file to be in .txt format:

    log using *filename*,text
- Open Stata built-in data set for today's examples:

  sysuse auto,clear

Command Syntax

- The general syntax:

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

- An illustrative example with the command **list**

  list [*varlist*] [*if*] [*in*] [, *options*]

  - Anything inside square brackets is optional
  - Optional pieces do not preclude one another unless explicitly stated
  - If a part of a word is underlined, the underlined part is the minimum abbreviation

Variable list

- List the variables explicitly:
  - One single variable: **myvar**
  - A few variables: **myvar thisvar thatvar**
- List variables with wildcards:
  - * matches one or more characters and returns all variables that match the pattern: **myvar***; **var***; **my*var**;
  - ~ matches one or more characters but only one variable is allowed to match: **my~var**;
  - ? matches one character and returns all variables that match the pattern: **my?var**;
  - - matches all variables in the data set between the two specified variables: **this-that**;
- Examples

The if qualifier

- **if** uses a logical expression to determine which observations to use. It determines whether the following operators are true or false:

<	less than
<=	less than or equal
==	equal
>	greater than
>=	greater than or equal
!=	not equal
&	and
|	or
!	not (logical negation)
()	parentheses are for grouping to specify order of evaluation

- Quotes "" are required for string variables
- Examples

The in qualifier

- **in** uses a *numlist* to give a range of observations that should be listed.
  - Positive numbers count from the beginning of the dataset
  - Negative numbers count from the end of the dataset
- Examples

Options

- **sepby()** separates observations by variable
- **abbreviate()** specifies the minimum number of characters to abbreviate a variable name in the output
- **divider** draws a vertical line between the variables in the list
- Examples

Open and Save

- Stata formatted data set: .dta
  - To save Stata data set to disk: **save filename,replace**
  - To open Stata data set from disk: **use filename,clear**
- Examples

How about non-Stata data sets?

- Copy and paste to Data Editor (type **edit** to launch it)
  - You can also edit the data in Data Editor, though I do not recommend doing so
- **Import** (the more recommended method)
  - From .txt or .csv: **import delimited [using] filename,[options]**
  - From Excel (.xls or .xlsx): **import excel [using] filename, [firstrow] [other_options]** - The **firstrow** option specifies the first row in the Excel file as the variable name in Stata
  - Remember to save the data in Stata format after importing it

Preserve and restore

- **preserve** preserves the data, guaranteeing that data will be restored after program termination
- **restore** forces a restore of the data now

Change variable names and labels

- Rename a variable: **rename** *oldname newname*
  - Rules for variable names:
    - Stata is case sensitive.
      `Make`, `make`, and `MAKE` are all different names to Stata. If you had named your variables `Make`, `Price`, `MPG`, etc., then you would have to type them correctly capitalized in the future. Using all lowercase letters is easier.
    - A variable name must be 1–32 characters long.
    - The characters can be letters (`A`–`Z`, `a`–`z`), digits (`0`–`9`), underscores (`_`), or Unicode characters that are not symbols.
    - Spaces or other characters are not allowed.
    - The first character of a variable name must be a letter, an underscore, or a Unicode character. Although you can use an underscore to begin a variable name, it is highly discouraged. Such names are used for temporary variable names in Stata. You would like your data to be permanent, so using a temporary name could lead to great frustration.
- Label a variable: **label variable** *varname "the label"*
- Label the values of a variable:
  - Step 1 define the value label: **label define** *lbname* 0 "FirstLabel" 1 "SecondLabel" [...]
  - Step 2 assign the value label to the variable: **label values** *varname lbname*
- Examples

Creating New Variables

- **generate** creates new variables
- **replace** replaces the values of an existing variable
- Basic command: **generate newvar=esp**

Arithmetic		Logical		Relational (numeric and string)	
+	addition	!	not	>	greater than
-	subtraction	|	or	<	less than
*	multiplication	&	and	>=	> or equal
/	division			<=	< or equal
^	power			==	equal
				!=	not equal
+	string concatenation				

- String variables and string functions (**help string_functions** for details)
- **egen** is an extension of **generate** with more functions and expressions
- Examples

Deleting variables

- **clear** removes the current dataset from the memory
- **drop** removes variables or observations from the dataset in memory
  - To remove variables, use **drop varlist**
  - To remove observations, use **drop** with an *if* or an *in* qualifier or both
- **keep** tells Stata to drop all variables except those specified explicitly or through the use of an an *if* or an *in* qualifier
- Examples

Try Reproducing the Examples

```
cd D:\Stata // this should be replaced by the path of your own working directory
log using week2,text replace
sysuse auto,clear
// List with varlist:
list make mpg price
l make mpg price
list m*
li price-weight
list ma?e
l gear_r~o
// List with if qualifier:
list if mpg>22
list if (mpg>22) & !missing(mpg)
list make mpg price gear if (mpg>22) | (price>8000 & gear<3.5)
list if make=="Datsun 510"
list if foreign==0
// List with in qualifier:
list in 1
list in -1
list in 2/4
list in -3/-2
// List with some options:
list ma p g f, sepby(foreign)
list make weight gear, abbreviate(3)
list,divider
// Save and open data to/from disk:
save autodata,replace
use autodata,clear
// Change variable names and labels:
preserve
rename make Make
gen domestic=1-foreign
label variable domestic "Domestic or not"
label define lbdomestic 0 "Foreign" 1 "Domestic"
label values domestic lbdomestic
restore
// Create New Variables:
preserve
generate lphk = 3.7854 * (100/1.6093) / mpg
gen lnprice=ln(price)
gen huge=(weight>=300) if !missing(weight)
replace weight=weight/1000
gen predprice=1.05*price if foreign==0
replace predprice=1.10*price if foreign==1
gen predprice2=(1.05+0.05*foreign)*price
gen where="D" if foreign==0
replace where="F" if foreign==1
gen model = substr(make,strpos(make, " ")+1,.)
gen modelwhere=model+ " " + where
restore
// Delete variables and observations
preserve
drop in 1/50
drop if mpg>21
drop gear_ratio
drop m*
restore
preserve
keep in 40/70
keep if mpg<=21
keep m*
clear
restore
// Close log file
log close
```

More to Explore: Advanced Data Management

- I provide a list of the most useful commands for data management
  - The full list can be found in *[D]Data Management* of the PDF documentation
  - Or, simply type: **help data management**
- Read the help document to learn what they are used for
- Refer to the help document for the syntax and detailed options when you actually need to work with them

For inputting or Importing data:

- Load Stata dataset: **use**
- Use Stata built-in dataset: **sysuse**
- Use dataset from Stata website: **webuse**
- Enter data from keyboard: **input**
- Importing data into Stata from:
  - delimited text data: **import delimited**
  - excel: **import excel**

For exporting data:

- Save Stata dataset: **save**
- Export data from Stata to:
  - delimited text data: **export delimited**
  - excel: **export excel**

To reorganize or combine data:

- Append datasets: **append**
- Merge datasets: **merge**
- Convert data from wide to long or vice versa: **reshape**
- Make dataset of summary statistics: **collapse**
- Rectangularize dataset: **fillin**
- Make duplicated observation: **expand**
- Reorder variables in dataset: **order**
- Sort data (ascending only): **sort**
- Sort data (ascending or descending): **gsort**

Other useful commands

- Encode string into numeric and vice versa: **encode**
- Recode categorical variables: **recode**
- Convert string variables to numeric variables and vice versa: **destring**
- Report, tag, or drop duplicate observations: **duplicates**

Next Week

- Data Visualization