Introduction to Ordinary Least Squares Regression

Irfan Kanat

August 9, 2017

Ordinary Least Squares (OLS) forms the foundation of estimation in a number of fields ranging from econometrics to electronics engineering. As such, this is very often the go to model for a first stab at modeling.

In this activity, I will try to introduce the basic idea of OLS without going into details. As always, I will remain practical and only briefly touch the theory if needed.

We will use the motor trends dataset we used in Module 2, activity 1. Go take a look if you can't remember.

```
data(mtcars) # Loading mtcars dataset from datasets package
```

What is the Big Idea?

Let us say you are interested in fuel efficiency. You would be interested in the measure mpg. In this instance mpg is the *dependent variable, y, variable of interest, or outcome variable . . .*

What can you tell us about mpg with what we covered in previous weeks? What kind of insight can you provide us?

As we discussed, without knowing anything else you can tell us the average mpg. Which is the simplest model you can fit to a variable.

As you get your hands on more data you can run correlations. For example you can tell me that there is a negative relation between the engine volume and fuel efficiency. Larger the engine the less efficient the machine will be. Now we are getting somewhere.

How about making predictions based on what we know about fuel efficiency and engine volume? That would be the domain of regression.

We can even go a step further to look at the joint effect of a number of variables on the dependent variable. We call these other variables *independent variables, x, or input variables . . .* Basically we estimate how these independent variables act together to shape the independent variable.

