R news and tutorials contributed by hundreds of R bloggers

Welcome!

Here you will find daily **news and tutorials about R**, contributed by hundreds of bloggers. There are many ways to **follow us -**

By e-mail:

On Facebook:

If you are an R blogger yourself you are invited to add your own R content feed to this site (**Non-English** R bloggers should add themselves- here)

Jobs for R-users

- (Junior) Data Analyst – Berlin
- Associate Researcher (Data Analytics)
- Lead Data Scientist @ Stabio, Ticino, Switzerland
- Senior Scientist, Translational Informatics
- Senior Scientist, Translational Informatics @

Recent Posts

- Gold-Mining Week 9 (2019)
- A brief primer on Variational Inference
- 81st TokyoR Meetup Roundup: A Special Session in {Shiny}!
- The Mysterious Case of the Ghost Interaction
- Non-randomly missing data is hard, or why weights won't solve your survey problems and you need to think generatively
- Enlarging the eBook supply
- What's new in DALEX v 0.4.9?
- Extracting basic Plots from Novels: Dracula is a Man in a Hole
- (Re)introducing skimr v2 – A year in the life of an open source R project
- An Amazon SDK for R!?
- Sept 2019: "Top 40" New R Packages
- Mocking is catching
- Any one interested in a function to quickly generate data with many predictors?
- Dogs of New York
- Spelunking macOS 'ScreenTime' App Usage with R

Other sites

- Jobs for R-users
- SAS blogs

Ordinary Least Squares (OLS) Linear Regression in R

July 4, 2017
By S. Richter-Walsh

Ordinary Least Squares (OLS) linear regression is a statistical technique used for the analysis and modelling of linear relationships between a response variable and one or more predictor variables. If the relationship between two variables appears to be linear, then a straight line can be fit to the data in order to model the relationship. The linear equation (or equation for a straight line) for a bivariate regression takes the following form:

$$y = mx + c$$

where y is the response (dependent) variable, m is the gradient (slope), x is the predictor (independent) variable, and c is the intercept. The modelling application of OLS linear regression allows one to predict the value of the response variable for varying inputs of the predictor variable given the slope and intercept coefficients of the line of best fit.

The line of best fit is calculated in R using the lm() function which outputs the slope and intercept coefficients. The slope and intercept can also be calculated from five summary statistics: the standard deviations of x and y, the means of x and y, and the ***Pearson correlation coefficient*** between x and y variables.

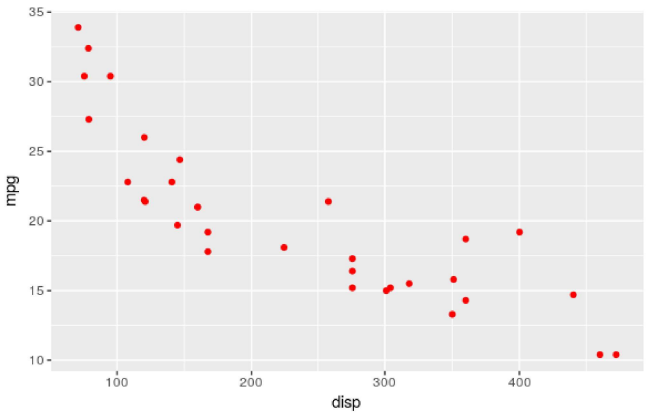
```
slope <- cor(x, y) * (sd(y) / sd(x))
intercept <- mean(y) - (slope * mean(x))
```

The scatterplot is the best way to assess linearity between two numeric variables. From a scatterplot, the strength, direction and form of the relationship can be identified. To carry out a linear regression in R, one needs only the data they are working with and the lm() and predict() base R functions. In this brief tutorial, two packages are used which are not part of base R. They are ***dplyr*** and ***ggplot2***.

The built-in mtcars dataset in R is used to visualise the bivariate relationship between fuel efficiency (mpg) and engine displacement (disp).

```
library(dplyr)
library(ggplot2)

mtcars %>%
 ggplot(aes(x = disp, y = mpg)) +
 geom_point(colour = "red")
```



Upon visual inspection, the relationship appears to be linear, has a negative direction, and looks to be moderately strong. The strength of the relationship can be quantified using the Pearson correlation coefficient.

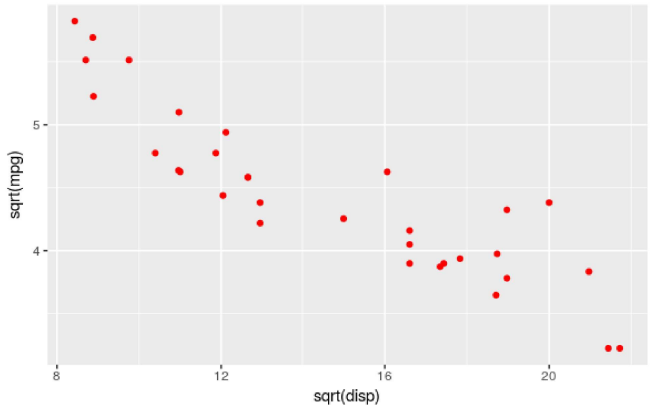
```
cor(mtcars$disp, mtcars$mpg)
[1] -0.8475514
```

