Data Management in Stata

November 2, 2020

Housekeeping

- We are recording
- Please feel free to ask questions!

01 Data Management Tips

02 Do Files

03 Continuous vs. Categorical Variables

04 Basic Data Cleaning

Data Management Tips

- Tip sheet sent out with Zoom link (thanks Dr. Fitzpatrick!)
- Keep raw data pristine
- Comments
- Use alignment and spacing to make things readable

Project Folder Structure

Raw Data Folder

Raw, pristine data; subfolders for original data

Build Data

Do files that clean data

Programs Folder

Programs that manipulate the data

Output Folder

Data that is the result of the programs in the Build Data Folder

Please watch the part of the recording from the workshop on do files and comments.

Using the lookfor command, we can look for variables by searching among all variable names and labels.

lookfor height

```
. lookfor height

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
an2             float   %9.0g      an2        child's length or height
hap             float   %9.0g                 height for age percentile
haz             float   %9.0g                 height for age z-score
ham             float   %9.0g                 height for age percent of reference median
whp             float   %9.0g                 weight for height percentile
whz             float   %9.0g                 weight for height z-score
whm             float   %9.0g                 weight for height percent of reference median
```

One of the most useful Stata commands uses the tabulate and summarize commands together. For example, we might want to know the average weight of children at different ages. We can type

tab uf11, sum(an1)

```
. tab uf11, sum(an1)
```

age of child	Summary of child's weight (kilograms) Mean	Std. Dev.	Freq.
0	9.4496468	15.197266	5,096
1	11.071011	11.74525	5,026
2	13.439275	12.485567	5,049
3	15.644188	13.846132	4,542
4	17.336117	14.024042	3,281
Total	13.029003	13.75435	22,994

We can start to look at issues related to the child's weight and their sex. Let's limit our analysis to one age group, let's say we're interested in 4 year olds.

tab hl4 if uf11 == 4, sum(an1)

```
tab hl4 if uf11 == 4, sum (an1)
```

sex	Summary of child's weight (kilograms) Mean	Std. Dev.	Freq.
male	17.265359	13.438022	1,631
female	17.406061	14.583932	1,650
Total	17.336117	14.024042	3,281

We can also use the tabulate and sum in a two way table. For example we could look at child's weight by both gender and wealth index at the same time. Here we would type:

```
tab wlthind5 hl4 if uf11 == 4, sum(an1)
```

The output that the table gives us means, standard deviations and frequencies for each cell. This can be overwhelming. If we are interested only in means, we can request this by typing:

```
tab wlthind5 hl4 if uf11==4, sum (an1) means
```

```
tab wlthind5 hl4 if uf11==4, sum (an1) means

                Means of child's weight (kilograms)

  wealth |
   index |        sex
quintile |    male     female      Total
---------+----------------------------------
  lowest | 17.274778  17.604244   17.44874
  second | 17.202744  17.727679  17.468374
  middle | 16.076657  17.177955  16.598939
  fourth | 17.712195  17.777778  17.744785
 highest | 18.238832  16.633333  17.423858
---------+----------------------------------
   Total | 17.265359  17.406061  17.336117
```

Continuous Variables

Continuous variables have an infinite number of possible values that fall between any two observed values. For example, consider age. In this data set, children's age is recorded in years (0 – 4). But it could have been recorded in months, days, minutes, or even seconds. A continuous variable is ordinal in the sense that its values have an inherent order. In the age example, an age of 3 is one year older than the age of 2 years, thus the unit of measurement in between these two lines is in itself meaningful. This may seem like common sense, but when we consider categorical values, this will no longer be true. Examples of continuous variables in this dataset include child's age (uf11), child's weight (an1), and child's height (an2). We are not being terribly careful with our definitions. We are going to treat variables like household size as continuous variables: continuous variable are where taking the average gives an answer that is readily interpreted. Taking the average of a categorical variable, on the other hand, usually yields nonsense.

Categorical Variables

Categorical variables are made up of separate and distinct categories that do not have an inherent order. To code these variables, each category is typically assigned a value, but this assignment is completely arbitrary. Take for example the ethnicity (lookfor ethnicity, codebook hc1b) variable. The sex variable (hl4) arbitrarily assigns a 1 to males and a 2 to females. Other examples of what we will consider categorical variables include the variable for religion (hc1a), district (hhdis), and region (hhreg). It generally does not make sense for us to summarize for these variables, since the mean and standard deviation have no real meaning, though it may be useful purely as a way to understand how the variable is coded.

