Introduction to Linear Regression Using R

Jing Wang

11/31/2017

Introduction

After we are done with Exploratory Data Analysis and got a rough visualization of correlated variables of a dataset, as I have shown in my first post, it's time for us to do some 'real' modeling to obtain relations between variables.

> ### The General Problem
>
> **Given**: a collection of variables
>
> **Problem**: In what way does a veriable Y depend on other variables X1, …, Xn in the given dataset.
>
> **Definition**: A statistical model defines a mathematical relationship between the Xi's and Y. The *model* is a representation of the real Y that aims to replace it as far as possible. The *response variable* is the variable that we are trying to predict (Y in our problem). The *explanatory variables* are factors that affect the output of the response variable (Xi's).

There are so many types of statistical model that we can choose based on the types of our explanatory and response variables, such as regression, analysis of variance, and analysis of covariance. For the purpose of this post, let's only consider the case with continuous variables, using lineary regressional models. In Homework 1 we have already had a little flavor of linear regression. My goal is to demonstrate how to build up a linear regression model with multivariables and visualize the result of our fitting.

Fitting the Model

> Multiple Linear Regression Example
>
> fit <- lm(y ~ x1 + x2 + x3, data=mydata)
>
> summary(fit) # show results

> Other useful functions
>
> coefficients(fit) # model coefficients
>
> confint(fit, level=0.95) # CIs for model parameters
>
> fitted(fit) # predicted values
>
> residuals(fit) # residuals
>
> anova(fit) # anova table
>
> vcov(fit) # covariance matrix for model parameters
>
> influence(fit) # regression diagnostics

Example of Simple Linear Regression

For this analysis, we will use the *mtcars* dataset that comes with R by default. *mtcars* is a standard built-in dataset, that makes it convenient to demonstrate linear regression in a simple and easy to understand fashion. Let's take a look

```
# Loading and viewing
mtcars
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles. If you want to learn more about the dataset, type in:

> ?mtcars

With the *lm()* function, we can explore the linear relation between fuel consumption and any one of the ten factors, or even better, any few of the factors.

```
# Relation between mpg (Mile Per Gallon) and cyl (Number of Cylinders)
# Simple Linear Regression model
linear_model <- lm(mpg~cyl, data = mtcars)
linear_model
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
##
## Coefficients:
## (Intercept)          cyl
##      37.885       -2.876
```

So our linear function is *mpg* = -2.876 *cyl* + 37.885

```
# Now if you want a more detailed analysis of the linear model
summary(linear_model)
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9814 -2.1185  0.2217  1.0717  7.5186
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.8846     2.0738   18.27  < 2e-16 ***
## cyl          -2.8758     0.3224   -8.92 6.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.206 on 30 degrees of freedom
## Multiple R-squared:  0.7262, Adjusted R-squared:  0.7171
## F-statistic: 79.56 on 1 and 30 DF,  p-value: 6.113e-10
```

Wow, what do all these mean? How we should read these?

Residuals are the differences between the observed value of the dependent variable and the predicted value. So that means, if we plug in one observation of *cyl* into the linear function that we generated with *lm()*, what we get is highly likely to be different from the actual *mpg*. In a sense, residuals measure how 'good' our model is.

Coefficients tell us the correlation between our variables. Estimate values are what we should plug in to our function. It's really nice that we are also given the standard error of our coefficients. And they are very handy when we need to contruct confidence intervals.

T value also measures how "good" our estimated coefficients are. It's caculated by dividing the estimated coefficients with the standard errors.

Pr(>|t|), also known as the p value. In hypothesis testing, a small p-value indicates strong evidence against the null hypothesis, so you reject the null hypothesis.

So now we have some basic understanding of linear models, we can proceed to more complicated examples.

Multiple Linear Regression

```
# Let's compare some results using 2 and 3 factors
model1 <- lm(mpg~cyl + disp + hp, data = mtcars)
model2 <- lm(mpg~cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb, data = mtcars)
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0889 -2.0845 -0.7745  1.3972  6.9183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.18492    2.59078  13.195 1.54e-13 ***
## cyl         -1.22742    0.79728  -1.540   0.1349
## disp        -0.01884    0.01040  -1.811   0.0809 .
## hp          -0.01468    0.01465  -1.002   0.3250
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.055 on 28 degrees of freedom
## Multiple R-squared:  0.7679, Adjusted R-squared:  0.743
## F-statistic: 30.88 on 3 and 28 DF,  p-value: 5.054e-09
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
##     am + gear + carb, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