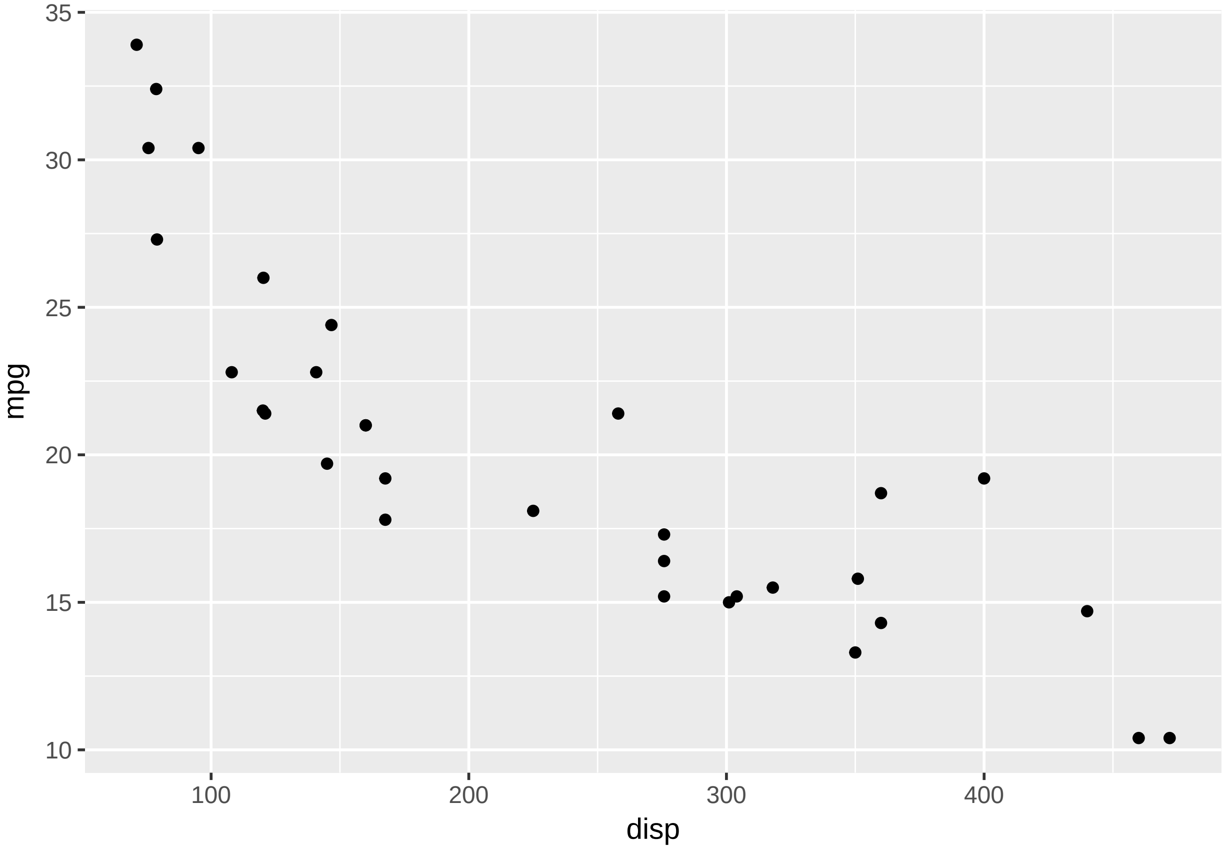
A Simple Example

Let us say we want to run a very simple regression with a single independent variable. We want to estimate y (mpg) based on x (disp).

Visual Representation

Let us start with a graphical representation (visualization is all the craze these days).

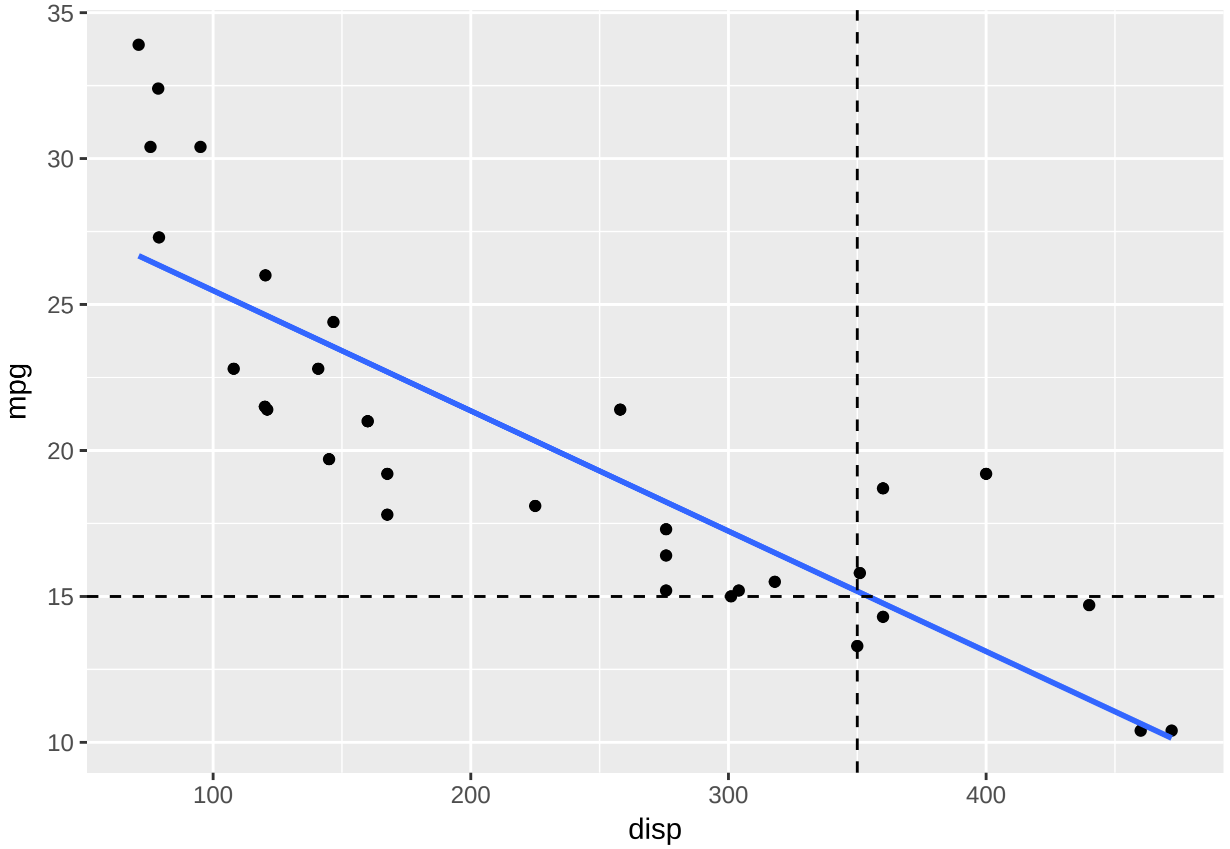
```
qplot(x = disp, y = mpg, data = mtcars)
```



In this plot you can see the negative correlation, but if you wanted to establish a linear relation between disp and mpg what can you do?

