**SALARY\_HIKE and CHURN\_OUT\_RATE**

Emp\_data -> Build a prediction model for Churn\_out\_rate

Do the necessary transformations for input variables for getting better R^2 value

**Inferences from the Data Set:**

Data Set talks about the Years of experience with respect to Salary with 30 Observations

**Columns:**

Salary\_hike

Churn\_out\_rate

**Data Set Size:** 10

Data give is found to be a continuous data for which a simple linear regression can be performed getting deeper into the data analysis and its behavior

**Salary\_hike:**

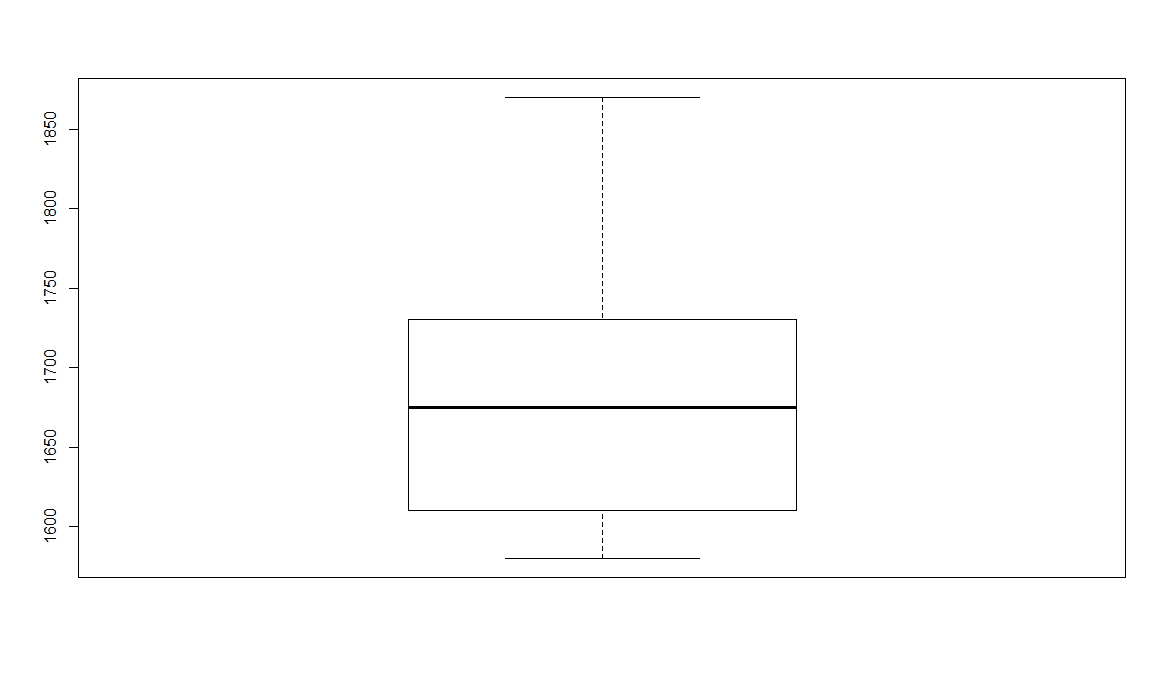
Ranges between 1580 - 1870

For this Salary\_hike the mean is 1689, it is just the average of the Salary\_hike data

The median for the given data is 1675, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=1675 < mean=1689), if data was not skewed, we would have considered mean but hear it is skewed so we take Median to talk about data.

