R Notebook

## Linear Regression

# Theory

Linear Regression is a linear algorithm to analyze/predict linear relation between a dependent and one or many independent variables. Let’s take an example of house price. In this case price of the house is the dependent variable and factors like size of house, locality, season of purchase might act as independent variables. To start using LR or any other algorithm, first and foremost step is to generate a Hypothesis. For this tutorial, we would be using a default data set in R. R comes with many default data sets and it can be seen by using MASS library.

Install.packages(“MASS”)

Library(MASS)

Data()

This will give you a list of available data sets using which you can get can a clear idea of linear regression problems. For this tutorial we would be using “airquality” data which is data set of measuring various parameters of air quality in New York. The data is daily temperature data of May to August

Aim of the tutorial: To analyze and create a linear regression model using R programming and understanding concepts related to hypothesis testing and prediction.

Along with temperature, it has Solar radiation, ozone and wind data

So, to start with, we would formulate null and alternate hypothesis.

The hypothesis is: "Temperature of house depends on ozone, wind and solar radiations" Now, the null hypothesis of linear regression says there is no relation between of dependent and independent variables and all coefficients are zero. i.e. if equation is

Temp= a1.Solar.R +a2.Ozone + a3.Wind + error and alternate hypothesis is: There is at least one non zero coefficient and hence relationship exists between dependent and independent variable.

In mathematical notations it can be written as:

H0: a1=a2=a3=0

Ha: a1≠a2≠a3≠0

Now we would be testing these hypothesis using Linear regression and draw a conclusion. To test the hypothesis, we would check the level of significance of variables in support of out hypothesis. If the significance is higher than accepted level (generally 95%), we would say that “We can reject the null hypothesis and hence there is a relation between dependent and independent variables” and if significance is less than the accepted level we would fail to reject null hypothesis and hence it would be proved that there is no relation between dependent and independent variables.

Before making a LR model let’s take little more understanding of the data looking few rows of data

data(airquality)# to call the data  
attach(airquality)  
head(airquality,10)# to see first 10 rows

## Ozone Solar.R Wind Temp Month Day  
## 1 41 190 7.4 67 5 1  
## 2 36 118 8.0 72 5 2  
## 3 12 149 12.6 74 5 3  
## 4 18 313 11.5 62 5 4  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6  
## 7 23 299 8.6 65 5 7  
## 8 19 99 13.8 59 5 8  
## 9 8 19 20.1 61 5 9  
## 10 NA 194 8.6 69 5 10

Now let's have a look at the summary of data, Summary gives five point summary (25 quartile, mean,median,) for numerical variables and frequency distribution for categorical variables

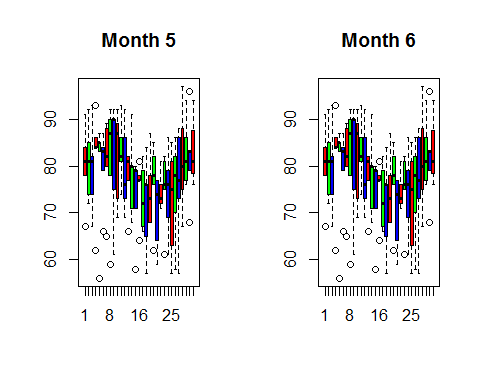
summary(airquality)

## Ozone Solar.R Wind Temp   
## Min. : 1.00 Min. : 7.0 Min. : 1.700 Min. :56.00   
## 1st Qu.: 18.00 1st Qu.:115.8 1st Qu.: 7.400 1st Qu.:72.00   
## Median : 31.50 Median :205.0 Median : 9.700 Median :79.00   
## Mean : 42.13 Mean :185.9 Mean : 9.958 Mean :77.88   
## 3rd Qu.: 63.25 3rd Qu.:258.8 3rd Qu.:11.500 3rd Qu.:85.00   
## Max. :168.00 Max. :334.0 Max. :20.700 Max. :97.00   
## NA's :37 NA's :7   
## Month Day   
## Min. :5.000 Min. : 1.0   
## 1st Qu.:6.000 1st Qu.: 8.0   
## Median :7.000 Median :16.0   
## Mean :6.993 Mean :15.8   
## 3rd Qu.:8.000 3rd Qu.:23.0   
## Max. :9.000 Max. :31.0   
##

