Version: October 5, 2021 Fall 2021
EC200: Econometrics and Applications

Lab 4: Loops and Multivariate Regression

Materials

- `cps_2016.dta`
- Do-file template `labtemplate.do`
- Looping exercise `loop_example.do`

Objectives

Today we're going to work with some new data, `cps_2016.dta`, which contains information from the 2016 Current Population Survey.

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

- Estimate and interpret multiple linear regression in levels, using continuous and binary independent variables, and use heteroskedasticity-robust standard errors.
- Interpret the results of multivariate linear regressions in terms of statistical and economic significance
- Practice generating binary variables from categorical measures
- Set up basic loops
- Use `xi` and `i.` prefix to include a lot of binary indicator variables at once.

Key commands

command	description
`regress var1 var2...`	Estimate a regression, with `var1` as the dependent variable and all others as the independent
variable(s)	
`tabulate var1,nolabel`	Tabulate variables *without* labels
`replace var1 = . if var1 == 999999`	Replace `var1` as missing (using a dot) if `var1` is equal to 999999. Can be replaced with any other values or logical expressions.

Creating binary variables

Recall that there are two easy ways to make binary variables out of categorical or continuous variables. Consider the variable `race`, where 100 = White, 200 = Black, 300 = Native American, 651 = Asian, etc. Suppose you want to generate a binary indicator for whether a person is White.

- `gen white = race == 100`: generates a variable equal to 1 if `race` is 100, and 0 otherwise
- `gen white = 1 if race == 100`: generates a variable equal to 1 if `race` is 100, and **missing** otherwise. To complete this you need two lines of code:
  ```
  gen white = 1 if race == 100
  replace white = 0 if race != 100
  ```

Working with loops

Loops can help us (1) avoid errors and (2) code super fast!

I've uploaded a sample from our class `here, as lab4_sample.do`

Stata has two types of looping setups, using the `forval` or `foreach` command. The first is simpler, and the second is more versatile. Recall that you can always use `help forval` or `help foreach` if your code isn't working or if you have a vision you're not sure how to realize.

Looping with `forval`

```
forvalues lname = range {
                Stata commands referring to ‘lname’
        }
```

What does each component mean?

- `forvalues`: this is the command. You can abbreviate it as `forval`.
- `lname`: this is a variable you make up. Often, people will just use `i`, becuase we're just counting. It will take on the values in `range` as it increments through the loop. It is a **local** variable, meaning that you have to call it using 'lname', and not as lname (need those punctuation marks!) and that it is only saved as long as your do-file is running.
- `range`: this is the set of values that the local variable will iterate over
- Brackets: Open bracket needs to be on same line as the `forval` command. Close bracket needs to be on its own line.

```
forval i = 0/2{
gen labfor‘i’ = labforce == ‘i’
}
```

What does this do? It creates a loop for which local variable '`i'` is first 0, then 1, then 2. Within the loop, it generates `labfor0`, which is equal to 1 if `labforce` equals 0 (not in universe), it generates `labfor1`, which is equal to 1 if `labforce` equals 1 (not in labor force), and it generates `labfor2`, which is equal to 1 if `labforce` equals 2 (in labor force).

The choice of ranges can be done in other ways:

- `forval i = 0/10`: hits every integer between 0 and 10 - 0, 1, 2, … 10
- `forval i = 1(10)100`: starts at 1, then increments by 10, stopping at 100: 1, 11, 21, 31, … 91
- `forvalues k = 5 10 to 300`: starts at 10, then increments by 5 until 300: 5, 10, 15, …

See `help forval` for more options

Looping with `foreach` This command lets you loop through number lists (like above), but also through sets of variables, values, names, etc. You can approach it two ways:

- Do not specify the type of list, use **in**: `foreach lname in list`:
- Specify the type of list (`listtype`), use **of**: `foreach lname of listtype list`

This is confusing until we see examples:

```
foreach x in "rice wheat corn rye barley oats" {
          display "‘x’"
        }
```

This will start with `x` equal to the string “rice”. Then, it will run with `x` equal to “wheat”, etc.

```
    foreach num of numlist 1 4/8 13(2)21 103 {
        display ‘num’
 }
```

This will loop over 1, 4, 5, 6, 7, 8, 13, 15, 17, …

You can loop over variable names too!

```
foreach var of varlist inc* {
      summarize ‘var’,d
        }
```

This summarizes (with detail) each variable that starts with `inc`

Working with binary independent variables

When you are representing a categorical variable with a set of binary variables, there is a slow way and a fast way to integrate them.

