« TikZ diagrams for economists: A price ceiling | The Chow test in R: A case study of Yellowstone's Old Faithful Geyser »

Calculate OLS regression manually using matrix algebra in R

The following code will attempt to replicate the results of the `lm()` function in R. For this exercise, we will be using a cross sectional data set provided by R called "women", that has height and weight data for 15 individuals.

The OLS regression equation:

$$Y = X\beta + \varepsilon$$

where $\varepsilon =$ a white noise error term. For this example $Y =$ weight, and $X =$ height. $\beta =$ the marginal impact a one unit change in height has on weight.

```
## This is the OLS regression we will manually calculate:
reg = lm(weight ~ height, data=women)
summary(reg)
```

Recall that the following matrix equation is used to calculate the vector of estimated coefficients $\hat{\beta}$ of an OLS regression:

$$\hat{\beta} = (X'X)^{-1}X'Y$$

where $X =$ the matrix of regressor data (the first column is all 1's for the intercept), and $Y =$ the vector of the dependent variable data.

Matrix operators in R

- `as.matrix()` coerces an object into the matrix class.
- `t()` transposes a matrix.
- `%*%` is the operator for matrix multiplication.
- `solve()` takes the inverse of a matrix. Note, the matrix must be invertible.

For a more complete introduction to doing matrix operations in R, check out this page.

Back to OLS

The following code calculates the 2 x 1 matrix of coefficients, $\hat{\beta}$:

```
## Create X and Y matrices for this specific regression
X = as.matrix(cbind(1,women$height))
Y = as.matrix(women$weight)

## Choose beta-hat to minimize the sum of squared residuals
## resulting in matrix of estimated coefficients:
bh = round(solve(t(X)%*%X)%*%t(X)%*%Y, digits=2)

## Label and organize results into a data frame
beta.hat = as.data.frame(cbind(c("Intercept","Height"),bh))
names(beta.hat) = c("Coeff.","Est")
beta.hat
```

Calculating Standard Errors

To calculate the standard errors, you must first calculate the variance-covariance (VCV) matrix, as follows:

$$Var(\hat{\beta}|X) = \frac{1}{n-k}\hat{\varepsilon}'\hat{\varepsilon}(X'X)^{-1}$$

The VCV matrix will be a square k x k matrix. Standard errors for the estimated coefficients $\hat{\beta}$ are found by taking the square root of the diagonal elements of the VCV matrix.

```
## Calculate vector of residuals
res = as.matrix(women$weight-bh[1]-bh[2]*women$height)

## Define n and k parameters
n = nrow(women)
k = ncol(X)

## Calculate Variance-Covariance Matrix
VCV = 1/(n-k) * as.numeric(t(res)%*%res) * solve(t(X)%*%X)

## Standard errors of the estimated coefficients
StdErr = sqrt(diag(VCV))

## Calculate p-value for a t-test of coefficient significance
P.Value = rbind(2*pt(abs(bh[1]/StdErr[1]), df=n-k,lower.tail= FALSE),
2*pt(abs(bh[2]/StdErr[2]), df=n-k,lower.tail= FALSE))

## concatenate into a single data.frame
beta.hat = cbind(beta.hat,StdErr,P.Value)
beta.hat
```

Search this blog
Search...