This is a strong negative correlation. Note that ***correlation does not imply causation***. It just indicates whether a mutual relationship, causal

If the relationship is non-linear, a common approach in linear regression modelling is to ***transform*** the response and predictor variable in order to coerce the relationship to one that is more linear. Common transformations include natural and base ten logarithmic, square root, cube root and inverse transformations. The mpg and disp relationship is already linear but it can be strengthened using a square root transformation.

```
mtcars %>%
 ggplot(aes(x = sqrt(disp), y = sqrt(mpg))) +
 geom_point(colour = "red")

cor(sqrt(mtcars$disp), sqrt(mtcars$mpg))
[1] -0.8929046
```

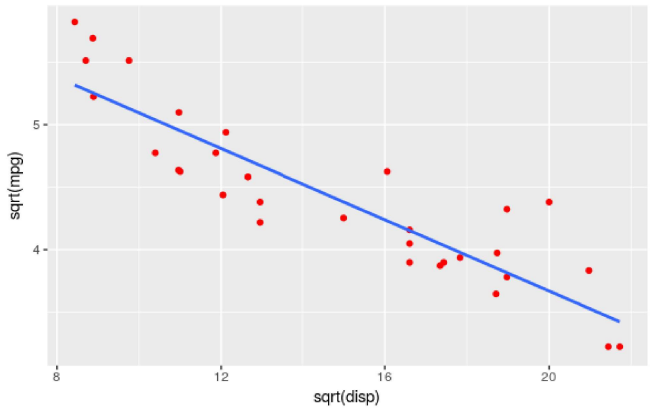


The next step is to determine whether the relationship is ***statistically significant*** and not just some random occurrence. This is done by investigating the variance of the data points about the fitted line. If the data fit well to the line, then the relationship is likely to be a real effect. The goodness of fit can be quantified using the ***root mean squared error*** (RMSE) and ***R-squared*** metrics. The RMSE represents the variance of the model errors and is an absolute measure of fit which has units identical to the response variable. *R*-squared is simply the Pearson correlation coefficient squared and represents variance explained in the response variable by the predictor variable.

The number of data points is also important and influences the ***p-value*** of the model. A rule of thumb for OLS linear regression is that at least 20 data points are required for a valid model. The *p*-value is the probability of there being no relationship (the null hypothesis) between the variables.

An OLS linear model is now fit to the transformed data.

```
mtcars %>%
 ggplot(aes(x = sqrt(disp), y = sqrt(mpg))) +
 geom_point(colour = "red") +
 geom_smooth(method = "lm", fill = NA)
```



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

The slope and the intercept can be obtained.

```
lmodel$coefficients

(Intercept) sqrt(disp)
 6.5192052 -0.1424601
```

And the model summary contains the important statistical information.

```
summary(lmodel)

Call:
lm(formula = sqrt(mpg) ~ sqrt(disp), data = mtcars)

Residuals:
 Min 1Q Median 3Q Max
-0.45591 -0.21505 -0.07875 0.16790 0.71178

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.51921 0.19921 32.73 < 2e-16 ***
sqrt(disp) -0.14246 0.01312 -10.86 6.44e-12 ***
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3026 on 30 degrees of freedom
Multiple R-squared: 0.7973, Adjusted R-squared: 0.7905
F-statistic: 118 on 1 and 30 DF, p-value: 6.443e-12
```

The *p*-value of 6.443e-12 indicates a statistically significant relationship at the $p<0.001$ cut-off level. The multiple *R*-squared value (*R*-squared) of 0.7973 gives the variance explained and can be used as a measure of predictive power (in the absence of overfitting). The RMSE is also included in the output (Residual standard error) where it has a value of 0.3026.

The take home message from the output is that for every unit increase in the square root of engine displacement there is a -0.14246 decrease in the square root of fuel efficiency (mpg). Therefore, fuel efficiency decreases with increasing engine displacement.

Comments are closed.

Search R-bloggers

Search.. Go

Most visited articles of the week

1. 5 Ways to Subset a Data Frame in R
2. How to write the first for loop in R
3. A Comprehensive Introduction to Command Line for R Users
4. R – Sorting a data frame by the contents of a column
5. Date Formats in R
6. Using apply, sapply, lapply in R
7. How to perform a Logistic Regression in R
8. Installing R packages
9. Sept 2019: "Top 40" New R Packages

Sponsors


Jobs for R users

- (Junior) Data Analyst – Berlin
- Associate Researcher (Data Analytics)
- Lead Data Scientist @ Stabio, Ticino, Switzerland
- Senior Scientist, Translational Informatics
- Senior Scientist, Translational Informatics @ Vancouver, BC, Canada
- Senior Principal Data Scientist @ Mountain View, California, United States
- Technical Research Analyst – New York, U.S.