The Data is Right Skewed, Skewness= 0.72



**Churn\_out\_rate:**

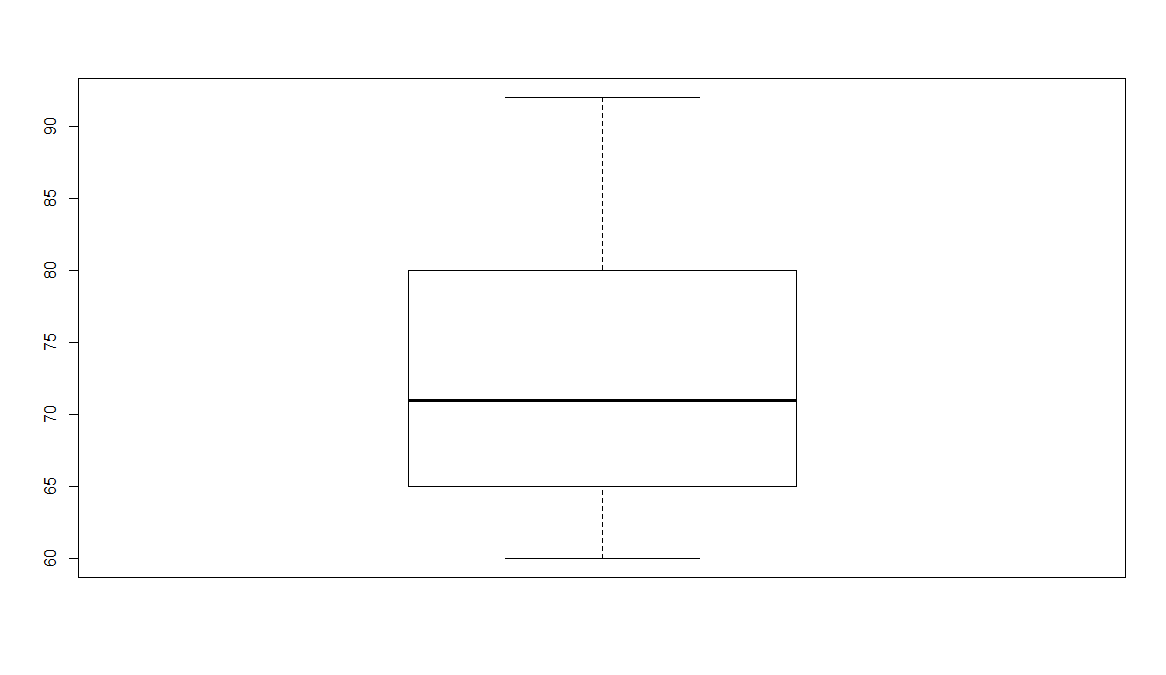
Ranges between 60 - 92

For this Churn\_out\_rate the mean is 72.90 , it is just the average of the Churn\_out\_rate data

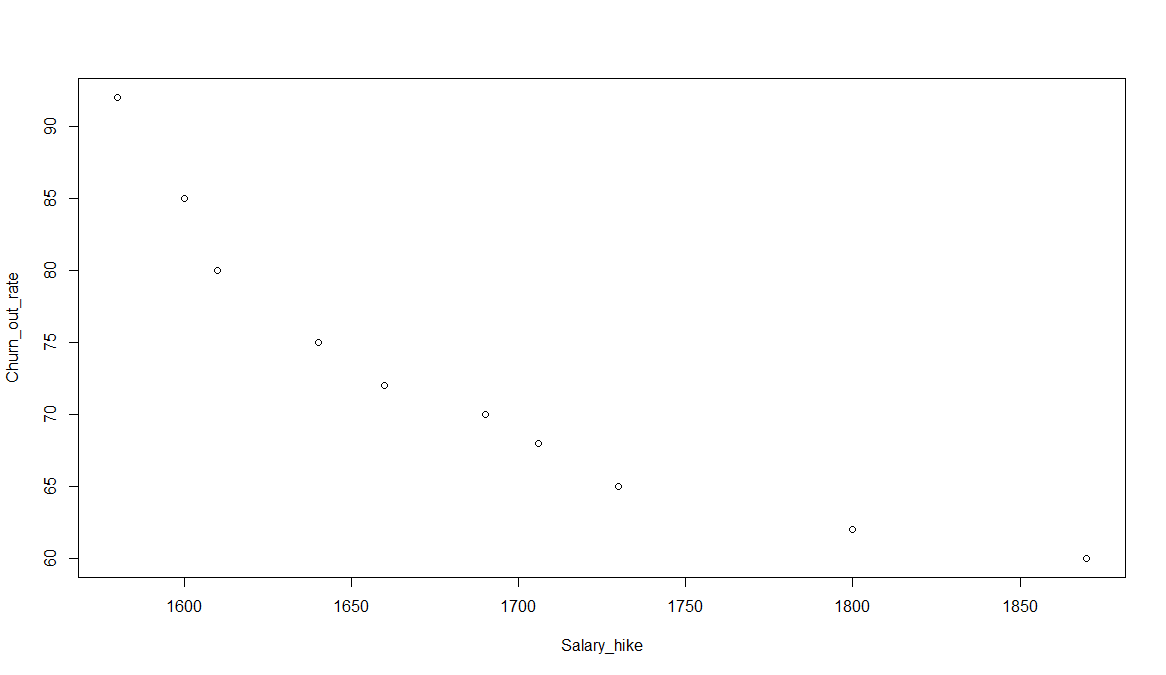
The median for the given data is 71, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=71<mean=72.90), if data was not skewed, we would have considered mean but hear it is skewed so we take Median to talk about data.