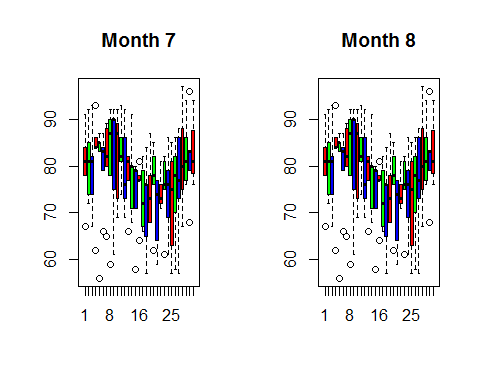
# Some Visializations

boxplot: Temeprature variation on daily basis (per month)

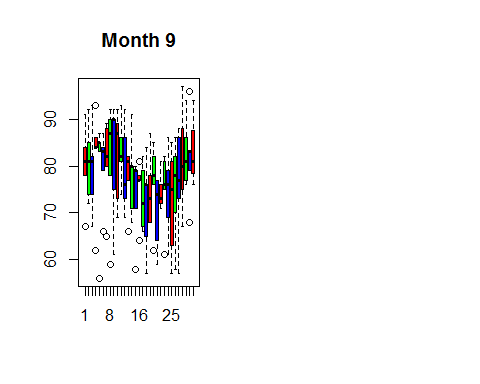
month5=subset(airquality,Month=5)  
month6=subset(airquality,Month=6)  
month7=subset(airquality,Month=7)  
month8=subset(airquality,Month=8)  
month9=subset(airquality,Month=9)  
  
par(mfrow = c(1,2)) # 3 rows and 2 columns  
boxplot((month5$Temp~airquality$Day),main="Month 5",col=rainbow(3))  
boxplot((month6$Temp~airquality$Day),main="Month 6",col=rainbow(3))



boxplot((month7$Temp~airquality$Day),main="Month 7",col=rainbow(3))  
boxplot((month8$Temp~airquality$Day),main="Month 8",col=rainbow(3))

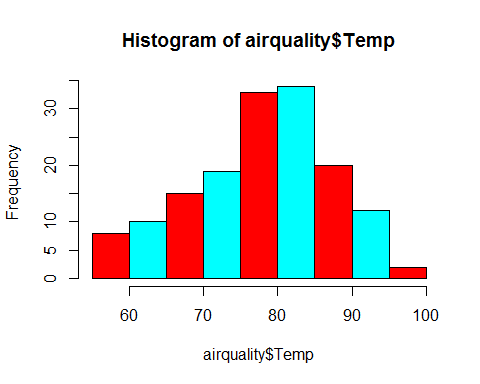


boxplot((month9$Temp~airquality$Day),main="Month 9",col=rainbow(3))