Easiest solution would be to run a line as close to all the points as possible. A line like the blue one below:

```r
qplot(x = disp, y = mpg, data = mtcars) +
  geom_smooth(method = "lm", se = FALSE) + # Fit regression line
  # add guide lines
  geom_hline(yintercept = 15, lty = 2) + geom_vline(xintercept = 350, lty = 2)
```



Now if somebody says "my car's displacement (disp) is 350", you can look at where the blue line crosses 350 on x axis and guess that the guy's car will have about 15 miles per galon (look at the dashed lines). Now you are cooking with gas. You are making a prediction based on a model.

In placing the line, we try to minimize the total distance between all the dots and the line (hence the model is called least squares). Ask me during virtual office hours if you are interested.

Quantitative Estimation

Now let us move beyond the visual representation of this simple example and let us look at how this model can be fit in R and how to interpret the results.

The formula for this model would be as follows:

$y = \beta_0 + \beta_1 \times X_1 + \epsilon$

Let us go over the terms:

- y is the dependent variable.
- x_1 is the independent variable.
- β_1 is the coefficient of x_1.[1]
- β_0 is the intercept.
- ϵ is the error in our estimation.

What this tells us is y (mpg) is determined by x_1 (displacement) and some random error (ϵ). So you take x_1, multiply it by β_1, add β_0 and you will get your best estimate of y. You do not touch the ϵ as epsilon is unknown, it is just allowed for in the model.

[1] In a proper statistics text book β_1 would be used to refer to the true value of the coefficient and b_1 to refer to the estimate of the coefficient. In this material, I won't make the distinction as I believe it can get confusing for students.

Let us fit an ordinary least squares regression model to estimate gas milage based on displacement.

We use lm() to fit these kinds of **l**inear **m**odels. Please skim the manual page for lm (?lm) before virtual office hours.

```
mtcars_lm_0 <- lm(mpg ~ disp, data = mtcars)
```

Let us see what we have done thus far:

1. mtcars_lm_0 <-: assigns the estimation results into a variable named mtcars_lm_0[2].
2. lm(mpg~disp, data=mtcars): calls lm function with two parameters
   - mpg~disp: is the formula specification, left of ~ is the dependent variable, and right of ~ is the independent variables.
   - data=mtcars: name of the dataset to draw the variables from.

So the model is estimated. How do we see the estimation results? We use the summary function (you should be familiar with it from descriptive statistics activity).

```
summary(mtcars_lm_0)
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8922 -2.2022 -0.9631  1.6272  7.2305
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.599855   1.229720  24.070  < 2e-16 ***
## disp        -0.041215   0.004712  -8.747 9.38e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.251 on 30 degrees of freedom
## Multiple R-squared:  0.7183, Adjusted R-squared:  0.709
## F-statistic: 76.51 on 1 and 30 DF,  p-value: 9.38e-10
```

The model results start with a reiteration of function call. This is helpful in case you forget the parameters of a model.

Residuals is a summary of the random error (ϵ) the model left after estimation. By definition the mean of residuals will be 0. Ideally you would like the variance of residuals to be small.

The part most people are most excited about is the Coefficients. This part tells you if the relationship is significant or not and the magnitude of the relationship. We will talk about this in greater detail later.

Finally the bottom three lines are the model fit statistics, how good is the model at explaining data.

[2] I often save model estimates in a certain way. mtcars indicates on which dataset this model was estimated. lm inicates the type of model that was fit. The number 0 means this is the first model in a series of models of increasing complexity. I recommend you determine a naming standard that suits your purposes.

How to Evaluate Model Estimation Results

Model Fit

While people get overly excited and jump to coefficients, I recommend starting with how good the model is fit for the data.

F Statistic

This is an overall evaluation of the model compared to using just the mean value (sometimess referred to as null model). If your model is good, it needs to be at least better than just guessing the mean value anytime somebody asks you to guess the mpg of a car.

Rule of thumb is look at the p value. You want p value to be really small[^3] (remember t tests?). So any p value below (.05) means your model is statistically significantly different than the null model.

