***SALARY and YEARS EXPERIENCE***

**Salary\_hike -> Build a prediction model for Salary\_hike**

**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:**

YearsExperience

Salary

**Data Set Size:** 30

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

**Years of experience:**

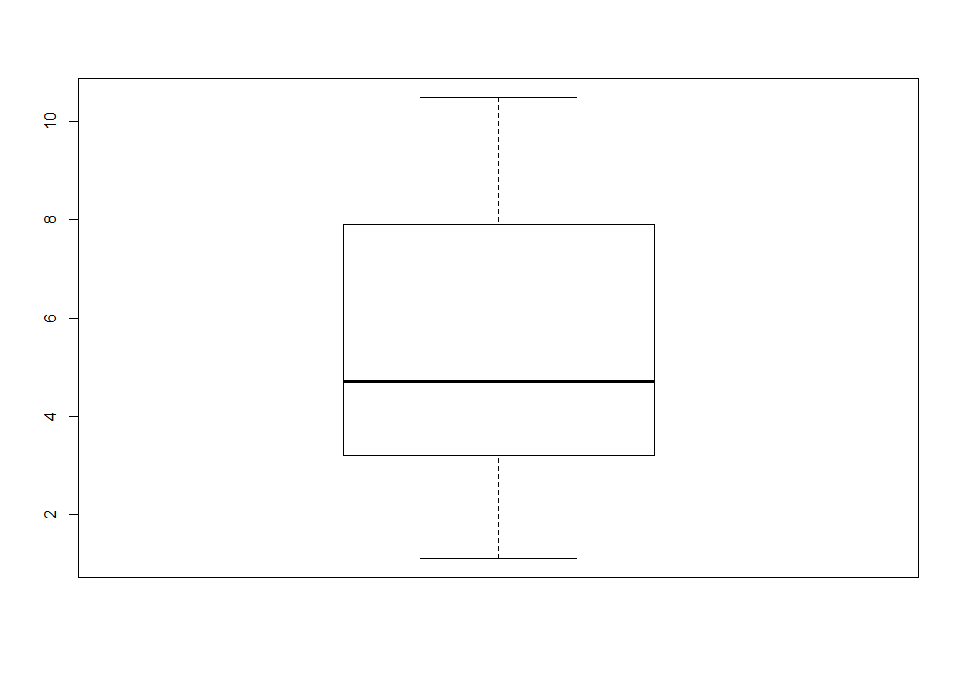
Ranges between 1.1-10.5

For this Years of experience the mean is 5.31, it is just the average of the Years of experience data

The median for the given data is 4.7, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=4.7>mean-5.31), 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= 0.36



**Salary:**

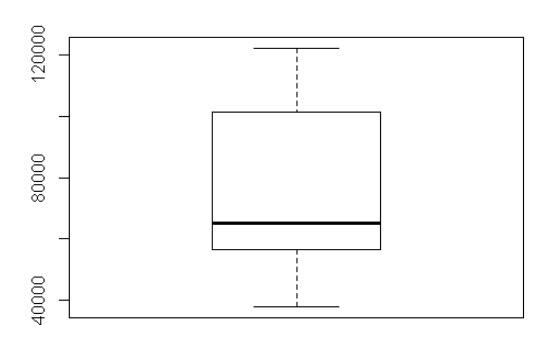
Ranges between 37731-122391

For this Years of experience the mean is 76003 , it is just the average of the Years of experience data

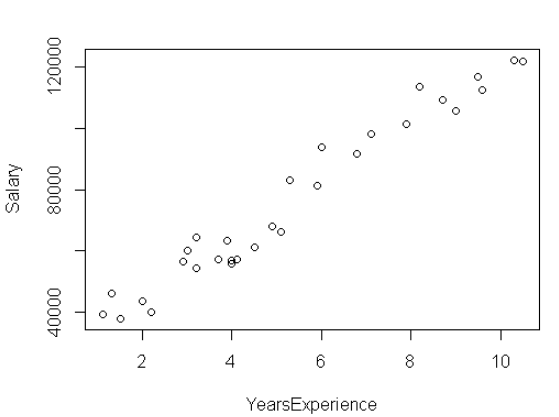
The median for the given data is 65237, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=65237<mean-76003), if data was not skewed, we would have considered mean but hear it is skewed so we take Median to talk about data.

Skewness =0.336



**Plot for Salary vs Years of experience**



The above scatter diagram infer that the Salary and Years of experience are moderately positive correlated.

**Correlation Coefficient:**

Let’s see the relationship between the Salary and Years of experience

**cor(Salary\_exp$YearsExperience,Salary\_exp$Salary)**

**0.9782416**

Based on the correlation value obtained which is 0.97(approx.) also tells that it is Positive correlation

We use **lm() function from Base Package in R-Studio** to estimate the Years of experience using the other variable Salary whereas in **python LinearRegression() is used from the sklearn package**

Call:

lm(formula = Salary\_exp$Salary ~ Salary\_exp$YearsExperience,

data = Salary\_exp)

Residuals:

Min 1Q Median 3Q Max

-7958.0 -4088.5 -459.9 3372.6 11448.0

Coefficients:

