Cleaning Data With Stata

Andy Grogan-Kaylor

18 Feb 2022 11:29:29

Background

> It sometimes seems like 80% of our work as data analysts is cleaning the data, while only 20% is the actual analysis. Here are some Stata commands that are useful in cleaning data.

First we simulate some data to work with, and to clean.

Simulate Some Data

> This section is provided for illustration only, as it may be helpful to see *how* the data was simulated, and the decisions that went into simulating the data. You may also *safely ignore* this section if you like.

Show / Hide Data Simulation Code

```
. clear all

. set obs 100 // 100 observations
Number of observations (_N) was 0, now 100.

. generate id = _n // random id

. generate age = rnormal(50,10) // random generated age

. replace age = 200 in 1 // someone is 200 years old!
(1 real change made)

. generate happy = runiformint(1,5) // randomly generated happiness

. replace happy = 999 in 10 // simulate a missing value
(1 real change made)

. generate somethingelse = rnormal(0, 1) // something else!
```

Look At Some Of The Data

```
. list in 1/10 // list first 10 observations
```

	id	age	happy	somethi~e
1.	1	200	4	1.18464
2.	2	43.75589	1	.3102925

3.	3	42.28216	5	.1889931
4.	4	58.31932	1	1.007387
5.	5	54.96909	1	1.732758
6.	6	59.94831	3	-.5162272
7.	7	51.0313	5	.2641199
8.	8	49.48943	5	1.307504
9.	9	49.3417	1	.7795271
10.	10	39.16794	999	.0064139

Clean The Data!

Look at Variables (`describe`, `summarize`, `tabulate`, `codebook`)

When we look at variables we are looking for values that don't make sense, or that are outside the plausible range.

```
. describe // describe the data
Contains data
 Observations:           100
    Variables:             4
```

Variable name	Storage type	Display format	Value label	Variable label
id	float	%9.0g		
age	float	%9.0g		
happy	float	%9.0g		
somethingelse	float	%9.0g		

```
Sorted by:
     Note: Dataset has changed since last saved.

. summarize // descriptive statistics
```

Variable	Obs	Mean	Std. dev.	Min	Max
id	100	50.5	29.01149	1	100
age	100	51.55316	17.98757	24.9591	200
happy	100	12.99	99.60817	1	999
somethinge~e	100	.208863	1.018356	-2.771254	2.648567

```
. tabulate happy // tabulation of this particular categorical variable
```

happy	Freq.	Percent	Cum.
1	24	24.00	24.00
2	14	14.00	38.00
3	19	19.00	57.00
4	19	19.00	76.00
5	23	23.00	99.00
999	1	1.00	100.00
Total	100	100.00	

```
. codebook happy // VERY detailed view of this particular categorical variable
```

happy

```
                  Type: Numeric (float)
                 Range: [1,999]                       Units: 1
        Unique values: 6                          Missing .: 0/100
           Tabulation: Freq.  Value
                         24  1
                         14  2
```

```
 19  3
 19  4
 23  5
  1  999
```

Notice that...

- There are variables in which we may not have interest.
- None of the variables are labelled informatively.
- Variables do not seem to have informative value labels.
- Someone appears to 200 years old.
- There appear to be missing values in the variable `happy` that need to be recoded.

  Remember that the command `lookfor` is often very helpful in *looking for* a particular variable. e.g. `lookfor happy`.

Only `keep` The Variables Of Interest

We may only be interested in keeping some variables to keep our analytic data set more manageable.

For this particular analysis we may wish to drop the variable called `somethingelse`.

```
. keep id age happy // keep only relevant variables
```

We could also have said `drop somethingelse`.

Add *Variable* Labels (`label variable "..."`)

```
. label variable id "ID" // label variable

. label variable age "Age in Years" // label variable

. label variable happy "Happiness Scale" // label variable
```

Create *Value* Labels (`label define ...`)

```
. label define happy 1 "Rarely" 2 "Sometimes" 3 "Often" 4 "Always" // create value label
```

Attach *Value* Labels To *Variables* (`label values ...`)

Variables and *value labels* can have the same names but are different things. We add the variable label `happy` to the variable named `happy`.

```
. label values happy happy // attach VALUE LABEL happy to VARIABLE happy
```

Recode Outliers, Values That Are Errors, Or Values That Should Be Coded As Missing (`recode`)

```
. recode happy (999 = .) // recode values as missing
(1 changes made to happy)

. recode age (100/max = 100) // age is topcoded at 100 (may or may not be plausible)
(1 changes made to age)
```

We `describe` and `summarize` The Data And See The Changes That Have Been Made

```
. describe
Contains data
 Observations:           100
    Variables:             3
```

Variable name	Storage type	Display format	Value label	Variable label
id	float	%9.0g		ID
age	float	%9.0g		Age in Years
happy	float	%9.0g	happy	Happiness Scale

```
Sorted by:
     Note: Dataset has changed since last saved.

. summarize
```

Variable	Obs	Mean	Std. dev.	Min	Max
id	100	50.5	29.01149	1	100
age	100	50.55316	11.12025	24.9591	100
happy	99	3.030303	1.501391	1	5