Generating OLS Results Manually via R

Sujan Bandyopadhyay

Statistical softwares and packages have made it extremely easy for people to run regression analyses. Packages like lm in R or the reg command on STATA give quick and well compiled results. With this ease, however, people often don't know or forget how to actually conduct these analyses manually. In this article, we manually recreate regression results created via the lm package in R.

Using the mtcars data set, which is pre-loaded into R, we produce regression results using the lm package. We randomly select *mpg* as the independent variable, and *disp*, *hp*, and *wt* as the independent variables.

$$mpg = \beta_0 + \beta_1 disp + \beta_2 hp + \beta_3 wt + \epsilon \tag{1}$$

Find the R Output below:

\- -

```
> # Loading the Data
> data(mtcars)
># LM Package for OLS
> lm_package <- lm(mtcars$mpg ~ mtcars$disp + mtcars$hp + mtcars$wt)
> #show results
> summary(lm_package)

Call:
lm(formula = mtcars$mpg ~ mtcars$disp + mtcars$hp + mtcars$wt)

Residuals:
Min       1Q  Median      3Q     Max
-3.891  -1.640  -0.172  1.061   5.861

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 37.105505   2.110815  17.579  < 2e-16 ***
mtcars$disp -0.000937   0.010350  -0.091  0.92851
mtcars$hp   -0.031157   0.011436  -2.724  0.01097 *
mtcars$wt   -3.800891   1.066191  -3.565  0.00133 **
```

```
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 2.639 on 28 degrees of freedom
Multiple R-squared: 0.8268, Adjusted R-squared: 0.8083
F-statistic: 44.57 on 3 and 28 DF, p-value: 8.65e-11
```

– –

We start with getting the correct β coefficients for the independent variables.

Since there are 32 observations, and 3 independent variables, we will have the following model.

$$Y_{32\times 1} = X_{32\times 4}\beta_{4\times 1} + \epsilon_{32\times 1} \tag{2}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_{32} \end{bmatrix} = \begin{bmatrix} 1 & disp_1 & hp_1 & wt_1 \\ 1 & disp_2 & hp_2 & wt_2 \\ . & . & . & . \\ . & . & . & . \\ 1 & disp_{32} & hp_{32} & wt_{32} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ \epsilon_{32} \end{bmatrix}$$

We know that the formula to calculate the vector or β coefficients is as follows.

$$\beta = (X'X)^{-1}(X'Y) \tag{3}$$

For the vector X, we must remember to add the vector of ones as there is a constant in our model. Using the specification in equation (2), we can define the the matrices of X, and Y and use equation(3) to generate the vector of β.

Find the R Output below:

– –

```
>#Independent Variable (Y)
> Y <- matrix(mtcars$mpg)
> ind <- "mpg"

#Dependent variables (X)
> v <- mtcars$disp
> v <- rep (1, length(mtcars$disp))
> X <- matrix(v)
> X <- cbind (X,mtcars$disp , mtcars$hp, mtcars$wt )
> dep <- c("constant", "disp", "hp", "wt")


> #Matrix Operations
> #Transpose of X
> X_t <-t(X)
```

```
> #Generating X'Y
> first <- X_t%*%Y
> #Generating (X'X)
> second <- X_t %*% X
> #Inverse ((X'X)^-1)
> second <- solve(second)
> # Beta Vector = (X'X)^-1 (X'Y)
> manual <- second_2 %*% first
> manual
mpg
constant  37.1055052690
disp      -0.0009370091
hp        -0.0311565508
wt        -3.8008905826
```

So here we see that we have exactly reproduced the vector of β. The next step is to try and recreate some of the measures, starting with the R Squared measure.

$$TSS = RSS + ESS \tag{4}$$

$$TSS = \sum(y_i - \bar{y})^2 \tag{5}$$

$$ESS = \sum(y_i - \hat{y})^2 \tag{6}$$

$$R^2 = 1 - \frac{ESS}{TSS} \tag{7}$$

Find the R output below:

```
> #Calculating R Squared
>
> #Mean of Y (Y_bar)
>
> TSS_i <- (Y-Y_bar)^2
> TSS <- sum (TSS_i)
>
> #Error Sum of Squares (ESS)
>
> # Calculating the Predicted value of Y (Y_hat)
>
> Beta_1 <- manual[1]
> Beta_2 <- manual [2]
> Beta_3 <- manual [3]
```

```
> Beta_4 <- manual [4]
>
> Y_hat <- Beta_1 + (Beta_2 * mtcars$disp) + (Beta_3*mtcars$hp) + (Beta_4*mtcars$wt)
>
> ESS_i <- (Y - Y_hat)^2
> ESS <- sum (ESS_i)
>
> # Rsquared
>
> R_squared <- 1 - ESS/TSS
>
> R_squared
[1] 0.8268361
>
> #Summary Stats for Residuals
> summary(Y- Y_hat)
mpg
Min.    :-3.891
1st Qu.:-1.640
Median :-0.172
Mean   : 0.000
3rd Qu.: 1.061
Max.   : 5.861
```

– –

We move on to calculating the F Stat.

$$MSM = \frac{RSS}{DFM} \tag{8}$$

$$MSM = p - 1 \tag{9}$$

$$MSE = \frac{ESS}{DFE} \tag{10}$$

$$DFE = n - p \tag{11}$$

$$f = \frac{MSM}{MSE} \tag{12}$$

Find the R output below.

– –

```
> #Calculating F Stat
>
> # Regression Sum of Squares
> RSS = TSS - ESS
>
```

```
> # Regression Degrees of Freedom
> DFM = 4 - 1
>
> #Mean Squares of Model
> MSM = RSS/DFM
>
> # Error Degrees of Freedom
> DFE = 32 - 4
>
> #Mean Square Error
> MSE = ESS/ DFE
>
> #F Stata
> f = MSM/MSE
> f
[1] 44.56552
```

- -

Finally, we calculate the standard errors of the coefficients.

$$V[\hat{\beta}] = \sigma^2 (X'X)^{-1} \tag{13}$$

$$\hat{\sigma^2} = \frac{\epsilon'\epsilon}{n - p} \tag{14}$$

Find the R Output Below:

- -

```
> # Standard Errors
>
> # Residuals
> resid <- Y - X %*% manual
>
> # Estimating sigma_square (sigma_hat_square)
> sigma_hat_square <- (t(resid) %*% resid)/(32-4)
>
> # Variance Covariance Matric of Beta_hat
> vcov_beta <- c(sigma_hat_square) * solve(t(X) %*% X)
>
> # Standard Errors
> se <- sqrt(diag(vcov_beta))
> se
constant        disp          hp          wt
2.11081525  0.01034974  0.01143579  1.06619064
```

- -

Thus, we have manually reproduced all of the key statistics that had been produced by the lm package, namely - the coefficients, the R Squared statistic, the F Statistic, and the standard errors of the coefficients.

It is obvious that process of computing these statistics manually is time consuming and inefficient. The available software packages are much more efficient. Having said that, it would be useful to conduct this exercise from time to time to refresh one's theoretical knowledge of regression analysis.

Finally, if someone is mechanically using software packages without understanding the underlying theory behind regression analysis, then conducting such an exercise is highly recommend.