STATA Training 4

Objective 1: Combining Data Sets with Append and Merge

Objective 2: Using Loops

When we talk about combining data sets there are two ways we are going to do this:

1. Append – adds observations to the data set

Each data set has the same variables and we want to stack them onto each other. For example, you might have data on sales tax revenue for 1990. This data set has three variables: city name, month, and sales tax revenue. The other data set has the same three variables but is for the year 1991. To combine these data sets, you want to stack them on top of each other. This will keep the same number of columns (variables) but increase the number of rows (observations).

1. Merge data – adds variables to a data set

Each data set contains a unique set of variable but have one or more shared variables. For example, we have a data set with the city name and sales tax revenue by month. Another data set has the same city name variable, but has variables on the number of sporting events in that city by month. In this case, we want to combine the data sets by matching the city name and then merge the two data sets together. This will increase the number of columns (variables) and will keep the same number of rows (observations).

Append Data

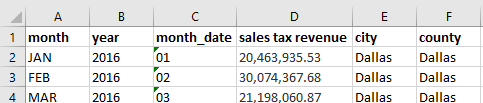
We have three files of data that have sales tax revenue by month for the city of Dallas. We want to append all three data sets into a single data set.

“dallas 2016.xlsx”

“dallas 2015.xlsx”

“dallas 2014.xlsx”

If you were to open these Excel files you will see that they have the same variables.



To start we select one of the files, it does not matter which one, and load it to STATA.

import excel using “dallas 2016.xlsx” , firstrow clear

We now save that data set into STATA format

save dallas\_2016.dta, replace

Now import the other two data sets and save them in STATA format.

import excel using “dallas 2015.xlsx” , firstrow clear

save dallas\_2015.dta, replace

import excel using “dallas 2014.xlsx” , firstrow clear

save dallas\_2014.dta, replace

With all data sets in STATA format we can now append. We can do it one at a time (notice that the 2014 data set is loaded, so that will be our master data set)

append using dallas\_2015

Now the two combined data sets of 2014 and 2015 are the master data, append using the last year

append using dallas\_2016

And now you are done.

You could append more than one data set at a time:

use dallas\_2014, clear

append using dallas\_2015 dallas\_2016

And do you are re-done.

Merge data

We want to create a data set that has the variable for sales tax revenue and a variable for precipitation. Maybe we want to run a simple model to look at how sales tax revenues changes with the amount of precipitation that falls in a month.

That model would look like this:

The “dallas 2014.xslx” has sales tax revenue and the data set “dallas stats 2014.xlsx” has statistics on precipitation. So we need to combine these data sets to add variables to the data set.

Let’s start with importing the “dallas stats 2014.xlsx” into STATA

import excel using “dallas stats 2014.xlsx”, firstrow clear

save dallas\_stats\_2014.dta, replace

We want to merge these two data sets by month; that is match Jan to Jan in the two data sets.

merge 1:1 month using dallas\_2014

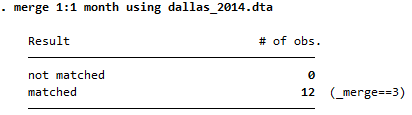
\*Note: We do a 1:1 merge because each data set has 1 january that we want to merge with.

Each time you merge data, STATA creates a variable *\_merge* which will give three values:

1. observations in the master data but not using
2. observations in the using data set but not the master
3. observations in both data sets

To see how this did, we type

tab \_merge



For this merge, we had 12 matches (1 for each month in the year)

Do it on your own: Now see if you can merge the 2015 and 2016 data together

Do it on your own: We appended all three data sets with sales tax revenue. Can you

1. append all three data sets with precipitation data
2. merge those two data sets together
3. estimate the following model
4. How does tax revenue vary with precipitation?

Do it on your own: use master\_sales\_tax\_revenue\_v2.xlsx and merge it with unemployment.xlsx

Objective: Loops

There are two common types of loops that you will do in STATA. One is looping by words and the other by numbers. Confused, don’t be.

Let’s look at the code that we just did above:

import excel using “dallas 2016.xlsx” , firstrow clear

save dallas\_2016.dta, replace

import excel using “dallas 2015.xlsx” , firstrow clear

save dallas\_2015.dta, replace

import excel using “dallas 2014.xlsx” , firstrow clear

save dallas\_2014.dta, replace

Notice that we really just repeated that code. The only difference is that we have different years or different numbers. We can then write a loop.

First step is to define a local variable i, that will start with 2014 the first time through the loop, then go to 2015, then 2016 and end the loop

Loops with numbers

forvalues i=2014/2016 {

import excel using “dallas `i’.xlsx”, firstrow clear

save dallas\_`i'\_2016, replace

Instead of writing the number, you write the local variable. We do this by typing ` which is located on the top left of your keyboard. Then the letter i, followed by an apostrophe ‘

}

Close the loop

Looping on words:

we have 4 data sets that have a tab with data for the years from 2014-2016

allen.xlsx

arlington.xlsx

dallas.xlsx

irving.xlsx

We want to combine these into a single data set in STATA format. To do this, we are going to loop first by city name and then by year. That is a loop within a loop. [inception!!!!!]

foreach x in allen arlington dallas irving {

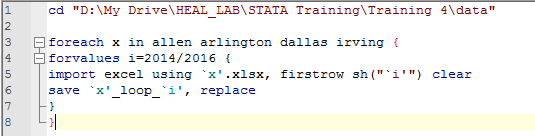
forvalues i=2014/2016 {

import excel using `x’.xlsx, firstrow sh(“`i’”) clear

save `x’\_loop\_`i’, replace

}

}



Now that they are all saved, simply append them all

Use allen\_loop\_2014, clear

Append using allen\_loop\_2015 allen\_loop\_2016 arlinton\_loop\_2014 arlington\_loop\_2015 arlington\_loop\_2016 dallas\_loop\_2014 dallas\_loop\_2015 dallas\_loop\_2016 irving\_loop\_2014 irving\_loop\_2015 irving\_loop\_2016

Boom Baby!!!!