Data Preparation using Stata

Econ 637: Econometrics 1

Markus Bjoerkheim

Department of Economics - George Mason University
mbjoerkh@gmu.edu

January 24, 2019

Outline

Why Stata?

- Still the most commonly used statistical software by economists
- Easy to learn (compared to alternatives), but for some purposes such as Network Analysis, Spatial Econometrics, Machine Learning, etc. you'll most likely want to learn R or Python
- My goal is to give an introduction to the most commonly used commands, functions, and data transformation procedures in Stata, but ultimately, most learning occurs through trial and error.
  `help 'command'` and google are your friends.

Three ways to Use Stata

- Point & click in drop-down menus (there's a time and a place for this)
- Simple code in command-line window
- `.do files` (which is where you spend most of your time)

Basic Stata Syntax

command varlist qualifiers, options

command	1, 2, or 3 words specifying the task
varlist	0, 1, 2 or more variables
= *exp*	A math or logical statement to set a value
if *exp*	A math or logical statement to limit cases
, *options*	A word or selector to alter the command

Exploring Data

- Let's import an extract from the Current Population Survey that I've have made available to you here.
- As we move through the next slides, try to execute the commands we discuss and please let me know if anyone is unable to download and open the dataset.

Data Exploration Commands

`browse` look at your data while working with it (second monitor is handy)
`edit` like `browse` but let's you edit data by hand as you might in excel
`describe` describes data in memory (storage type, display format, labeling)
`hist sex` plots basic histogram of age
`hist incwage` plots histogram of income - see anything odd?
`sum age sex` summary stats for variables `age sex`
`tab age` the frequency distribution of `age`
`tab age sex` two-way table of the frequency distribution of `age` and `sex`
`summarize` summary statistics
`count if sex==1 & age>40` displays the number of observations that meets criteria

`generate old = 0` generates a variable named `old` that takes the value 0
`replace old=1 if age>65` set value of `old` to 1 if `age` is greater than 65
`gen old=(age>=65)` evaluate expression (if true, `old=1`, if false, `old=0`)
`gen age_sq=age*age` create variable by multiplication(all operators work)
`drop x` drops x
`drop x*` drop all variables whose name begins with x
`drop *_sq` drop all variables with names that ends with _sq
`egen median_inc=median(inc), by(state year)` a variable whose value is the median `inc` by `state` and `year`. Check out `egen` for more

Some functions

`real(string)` converts a string(text) to numeric
`string(real)` converts numeric(say, 5) to string("5"). Also check out `encode, decode, tostring, destring`
`substr(stringvar,n1,n2)` the substring of `stringvar`, starting at place n1, for a length of n2.
`ln(var)` the log of `var`

Working in `.do` Files

Two "rules" from "Code and Data for Social Scientists" by Matt Gentzkow & Jesse Shapiro
1) Automate everything that can be automated
2) Write a single script that executes all code from beginning to end

- To do this you need to use `.do`files where you write(or paste) code that Stata will execute, which also allows you to write more complex code than through the `command line`
- Commands that prepare or analyze data should **end up** in `.do` files, otherwise you start from scratch in order to change your analysis, correct a mistake, etc.
- Small upfront investment, worth it the first time you need to redo something. However, I often like to work interactively and transfer the code to `.do` files

Log your work

- Good habit to log your work using `.log` files

```
capture log close

log using logfile.log, replace

cd S:\name

**********************

* stuff goes here

**********************

log close
```

Ctrl-**L:** Select **L**ine

Ctrl-**D:** Execute (**D**o)

Importing Data

- Stata has several different commands for different kinds of data sources, these are pretty self-explanatory.
- `import excel` for `.xls` or `.xlsx` spreadsheets, `import delimited` for text-files separated by a delimiter, etc. Occasionally you'll find a file that Stata can't import, find a way to convert the file to a different format.
- Data can (almost) always be downloaded to your computer and imported to Stata. This works fine, but involves some fixed costs.. Many commonly used sources have API's.