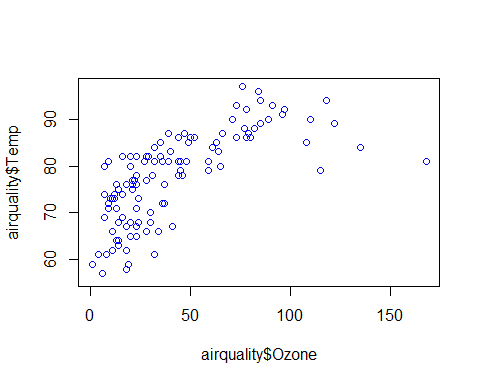
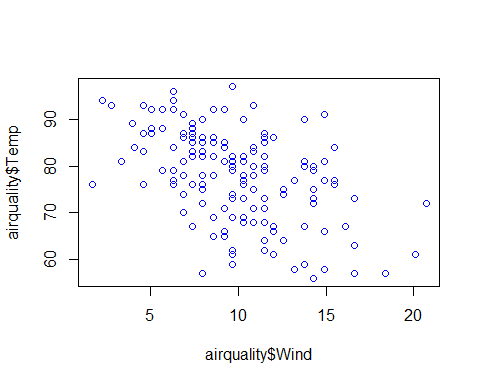
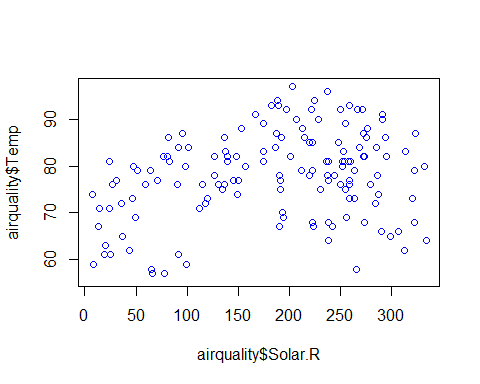
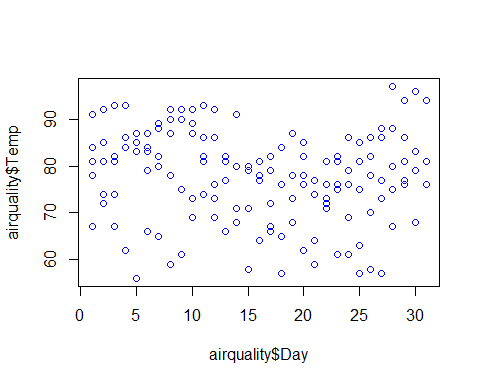
Distribution of temperature:

hist(airquality$Temp,col=rainbow(2))



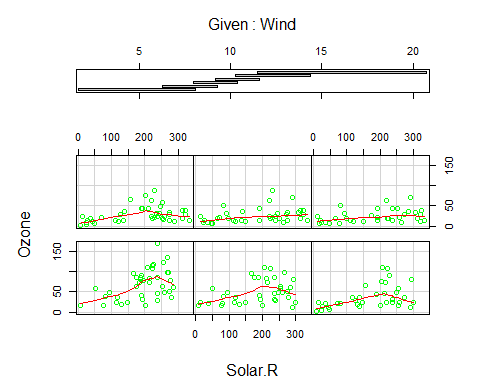
To check if there is some linear pattern in temerature rise and othet variables

plot(airquality$Temp~airquality$Day+airquality$Solar.R+airquality$Wind+airquality$Ozone,col="blue")



It seems that solar.R ,Ozone, and wind have a linear pattern with temperature(Solar and Ozone having a positive and wind a negative) Co-plot: to see effect of wind and solar radiations(combined) on Temperature

coplot(Ozone~Solar.R|Wind,panel=panel.smooth,airquality,col ="green" )



# 

# Linear Regression

Data preparation:

Before making the linear or any model, we should first prepare our data as the input. This involved missing value treatment, checking correlation and outliers.

While making the model, R inherentily takes care of the null values and drops the rows where the data is missing. but this eventually is a data loss. There are different methods to deal with null values like imputing mean for numerical variables and mode for categorical variables. Another methods like replacing null with any value which is way out of range is also useful in many cases. e.g. replacing a null value with -1 when variable is age.Since age can't be negative, R, while making the model takes it as outlier and don't consider it .

Additional command: To find column wise count of null values in the data

sapply(airquality,function(x){sum(is.na(x))})

## Ozone Solar.R Wind Temp Month Day   
## 37 7 0 0 0 0

So we can see that Ozone and Solar.R are having missing values. We can drop those rows but that would cause a data loss and since we have only 153 rows in our data, dropping 37 would be almost a 20% loss. Hence we can use another approach of replacing null values with mean (since both of the variables are numerical)

airquality$Ozone[is.na(airquality$Ozone)]=mean(airquality$Ozone,na.rm=T)

airquality$Solar.R[is.na(airquality$Solar.R)]=mean(airquality$Solar.R,na.rm=T)