R^2

R^2 (R-squared) is sometimes referred to the explanatory power of the model. Basically this is the percentage of variance explained by the model. The higher the better. You will see that sometimes OLS models with 20% or 30% explained being accepted as good models.

In our model the total ammount of variance was 100, just having displacement explained 71.8 units of variance. So the model is very good.

One trick with R^2 is that it is easy to inflate this number by just adding variables to the model. So a model that explains 30% of the variance with a single variable is obviously not the same as a model that explains 30% of variance with 30 variables. For these purposes we use adjusted R^2. Adjusted R^2 penalizes the number of variables in model. So adding variables that contribute very little explanatory power to the model will actually decrease adjusted R^2 value rather than increasing it. It is a good way to keep people honest.

Another thing to know is, while it is easy to calculate R^2 for OLS regression, other types of regression may not have an exact equivalent.

Residual Standard Error

Residual is the difference between observed value and estimated value. So let us say that you estimated that the car would have an mpg of 15 and the actual mpg was 17. This means there remains a residual of 2.

Ideally you want the residual to be as small as possible, but in all likelihood you can not estimate every observation perfectly (just look at that plot with regression line, there are no actual observations on that line). This means when you estimate all observations, you will end up with a range of residual values. The smaller the variability on residuals the more precise your estimations.

There is no simple rule of thumb to use residual standard error. It is needed if you want to diagnose problems with the model. So in all likelihood you won't immediately use this.

Coefficients

Once you determined that your model has some merit and is worth interpreting, coefficients is the place to look.

First thing you need to do is to look at the p values. If a variable significantly contributes to the model the p value will be small. Summary function places stars at the end of each row to indicate statistical significance. Look at the number of stars at the end of each line. In our example both the intercept and the displacement is statistically significant. Meaning they should be used in estimation.

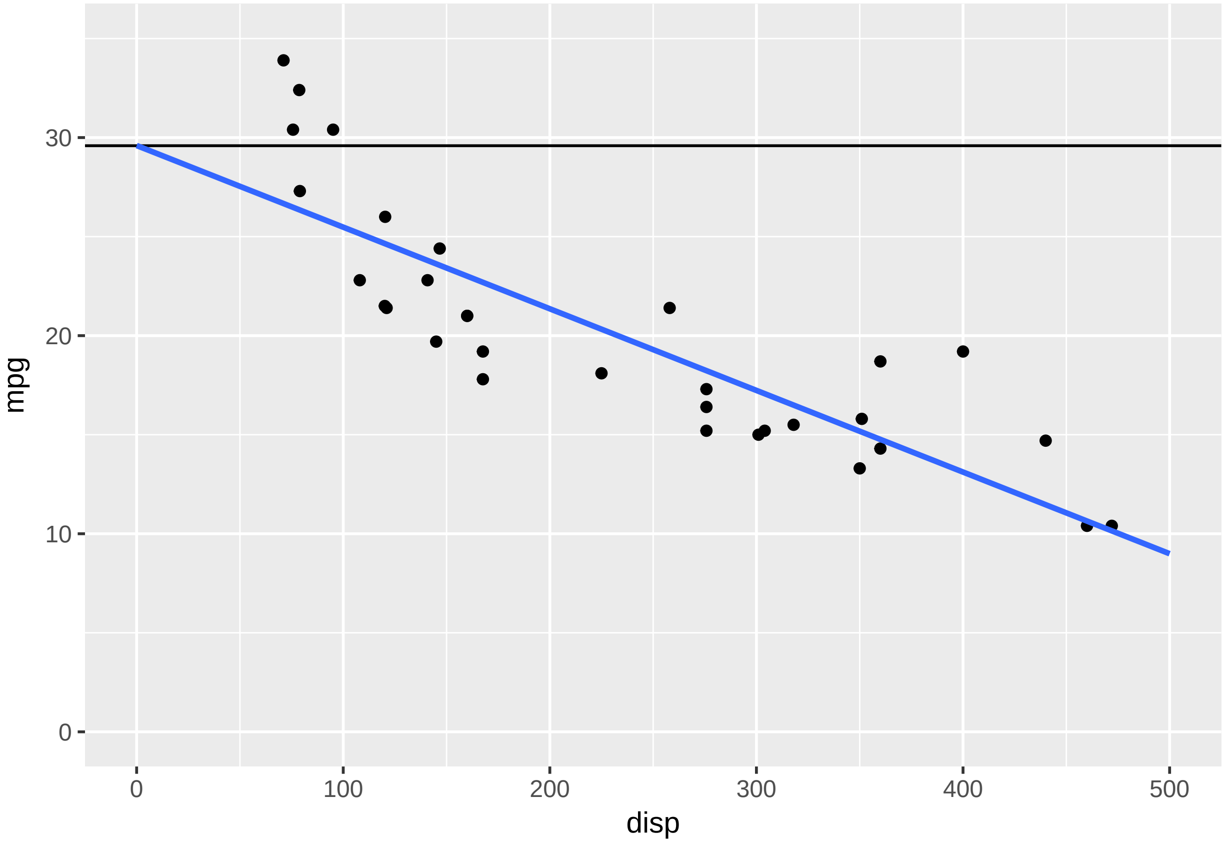
Second thing we need to at look is the coefficient estimate itself. Let us refresh our memory about what this model is (I will replace x and y with disp and mpg for demonstration purposes here).

