Surviving Graduate Econometrics with R: The Basics — 1 of 8

Introduction

The following is an introduction to statistical computing with R and STATA. In the future, I would like to include SAS. It is meant for the graduate or undergraduate student in Econometrics that may want to use one statistical software package, but his teacher, adviser, or friends are using a different one. I encountered this issue when I wanted to learn and use R, while both my econometrics courses were taught using SAS and STATA. I will be following the course homeworks for ECNS 562: Econometrics II taught by Dr. Christiana Stoddard in the Spring of 2011, so you may see reference to STATA in the actual questions. Read further for the R code.

ACKNOWLEDGMENTS

Special thanks to Dr. Christiana Stoddard for letting me use her homework assignments and class notes to structure this blog series. In a subject that is prone to dry class experiences, her econometrics course was incredibly engaging, useful, and challenging — a true pleasure. Also, thank you to Dr. Joe Atwood for his help in getting me started using R and providing insightful guidance on my code and supporting me in myriad ways. Roger Avalos, a fellow graduate student, provided his STATA code for this series — as well as encouragement in writing this blog. Thank you, Roger.

Let's Get Started

For this assignment, we will be using the data available available at www.montana.edu/stock/ecns403/rawcpsdata.dta – raw Consumer Pricing Index data.

The homework questions

1. Load your data into Stata.
2. Describe your data. Label any variables missing descriptors. If you have string variables that should be converted to numeric or missing values that should be recoded, do that.
3. Calculate the summary statistics for your dataset. Note any features of interest using comments in your do-file (e.g., potential outliers, low variance variables, etc.).
4. Create a two way table for some of your variables of interest.
5. Use the tab, sum command for a variable of interest.
6. Make a well designed graph for a key relationship of interest. (Use labels, titles, etc.)
7. Use the gen command and the egen command to make two variables of interest.
8. Create a relevant dummy variable for your data.
9. Run a regression and conduct a hypothesis test of interest to you. Explain how to interpret the results using comments in your do-file.
10. Make a table with the regression results using the outreg2 command

Loading Your data

The following is STATA code that you should include at the very top of all your ".do" files:

STATA:

```
/*Last modified 1/11/2011 */
/*This program illustrates some commonly used Stata commands*/

*************************************************************************
*The following block of commands go at the start of nearly all do files*/
*Bracket comments with /* */ or just use an asterisk at line beginning

clear                                /*Adjust this for your particular dataset*/
set mem 50m                          /*Adjust this for your particular dataset*/
cd "C:\DATA\Econ 562\homework"      /*Change this for your file structure*/
log using stata_assign1.log, replace    /*Log file records commands & results*/
display "$S_DATE $S_TIME"
set more off
insheet using rawcpsdata.dta, clear
*************************************************************************
```

R:

First off, using R involves downloading and loading packages. These packages have predefined functions that will be useful in a variety of tasks. In each new R session, you must load in the desired packages using the command `require(packagename)` . However, you must first download the package to your hard drive using the code `install.packages("packagename")` . You only need to download it once, but you must load it every new R session.

Search this blog
Search...

Contributors

Kevin Goulding

Categories
Econometrics
Econometrics with R
Numpy
Python
R tips & tricks
Surviving Graduate Econometrics with R
TikZ for Economists
Visualizing Data with R
White Papers

Twitter feed
RT @gappy3000: This post, apparently about #julialang and #pydata, explains why #rstats has become the standard of data analysis http:// ... 3 years ago
RT @justinwolfers: "if prediction markets are really as valuable as economists think, then..more experimentation could prove worthwhile. ... 3 years ago
RT @vsbuffalo: For me the biggest victory is for statistics and empiricism. Go Nate Silver and @fivethirtyeight for a brilliant forecast ... 3 years ago
Follow @baha_kev

Tag Cloud
cluster-robust Econometrics heteroskedasticity LaTeX Numpy Parallel Computing plots Python R STATA tex TikZ

In the code below, we will download and load the package "foreign". This package allows you to import a variety of different data files, including data files from STATA (.dta) and SAS. I have assumed that you have not yet downloaded the package. Note that you will be asked to select a CRAN download source – I always select WASHINGTON U.S.A. at the bottom of the list (because it's closest to Montana). The code also loads the desired data file rawcpsdata.dta

```
install.packages("foreign",dependencies=TRUE)
require(foreign)
cps = read.dta("http://www.montana.edu/stock/ecns403/rawcpsdata.dta")
```