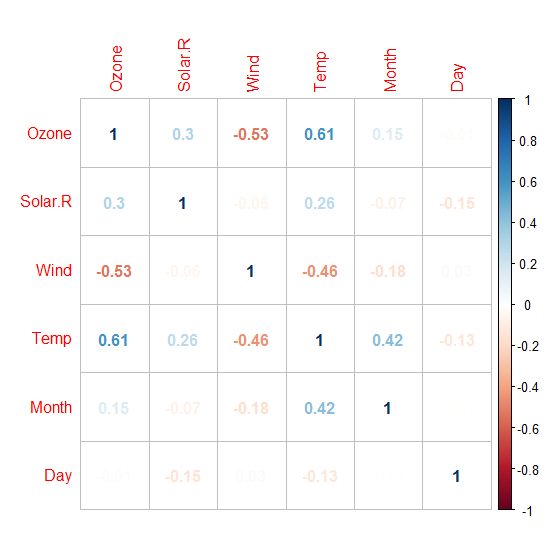
sapply(airquality,function(x){sum(is.na(x))})

## Ozone Solar.R Wind Temp Month Day   
## 0 0 0 0 0 0

Now we can check the correlation between variables. R has a inbuilt command for finding correlation but on the top of that we can use corrplot (from corrplot package) to get a good visualization.

**library(corrplot)**

**o=corrplot(cor(airquality),method='number') # this method can be changed try using method='circle'**



So Wind and Ozone are having a little high correlation (we are checking correlation between independent variables for model)

We can either drop one of the two variables or do take ratio, subtraction etc. to create a new variable using these two( this depends on the domain knowledge that how do we know about relation in variables. Eg. If a variable is income and one is expenditure. We can take the difference and create another variable)

Now we would make our linear model (without dropping the variable) and will see effect of multicollinearity on the model.

Model\_lm1=lm(Temp~.,data=airquality)  
summary(Model\_lm1)

##   
## Call:  
## lm(formula = Temp ~ ., data = airquality)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.110 -4.048 0.859 4.034 12.840   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 57.251830 4.502218 12.716 < 2e-16 \*\*\*  
## Ozone 0.165275 0.023878 6.922 3.66e-10 \*\*\*  
## Solar.R 0.010818 0.006985 1.549 0.124   
## Wind -0.174326 0.212292 -0.821 0.413   
## Month 2.042460 0.409431 4.989 2.42e-06 \*\*\*  
## Day -0.089187 0.067714 -1.317 0.191   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.159 on 105 degrees of freedom  
## (42 observations deleted due to missingness)  
## Multiple R-squared: 0.6013, Adjusted R-squared: 0.5824   
## F-statistic: 31.68 on 5 and 105 DF, p-value: < 2.2e-16

Tuning the model for a low AIC value: This can be done in two ways

1)Dropping some less significant variables and re-running the model

2)Using Step funtions in R: This runs all the possible parameters and check the lowest value

Model\_lm\_best=step(Model\_lm1)

## Start: AIC=409.4  
## Temp ~ Ozone + Solar.R + Wind + Month + Day  
##   
## Df Sum of Sq RSS AIC  
## - Wind 1 25.58 4008.2 408.11  
## - Day 1 65.80 4048.5 409.22  
## <none> 3982.7 409.40  
## - Solar.R 1 90.96 4073.6 409.91  
## - Month 1 943.90 4926.6 431.01  
## - Ozone 1 1817.27 5799.9 449.12  
##   
## Step: AIC=408.11  
## Temp ~ Ozone + Solar.R + Month + Day  
##   
## Df Sum of Sq RSS AIC  
## - Day 1 71.5 4079.8 408.07  
## <none> 4008.2 408.11  
## - Solar.R 1 82.1 4090.3 408.36  
## - Month 1 997.2 5005.5 430.77  
## - Ozone 1 3242.4 7250.6 471.90  
##   
## Step: AIC=408.07  
## Temp ~ Ozone + Solar.R + Month  
##   
## Df Sum of Sq RSS AIC  
## <none> 4079.8 408.07  
## - Solar.R 1 92.1 4171.9 408.55  
## - Month 1 1006.3 5086.1 430.55  
## - Ozone 1 3225.5 7305.3 470.74

summary(Model\_lm\_best)