Contributors

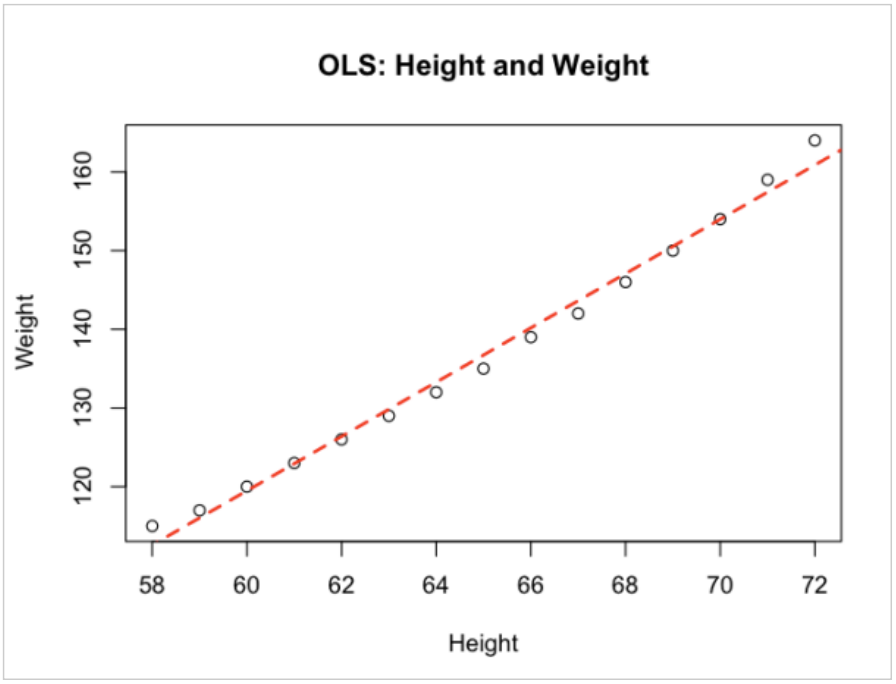
Kevin Goulding

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

A Scatterplot with OLS line

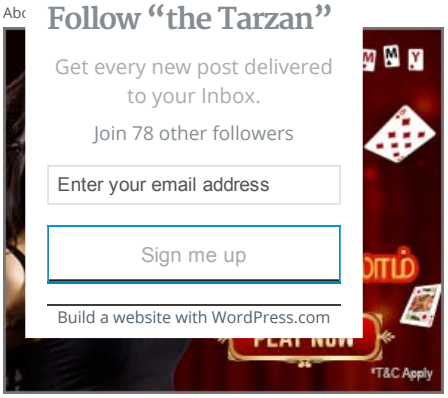


Women's height vs. weight using plot() and abline() functions in R.

```
## Plot results
plot(women$height,women$weight, xlab = "Height", ylab = "Weight",
                main = "OLS: Height and Weight")
abline(a = bh[1], b = bh[2], col = 'red', lwd = 2, lty="dashed")
```

Now you can check the results above using the canned `lm()` function:

```
summary(lm(weight ~ height, data = women))
```

Share this:

Share

Like

Be the first to like this.

Related

Calculate an OLS regression using matrices in Python using Numpy
In "Econometrics"

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

Clustered Standard Errors in R
In "Econometrics with R"

Posted on June 13, 2011 at 8:36 am in Econometrics with R | RSS feed | Reply | Trackback URL

Tags: R

7 Comments to "Calculate OLS regression manually using matrix algebra in R"

Vicente Sanchis

August 8, 2011 at 12:37 pm

Hi.

Thank you very much for this interesting blog about econometrics topics. I liked this contiiribution but I think ther is a mistake when you calculate the p-values of standard errors, because I think the code would be:

P.Value = rbind(2*pt(abs(bh[1]/StdErr[1]), df=n-k,lower.tail= FALSE),2*pt(abs(bh[2]/StdErr[2]), df=n-k,lower.tail= FALSE))

Reply

Kevin Goulding
August 15, 2011 at 12:32 pm

Hi Vicente – Great catch; I have updated the code to reflect your comment. Thanks for reading!

Reply

Kyle
January 31, 2012 at 6:29 am

Hi Kevin, this is really great! Thanks for making it available. I'm new to R and having my first crack trying to build the OLS model. Just curious how one might extend this to the multivariate case? Alternatively, might there be a way to examine the 'souce code' for the lm() function. I say 'souce code' because I'm thinking in Stata mode…I want to see what the lm() does behind the scenes!

Reply

Kevin Goulding
January 31, 2012 at 3:10 pm

Kyle – that's the beauty of matrix notation, it's exactly the same for the multivariate case: (X'X)^(-1)X'Y. Note that you could think of the example as a 2-variable regression, with one regressor that doesn't vary (the intercept). So the X matrix has a column of 1's and then a column with the data women$height. See the code: X = as.matrix(cbind(1,women$height)).

For additional regressors, just extend that line of code by appending new columns of data: X = as.matrix(cbind(1,women$height, newdata$x2, newdata$x3, …)) and the rest should work as shown.

To see the inner workings of any function in R, just execute the function without the parentheses, e.g. lm. I don't think it will be too much help, though, because R uses a "QR decomposition" to do OLS, which basically is a different approach that is more computationally efficient. Cheers-

Reply

Kyle
February 1, 2012 at 12:54 am

Fantastic!!! I've done it! Man, the learning curve is huge but well worth it. Thanks very much!

And yes, it turns out the source code is useless for these purposes! Cheers, Kyle

Reply

Fred
November 29, 2013 at 8:40 am

Dear Kevin,
great code! I wonder what n and k would be in the case of a fixed-effects panel data model? I have 744 observations in 24 countries over 31 years using 6 continuous variables. Not sure how to compute the p-values in such a case, given the standard errors and beta.hat.

Reply

Metin CALISKAN
January 3, 2014 at 11:03 am

Hello, this post is really great.
This makes me understand what's going on in detail.
I would like to know if it's possible to make a similar post for a logistic regression.
Thanks.

Reply

Leave a Reply

Enter your comment here...

Tags

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

Calendar

June 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			

« May Jul »

Archives

October 2012
February 2012
July 2011
June 2011
May 2011

Blogroll

Documentation
Plugins
Suggest Ideas
Support Forum
Themes
WordPress Blog
WordPress Planet