We can see that our coefficients are different in these two models. To compare models in a more quantative way, use the *anova()* function

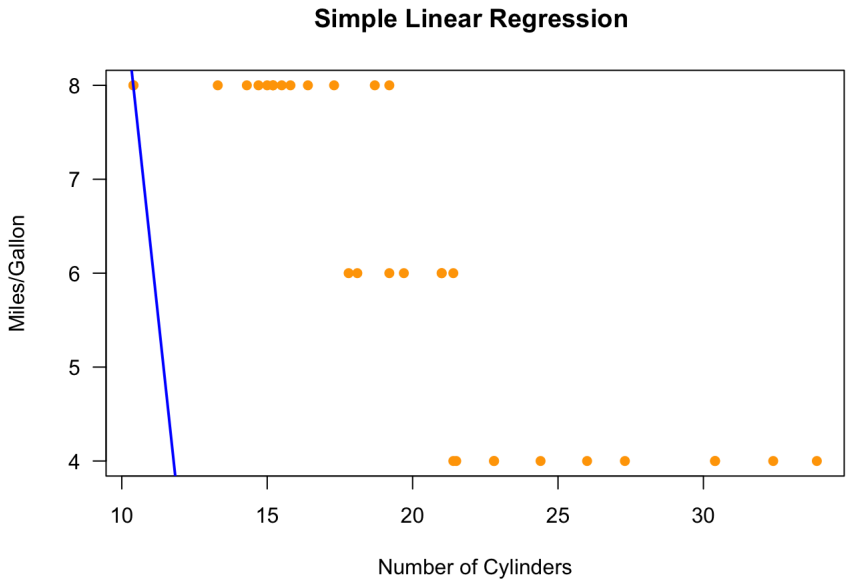
```
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ cyl + disp + hp
## Model 2: mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     28 261.37
## 2     21 147.49  7    113.88 2.3162 0.06443 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Visualization of Single Linear Regression

If you still find all these numbers confusing, don't worry, let's visualize them.