Importing Data: FRED

- `import fred` gives you access to 528,000 time series from 87 sources.
- To set up Stata with FRED: go to www.fred.stlouisfed.org, register, get API key, and open Stata.
- Typing `help import fred` will open the viewer and show you how to set the `fredkey`
- `set fredkey key, permanently`
- Import data from FRED by point & click; highlight commands in the `Command Review Window`, right-click and choose `Send selected to do-file`
- Similar source is the NBER: www.nber.org/data/

Importing Data: FRED

- `import fred` gives you access to 528,000 time series from 87 sources.
- To set up Stata with FRED: go to www.fred.stlouisfed.org, register, get API key, and open Stata.
- Typing `help import fred` will open the viewer and show you how to set the `fredkey`
- `set fredkey key, permanently`
- Import data from FRED by point & click; highlight commands in the `Command Review Window`, right-click and choose `Send selected to do-file`
- Similar source is the NBER: www.nber.org/data/

Importing Data: FRED

- `import fred` gives you access to 528,000 time series from 87 sources.
- To set up Stata with FRED: go to www.fred.stlouisfed.org, register, get API key, and open Stata.
- Typing `help import fred` will open the viewer and show you how to set the `fredkey`
- `set fredkey key, permanently`
- Import data from FRED by point & click; highlight commands in the `Command Review Window`, right-click and choose `Send selected to do-file`
- Similar source is the NBER: www.nber.org/data/

Importing Data: Comparative Manifesto Project

- 90% of current data was generated in the past 2 years. This is what's fun with empirical work, we can ask new questions, as well as old questions in new ways.
- Content analysis of 1000+ political parties' election manifestos, in 55+ countries, going back to 1920's.
- `net from https://manifesto-project.wzb.eu/manifestata`
- `net install manifestata , replace`
- Load with: `mp_maindataset, api(key) clear`

Appending Data

- You use `append` to add more `rows` to your dataset, for instance when another year of data has been released
- `append` is the simplest of the data combination commands
- Typical syntax: `append using filename`
- The rows in `filename` will now be appended to the end of the dataset in memory
- Note that filename needs to be in `.dta` format (or in a file that you can import and save to `.dta`
  Worst case Stat/Transfer is $39/year.

Merging Data

- `merge` add `columns` (variables) to your dataset. In order to merge you need to identify a logical observation in both datasets.
- Syntax depends on the kind of merge you have; simplest is a `1:1` merge where the rows in your two datasets represent the same level (i.e. state-year). Can also be a `m:1` or `1:m` merge. I'll show you the first two real quick.
- Warning: If you're about to do an `m:m` merge, stop. You're almost certainly doing it wrong.

Data Structures

Data is structured as a `observations x variables` matrix, where rows = observations, columns = variables

	id	year	sex	inc	ue
1	1	80	0	5000	0
2	1	81	0	5500	1
3	1	82	0	6000	0
4	2	80	1	2000	1
5	2	81	1	2200	0
6	2	82	1	3300	0
7	3	80	0	3000	0
8	3	81	0	2000	0
9	3	82	0	1000	1

Same underlying data can also be structured in “wide” format..

	id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1	1	0	5000	5500	6000	0	1	0
2	2	1	2000	2200	3300	1	0	0
3	3	0	3000	2000	1000	0	0	1

Kevin DeLuca 🤝
@cantstopkevin

There are literally only two qualities you need to be an economist: 1) the intellectual curiousity, math ability, programming skills, and work ethic to take on extremely hard problems in a quantitatively rigorous way and 2) the willingness to type "help reshape" on a daily basis

`reshape` let's you reshape data from wide to long, and vice versa. Syntax:

```
reshape long[wide] stub , i(i) j(j)
```

so to reshape the below (wide) dataset to long:

```
reshape long inc ue , i(id) j(year)
```

where "i" identifies a logical observation, the "stub(s)" becomes the variable names of the new "combined" `inc` and `ue` variables, and what comes after the "stub(s)" becomes the values of the "j" variable you create(`year` in this case)

	id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1	1	0	5000	5500	6000	0	1	0
2	2	1	2000	2200	3300	1	0	0
3	3	0	3000	2000	1000	0	0	1

`reshape` let's you reshape data from wide to long, and vice versa. Syntax:

```
reshape long[wide] stub , i(i) j(j)
```

so to reshape the below (wide) dataset to long:

```
reshape long inc ue , i(id) j(year)
```

where "i" identifies a logical observation, the "stub(s)" becomes the variable names of the new "combined" `inc` and `ue` variables, and what comes after the "stub(s)" becomes the values of the "j" variable you create(`year` in this case)

	id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1	1	0	5000	5500	6000	0	1	0
2	2	1	2000	2200	3300	1	0	0
3	3	0	3000	2000	1000	0	0	1

Paid Family Leave Mandates and Female Earnings

- Research question: what is the impact of Family Leave Mandates on Female Earnings?
- Simplified research design: compare female earnings in states that impose mandates, to those that don't, using data from Current Population Survey(CPS)
- Problem: unit of observation in CPS is person-year, but analysis requires data on state-year level
- Solution: `collapse` converts your dataset to a dataset of means, sums, medians, etc. on the level you specify

