Econometrics
Basics of 'R'

[Disclaimer: This is not an R class or a rigorous R tutorial. I will not discuss data/variable types or discuss writing functions etc. This is just enough so that you can do some regressions and data analysis using R. If you want to learn how to 'really' use R there are many free books/tutorials.]

NOTE: You will need the package 'car' for this exercise. So please **before class** open up RStudio, go to the top panel and select Tools, then select Install Packages, then type in car and hit Install. This will save time as the computers in class can be very slow.

We will also be using some data that you will need to download. First you need to save the data. Of course you need to save it in a place where the program can find it. The program will only 'look' in its working directory. This is the place where you will save all the data you will use. What is your working directory? Type getwd() and it will tell you. Again, this should be that folder you made in your USB drive.

Now go to my webpage and download the folder with data sets ('R Files'), now open and save all of these files in your working directory.

Open up RStudio

Now go to '$File \rightarrow NewFile \rightarrow RScript$'

This will open up a new panel in the top left. It should look like this (though yours will be empty):

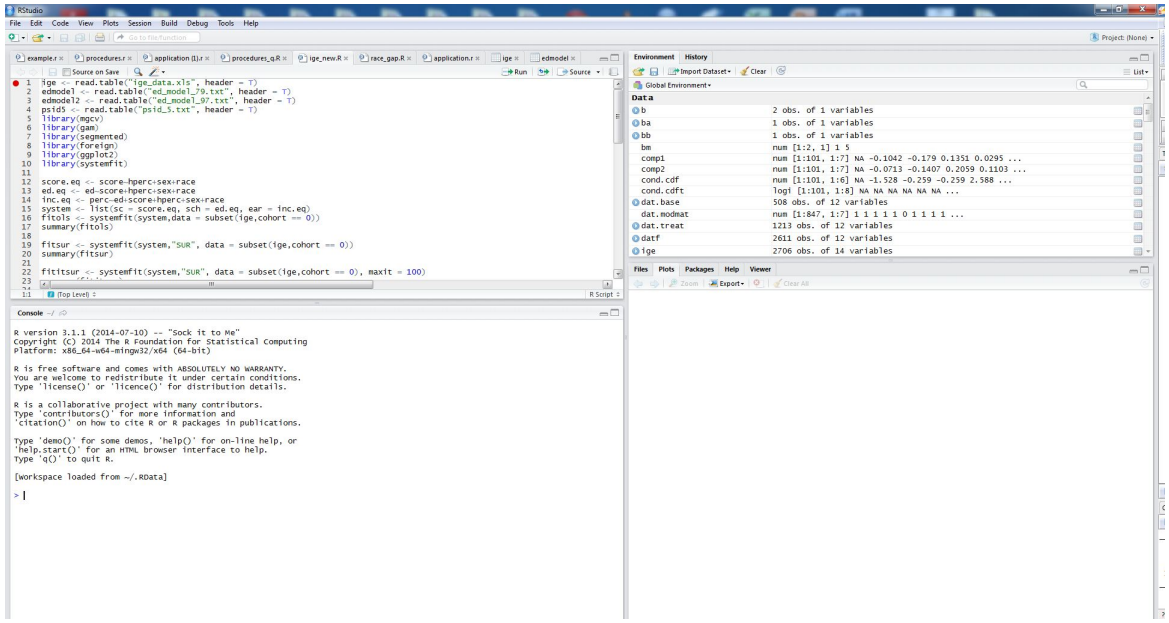

Figure 1: RStudio Screen Shot

The top left panel will be where you have opened 'scripts' (ie. code for larger projects) and data

sets that you can visualize. You see I have several open (notice the tabs), yours will be empty.

The top right is lists of data and 'objects' opened in R.

The bottom right has multiple tabs. Such as 'files' where you can see saved data (note that file tab should be your folder on your USB drive 'myrfiles'), or saved scripts. There is also a tab for any 'plots' opened that you may have made from data. etc.

The bottom left is the 'console' or 'command line' if you will.

You will 'work in' the console for simple things, and work in and save script files for larger projects.

First, don't get overwhelmed.

R is just a tool. For example its a calculator:
Go to the console (bottom left) and type: 3 + 5 in the console and hit enter

Or multiply: 3*5 and hit enter

Or raise to power: 3∧5 and hit enter

Parentheses work in normal ways: 2*(3+2) hit enter

Note you are typing next to prompt >

If you see a + instead, it means R is expecting you to keep going, so either continue what you are doing or you made a mistake.

For example, type 5* and hit enter. You see the prompt is now + because * is not a valid end of a command - 5 times what?? Now if at the + prompt you hit 3 and enter you get the answer to 5*3. This is nice because sometimes maybe you have a long equation, well you can just continue on the next line with no worries.

So if you get the + prompt in error, either hit the ↑ key and return to the previous line, or hit esc and start over.

Ok, that was easy. But we will be wanting to use some data. For this course I will be giving you data in nice formats for easy use. And in general I will give them in the form of 'csv' files with variable names.

Data

Hopefully you already downloaded and saved the needed data files. If not, go to the top of this file and follow the directions for that. Now you should see them all listed in your bottom right panel under the 'Files' tab.

Now I want you to load the cps_2009.csv file.