Alternately, you could simply download rawcpsdata.dta to your hard drive, and load it in like this:

```
cps = read.dta("C:\DATA\Courses\Econ 562\homework\rawcpsdata.dta")
```

If you are on a Mac (like me), the file extension will look something like this:

```
cps = read.dta("/Users/kevingoulding/DATA/Econ 562/rawcpsdata.dta")
```

You still need to download and load in the "foreign" package. When R imports your data, it creates a data frame. You can think of a data frame as a two dimensional matrix that has column headings.

```
class(cps)
```

Looking At Your Variables

After you've loaded in your data set, you may be interested to see how your statistical program interpreted the variables. To do this in STATA, it is ridiculously simple:

STATA:

```
sum
```

R:

In R, each column of your data is assigned a class which will determine how your data is treated in various functions. To see what class R has interpreted for all your variables, run the following code:

```
sapply(cps,class)
```

Or, to organize it visually, you can coerce the results into a data frame that is easier to look at:

```
as.data.frame(sapply(cps,class))
```

Later, we may be interested in changing a class. Let's say we wanted to change the variable "year" from an integer to a numeric. We would then do the following code:

```
class(cps$year)
cps$age = as.integer(cps$age)
class(cps$year)
```

Note how the class changed from before and after running the function `as.numeric()` .

Describing Your Data

To summarize your data, we like to look at summary statistics. In STATA, it is ridiculously simple:

STATA:

```
des
```

In R, it is also simple:

```
summary(cps)
```

However, the included function in R does not look very much like the results from STATA. Instead, you can use the bit of code I wrote in an earlier post, the sumstats function:

```
sumstats(cps)
```

You can also isolate a single variable for summary statistics:

```
summary(cps$perwt)
```

Create a Two-way Table

Right now, I do not have a good way to make a two-way table in STATA…

STATA:

```
entercode
```

However, in R you can choose two factors to see how many are in different "buckets". For example, say we are interested in how many males and females were in each racial group:

R:

```
twoway = table(cps$race,cps$sex)
twoway
```

Note: you can find more information on tables at http://www.cyclismo.org/tutorial/R/tables.html

Summarize on a subset of the data

Say you are interested in the summary statistics of your entire data for the year 2003:

STATA:

```
summarize if year==2003
```

R:

```
sumstats(cps[cps$year == 2003, ])
```

or,

```
summary(cps[cps$year == 2003, ])
```

You can also add additional conditions by adding "or" `|` or "and" `&` . Different conditions could be "equal to" `==` , "greater than" `>` , "greater than or equal to" `>=` , "not equal to" `!=` , etc.
For example:

```
sumstats(cps[cps$year >= 2003 & cps$race == 'White', ])
```

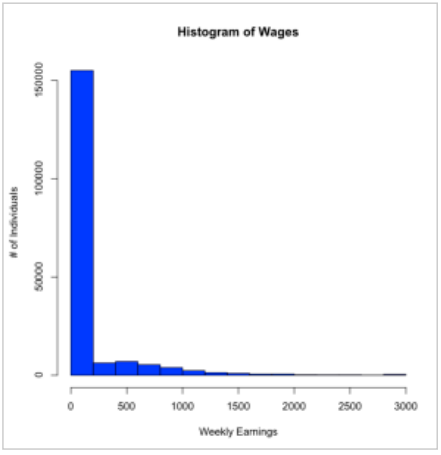
Make a well designed graph

There are many options for creating aesthetically pleasing charts in R, but for now we will focus on the simplest examples.

STATA:

```
entercode
```

Here is an example of a plot and a histogram in R:



R:

```
hist(cps$earnweek,xlab="Weekly Earnings",ylab="# of Individuals",main = "Histogram of Wages",col="blue")
```

And, just a regular plot:

```
plot(cps$earnweek,cps$age,xlab="Weekly Earnings",ylab="Age of Worker",col="red")
```

Create some variables of interest

Let's say we would like to create a variable that's a numerical proxy for level of education, so that if you have at least a high school diploma, it's equal to 1, an associate's degree =2, bacheler's degree = 3, masters or professional degree = 4, doctorate degree = 5. Our new variable will be entitled "edproxy".

