Linear Regression - Part 1 - Edx Analytical Edge

FdR

3 January 2015

This use the EDx course - Analytical Edge

## Single variable regression.

The general equation for a linear regression model

where:

* is the observation of the dependent variable
* is the intercept coefficient
* is the regression coefficient for the dependent variable
* is the observation of the independent variable
* is the error term for the observation. It basically is the difference in therm of y between the observed value and the estimated value. It is also called the residuals. A good model minimize these errors.

One way to assess how good our model is to:

1. compute the SSE (the sum of squared error)
   * SSE = =
   * problem: SSE is dependent of N. SSE will naturally increase as N increase
2. compute the RMSE (the root mean squared error)
   * RMSE =
   * It depends of the unit of the independent variable
3. compute
   * It compare the models to a baseline model
   * is **unitless** and **universaly** interpretable
   * SST is the sum of the squared of the difference between the observed value and the mean of all the observed value

### In practice.

#### First example. Predicting wine price.

The wine.csv file is used in the class. The *AGST* is the independent variable while the *price* is the dependent variable.

Let's load it and then have a quick look at its structure.

wine = read.csv("wine.csv")  
str(wine)

## 'data.frame': 25 obs. of 7 variables:  
## $ Year : int 1952 1953 1955 1957 1958 1959 1960 1961 1962 1963 ...  
## $ Price : num 7.5 8.04 7.69 6.98 6.78 ...  
## $ WinterRain : int 600 690 502 420 582 485 763 830 697 608 ...  
## $ AGST : num 17.1 16.7 17.1 16.1 16.4 ...  
## $ HarvestRain: int 160 80 130 110 187 187 290 38 52 155 ...  
## $ Age : int 31 30 28 26 25 24 23 22 21 20 ...  
## $ FrancePop : num 43184 43495 44218 45152 45654 ...

head(wine)

## Year Price WinterRain AGST HarvestRain Age FrancePop  
## 1 1952 7.4950 600 17.1167 160 31 43183.57  
## 2 1953 8.0393 690 16.7333 80 30 43495.03  
## 3 1955 7.6858 502 17.1500 130 28 44217.86  
## 4 1957 6.9845 420 16.1333 110 26 45152.25  
## 5 1958 6.7772 582 16.4167 187 25 45653.81  
## 6 1959 8.0757 485 17.4833 187 24 46128.64

We use the lm function to find our linear regression model.

model1 = lm(Price ~ AGST,data=wine)  
summary(model1)

##   
## Call:  
## lm(formula = Price ~ AGST, data = wine)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.78450 -0.23882 -0.03727 0.38992 0.90318   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.4178 2.4935 -1.371 0.183710   
## AGST 0.6351 0.1509 4.208 0.000335 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4993 on 23 degrees of freedom  
## Multiple R-squared: 0.435, Adjusted R-squared: 0.4105   
## F-statistic: 17.71 on 1 and 23 DF, p-value: 0.000335

The summary function applied on the model is giving us a bunch of important information

* the stars next to the predictor variable indicated how significant the variable is for our regression model
* it also gives us the value of the R coefficient

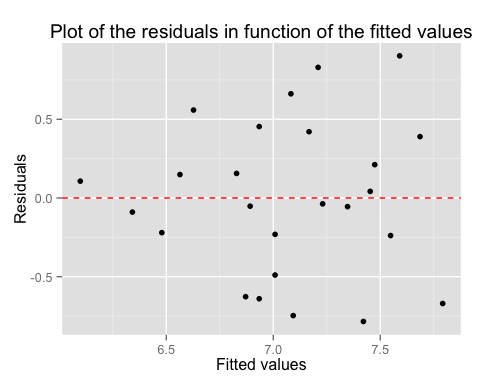
We could have calculated the R value ourselves:

SSE = sum(model1$residuals^2)  
SST = sum((wine$Price - mean(wine$Price))^2)  
r\_squared = 1 - SSE/SST  
r\_squared

## [1] 0.4350232

It is always nice to see how our residuals are distributed.  
We use the ggplot2 library and the fortify function which transform the summary(model1) into a data frame usable for plotting.

library(ggplot2)  
model1 <- fortify(model1)  
ggplot(model1, aes(.fitted, .resid)) + geom\_point() + geom\_hline(yintercept = 0, col = "red", linetype = "dashed") + xlab("Fitted values") + ylab("Residuals") + ggtitle("Plot of the residuals in function of the fitted values")



## Multi-variables regression.

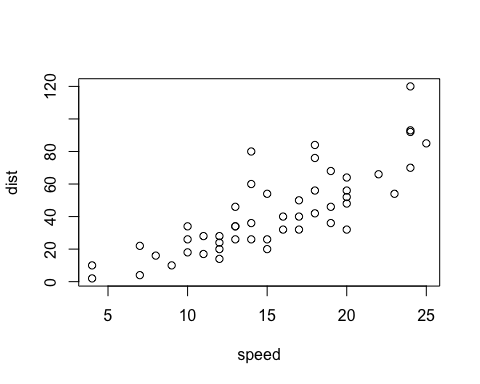
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.