Version: March 14, 2022 Spring 2022
EC200: Econometrics and Applications

Lab 5: Merging and hypothesis tests

Objectives

Today we're going to keep working with `cps_2016.dta`, which contains information from the 2016 Current Population Survey.

We're going to merge in county-level unemployment rates from the BLS

By the end of this lab, you should be able to complete the following tasks in Stata:

- Import data from Excel
- Merge data sets
- Test hypotheses for linear combinations of coefficients
- Test the general significance of a regression

Key commands

command	description
Importing data	
`import excel using "file1.xlsx", firstrow clear`	Import the data set file1.xlsx from excel into Stata. The `firstrow` option tells Stata to use the first row as the variable name. The `clear` option tells Stata to erase any data already in the set
Identifying duplicates	
`duplicates list var1 var2`	List any observations that are duplicates on the listed variables, *var1* `var2`, etc.
`duplicates tag var1 var2, gen(d1)`	Generate a new variable, `d1` that indicates which variables are duplicates for `var1` and `var2`
Merging datasets	
`merge 1:1 var1 var2 using file2`	Perform a one-to-one merge based on `var1` and `var2`. There cannot be any duplicates on the variables you are using to merge
`merge m:1 var1 var2 using file2`	Perform a many-to-one merge based on `var1` and `var2`. There can be duplicate identifiers in the master data set (like if merging state data to individuals), but there should be no duplicates in the using data set

command	description
Converting between string and numeric variables	
`decode var1, gen(newvar)`	Take a numeric variable with labels and generate a new string variable that is equal to the values of those labels. (You can do the opposite with `encode`).
`destring var1, replace`	Take a string variable, `var1` and convert it to a numeric variable, replacing the old variable
`tostring var2, gen(string_var)`	Take a numeric variable, `var2` and make it a string, but make that into a new variable called `string_var`
Statistical tests	
`test var1 = var2`	Run after estimating a regression. Tests the null hypothesis that the coefficient on `var1` equals the coefficient on `var2`, against the two-sided alternative.
`testparm var1 var2 ...`	Run after estimating a regression. Tests the whether all listed variables, `var1` , `var2`, etc., are jointly equal to zero, against the two-sided alternative.

0.0.1 A note on temporary files (optional)

This exercise works by having two data sets stored on your hard drive, then running a `merge` command to unite them. You might notice that the workflow feels clunky and generates extra files - open a data set, save it, open another data set, then merge in the first data set.

You can use temporary files to speed things up! Basically, you can save files in your local memory, and call those files the same way we called local variables. Everything has to be run in the do-file for this to work.

A short example (you can paste this in a do-file and run it, as it uses Stata files) :

```
tempfile tempauto       // Declare tempfile (needs to run before you try to save)

webuse autosize,clear

save ‘tempauto’, replace    // save to temp file t1

webuse autoexpense, clear

merge 1:1 make using ‘tempauto’   // call tempfile

tab _merge    // check out merge

list
```

```
r htmltools::HTML("{{< youtube 8yfXvk8QYy0 >}}")
```

0 Lab 5 Worksheet

0.1.1 What do I submit?

- Your written up answers to the exercise questions. This can be typed or written out then scanned (or photographed), in any reasonable format.
- The do-file you've created that runs this analysis
- A log file that contains the results from this exercise.

0.1.2 Exercises

1. Visit https://www.bls.gov/lau/tables.htm to access 2016 annual **county-level** *annual* unemployment rates.
    1. Download the appropriate table.
    2. Rename variables as needed, and delete any unnecessary cells. If you want your life to be easier, make the first row include your variable names, and then have the data start in second row.[1]
    3. Save your revised file.
2. Open Stata, start a new do-file (or bring in a template). Make sure you add code to start (and end) a log.
3. Import your unemployment excel into Stata and save it as a data file, `unemp.dta`.
4. Open `cps_2016.dta` and restrict the sample to adults (age 18+).
5. Now, merge your unemployment data into `cps_2016.dta` by county. This may not be smooth. A few tips:
    1. The FIPS codes are in different formats between the two data sets. A county code like this "55083" contins a state part (55) and a county part (083).
    2. You can convert a variable to and from a string using the commands `destring var1,replace` and `tostring var2,replace`, respecitvely.
    3. You can concatenate string variables by adding them like this: `gen newvar = string1 + string2`
    4. Determine whether you need a one-to-one or many-to-one merge.
    5. You may get errors, and you'll need to fix these to have a successful merge.
6. You've done it! Tabulate the new variable `_merge`. What share of observations successfully merge?[2]

[1] You can also sort this out w/ Stata commands if you'd rather work with the raw, unedited file

[2] To get a sense if you've done this right, about 40-45% of observations should match. This is because the CPS will withhold county-level identifiers for very small counties to protect confidentiality.

7. Drop any unmatched observations (you can use `drop if`, as we'll retain this restriction for the rest of the exercise.) What is the average unemployment rate for the entire sample - why would this be different than taking the average of county-level unemployment rates in your excel file?

8. Why can't we use education as a linear variable?

9. Generate three dummy variables. These three variables should be mutually exclusive, and they should not be missing for any people.

   - `lesshs`, a variable equal to one if a person completed *less than* a high school diploma
   - `hsgrad`, a variable equal to one if a person completed at least a high school but less than a Bachelor's degree
   - `colgrad`, a variable equal to one if a person completed a Bachelor's degree or higher

   Note: Education is coded with **labels,** which means that it is numeric data with a description of what each number means on top. These show up as blue in the Stata browser. To view variables without the labels, add the no-label option:`tab educ, nolabel`.

10. What is the mean of `lesshs`, `hsgrad`, and `colgrad`?

11. Estimate a regression of total personal income on education, usingthe binary variables you just created. Omit `lesshs`.

12. Set up a hypothesis test for whether both `hsgrad` and `colgrad` are jointly significant. Report the null hypothesis, alternative hypothesis, test statistic, and conclusion.

13. Set up a hypothesis test for whether the returns to being a high-school graduate are the same as the returns to being a college graduate. Report the null hypothesis, alternative hypothesis, test statistic, and conclusion.

14. Is this regression significant overall? Explain how you know.

15. Now add county-level unemployment rate to the previous equation. What is the interpretation of the coefficient on unemployment? Is it statistically significant?

16. Estimate the same equation by regressing total personal income on the education binary variables and county-level unemployment, restricting to those who are currently in the labor force. How does this change the coefficient on unemployment?

17. Identify three *state* or *county-level* variables that are likely to cause omitted variable bias if you want to know whether unemployment affects individual wages.

18. For *one* of the variables you listed above, find the data online, import into Stata, and merge it in.

19. Regress total personal income on the education binary variables, county-level unemployment, and the new variable you found. Restrict your sample to those who are currently in the labor force. How does the inclusion of your new variable affect the coefficient on unemployment?