the Tarzan

[R] + applied economics.

About | ECNS 561 | Nuts'n Bolts | Resources

« Surviving Graduate Econometrics with R: Difference-in-Differences Estimation — 2 of 8 | Surviving Graduate Econometrics with R: Advanced Panel Data Methods — 4 of 8 »

Surviving Graduate Econometrics with R: Fixed Effects Estimation — 3 of 8

The following exercise uses the `CRIME3.dta` and `MURDER.dta` panel data sets from Jeffrey Wooldridge's econometrics textbook,

> Wooldridge, Jeffrey. 2002. *Introductory Econometrics: A Modern Approach*. South-Western College Pub. 2nd Edition.

If you own the textbook, you can access the data files here.

Load and summarize the data

STATA:

```
use "C:\Users\CRIME3.dta"
des
sum
```

R:

```
require(foreign)
crime = read.dta(file="/Users/CRIME3.dta")
sumstats(crime)
as.matrix(sapply(crime,class))
```

If you haven't yet loaded in the `sumstats` function, I suggest you do – you can find the code here.

A hypothesis test

See Part 2 of this series for a primer on hypothesis testing. Here, we will do one more example of testing a hypothesis of a linear restriction. Namely, from the regression equation:

$$\log(crime_{it}) = \beta_0 + \delta_0 d78_t + \beta_1 clrprc_{i,t-1} + \beta_2 clrprc_{i,t-2} + \alpha_i + \varepsilon_{it}$$

where α_i are "district" fixed effects, and ε_{it} is a white noise error term.

We would like to test the following hypothesis:

$$H_0 : \beta_1 = \beta_2$$
$$H_a : \beta_1 \neq \beta_2$$

This can be re-written in matrix form:

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

Where:

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

$$\beta = \begin{bmatrix} \beta_0 \\ \delta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

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

STATA:

```
reg clcrime cclrprc1 cclrprc2
cclrprc1= cclrprc2
```

R:

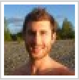
```
# Run the regression
reg1a = lm(lcrime ~ d78 + clrprc1 + clrprc2, data=crime)

# Create R and q matrices
R = rbind(c(0,0,1,-1))
q = rbind(0)

# Test the linear hypothesis beta_1  = beta_2
require(car)
linearHypothesis(reg1a,R,q)

# Equivalently, we can skip creating the R and q matrices
# and use this streamlined approach:
linearHypothesis(reg1a,"clrprc1 = clrprc2")

# Or, we can use the glhtest function in gmodels package
require(gmodels)
glh.test(reg1a, R, q)
```

First-Differenced model

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

As a review, let's go over two very similar models that take out individual-specific time-invariant heterogeneity in panel data analysis. Our example regression is:

$$Y_{it} = X_{it}\beta + \varepsilon_{it}$$

where individual and time period are denoted by the i and t subscripts, respectively.

The within estimator — a.k.a the "fixed effects" model, wherein individual dummy variables (intercept shifters) are included in the regression. All variation driving the coefficients on the other regressors is from the differences from individual specific means (= individual dummy estimates). The new model is:

$$Y_{it} = X_{it}\beta + \alpha_i + \varepsilon_{it}$$

where α_i represents the individual dummy variables.

The first-differenced model — The first-differenced model creates new variables reflecting the one-period change in values. The regression then becomes $\Delta Y_i = \Delta X_i\beta + \Delta\varepsilon_i$ where $\Delta Y_i = Y_{it} - Y_{i,t-1}$.

Note: These two models are very similar because they "strip out" / "eliminate" / "control for" the variation "between" individuals in your panel data. To do this, they use slightly different methods. The variation left over, and therefore identifying the coefficients on the other regressors, is the "within" variation — or the variation "within" individuals.