$mpg = \beta_0 + \beta_1 \times disp + \epsilon$

In the model summary summary, β_0 is the intercept and β_1 is the coefficient of disp. We will talk about how to use these numbers below.

Going back to the visual representation:

```
ggplot(aes(x = disp, y = mpg), data = mtcars) +
  geom_point() +
  xlim(0, 500) + ylim(0, 35) +
  geom_hline(yintercept = 29.59) +
  geom_smooth(method = "lm", se = F, fullrange = T)
```



Look at that black line at 29.6, that is the intercept. You can consider that the value of y where x is equal to 0. That is the point where the regression line crosses the y axis.

The slope of the blue line is the coefficient estimate of displacement. The coefficient estimate for disp is -0.04. That means for every unit of displacement, mpg is going to shrink by 0.04.

Placing coefficients into the formula, the formula now reads:

$mpg = 29.6 - 0.04 \times disp$

Let us plug some disp numbers in place to estimate the mpg.

For a car with a displacement of 100, the mpg estimate will be:

$mpg = 29.6 - 0.04 \times 100$
$mpg = 25.6$

Here is a simple exercise for you, try to estimate the mpg for a car with (A) disp = 150, (B) disp = 400.

Multiple Regressions

The simple idea was to model a relation between a single dependent variable and a single independent variable. What if we want to integrate more than one independent variable? To see the joint effect of multiple variables?

We can definitely fit more variables into the model. The new formula would look something like this:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$

Let us assume that we are interested in the effect of displacement (disp), and weight (wt) on the fuel efficiency. The formula would be:

$mpg = \beta_0 + \beta_1 \times disp + \beta_2 \times wt + \epsilon$

And the R code for the model with the three variables we have in mind would be:

```r
mtcars_lm_1 <- lm(mpg ~ disp + wt, data = mtcars)
```

The only difference is in the formula parameter:

`mpg~disp+wt`

I use + signs to add variables to the model.

Interpreting Results

```r
summary(mtcars_lm_1)
```

```
##
## Call:
## lm(formula = mpg ~ disp + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4087 -2.3243 -0.7683  1.7721  6.3484
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.96055    2.16454  16.151 4.91e-16 ***
## disp        -0.01773    0.00919  -1.929  0.06362 .
## wt          -3.35082    1.16413  -2.878  0.00743 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 29 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7658
## F-statistic: 51.69 on 2 and 29 DF,  p-value: 2.744e-10
```

F Statistic indicates that the model is significant. The correct way to report it is as follows: F(2, 29) = 51.69, $p < 0.001$.

R^2 Indicates the model explains 78% of variance in fuel efficiency. You can say that this model is an improvement over mtcars_lm_0 as it explains more of the variance.

We note that displacement is no longer statistically significant. Now remember, most probably there are (hopefully small) correlations between independent variables, meaning coefficient estimates will depend on what variables are in the model. So don't be surprised when you realize the coefficient estimates and their significances change depending on what variables are included. This means the variance explained by displacement, after controlling for car's weight is not statistically significant. The correct way to report this is to say "effect of displacement is fully mediated by car's weight". **We need not report the displacement in this model as it is not significant. In fact we need to ignore the displacement in our estimates.**

We see the weight is significant. The correct way to report this is as follows: "Weight was a significant predictor of fuel efficiency ($\beta_2 = -3.35, p < 0.01$)".