```
# Sactter plot for simple linera regression
plot(mtcars$mpg, mtcars$cyl, las = 1, pch = 19, cex = 0.9, col = "orange", xlab = "Number of Cylinders", ylab = "Miles/Gallon", main = "Simple Linear Regression ")
abline(linear_model, col = "blue", lwd = 2)
```

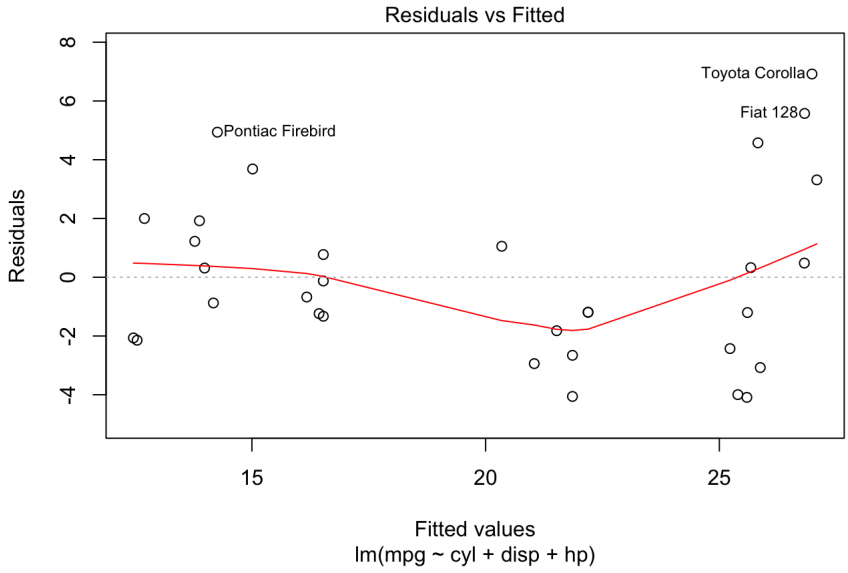


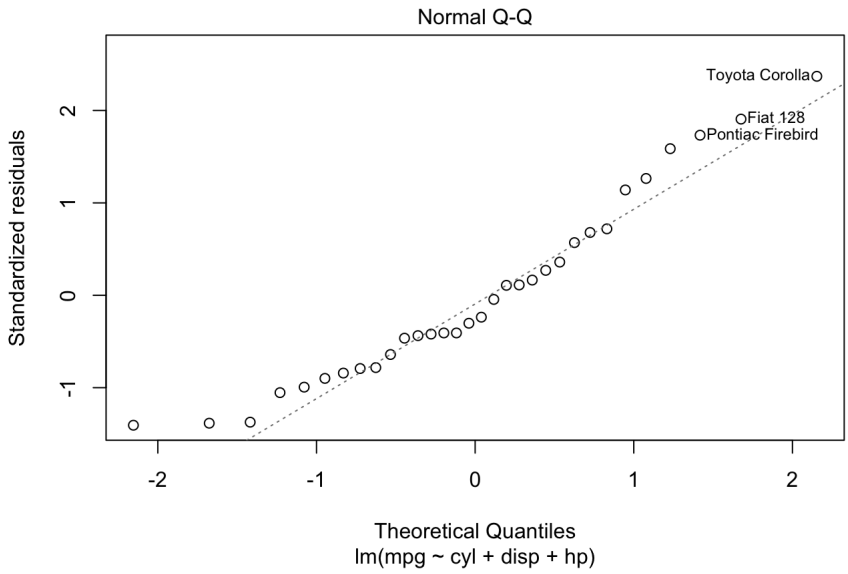
Now after actually visualize it, we can see that the correlation between the two is not that strong.

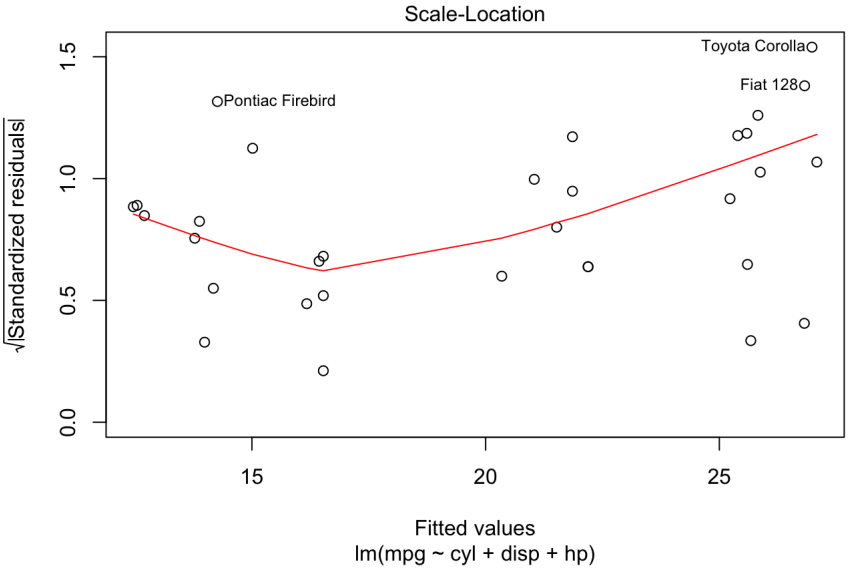
Graphical Analysis of Multiple Linear Regression

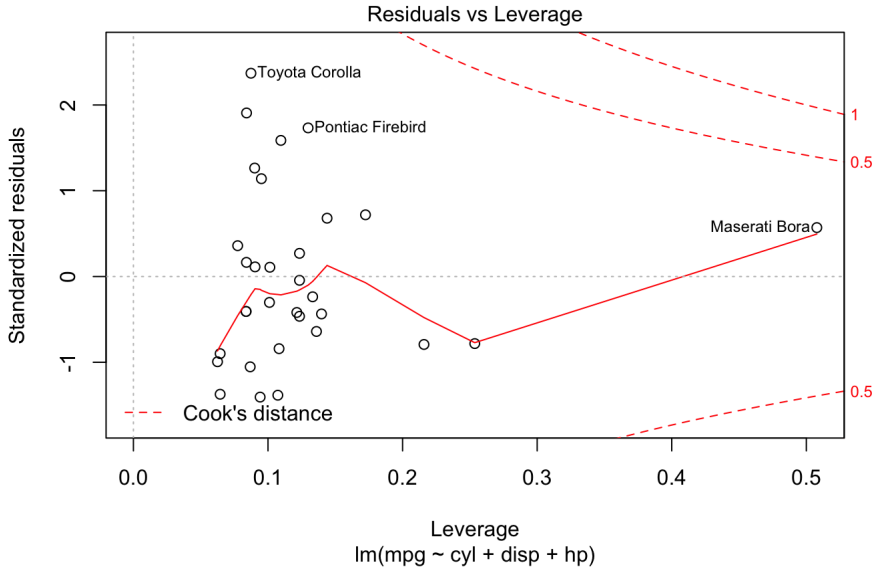
We can visualize our models using 3D scatterplots. But to diagnose our model in a quantative way, obtain diagnostic scatterplots and I'll explain them one by one.