STATA:

```
reg clcrime cavgclr
outreg2 using H3_1312, word replace
```

There are two ways we can calculate the first-differenced model, given the variables included in `CRIME3.dta` . Since the data set included changed variables with a "c" prefix (e.g. "clcrime" = change in "lcrime"; "cavgclr" = change in "avgclr") we can do a simple OLS regression on the changed variables:

R:

```
reg2 =  lm(clcrime ~ cavgclr, data=crime)
summary(reg2)
```

Or, we can take a more formal approach using the `plm` package for panel data. This approach will prepare us for more advanced panel data methods.

```
require(plm)        # load panel data package

# convert the data set into a pdata.frame by identifying the
# individual ("district") and time ("year") variables in our data
crime.pd = pdata.frame(crime, index = c("district", "year"),
             drop.index = TRUE, row.names = TRUE)

# Now, we can run a regression choosing the
# first-differenced model ("fd")
reg.fd = plm(lcrime ~ avgclr, data = crime.pd, model = "fd")
summary(reg.fd)
```

Back to Pooled OLS

Let's switch over to the `MURDER.dta` data set to do some further regressions and analysis. First, we'll compute a pooled OLS model for the years 1990 and 1993:

$$mrdrte_{it} = \delta_0 + \delta_1 d93_t + \beta_1 exec_{it} + \beta_2 unem_{it} + \alpha_{it} + \varepsilon_{it}$$

By using pooled OLS, we are disregarding the term α_{it} in the regression equation above.

STATA:

```
reg mrdrte d93 exec unem if year==90|year==93
```

R:

```
crime = read.dta(file="/Users/CRIME3.dta")
sumstats(crime)

mrdrYR = subset(mrdr, year == 90 | year == 93)

reg3 = lm(mrdrte ~ d93 + exec + unem, data=mrdrYR)
summary(reg3)

# convert the data set into a pdata.frame (panel format) by identifying the
# individual ("state") and time ("year") variables in our data
require(plm)
mrdr.pd = pdata.frame(mrdrYR, index = c("state", "year"),
             drop.index = TRUE, row.names = TRUE)

# Run a pooled OLS regression - results are the same as reg3
reg3.po = plm(mrdrte ~ d93 + exec + unem, data = mrdr.pd, model = "pooling")
summary(reg3.po)
```

Another First-Differenced Model

STATA:

```
reg cmrdrte cexec cunem if year==93
```

R:

```
# We can run the regression using the variables
# provided in the data set:
reg4 = lm(cmrdrte ~ cexec + cunem, data = subset(mrdrYR,year == 93))
summary(reg4)

# Or, we can run a regression using the plm package by choosing the
# first-differenced model ("fd")
reg4.fd = plm(mrdrte ~ d93 + exec + unem, data = mrdr.pd, model = "fd")
summary(reg4.fd)

# Note: we don't need the d93 dummy anymore, so it's equivalent
# to running the regression without it:
summary(plm(mrdrte ~ exec + unem, data = mrdr.pd, model = "fd"))
```

The Fixed Effects model

Another way to account for individual-specific unobserved heterogeneity is to include a dummy variable for each individual in your sample – this is the fixed effects model. Following from the regression in the previous section, our individuals `MURDER.dta` are states (e.g. Alabama, Louisiana, California, Montana…). So, we will need to add one dummy variable for each state in our sample but exclude one to avoid perfect collinearity — the "dummy variable trap".

In STATA, if your data is set up correctly (e.g. individual in first column, time variable in second column), it is accomplished by adding `, fe` to the end of your regression command.

STATA:

```
reg mrdrte exec unem, fe
```

In R, we can add dummy variables for each state in the following way:

R:

```
reg5 = lm(mrdrte ~ exec + unem + factor(state), data=mrdr)
summary(reg5)
```

See Part 4 of this series for more attention to fixed effects models, inference testing, and comparison to random effects models.

The Breusch-Pagan test for Heteroskedasticity

The Breusch-Pagan (BP) test can be done via a LaGrange Multiplier (LM) test or F-test. We will do the LM test version; this means that only one restricted model is run.

$Var(\varepsilon_{it}|X_{it}) = \Omega\sigma^2$
$H_O : \Omega =$ identity matrix, $\Rightarrow Var(\varepsilon_{it}|X_{it}) = \sigma^2 \Rightarrow$ homoskedasticity
$H_a : \Omega \neq$ identity matrix, e.g. heteroskedasticity

First, we will run the test manually in three stages:

1. Square the residuals from the original regression $\rightarrow \hat{\varepsilon}^2$.
2. Run an auxiliary regression of $\hat{\varepsilon}^2$ on the original regressors.
3. Calculate the BP LM test statistic $= nR^2$, where R^2 is r-squared fit measure from the auxiliary regression, and n is the number of observations used in the regression.

