Stata Brush Up Course

Session 2

Sergi Quintana

IDEA

September 3, 2024

Plan for the session

We will see:

- Open and store data.
- Duplicates and missings.
- Logical operations.
- Generate variables.
- String manipulation.
- Cleaning a Data Set!

Open Data

To open data in stata we distinguish two different procedures:

- When the data is in .dta format. In this case we just use the data.
- When the data is not in .dta format. In this case we will have to import the data.

Import Data

Stata has various commands for importing data. The three main commands for reading non–Stata datasets in plain text (ASCII) are

- import delimited, which is made for reading text files created by spreadsheet or database programs or, more generally, for reading text files with clearly defined column delimiters such as commas, tabs, semicolons, or spaces.
- infile, which is made for reading simple data that are separated by spaces or rigidly formatted data aligned in columns.
- infix, which is made for data aligned in columns but possibly split across rows.

Import Data

There are other commands:

- The import excel command can read Microsoft Excel files directly, either as an .xls or as an .xlsx file.
- The import sas command can read native SAS files, so data can be transferred from SAS to Stata in this fashion.
- The import spss command can read IBM SPSS Statistics files.

Save Data

To save data we will apply the same logic as before.

- If we want to save data in .dta format we will use save command.
- If we want to save to any other format we will use export and apply the same logic as import.

Exercices - Import and Export files

Using the example data "auto.csv". Do the following exercices:

- Load the data set and save it as ".dta" file but just the first 20 observations.
- Now export the file as an excel file. Make sure you keep the original variable names and labels.
- Finally, import the ".xls" file with the corresponding variable names and labels, but just up to cell D70.

Duplicates and Missings

The duplicates command allows to

- identify
- report
- tag
- list
- drop

duplicate observations.

The missing commands will allow us to identify and understand missing patterns.

Logical Operations

a == b	true if a equals b
a != b	true if a not equal to b
a > b	true if a greater than b
a >= b	true if a greater than or equal to b
a < b	true if a less than b
a <= b	true if a less than or equal to b
!a	logical negation; true if a==0 and false otherwise
a & b	true if a!=0 and b!=0
a | b	true if a!=0 or b!=0
a && b	synonym for a & b
a || b	synonym for a | b

Exercices - Logical Operations

Using the NLSY data set "data1":

- Report the fraction of white individuals born each year by gender.
- Keep only female individuals younger than 43 years old.
- Report the average earnings by ethnicity of females that are not hispanic.
- What is the correlation of earnings and asvab if the ethnicity is white? And if it is black? And if it is white and the age is 42?

Variable Generation

The most basic form for creating new variables is generate newvar = exp, where exp is any kind of expression. We can use:

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				

Replace

Whereas generate is used to create new variables, replace is the command used for existing variables.

The replace command cannot be abbreviated. Stata generally requires you to spell out completely any command that can alter your existing data.

Exercice

Exercice 1

Using the auto data from webuse. Suppose the cost of manufacturing a car is the sum of the following:

- \$1.50 per pound of weight
- \$0.25 per pound to ship if it is foreign.
- \$100 if its rep78 is 5

Calculate the profit of selling each car.

Report the average profit for quartiles of the weight distribution.

Exercice

Exercice 2

Suppose an automobile's rating is defined by the following:

- Start by multiplying the cars weight by its mpg and dividing by 1000
- Subtract the car's price divided by 1000 unless the following exception applies: For foreign cars weighing more than 2800 pounds, subtract the car's price divided by 1200.
- Add 10 if the cars rep78 is creater than 3.
- If any of the inputs to the ranking score are missing, the score should be missing.

Create the ranking and export the data to excel.

Variable Generation

Explicit subscripting

When you type something like

```
. generate y = x
```

Stata interprets it as if you typed

```
. generate y = x[_n]
```

which means that the first observation of **y** is to be assigned the value from the first observation of **x**, the second observation of **y** is to be assigned the value from the second observation on **x**, and so on. If you instead typed

```
. generate y = x[1]
```

you would set each observation of **y** equal to the first observation on x. If you typed

```
. generate y = x[2]
```

you would set each observation of **y** equal to the second observation on x. If you typed

```
. generate y = x[0]
```

Stata would merely copy a missing value into every observation of **y** because observation 0 does not exist. The same would happen if you typed

```
. generate y = x[100]
```

and you had fewer than 100 observations in your data.