```
plot(model1)
```









Residual vs Fitted Plots

A residual vs fitted plot is a scatter plot of residuals on the y axis and fitted values (estimated responses) on the x axis. The plot is used to detect non-linearity, unequal error variances, and outliers.

Here are the characteristics of a well-behaved residual vs. fits plot and what they suggest about the appropriateness of the simple linear regression model:

- The residuals “bounce randomly” around the 0 line. This suggests that the assumption that the relationship is linear is reasonable.
- The residuals roughly form a “horizontal band” around the 0 line. This suggests that the variances of the error terms are equal.
- No one residual “stands out” from the basic random pattern of residuals. This suggests that there are no outliers.

Normal Quantile-Quantile Plots

The Q-Q plot, or quantile-quantile plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential. For example, if we run a statistical analysis that assumes our dependent variable is Normally distributed, we can use a Normal Q-Q plot to check that assumption. It’s just a visual check, not an air-tight proof, so it is somewhat subjective. But it allows us to see at-a-glance if our assumption is plausible, and if not, how the assumption is violated and what data points contribute to the violation.

A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a line that’s roughly straight.

Scale Location Plots

It’s also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors. This is how you can check the assumption of equal variance (homoscedasticity). It’s good if you see a horizontal line with equally (randomly) spread points.

Residuals vs Leverage Plots

This plot helps us to find influential cases (i.e., subjects) if any. Not all outliers are influential in linear regression analysis (whatever outliers mean). Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn’t be much different if we either include or exclude them from analysis. They follow the trend in the majority of cases and they don’t really matter; they are not influential. On the other hand, some cases could be very influential even if they look to be within a reasonable range of the values. They could be extreme cases against a regression line and can alter the results if we exclude them from analysis. Another way to put it is that they don’t get along with the trend in the majority of the cases.

Unlike the other plots, this time patterns are not relevant. We watch out for outlying values at the upper right corner or at the lower right corner. Those spots are the places where cases can be influential against a regression line. Look for cases outside of a dashed line, Cook’s distance. When cases are outside of the Cook’s distance (meaning they have high Cook’s distance scores), the cases are influential to the regression results. The regression results will be altered if we exclude those cases.

Conclusion

After reading this, you should feel confident to use *lm()* to run a linear model and be able to analyze if the model works well for the data. There are definitely more statistical modeling methods that can be done with R, and they are very practical and used a lot by data scientists in real life. It would also be fun to explore those.

Reference

https://www3.nd.edu/~steve/Rcourse/Lecture7v1.pdf

http://r-statistics.co/Linear-Regression.html

https://www.statmethods.net/stats/regression.html

https://www.statsdirect.com/help/basics/p_values.htm

https://rpubs.com/FelipeRego/MultipleLinearRegressionInRFirstSteps

http://data.library.virginia.edu/understanding-q-q-plots/

http://data.library.virginia.edu/diagnostic-plots/