University of Essex
2015-16

Department of Government
TA: Stavros Poupakis

GV903: Advanced Research Methods

Lab 4: Relationships Between Variables & Least Squares

Generate 1000 observations of $x_i \sim N(77, 9)$. If you are not sure about the arguments of the function, run `help rnormal` in Stata to see the help page of the command. Before you run the command, add before that the following line `set seed 12345` but pick another number, any number! This way each of us will have his own random-number seed.

Look at the descriptive statistics, does the mean and variance look as you expected?

Run a t-test to test the hypothesis $H_0 : \mu = 77$ by typing the following

```
ttest x==77
```

We will discuss everything from this table. It should be absolutely clear what each number means by everyone. **Generate x again and perform a new t-test.**

If we know how to derive everything by hand, we can also ask Stata to do the calculations for us. Run `sum x` and then `return list` to remember what `sum` returns. Using the `display` command we can easily print the t value by typing

```
dis (r(mean)-77) * sqrt(r(N)) / r(sd)
```

Replicate all the results from the t-test output in Stata using `dis`, the appropriate `r()` objects and `tprob` and `invt()`.

Run a Monte Carlo simulation to see how the statistic behaves. First we will write a program which returns the mean and standard deviation. Then we will generate some new variables `t`, `p`, `ci_lo` and `ci_up` and use them to measure the Type I error. Do you remember what is this?

```
clear all

program define mc_ttest, rclass
        drop _all
syntax [, n(int 100)]
set obs `n'
gen x = rnormal(77,3)
sum x
return scalar x_mean = r(mean)
return scalar x_sd= r(sd)
end

simulate mean=r(x_mean) sd=r(x_sd) , reps(10000): mc_ttest, n($n)
```

Generate 100 observations of $x_i \sim U(0, 10)$ and then create $y_i = 2 + 3x_i + \epsilon_i$ as

```
gen y = 2 + 3*x + rnormal(0,3)
```

We know the true values of β_0 and β_1 as we created them, but let's pretend we don't know and all we have is x and y, our data.

Plot the two variables using the command `scatter y x` and **run a regression** of y on x using `reg y x`. Again, everyone need to understand every element of the output table.
If you type `return list` after `reg` you will notice that the `r` obects are different from before. An easy way to access the coefficients and SEs for each covariate is typing

```
dis _b[x]
dis _se[x]
```

Calculate the t, p and 95%CI for the estimate of β_1 using the `display` command.
Another useful command in Stata, which can be run after `reg` is `predict`. You can instantly calculate the predicted values of the model, or the residuals by typing

```
predict think_a_name_to_put , xb
predict similar_here , residuals
```

The `xb` is the default option and is not needed to be typed. **What is the mean and variance of the residuals? How is their distribution.**

Plot two scatterplots together, y on x and predicted values on x. To do that, run:

```
twoway (scatter y x) (scatter think_a_name_to_put x)
```

But it would be nice if in one of the two we had different colors. Use `help scatter` to find out how.

To see the distribution of b_1, the OLS estimate of β_1 again we need Monte Carlo simulation.

Write a program as before - just remember that now you will get what you want using this:

```
return scalar b1 = _b[x]
```

How is b_1 distributed? Does it make sense from what you learned so far?