You will use the 'read.csv' code to import the data.

Decide what you want to call this data, maybe cps1, maybe mycps, whatever. Now import it (just do this in the console):

cps1 <− read.csv("cps_2009.csv")

The <− (no space between them) assigns the data to the name I chose (cps1).

Now if you look in the top right in environment you should see your data set listed. Click on it.

Now you should see it in the top left panel with variable names at the top (shaded grey).

The data should have 8 columns
Variables:
wage: hourly earnings in 2004 $s
ed: years of education
sex: sex, =1 if female, =0 if male
age: age in years
neast: = 1 if rom North East
midw: = 1 if from MidWest
south: = 1 if from South
west: = 1 if from West

Now lets look at some data summaries. We can do this by using the 'summary' call, again in the console:

summary(cps1)

This gives me the mean, median, 1st and 3rd quartile, min and max of each variable. I can also just ask for a summary of one variable. To do this I need to tell R what data set and what variable, like this using the $ code:

summary(cps1$wage)

Note I first tell it the data set, then the variable name. Now I could avoid this by 'attaching' the data set. To do this you simply say attach(cps1) and it only looks at this data set. But lets not do this in case you forget to 'detach' it but want to use a different data set.

Now we can look at some things we learned like looking at the covariances or correlations of the variables.

cov(cps1$wage,cps1$sex), or cor(cps1$wage,cps1$sex)

Or look at the whole covariance or correlation matrix

cov(cps1), or cor(cps1)

This gives us all the covariances (or correlations in one matrix)

Linear Regression

OLS is in a already loaded package so you can just use it. The code is just 'lm'. But you need to tell it what the regression looks like. Here is how.

Say I want a regression like: $wage = \beta_0 + \beta_1 sex + \epsilon$ because I want to know how wage is related to sex. I enter (it automatically adds an intercept unless you tell it not to):

cps.reg < − lm(wage∼sex,data = cps1)

But nothing happened! Yes it did, your results are just stored in cps.reg - an 'object' - and you should see it listed in the top right panel in your 'environment' under 'Values'. So now enter

summary(cps.reg)

```
Call:
lm(formula = wage ~ sex, data = cps1)

Residuals:
    Min      1Q  Median      3Q     Max
-22.223  -6.936  -1.771   4.833  58.191

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  28.0338     0.2580  108.67   <2e-16 ***
sex          -3.8073     0.1652  -23.05   <2e-16 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 10.24 on 15391 degrees of freedom
Multiple R-squared:  0.03337, Adjusted R-squared:  0.03331
F-statistic: 531.3 on 1 and 15391 DF,  p-value: < 2.2e-16
```

Figure 2: Regression results

And there you go. Everything we have done so far. You have your coefficients $\hat{\beta}$, your standard errors $SE(\hat{\beta})$, your t-statistics, your p-values, your SER, your R^2,

Now if you recall I made a big deal about heteroskedasticity and correct SEs. Well this is also an issue for R, these SEs are not heteroskedastic robust. So we will use a different summary package. This code was written posted to the R community by Professor John Fox of McMaster University in Canada.

If you already correctly saved the data files you will have also already saved this code. Now you should see it in your 'Files' in the bottom right panel (summaryHCCM.R). Click on it. Now it should pop up in the top left panel and you should see the code. All you need to do is hit 'Source'. Now you should see it listed under 'Functions' in the top right panel just like you have cps1 and cps.reg.

Now as long as you save your workspace when you close R (it will ask you when you close it down, just say yes) you can always use this (ie. the function, your data, your results will all be stored in your environment unless you choose to clear it). Lets try:

summaryHCCM(cps.reg)

Now you will notice the SEs are somewhat different, but not much, at least not with this data set.

You can also run regressions on subsets of the data. Say I wanted to run a regression on only men.

cps.reg.men < − lm(wage∼ed,data = subset(cps1, sex == 0))

Notice the double =, this is necessary. Also if you want many restrictions on the subset this is simply done with the & symbol:

cps.reg.men2 < − lm(wage∼ed,data = subset(cps1, sex == 0 & ed > 12))

Lets wrap up with a quick 'data check' of something we saw in class.

TURN IN

We saw that the R^2 of a simple regression equaled the correlation coefficient squared between Y and X. Lets check this.

In this, **as in all exercises we do**, I want you to write your 'code' as a new 'script'

Scripts

Everything we have done so far has been in the console. But maybe its better to write this in a 'script' so that you can save it for later. Now for simple things maybe doesn't matter, but usually is better and I will ask you to submit a copy of your script file (copied into word).

Just go to the 'File' tab all the way at the top, select 'new file' and then 'R-script'. It will open up a new tab in the top left panel.

Here you can write code to call open data, maybe run a test or two and save it all for later.
In this script write code to run a regression using the 'cps_2009' data of wages on age. Also write the line of code to print the results (summary). Also write a line to give you the square of the correlation between the two variables.

Now maybe save this as 'basics1'

Now to 'run' this, you can just highlight all three lines and hit the 'run' tab at the top right of your script. This will display the regression results and the correlation squared.

Now copy and past your script and the results into a word file (I will want you to do this for all of your exercises we do) and show me that the R^2 equals the square of the correlation coefficient. Make sure your name is on this and turn it in.