As you have seen in the example of displacement, the independent variables included in the model determine the coefficients and significance of each other. One way to think about these coefficients is to assume that these effects are when everything else is held constant. That is why when we use these coefficients, we say things like: "All other variables held constant, a unit change in weight (x_2) results in -3.35 (β_2) unit change in mpg (y)."

A word of warning, do not compare the estimates directly unless the variables are standardized or are in the same range. Think back to our zillow dataset. Assume that x_1 is rent in dollars and x_2 is number of bedrooms. β_1 will be the effect of $1 change on outcome variable ranging possibly from zero to several thousand, whereas β_2 will be the number of rooms ranging from one to possibly a dozen at most. These coefficients won't be directly comparable. If you want to compare coefficients and say one is greater than the other, standardize the independent variables as discussed in Module 2, Learning Activity 5 (Transformations).

Here is another exercise for you: Fit a model with number of carburateurs and displacement; (A) Interpret the results, (B) estimate mpg for a car with 2 carburateurs and a displacement of 200.

Model Comparison

Sometimes you will add or remove variables from a model. Is there a way beyond simply saying R^2 is higher or lower to compare models? Yes there is. If the models are nested you can compare the models with an ANOVA. Nested here means the models are fit with the same data, AND one model is simply missing a few variables.

So in our example mtcars_lm_0 and mtcars_lm_1 are nested as mtcars_lm_1 can be obtained by adding variable wt to model mtcars_lm_0.

Let us see how to compare models:

```r
anova(mtcars_lm_0, mtcars_lm_1)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ disp
## Model 2: mpg ~ disp + wt
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     30 317.16
## 2     29 246.68  1    70.476 8.2852 0.007431 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Degrees of freedom (df) tells us there is a 1 variable difference between the two models. The residual sums of squares (RSS - error) is smaller in second model. This difference in RSS is statistically significant since the p value being smaller than 0.01. This means the second model is statistically significantly better than the first.

What is Next?

In the next learning activity, we will be learning about interaction effects.

Solutions to Exercises

1 - Estimate the mpg for a car with (A) disp = 150, (B) disp = 400.

A - Where disp = 150

```
disp <- 150
# Manual calculation
29.6 - 0.04 * disp
```

```
## [1] 23.6
```

B - Where disp = 400

```
disp <- 400
# Manual calculation
29.6 - 0.04 * disp
```

```
## [1] 13.6
```

2 - Fit a model with number of carburateurs and displacement.

```
mtcars_lm_2 <- lm(mpg ~ disp + carb, data = mtcars)
```

A - Interpret the results:

```
summary(mtcars_lm_2)
```

```
##
## Call:
## lm(formula = mpg ~ disp + carb, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3379 -2.0849 -0.3448  1.5118  6.2836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 31.152710   1.263620  24.654  < 2e-16 ***
## disp        -0.036296   0.004676  -7.762 1.47e-08 ***
## carb        -0.955677   0.358789  -2.664   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.964 on 29 degrees of freedom
## Multiple R-squared:  0.7737, Adjusted R-squared:  0.7581
## F-statistic: 49.58 on 2 and 29 DF,  p-value: 4.393e-10
```

The model significantly explains 77% of the variance ($R^2 = .7737$, F(2, 29) = 49.58, $p < 0.001$).

We see that both displacement and number of carburateurs are statistically significant in explaining fuel efficiency.

All other variables held constant, a unit change in number of carburateurs decreases the gas milage by .96.

B - Estimate mpg for a car with 2 carburateurs and a displacement of 200.

$mpg = 31.15 - .036 \times disp - .956 \times carb$

```
disp <- 200
carb <- 2
```

```
# Using the predict function instead of manual calculation
predict(object = mtcars_lm_2, newdata = data.frame(disp, carb))
```

```
##        1
## 21.98219
```

House Keeping

Forgive our dust, here I export some models so we can import them back later.

```
save(list = ls(pattern = "mtcars_lm_*"), file = "data/mtcars_fit")
```