##   
## Call:  
## lm(formula = Temp ~ Ozone + Solar.R + Month, data = airquality)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -21.2300 -4.3645 0.6438 4.1479 11.3866   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 53.262897 3.268761 16.295 < 2e-16 \*\*\*  
## Ozone 0.176503 0.019190 9.198 3.34e-15 \*\*\*  
## Solar.R 0.010807 0.006953 1.554 0.123   
## Month 2.092793 0.407364 5.137 1.26e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.175 on 107 degrees of freedom  
## (42 observations deleted due to missingness)  
## Multiple R-squared: 0.5916, Adjusted R-squared: 0.5802   
## F-statistic: 51.67 on 3 and 107 DF, p-value: < 2.2e-16

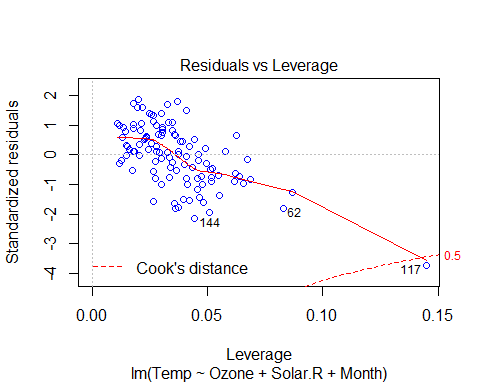
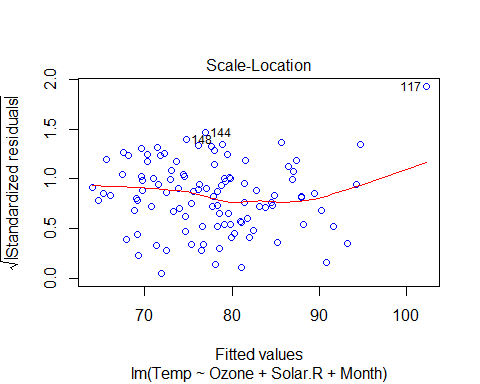
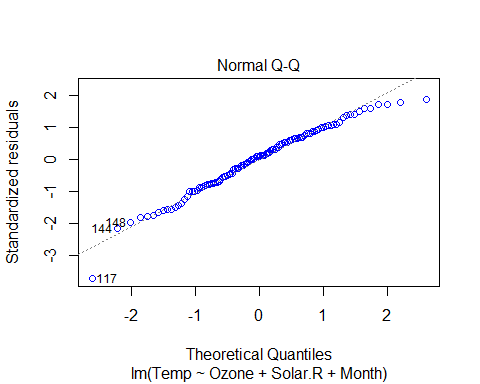
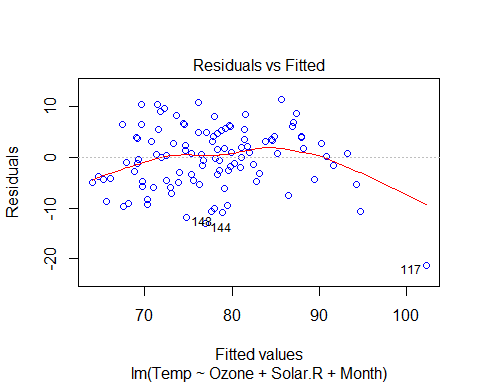
Significance level

This summary gives values for coefficients of dependent variables and error term with the significance level (confidence level). Highlighted line in the result shows how to read level of significance. A three star means 99.99 % significant (check corresponding P value. If it is less than 0.01 it means variable is 99.99 % significant)

Along with this this gives R square and adjusted R square values which defines how much variance of the dependent variable is explained by the model (rest is explained by the error term) and hence higher the R square/adjusted R square better the model.