Skewness =0.54



**Plot for Salary\_hike vs Churn\_out\_rate:**



The above scatter diagram infer that the Salary\_hike and Churn\_out\_rate are moderately negative correlated.

Correlation Coefficient:

Let’s see the relationship between the Salary\_hike and Churn\_out\_rate

**cor(Salary\_hike,Churn\_out\_rate)**

**-0.9117216**

Based on the correlation value obtained which is -0.91(approx.) also tells that it is negative correlation

We use **lm() function from Base Package in R-Studio** to estimate the Salary\_hike using the other variable Churn\_out\_rate whereas in **python LinearRegression() is used from the sklearn package**

Call:

lm(formula = Churn\_out\_rate ~ Salary\_hike, data = SH\_CO)

Residuals:

Min 1Q Median 3Q Max

-3.804 -3.059 -1.819 2.430 8.072

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 244.36491 27.35194 8.934 1.96e-05 \*\*\*

Salary\_hike -0.10154 0.01618 -6.277 0.000239 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.469 on 8 degrees of freedom

Multiple R-squared: 0.8312, Adjusted R-squared: 0.8101

F-statistic: 39.4 on 1 and 8 DF, p-value: 0.0002386

**P-values:**

coefficient p-values are used to determine which terms to keep in the regression model

Look at the r-squared values are 0.8312

Lets apply some transformation on the data to get a better transformation, there are different types of transformation techniques like log transformation, exponential transformation, Quadratic model..

Lets also look into the plots how they are behaving

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **output** | **input** | **cor** | **R^2** | **RMSE** | **Model** | **Plot** |
| CO | SH | -0.91 | 0.83 | 3.99 | SLR | C:\Users\RAVI\Desktop\C1 |
| CO | log(SH) | -0.92 | 0.84 | 3.78 | LT | C:\Users\RAVI\Desktop\C2 |
| CO | SH\*SH |  | 0.97 | 1.57 | QM-2D | C:\Users\RAVI\Desktop\C3 |
| CO | SH\*SH\*SH |  | 0.98 | 1 | QM-3D | C:\Users\RAVI\Desktop\C4 |

CO = Churn\_out\_rate SH= Salary\_hike ET= Exponential Transformation

QM-2D= QM2D=Quadratic model 2Degree QM-3D= QM2D=Quadratic model 3Degree

Based on obtained R-squared values and the plot the best transformation technique is Polynomial 3Degree with 0.98 R-squared value and RMSE vaue 1

**Packages:**

**R Studio**

* readr
* ggplot2
* moments

**Python**

* import pandas as pd
* import numpy as np
* import matplotlib.pyplot as plt
* from sklearn.linear\_model import LinearRegression
* import statsmodels.api as sm
* import statsmodels.formula.api as smf
* from sklearn import metrics

**CODES:**

**R code:**

**# Simple Linear Regression Assignment #**

**# 3) Emp\_data -> Build a prediction model for Churn\_out\_rate**

**# Do the necessary transformations for input variables for getting better R^2 value for the model prepared.**

library(readr)

library(ggplot2)

library(moments)

SH\_CO <- read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/emp\_data.csv")

View(SH\_CO)

attach(SH\_CO)

summary(SH\_CO)

range(SH\_CO$Salary\_hike)

range(SH\_CO$Churn\_out\_rate)

skewness(SH\_CO$Salary\_hike)

skewness(SH\_CO$Churn\_out\_rate)

**#Exploratory Data Analysis**

boxplot(SH\_CO$Salary\_hike)

boxplot(SH\_CO$Churn\_out\_rate)

**#scatter plot for Caloriesconsumed vs Weightgained (Plot x,y)**

plot(Salary\_hike,Churn\_out\_rate)

**#calculate correlation coefficient**

cor(Salary\_hike,Churn\_out\_rate)

**#Simple Regression model**

reg <- lm(Churn\_out\_rate~Salary\_hike,data =SH\_CO)

summary(reg)

**#values prediction**

**#Confidence interval Calculation**

confint(reg,level = 0.95)

pred <- predict(reg,interval = "predict")