Paid Family Leave Mandates and Female Earnings

- Research question: what is the impact of Family Leave Mandates on Female Earnings?
- Simplified research design: compare female earnings in states that impose mandates, to those that don't, using data from Current Population Survey(CPS)
- Problem: unit of observation in CPS is person-year, but analysis requires data on state-year level
- Solution: `collapse` converts your dataset to a dataset of means, sums, medians, etc. on the level you specify

Paid Family Leave Mandates and Female Earnings

- Research question: what is the impact of Family Leave Mandates on Female Earnings?
- Simplified research design: compare female earnings in states that impose mandates, to those that don't, using data from Current Population Survey(CPS)
- Problem: unit of observation in CPS is person-year, but analysis requires data on state-year level
- Solution: `collapse` converts your dataset to a dataset of means, sums, medians, etc. on the level you specify

Programming: foreach - loop over items

- This is basically where it gets fun. Learning a few programming commands can save you hours..
- Say you want to repeatedly execute the same commands to a lot of variables (take logs, square, etc.).
  foreach repeatedly sets local macro var to each element of the list and executes the command(s)

```
foreach var in y x x2 x3 x4 x5 x6 x7 x8 x9 {
    gen `var'_sq = `var' * `var'
    gen `var'_ln = ln(`var')
}
```

Programming: foreach - loop over items

- This is basically where it gets fun. Learning a few programming commands can save you hours..
- Say you want to repeatedly execute the same commands to a lot of variables (take logs, square, etc.).
  foreach repeatedly sets local macro var to each element of the list and executes the command(s)

```
foreach var in y x x2 x3 x4 x5 x6 x7 x8 x9 {
      gen `var'_sq = `var' * `var'
      gen `var'_ln = ln(`var')
}
```

copy

- Say you need to download data from the web. `copy` is a handy command, syntax:
  `copy filename1 filename2 , options`
- Where `filename1` can be a URL.. Code below copies (downloads) the 2010 version of the Behavioral Risk Factor Surveillance System (BRFSS) survey from the CDC to a file named `brfss2010.xpt.zip`

```
copy www.cdc.gov/brfss/annual_data_2010.htm brfss2010.xpt.zip
```

Programming: forval - loop over consecutive values

- forval is similar to foreach but instead of looping over variables, it loops over values. This can be very convenient when combined with copy..
- If you were to download and unzip many years of data, you could write something like this.

```
forval i = 1984/1999 {
  cap copy www.cdc.gov/brfss/annual_data_`i'.htm     `i'.xpt.zip
  cap unzipfile `i'.xpt.zip , replace
}
```

Note: Use the code in the example .do-file (I trimmed this URL for illustrative purposes), and where I show how to import, append, compress, and save all years to a single .dta-file.

Macros

- Instead of repeatedly writing `reg y age gender race year education` etc. Write a local or global macro that you can call `global x age gender race year education` defines the global macro `x` as age gender race year education
- Next time you run a regression, you write `reg y $x` where the $ allows you to access the macro, which means that Stata substitutes in `age gender race year education` for $x..

Additional Commands

If you find yourself repeating something over and over, there's almost always a better way.. Google for user-written packages:

- `statastates` adds U.S. state identifiers (abbreviation, FIPS code, and name) to your dataset –> if you have one of those, you now have them all
- `unique` reports the number of unique values for `varlist` - convenient for records at multiple levels i.e. person, state, year like in the CPS. `unique state` tells you how many unique values there are for state; `unique state year` tells you how many unique state-year values there are
- `duplicates` report, tag, or drop duplicate observations

Producing Publication-Style Tables

"Social Science is 60% substance and 40% style, let's get that marginal product of style down to zero straight away"

(Garett Jones)

- `estout` and `outreg2` are popular packages. Open Stata, –> `help outreg2` –> scroll to "basic game plan" and click the 5 lines of code
- Presenting the "tex" version from line 5 just looks that much better than some copy/pasted output from results window

Potentially Useful Resources

- Digital Scholarship Center in Fenwick.
  Consultations with guidance for solving your problem.
- Useful Stata Commands by Kenneth Simons
  Walks you through lot's of Stata commands, including estimating models and how to interpret them for an Econ audience.
- Code and Data for the Social Sciences: A Practitioners Guide by Matthew Gentzkow & Jesse Shapiro. How the cool kids do this.