Variable Generation

The built-in underscore variable `_n` is understood by Stata to mean the observation number of the current observation. That is why

```
. generate y = x[_n]
```

results in observation 1 of `x` being copied to observation 1 of `y` and similarly for the rest of the observations. Consider

```
. generate xlag = x[_n-1]
```

`_n-1` evaluates to the observation number of the previous observation. For the first observation, `_n-1` = 0 and therefore `xlag[1]` is set to missing. For the second observation, `_n-1` = 1 and `xlag[2]` is set to the value of `x[1]`, and so on.

Similarly, the lead of `x` can be created by

```
. generate xlead = x[_n+1]
```

Here the last observation on the new variable `xlead` will be *missing* because `_n+1` will be greater than `_N` (`_N` is the total number of observations in the dataset).

Variable Generation

Subscripting within groups

When a command is preceded by the by varlist: prefix, subscript expressions and the underscore variables n and N are evaluated relative to the subset of the data currently being processed. See an example:

```
by bvar, sort: generate small_n=_n
by bvar: generate big_n =_N
by bvar: generate newvar=oldvar[1]
list
```

	bvar	oldvar	small_n	big_n	newvar
1.	1	1.1	1	3	1.1
2.	1	2.1	2	3	1.1
3.	1	3.1	3	3	1.1
4.	2	4.1	1	2	4.1
5.	2	5.1	2	2	4.1

Variable Generation

Indicator variables

There are multiple ways to generate indicator variables:

- using replace.
- using logical operators.
- using factor variables.

Variable Generation

Other generation commands:

- The egen command is useful for working across groups of variables or within groups of observations.
- The encode command turns categorical string variables into encoded numeric variables, while its counterpart decode reverses this operation.

Exercices - Variable Generation

Using the auto data. Write it as short as possible.

- Generate a new variable demeaning price. Now generate a new variable demeaning at the rep78 level.
- Generate the first and second differences of the variable price at the rep78 level.
- Generate a dummy variable for rep78 being 4 and price being smaller than 4000. Preserve the missings.
- Sum mpg for each level of rep78 if mpg is greater than 30.
- Generate a new variable that includes only the first observation of earnings for each individual.

Exercices - Variable Generation

Now using the census from webuse:

- Generate a new variable including the region divorce average excluding regions with popurban higher than 5 million.
- Generate an indicator variable if the divorce rate is higher than 1% at the region level.
- What is the region with the highest difference in divorce and marriage rates ?

Exercice - Variable Generation

Using 2000_acs_cleaned.dta :

- Generate a variable with total household income and another with the average.
- Find and summarize household size.
- Generate the total household income only for individuals under 18.
- Find the share of individuals under 18 within each household.
- Find the minimum age within each household and summarize their frequencies.

Exercice

Using nnlsy_extract

- Generate the total income of each individual.
- Find the initial age of the individuals.
- Get the initial income of the individuals.
- Compute the first difference on the income variable.
- Identify the year an individual graduated from highshool (educ $= 12$).
- Identify a break in an individual's education:
  - They have started college (educ > 13).
  - They have not finished (educ < 16).
  - The years of education completed ist the same as the year before.

Working with Strings

There are some important functions when dealing with strings:

- destring: converts variables from string to numeric.
- toostring: converts variables from numeric to string.
- substr: allows to extract a subset from a string.
- subinstr: allows to perform character substitution from a string.
- For more functions visit here.

Exercices - Strings

Using the NLSY data "data1.dta":

- Transform ethnicity to apply the following command: list i.var
- Can you generate a new variable being ethnicity*2?
- Using your previous generated variable, replace the first white occurrence by "hello".
- Generate a new string variable with the last two characters of the variable ethnicity. Can you sum ethnicity and the newly created variable?

Exercice

Using data1. Save all the output in a pdf log file.

- Open the data set and store in an excel file a copy containing only the first 4 variables.
- Demean income by gender and produce a table of the standard deviation of income by gender and race.
- Generate a new variable which is income over experience only if the individual is a female. Produce a table of the average of this variable across race.
- Are male respondents more likely to work full-time (i.e. 40 hours or more)?
- In the data, non-white respondents have lower earnings on average. Is it affected by growing up poor?
- Show the relationship of asvab and earnings. Does it change by race?

Data Cleaning Example

What to do when we open a data set for the first time (order might be different):

- Describe the data.
- Analyze labels, duplicates and others.
- Get rid of the data that we know we will not use.
- Set the right data types.
- Recode variables if necessary.
- Set labels.
- Understand the distribution of the data and missing values.

Exercice.

Repeat the previous process on the 2000_acs_sample_harmonized data set.

Missing From Previous Lecture

- Preserve and restore
- Running other do files.
- Dealing with errors (capture).