**#predict function gives fit value and its lower and upeer values as a range**

pred <- as.data.frame(pred)

pred

**#####Plot Graph for both Actual values and also the predicted linear Graph(Actual:Red,Predicted:Blue)#########**

ggplot() +

geom\_point(aes(x =`Salary\_hike` , y =`Churn\_out\_rate` ),colour='red') + geom\_line(aes(x = `Salary\_hike`, y = predict(reg, newdata=SH\_CO)),colour='blue') + ggtitle('Salary\_hike vs Churn\_out\_rate') +

xlab('Salary\_hike') +ylab('Churn\_out\_rate')

cor(pred$fit,`Churn\_out\_rate`)

**#Calculate Residuals "Errors"**

reg$residuals

reg$residuals^2

mean(reg$residuals^2)

rmse <- sqrt(mean(reg$residuals^2))

rmse

**############ Applying transformations##############**

**############ lOGORITHMIC MODEL x = log(Salary\_hike); y = Churn\_out\_rate ############**

plot(log(`Salary\_hike`),`Churn\_out\_rate`)

cor(log(`Salary\_hike`),`Churn\_out\_rate`)

log\_reg <- lm(`Churn\_out\_rate` ~ log(`Salary\_hike`),data = SH\_CO)

summary(log\_reg)

**#values prediction**

**#Confidence interval Calculation**

confint(log\_reg,level = 0.95)

pred\_log <- predict(log\_reg,interval ="predict")

#predict function gives fit value and its lower and upeer values as a range

pred\_log <- as.data.frame(pred\_log)

pred\_log

rmse\_log <- sqrt(mean(log\_reg$residuals^2))

rmse\_log

**##########Plot Graph for both Actual values and also the predicted linear Graph(Actual:Red,Predicted:Blue)#########**

ggplot() + geom\_point(aes(x =`Salary\_hike` , y =`Churn\_out\_rate` ),colour='red') +

geom\_line(aes(x =`Salary\_hike`, y = predict(log\_reg, newdata=SH\_CO)),colour='blue') +

ggtitle('Salary\_hike vs Churn\_out\_rate') + xlab('Salary\_hike') +ylab('Churn\_out\_rate')

**############Polynomial model with 2 degree (quadratic model) ;x =Salary\_hike^2 ; y = Churn\_out\_rate ############**

**#### input=x & X^2 (2-degree); output=y ####**

reg\_quad2<- lm(`Churn\_out\_rate` ~ `Salary\_hike`+I(`Salary\_hike`\*`Salary\_hike`),data = SH\_CO)

summary(reg\_quad2)

**#prediction**

**#Confidence interval Calculation**

confint(reg\_quad2,level = 0.95)

pred\_quad2<-predict(reg\_quad2,interval = "predict")

pred\_quad2 <- as.data.frame(pred\_quad2)

pred\_quad2

resq=`Churn\_out\_rate`-pred\_quad2$fit

rmse\_quad<-sqrt(mean(resq^2))

rmse\_quad

**##########Plot Graph for both Actual values and also the predicted linear Graph(Actual:Red,Predicted:Blue)#########**

ggplot() + geom\_point(aes(x =`Salary\_hike` , y =`Churn\_out\_rate` ), colour='red') + geom\_line(aes(x = `Salary\_hike`, y = predict(reg\_quad2, newdata=SH\_CO)), colour='blue') + ggtitle('Salary\_hike vs Churn\_out\_rate') +xlab('Salary\_hike') +ylab('Churn\_out\_rate')

**############Polynomial model with 3 degree (quadratic model) ;x = Salary\_hike^3; y = Churn\_out\_rate ############**

**#### input=x & X^2 & x^3 (3-degree); output=y ####**

reg\_quad3<- lm(`Churn\_out\_rate` ~ `Salary\_hike`+I(`Salary\_hike`\*`Salary\_hike`)+I(`Salary\_hike`\*`Salary\_hike`\*`Salary\_hike`),data = SH\_CO)

summary(reg\_quad3)