Generating New Variables

Often, we want to create new variables from the variables in the data set. This could be because we do not like the way the variable was constructed originally and/or because we need to have the variable in an alternative form to analyze it; because we need to create combined categorical variables or because we simply need another new variable. Commands we use are generate and replace.

Suppose we want to create a simpler education variable for mother's education that only has three categories: less than none, primary, or secondary school.

We use the generate command, which can be shortened to gen followed by replace .

```
gen ednew=.
replace ednew=1 if melevel==1
replace ednew=2 if melevel==2
replace ednew=3 if melevel==3
replace ednew=3 if melevel==4
```

We assign a variable label – which these are the things in the right column of our variable view in the main Stata interface to our new variable like this:

```
label var ednew "Education of mother"
```

We can attach value labels to our new variable like this:

```
label define ednewlab 1 "None" 2 "Primary" 3 "Secondary"
label values ednew ednewlab
```

ednewlab is the name of the collection of value labels, like we had value labels of domestic and foreign last week. The second command assigns the new variable the value labels.

Let's see how this worked.

tab ednew

```
. tab ednew
```

Eduation of mother	Freq.	Percent	Cum.
None	5,222	22.48	22.48
Primary	15,467	66.59	89.07
Secondary	2,539	10.93	100.00
Total	23,228	100.00	

Let's see how this worked.

tab melevel ednew

```
. tab melevel ednew
```

mother's education	Eduation of mother None	Primary	Secondary	Total
none	5,222	0	0	5,222
primary	0	15,467	0	15,467
secondary	0	0	2,479	2,479
non-standard curricul	0	0	60	60
Total	5,222	15,467	2,539	23,228

Note that once a variable has been generated we cannot generate it again. So we have to use the replace command to make additional changes to the variable. Usually it will take a generate command followed by one or more replace commands to completely define the new variable. If you make a mistake and want to start over you may want to drop the variable and begin with a new generate command:

```
drop ednew
```

What happened to the values of melevel that did not get assigned, specifically the cases in which melevel==9 or melevel==4? Let's browse the old and new education variables:

browse melevel ednew

Cases that were not assigned a value for ednew have a Stata missing value code, which is a period (.). The tabulate command does not show these unless we tell it to with the missing option:

tab melevel ednew, missing

tab melevel ednew, m

```
. tab melevel ednew, m
```

mother's education	Eduation of mother None	Primary	Secondary	.	Total
none	5,222	0	0	0	5,222
primary	0	15,467	0	0	15,467
secondary	0	0	2,479	0	2,479
non-standard curricul	0	0	60	0	60
missing/dk	0	0	0	10	10
Total	5,222	15,467	2,539	10	23,238

To demonstrate how to deal with missing values, let's go back to the weight variable (an1) .

sum an1

```
. sum an1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
an1	22,994	13.029	13.75435	1.7	99.9

Look at the maximum value of 99.9. We know that this variable is in kilograms. So, doing some math, this tell us that the maximum weight of these children (remember the oldest children are 4 years old) is 220 pounds.

So, and from any survey documentation/codebook we could see this, the 99 is the "missing" code for this variable, and is a value that you need to change in order for your analysis to be correct. For now we can deal with this in several ways:

```
replace an1 = . if an1>50
sum an1
```

```
sum an1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
an1	22,463	10.97547	3.322175	1.7	31.2

You can also see that this changes the average weight compared to the previous output.

A special type of a categorical variable is a dummy variable, also called an indicator variable. A dummy variable typically takes on a value of one if the observation meets specified criteria and a value of zero if otherwise. Observations that don't have the required information are assigned a missing value.

```
gen male = 1 if hl4==1
replace male =0 if hl4==2
tab male
sum male
```

Dummy Variables

And a dummy for having a mother with some education.

```
gen educ= 0 if melevel==1
replace educ=1 if melevel==2
replace educ=1 if melevel==3
replace educ=1 if melevel==4
tab educ
sum educ
```

Programming tip: Dummy variables should always be named such that if you were to ask the variable name in the form of a question, a "yes" always corresponds to "1".

Instead of generating new variables, let's rename ones that we already have. Let's rename the child weight and age variables (an1, uf11)

```
rename an1 weight
rename uf11 age
```

Renaming Variables

We type first after rename the variable that we want to rename, and then the new name that we want to rename it to.

Thanks!

Does anyone have any questions?
ellends@udel.edu

Next Stata Workshop:
Monday, November 9 - Data Manipulation