Adjusted R square is a better indicator of explained variance because it considers only important variables and if extra variables are deliberately used to explained more variance, adjusted R square will drop. In other terms adjusted R square penalize the attempt of inclusion of many variables in the model for the sake of high percentage of variance explained.

plot(Model\_lm\_best,col="blue")



# VIF and multicollinearity

Variable Inflation factor is an important parameter regarding value of coefficient of determination (R2). If two independent variables are highly correlated than it inflates the variance of the (estimated error) model.

To deal with this, we can check VIF of the model and then VIF of the model after dropping one of the two highly correlated variables.

Formula for VIF:

VIF(k)= 1/1+Rk^2

where R2 is the R2-value obtained by regressing the kth predictor on the remaining predictors.

So to calculate VIF we make model for each independent variable and taking all other variables as predictors. Then we calculate VIF for each variable. Whenever VIF is high it means that set of variables have high correlation with the selected variable.

R comes with many packages to calculate the VIF and we would be using ‘fmsb’ package.

So we can check VIF for our final linear model.

Library(fmsb)

Model\_lm1=lm(Temp~ Ozone+Solar.R+Month,data=airquality)

VIF(lm(Month ~ Ozone+Solar.R,data=airquality))

[1] 1.039042

VIF(lm(Ozone ~ Solar.R+Month, data=airquality))

[1] 1.137975

VIF(lm(Solar.R ~ Ozone +Month, data=airquality))

[1] 1.118629

# As a general rule, if VIF < 5 is acceptable (VIF = 1 means there is no multicollinearity) and for VIF >5 and < 10 it is alarming. VIF >10 is generally not acceptable and we need to check it.

In our example, VIF < 5 and hence there is no need of any additional step.

# interpretation of results

Basic assumptions of Linear regression:

* Linear relationship between variables
* Normal distribution of residuals
* No or little multi-collinearity: We have seen this using VIF
* Homoscedasticity: Variance across the regression line should be uniform

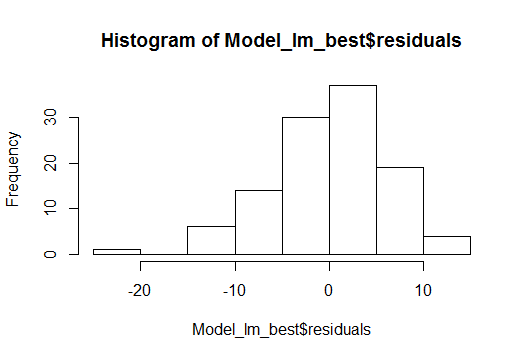
R comes with very nice way to show results; the summary of the model gives intercept values of all the independent variables along with error term (or residuals)

The Linear relationship between variables has been verified by the significance (p value) for variables.

In ‘Residuals vs fitted values’ plot it can be seen that residuals are linearly distributed and hence variance is uniform

In ‘Normal Q-Q’ plot it can be seen that residuals are normally distributed. It can be seen by plotting histogram of residuals also

hist(Model\_lm\_best$residuals)



To measure the quality of the model there are many ways and residual sum of squares is the most common one.

It that to draw a line of best fit in scattered data points so that that line has least error with respect to the actual data points. if Y is actual data point and Y is the predicted value by the equation of the line then error = y-y but this has a bias towards sign as while adding positive and negative values would cancel each other and resultant error would be less than the actual value. To overcome this, a general method is to take square which surves two purposes:

1) Cancel out the effect of signs

2) Penalize the error in prediction.

## Prediction

For prediction lets make a data frame for new values of Solar.R,Wind and Ozone Solar.R=185.93 Wind=9.96 Ozone=42.12

Solar.R=185.93  
Wind=9.96  
Ozone=42.12  
Month=9  
new\_data=data.frame(Solar.R,Wind,Ozone,Month)  
new\_data

## Solar.R Wind Ozone Month  
## 1 185.93 9.96 42.12 9

pred\_temp=predict(Model\_lm\_best,newdata=new\_data)

## [1] "the predicted temperature is: 81.54"

# Final notes

The regression algorithm takes in account the data is normally distributed and there is a linear relation between dependent and independent variables. It is a good mathematical model to analyze relationships and significance of various possible variables and once the model is ready, it can be used to predict the values.