STATA:

```
entercode
```

R:

```
cps$ed1 = as.numeric(cps$educ == "High school diploma or equivalent")
cps$ed2 = as.numeric(cps$educ == "Associate's degree, academic program")*2
cps$ed3 = as.numeric(cps$educ == "Bachelor's degree")*3
cps$ed4 = as.numeric(cps$educ == "Master's degree" |
                     cps$educ == "Professional school degree")*4
cps$ed5 = as.numeric(cps$educ == "Doctorate degree")*5

cps$edproxy = cps$ed1 + cps$ed2 + cps$ed3 + cps$ed4 + cps$ed5

cps$ed1 = NULL
cps$ed2 = NULL
cps$ed3 = NULL
cps$ed4 = NULL
cps$ed5 = NULL
```

Create a relevant dummy variable for your data.

Dummy variables are = 1 under certain conditions and = zero otherwise. Creating dummy variables is straighforward:

STATA:

```
gen div = 1 if marst == 'divorced'
replace div = 0 if marst != 'divorced'
```

In R, you basically give it a statement that it will report a 1 if the statement is true and a zero if it is false:

R:

```
cps$div.dummy <- as.numeric(cps$marst == 'Divorced')
```

*Note: In R, using = or using the "arrow" or "carrot" `<-` is equivalent.

Run a regression and conduct a hypothesis test

Ordinary Least Squares (OLS) regression

Let's run the following cross-sectional regression in OLS:

$$earnweek = \beta_0 + \beta_1 age + \delta div + \varepsilon$$

where div is the dummy variable we created earlier = 1 if the individual is divorced.

STATA:

```
reg earnweek age div
```

R:

```
reg <- lm(earnweek ~ age + div, data = cps)
summary(reg)
```

Hypothesis testing

Say you wanted to do the following F-test for joint significance of all variables. This procedure can be generalized for any linear restrictions:

$H_0 : R\beta = q$
$H_a : R\beta \neq q$

Where:

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

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \delta \end{bmatrix}$$

$$q = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

STATA:

```
entercode
```

R:

```
require(gmodels)
R <- rbind(c(0,1,0),c(0,0,1))
q <- rbind(0,0)
FT <- glh.test(reg, R, q)
R
q
FT
```

But wait — do you notice anything about the F-statistic? It should exactly match the results shown in the canned regression summary output `summary(reg)` ! But I have included it here so that you can customize your tests using any linear restrictions — just modify the R and q matrices, and then re-run the test.

Make a table with the regression results

STATA:

```
entercode
```

To output the table to LaTeX, you need to first (download and) load the "xtable" package.

R:

```
require(xtable)
xtable(summary(reg))
```

Then you can copy the output into a .tex document, and compile.

If you have any questions or find problems with my code, you can e-mail me directly at **kevingoulding {at} gmail [dot] com**.

To continue on to Part 2 of our series, Difference-in-Differences esti here.

Share this:

Share

Like

Be the first to like this.

Related

Surviving Graduate Econometrics with R: Fixed Effects Estimation -- 3 of 8
In "Surviving Graduate Econometrics with R"

Surviving Graduate Econometrics with R: Difference-in-Differences Estimation -- 2 of 8
In "Surviving Graduate Econometrics with R"

Surviving Graduate Econometrics with R: Advanced Panel Data Methods -- 4 of 8
In "Surviving Graduate Econometrics with R"

Posted on May 24, 2011 at 5:57 pm in Surviving Graduate Econometrics with R | RSS feed | Reply | Trackback URL

Tags: R, STATA

2 Comments to "Surviving Graduate Econometrics with R: The Basics — 1 of 8"

Prasanna
September 10, 2012 at 11:50 pm

Hi Kevin,
Thank you so much for this blog. Its helping me get acquainted with R for econometrics, starting from the basic steps like downloading packages. I don't think anyone would explain these steps, as these are usually taken for granted, while someone like me who is not so bright would get lost.
Thanks again
Prasanna

Reply

Chris
February 20, 2015 at 10:40 am

Hi Kevin, this blog is amazing – thanks so much for putting it out there! I found the google doc you uploaded for the Diff-in-Diff estimation in the next post, but i cannot access the dataset for this exercise. Any chance you could post it also in a google doc?

Reply

Leave a Reply

Enter your comment here...

Tags

cluster-robust econometrics heteroskedasticity latex numpy parallel computing plots python r stata tex tikz

Calendar

May 2011

M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Jun »

Archives

October 2012
February 2012
July 2011
June 2011
May 2011

Blogroll

Documentation
Plugins
Suggest Ideas
Support Forum
Themes
WordPress Blog
WordPress Planet