**#prediction**

**#Confidence interval Calculation**

confint(reg\_quad3,level = 0.95)

pred\_quad3<-predict(reg\_quad3,interval = "predict")

pred\_quad3 <- as.data.frame(pred\_quad3)

pred\_quad3

resq3=`Churn\_out\_rate`-pred\_quad3$fit

rmse\_quad3<-sqrt(mean(resq3^2))

rmse\_quad3

**##########Plot Graph for both Actual values and also the predicted linear Graph(Actual:Red,Predicted:Blue)#########**

ggplot() + geom\_point(aes(x =`Salary\_hike` , y =`Churn\_out\_rate` ),colour='red') +

geom\_line(aes(x = `Salary\_hike`, y = predict(reg\_quad3, newdata= SH\_CO)),colour='blue') +

ggtitle('Salary\_hike vs Churn\_out\_rate') +xlab('Salary\_hike') +ylab('Churn\_out\_rate')

################################################################################

**PYTHON:**

**# For reading data set**

**# importing necessary libraries**

import pandas as pd **# deals with data frame**

import numpy as **np # deals with numerical values**

SH\_CO = pd.read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/emp\_data.csv")

import matplotlib.pylab as plt **#for different types of plots**

SH\_CO.Salary\_hike

plt.scatter(x=SH\_CO['Salary\_hike'], y=SH\_CO['Churn\_out\_rate'],color='green'**)# Scatter plot**

np.corrcoef(SH\_CO.Salary\_hike, SH\_CO.Churn\_out\_rate) **#correlation**

help(np.corrcoef)

import statsmodels.formula.api as smf

plt.hist(SH\_CO["Salary\_hike"])

plt.hist(SH\_CO["Churn\_out\_rate"])

model = smf.ols('Churn\_out\_rate ~ Salary\_hike', data=SH\_CO).fit()

model.summary()

**#values prediction**

**#Confidence interval Calculation**

pred1 = model.predict(pd.DataFrame(SH\_CO['Salary\_hike']))

pred1

print (model.conf\_int(0.95)) **# 95% confidence interval**

res = SH\_CO.Churn\_out\_rate - pred1

sqres = res\*res

mse = np.mean(sqres)

rmse = np.sqrt(mse)

**######### Model building on Transformed Data#############**

**# Log Transformation**

**# x = log(Salary\_hike); y = Churn\_out\_rate**

plt.scatter(x=np.log(SH\_CO['Salary\_hike']),y=SH\_CO['Churn\_out\_rate'],color='brown')

np.corrcoef(np.log(SH\_CO.Salary\_hike), SH\_CO.Churn\_out\_rate**) #correlation**

model2 = smf.ols('Churn\_out\_rate ~ np.log(Salary\_hike)',data=SH\_CO).fit()

model2.summary()

pred2 = model2.predict(pd.DataFrame(SH\_CO['Salary\_hike']))

pred2

print(model2.conf\_int(0.95)) # 95% confidence level

res2 = SH\_CO.Churn\_out\_rate - pred2

sqres2 = res2\*res2

mse2 = np.mean(sqres2)

rmse2 = np.sqrt(mse2)

**############Polynomial model with 2 degree (quadratic model) ;x = Salary\_hike\*Salary\_hike; y = Churn\_out\_rate############**

**#### input=x & X^2 (2-degree); output=y ####**

model4 = smf.ols('Churn\_out\_rate ~ Salary\_hike+I(Salary\_hike\*Salary\_hike)', data=SH\_CO).fit()

model4.summary()

pred\_p2 = model4.predict(pd.DataFrame(SH\_CO['Salary\_hike']))

pred\_p2

print(model4.conf\_int(0.95)) # 95% confidence level

res4 = SH\_CO.Churn\_out\_rate - pred\_p2

sqres4 = res4\*res4

mse4 = np.mean(sqres4)

rmse4 = np.sqrt(mse4)

**###########Polynomial model with 3 degree (quadratic model) ;x = Salary\_hike\*Salary\_hike\*Salary\_hike; y = Churn\_out\_rate############**

**#### input=x & X^2 (2-degree); output=y ####**

model5 = smf.ols('Churn\_out\_rate ~ Salary\_hike+I(Salary\_hike\*Salary\_hike)+I(Salary\_hike\*Salary\_hike\*Salary\_hike)', data=SH\_CO).fit()

model5.summary()

pred\_p3 = model5.predict(pd.DataFrame(SH\_CO['Salary\_hike']))

pred\_p3

print(model5.conf\_int(0.95)) **# 95% confidence level**

res5 = SH\_CO.Churn\_out\_rate - pred\_p3

sqres5 = res5\*res5

mse5 = np.mean(sqres5)

rmse5 = np.sqrt(mse5)