STATA:

```
reg cmrdrte cexec cunem if year==93

predict resid , resid

gen resid2 = resid^2

reg resid2 cexec cunem if year==93
```

R:

```
# Breusch-Pagan test for heteroskedasticity

# Square the residuals
res4 = residuals(reg4)
sqres4 = res4^2

m4 = subset(mrdr,year == 93)
m4$sqres = sqres4

# Run auxiliary regression
BP = lm(sqres ~ cexec + cunem, data = m4)
BPs = summary(BP)

# Calculation of LM test statistic:
BPts = BPs$r.squared*length(BP$residuals)

# Calculate p-value from Chi-square distribution
# with 2 degrees of freedom
BPpv = 1-pchisq(BPts,df=BP$rank-1)

# The following code uses a 5% s
if (BPpv < 0.05) {
    cat("We reject the null hypc
    "BP = ",BPts,"\n","p-value =
} else {
    cat("We fail to reject the r                              sticity.\n",
    "BP = ",BPts,"\n","p-value =
}
```

Now, let's compare the results obtained ab   d in the R `lmtest` package:

```
require(lmtest)
bptest(reg4)
```

I Hope your results are exactly the same a   manually — they should be!

White's Test for Heteroskedas

White's test for heteroskedasticity is similar the Breusch-Pagan (BP) test, however the auxiliary regression includes all multiplicative combinations of regressors. Because of this it can be quite bulky and finding heteroskedasticity may simply imply model mispecification. The null hypothesis is homoskedasticity (same as BP).

So, here we will run a special case of the White test using the fitted values of the original regression:

$$\hat{\varepsilon}_{it}^2 = \hat{Y}_{it} + \hat{Y}_{it}^2$$

STATA:

```
reg cmrdrte cexec cunem if year==93

gen resid2 = resid^2

predict yhat

gen yhat2 = yhat^2

reg resid2 yhat yhat2 if year==93
```

R:

```
# White's test for heteroskedasticity: A Special Case

# Collect fitted values and squared f.v. from your regression
yhat = reg4$fitted.values
yhat2 = yhat^2
m4 = NULL           # clears data previously in m4

# create a new data frame with the three variables of interest
m4 = data.frame(cbind(sqres4,yhat,yhat2))

# Run auxiliary regression
WH = lm(sqres4 ~ yhat + yhat2, data = m4)
WHs = summary(BP)

# Calculation of LM test statistic:
WHts = WHs$r.squared*length(WH$residuals)

# Calculate p-value from Chi-square distribution
# with 2 degrees of freedom
WHpv = 1-pchisq(WHts,df=WH$rank-1)

# The following code uses a 5% significance level
if (WHpv < 0.05) {
    cat("We reject the null hypothesis of homoskedasticity.\n",
    "BP = ",WHts,"\n","p-value = ",WHpv)
} else {
    cat("We fail to reject the null hypothesis; implying homoskedasticity.\n",
    "BP = ",WHts,"\n","p-value = ",WHpv)
}
```

Heteroskedasticity-Robust Standard Srrors

If heteroskedasticity is present in our data sample, using OLS will be inefficient. See this post for details behind calculating heteroskedasticity-robust and cluster-robust standard errors.

To continue on to Part 4 of our series, Advanced Panel Data Methods, click here.

Share this:

Share

★ Like

One blogger likes this.

Related

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"

Surviving Graduate Econometrics with R: The Basics -- 1 of 8
In "Surviving Graduate Econometrics with R"

Posted on May 25, 2011 at 4:14 pm in Surviving Graduate Econometrics with R | RSS feed | Reply | Trackback URL

Tags: R, STATA

One Comment to "Surviving Graduate Econometrics with R: Fixed Effects Estimation — 3 of 8"

Bill Zhou
March 16, 2013 at 10:25 pm

There's a little problem in the part of First-Differenced Model.

First difference method eliminate the constant, so in R, it looks more like in this way:

reg.fd = plm(lcrime ~ avgclr-1, data = crime.pd, model = "fd")

Also, a little tip for Stata running first difference model: no need to calculate the difference, try this one:

reg D.(lcrime avgclr), noconstant

Then you'll get the result of first-differenced fixed effects model.

Thanks for other parts, really helpful!

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