Econometrics
Basics of 'R' II

Multiple Regression

We will use the data 'cps_2009.csv' again, but if you saved your workspace last time it should already be loaded and saved as whatever you saved it as. Do you see it? I will assume you have it saved and have it saved as cps1.

Say I want a regression like: $wage = \beta_0 + \beta_1 sex + \beta_2 ed + \epsilon$ because I want to know how wage is related to sex and education. :

cps.reg <− lm(wage∼sex+ed,data = cps1)

All you do is add the '+' symbol in between the variables you want to include

Then again (assuming you saved your workspace):

summaryHCCM(cps.reg)

Easy.

Creating Interactions or Square terms

Say I want to include an interaction between sex and ed. I have two ways to do this. One way is to do it within the regression command:

cps.reg.sed <− lm(wage∼sex+ed +I(sex*ed),data = cps1)

Note the I(·), this tells it to evaluate what is in the parenthesis. Or I could create a new variable in my data set first and then include that:

cps1$sex.ed <− cps1$sex*cps1$ed

Now if you click on cps1 in your environment panel (top right) it will refresh and you should see the new variable. Now you can just use it.

The same goes for things like wanting to add a square term or natural log - either make a new variable or call it in the regression with I().

Partial F-test

Maybe we want to test if two or more variables are jointly significant, say adding regions will improve the regression. This is done with an F-test using anova() command.

First we need to run both models:

out_base $<-$ lm(wage$\sim$sex+ed,data = cps1)
out_full $<-$ lm(wage$\sim$sex+ed+neast+midw+south, data = cps1)

Now compare them:

anova(out_base,out_full)

Now, this is the homoskedastic only F-statistic. Believe me? Check - no really, check (ie. do summary of each to get your R^2 for each and manually compute)

TURN IN

Copy and paste your R script code and your results in a word file. Also show your calculations showing that you can replicate this F-test with the two regressions R^2

If you have forgotten how to do this just look in your notes

So what about heteroskedasticity? Ok, we have to do it slightly different.

Now to do this we will need to use a package - so type :

library(car)

This loads the option we need (note you had already downloaded this package before, the 'library' command brings it up to be used).

Ok, now our regression (the unrestricted one) has 6 coefficients: β_0 - β_1 - β_2 - β_3 - β_4 - β_5, and we are testing three restrictions (if all three region dummies are mutually insignificant).

Think of it like this:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The first line says that the fourth coefficient (on neast) equals zero, and the second line says the fifth coefficient (on midw) equals zero, and the third line says the sixth coefficient (on south) equals zero.

The part on the left is our 'hypothesis matrix' (hm) - it is (one for each row) what we want to test.

The part on the right is what we are testing it against, just call it right hand side (rhs). Here all hypothesis are zero.

Side note

To do this we need to make these matrices - but its easy.

To make the rhs, we just say:

rhs $<-$ c(0,0,0)

the 'c()' command sort of puts the zeros together, and to make the hypothesis matrix we simply:

hm $<-$ rbind(c(0,0,0,1,0,0),c(0,0,0,0,1,0),c(0,0,0,0,0,1))

the 'rbind' puts the rows into actual rows.

Now we can redo our f-test

linearHypothesis(out_full,hm,rhs)

Now if you note the F-stat is the same as before! Because we need to tell it to use the robust version

linearHypothesis(out_full,hm,rhs,white.adjust = T)

This tells it to use a certain form of hetero-robust SEs when doing our test.

TURN IN

Copy and paste your R script code for the above different F test into your word file, along with the results.

Note, if your rhs is just all zeros, you can omit it as that is the default value anyway.

Multiple Restrictions

We can use what we just did to easily test things like $\beta_2 = \beta_3$.

Note this can be rewritten as $\beta_2 + (-\beta_3) = 0$, or in the above testing framework:

$$\begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

Predictions

The 'car' package also has code that can be used to predict outcomes. First you need model results, lets use the above out_full, but we also need to tell it what values of X we want to predict Y. To do this we first create a 'data.frame' with those values:

pred.val ¡- data.frame(ed = 12, sex = 1, neast = 1, midw = 0, south = 0)

Then use the 'predict' function (here I also tell it to report confidence intervals):
predict(out_full, pred.val , interval="confidence")

Make sure you recall this, because we will be using it later.