- Slow way: generate the binary variables you want, and include them. This is good when you want to be precise about your omitted variable, or when you want to create complicated binary categories

```
gen white_nh = race == 100 & hisp == 0
gen black_nh = race == 200 & hisp == 1
gen hisp = hisp == 1
gen other = white_nh == 0 & black_nh == 0 & hisp == 0
regress incwage black_nh hisp other
```

Here, white, non-Hispanic is the omitted "reference" category.

- Fast way: tell Stata to create a binary variable for each value of a categorical variable. This is good when you aren't trying to do anything complicated and when you want to be quick - very useful if you want something like state-level dummies.

```
regress incwage i.statefip
```

Note that this will work only if your categorical variable is numeric. If it's a string you'll get an error. You can fix it by adding a `xi:` prefix, like so:

```
xi: regress incwage i.statefip
```

When we include a dummy variable for every value of a categorical variable, like above, we call those "fixed effects." We'll talk about these more soon.

Reading regression tables (reminder!)

Lab 4 Worksheet

Download the do-file template and data files. Personalize the file paths so that you can run it and open your `cps_2016.dta` file. You can also work with a blank data file if you're more comfortable - just make sure you remember to include commands to start and close your log file.

Use robust standard errors in all regressions

1. Let's practice with loops! Download loop_example.do and paste the code into your sample. Run it and look at the output. In your do-file, write comments that describe what each loop is going.
2. Now, go back to your `cps_2016.dta` file and do-file template. Adjust your do-file template so that it loads `cps_2016.dta` and starts a log.
3. Restrict your sample to individuals ages 25-54.
4. Create a new variable, `birthyr`, equal to each individual's year of birth. Is there any potential imprecision or error in this variable?

5. Then, write a loop to generate a dummy variable for each possible value of birth year.[1]

6. Look through the available list of data (note, IPUMS has full documentation of all variables). Based on this data, think of a research question for your lab of the form, "What is the relationship between .... and ...?". Pick a dependent variable that is **continuous**. *(Because a later question asks you to explore race/ethnicity controls, please do not use a race/ethnicity variable for X.)*

   Research question:

   Dependent variable (Y):

   Key independent variable (X):

7. Using the data available, write a reasonable population model, including your key independent variable along with a set of likely relevant independent variables (somewhere between 2 and 5 additional variables). Before estimating your regression, you should tabulate each variable to make sure you are interpreting it correctly.

8. In words, what exactly will your estimated regression tell us?

9. What do you hypothesize the answer to your research question is? (i.e. strong positive, weak negative, none)

10. Before you estimate your model, make sure you don't have any N/A values coded. For example, if `incwage` is not applicable, it is coded as 9999999. Tabulate or summarize your data to check for any values like this. Replace any values as missing if they are equal to some N/A code (see above).

11. Estimate the relationship between X and Y using simple linear regression (excluding any other covariates). Write your results in equation form and report the R^2. How many degrees of freedom do you have?

12. Estimate the relationship between X and Y using multiple linear regression (including other covariates). That is, estimate the population model you wrote earlier. Write your results in equation form and report the R^2. How many degrees of freedom do you have?

13. Using your multivariate linear regression from the previous step, set up a hypothesis test for your parameter of interest, the β associated with your key independent variable, X. What do you find? What is the p-value? What is the interpretation?

14. Besides your key independent variable, which other variables are statistically significant at the five-percent level?

15. A lot of student research papers will look at differences in outcomes by gender and by racial/ethnic groups. U.S. surveys like the CPS, ACS, and Census treat race and ethnicity a little strangely, and it can take some practice to get comfortable.

    There are two variables commonly used to identify a person's race and ethnicity: the `race` and the `hispan` variable.

[1]There is a faster way to do this, using `xi i.birthyr`, but we're learning about loops, so just go with it.

1. What share of the sample is White, non-Hispanic?
2. What share of the sample is Hispanic/Latino?
3. A common way to summarize the racial/ethnic make-up of the U.S. is the following categories:
   - White, non-Hispanic
   - Black, non-Hispanic
   - Hispanic/Latino
   - Asian, non-Hispanic
   - Other

   Make a table that shows the distribution of the population into these five groupings.

16. Estimate your multiple linear regression model from earlier, but include the race/ethnicity variables that you created in the previous part. How do the inclusion of these factors affect your estimates of the relationship between Y and X?

17. Now, add "birth-year fixed effects" to your regression that you generated earlier. Because there is a set of binary 0/1 variables, one for each year of birth, they will essentially pull out any mean differences in your dependent variable at the birth-year level - so if your outcome variable is different for people born in 1971 vs 1971 on average, these variables will take care of it. What is the omitted birth cohort? How do the inclusion of these factors affect your estimates of the relationship between Y and X?