Topics in Stata

Chinmay Lohani

August 2019

Predoc Orientation 2019
Energy Policy Institute at UChicago

Items

- Fuzzy string matching: `reclink, matchit`
- Dealing with dates: `year(), month(), format %td`
- Random sampling and bootstrapping: `bsample, gsample`
- Parallel processing: `parallel`
- Table customisation : `file write`

Fuzzy string matching

- You will encounter many occasions where you will have to match the names of certain variables between two datasets.
- It will rarely be the case that names will match perfectly, making the need for fuzzy string matching.

Fuzzy matching- `reclink`

- reclink uses a bigram based method to join two separate datasets (can use many variables to match on).

```
. reclink v1 v2 v3 using set2, gen(myscore) idmaster(id)
    idusing(recno)
```

String matching- `matchit`

The other useful command for fuzzy string matching is `matchit`.

You can choose n-gram based matching, soundex (which is phonetic) and can even weigh between various gram methods and is highly flexible.

```
. matchit v1 v2, similmethod(soundex)
```

General guideline- tweak string matching methods to generate a large number of viable matches and skim through.

Date variables

Stata has an extensive package/function called date() to deal with date objects in your datasets.

You can extract various elements of a timestamp using functions such as month(), year(); merge together using dofm() etc.

Basic idea- machine format vs human readable format, and knowing how to switch between them.

Note- All data is actually stored numerically under the hood, however best practice would entail using human readable formats wherever possible.

Random sampling and bootstrapping

- Many tasks will require you to sample parts of your data randomly, to draw from a given distribution.
- `gsample` is a function you can use to draw from a data by using its instances as an empirical distribution function (or proportional to size).

```
 .gsample 6000, wor //sample without replacement, draw
6000 observations
```

Random sampling

- You can also draw from a standard distribution to generate a new variable in your dataset. eg. `runiformint(a,b)`
- Useful when you want to randomly sample a smaller part of the data to work with.

```
.gen sampler= runiformint(1, 100)
.keep if sampler<=5
.drop sampler
```

Bootstrapping

- No easy closed form package for the standard errors- use bootstrapping.
- Draw from your distribution of errors, use that draw as an instance of given data to re-estimate parameter, estimate parameters until you have a distribution.
- For generating these errors, use `bsample`.

Parallel programming- `parallel`

- Many times, you would like to utilise the processing power available to you.
- In principle, user made package `parallel` allows you to parallelise the control flow used to execute your stata code under the hood.
- In practice, not super robust. Worth testing in certain usecases.

Writing to file

- Of the programming functionality that Stata offers, one useful bit to know is writing to a file.
- Specially useful in cases when you wish to write to a different format, or don't wish to disturb current workspace.
- In particular, you can automate making highly customisable tables by writing Latex files.

```
    .file open myfile using "$dirpath_out/
Table_cost_various_responses.tex", write replace
    .file write myfile " & & & \multicolumn{5}{c}{Rebate(
Rs)} \\" _n
    .file write myfile " Treated \% & N(treated) &
Increase (kWh) & 0.5 & 1 & 1.5 & 2 & 2.5 \\" _n
    .file write myfile " \midrule " _n
```

Example Table

Notch (kWh)	Marginal price below and above the notch (Rs)	Initial estimates		Corrected estimates	
		Excess (bills)	Excess (ratio)	Excess (bills)	Excess (ratio)
Panel A: Standard domestic households					
200	4	-4900.6	-0.03	-4740.3	-0.03
	5.95	(8660.9)	(0.06)	(13913.2)	(0.17)
		[-0.57]	[-0.50]	[-0.34]	[-0.18]
400	5.95	2424.5	0.07	2391.9	0.07
	7.3	(27221.3)	(0.81)	(32076.6)	(0.81)
		[0.09]	[0.09]	[0.07]	[0.09]
800	7.3	-1392.0	-0.25	-1354.4	-0.24
	8.1	(24840.6)	(150.62)	(69051.9)	(56.88)
		[-0.06]	[-0.00]	[-0.02]	[-0.00]
1200	8.1	-1604.0	-0.59	-1593.5	-0.59
	8.75	(18628.3)	(8.11)	(62071.7)	(17.73)
		[-0.09]	[-0.07]	[-0.03]	[-0.03]
Panel B: Jhuggi Jhopri (JJ) households					