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

(Intercept) 25792.2 2273.1 11.35 5.51e-12 \*\*\*

Salary\_exp$YearsExperience 9450.0 378.8 24.95 < 2e-16 \*\*\*

---

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

Residual standard error: 5788 on 28 degrees of freedom

Multiple R-squared: 0.957, Adjusted R-squared: 0.9554

F-statistic: 622.5 on 1 and 28 DF, p-value: < 2.2e-16

**P-values:**

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

Look at the r-squared values are 0.957

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** |
| SL | YE | 0.98 | 0.957 | 5592.04 | SLR |  |
| SL | log(YE) | 0.92 | 0.853 | 10302.9 | LT | C:\Users\RAVI\Desktop\log |
| log(SL) | YE | 0.97 | 0.932 | 7213.24 | ET |  |
| SL | YE\*YE | 0.98 | 0.957 | 5590.84 | QM-2D | C:\Users\RAVI\Desktop\q2 |
| **SL** | **YE\*YE\*YE** | **0.98** | **0.963** | **5142.64** | **QM-3D** | C:\Users\RAVI\Desktop\q3 |

YE= Years Experience QM= quadratic model ET= Exponential Transformation

SLR=Simple linear Regression , LT= Logorithmic Transformation, SL=Salary

Based on obtained R-squared values, RMSE and the plot the best transformation technique is Polynomial 3Degree with 0.96 R-squared value

**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

**# Simple Linear Regression Assignment #**

**# 4) Salary\_hike -> Build a prediction model for Salary\_hike**

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

library(readr)

library(ggplot2)

library(moments)

Salary\_exp <- read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/Salary\_Data.csv")

View(Salary\_exp)

attach(Salary\_exp)

summary(Salary\_exp)

mean(YearsExperience)

skewness(YearsExperience)

skewness(Salary\_exp$Salary)

#Exploratory Data Analysis

boxplot(YearsExperience)

boxplot(Salary,horizontal = T)

**#scatter plot for YearsExperience vs Salary (Plot x,y)**

plot(Salary\_exp$YearsExperience,Salary\_exp$Salary)

**#calculate correlation coefficient**

cor(Salary\_exp$YearsExperience,Salary\_exp$Salary)

**#Simple Regression model**

reg <- lm(Salary\_exp$Salary ~ Salary\_exp$YearsExperience,data = Salary\_exp)

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\_exp$YearsExperience, y = Salary\_exp$Salary),

colour='red') +

geom\_line(aes(x = Salary\_exp$YearsExperience, y = predict(reg, newdata=Salary\_exp)),

colour='blue') +

ggtitle('Salary vs Experience (Salary\_exp)') +

xlab('Years of experience') +

ylab('Salary')

cor(pred$fit,Salary\_exp$Salary)

**#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(YearsExperience); y = Salary ############**

plot(log(Salary\_exp$YearsExperience),Salary\_exp$Salary)

cor(log(Salary\_exp$YearsExperience),Salary\_exp$Salary)

log\_reg <- lm(Salary\_exp$Salary ~ log(Salary\_exp$YearsExperience))

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

log\_reg$residuals

cor(pred\_log$fit,Salary)

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\_exp$YearsExperience, y = Salary\_exp$Salary),

colour='red') +

geom\_line(aes(x = Salary\_exp$YearsExperience, y = predict(log\_reg, newdata=Salary\_exp)),

colour='blue') +

ggtitle('Salary vs Experience (Salary\_exp)') +

xlab('Years of experience') +

ylab('Salary')

**####################END OF lOGORITHMIC MODEL ##########################**

**############ EXPONENTIAL MODEL x = YearsExperience; y = log(Salary) ############**

plot(Salary\_exp$YearsExperience,log(Salary\_exp$Salary))

cor(Salary\_exp$YearsExperience,log(Salary\_exp$Salary))

log\_reg2 <- lm(log(Salary\_exp$Salary) ~ Salary\_exp$YearsExperience)

summary(log\_reg2)

**#values prediction**

**#Confidence interval Calculation**

confint(log\_reg2,level = 0.95)

pred\_log2 <- predict(log\_reg2,interval ="predict")

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

pred\_log2 <- as.data.frame(pred\_log2)

log\_reg2$residuals **#output is log(AT) so we are getting less values apply antilog**

pred<- exp(pred\_log2) **#anti-log=exponential**

pred

cor(pred$fit,Salary\_exp$Salary)

res\_log2=Salary\_exp$Salary-pred$fit

rmse2 <- sqrt(mean(res\_log2^2))

rmse2

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

ggplot() +

geom\_point(aes(x = Salary\_exp$YearsExperience, y = Salary\_exp$Salary),

colour='red') +

geom\_line(aes(x = Salary\_exp$YearsExperience, y = predict(log\_reg2, newdata=Salary\_exp)),

colour='blue') +

ggtitle('Salary vs Experience (Salary\_exp)') +

xlab('Years of experience') +

ylab('Salary')

**####################END OF EXPONENTIAL MODEL ##########################**

**############Polynomial model with 2 degree (quadratic model) ;x = YearsExperience\*YearsExperience; y = log(Salary)############**

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

reg\_quad2<- lm(Salary\_exp$Salary ~ Salary\_exp$YearsExperience+I(Salary\_exp$YearsExperience\*Salary\_exp$YearsExperience),data =Salary\_exp)

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

cor(pred\_quad2$fit,Salary\_exp$Salary)

resq=Salary\_exp$Salary-pred\_quad2$fit

rmse\_quad<-sqrt(mean(resq$fit^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\_exp$YearsExperience, y = Salary\_exp$Salary),

colour='red') +

geom\_line(aes(x = Salary\_exp$YearsExperience, y = predict(reg\_quad2, newdata=Salary\_exp)),

colour='blue') +

ggtitle('Salary vs Experience (Salary\_exp)') +

xlab('Years of experience') +

ylab('Salary')

**############ END OF Polynomial model with 2 degree (quadratic model) ############**

**############Polynomial model with 3 degree (quadratic model) ;x = YearsExperience\*YearsExperience\*YearsExperience; y = Salary############**

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

reg\_quad3<- lm(Salary\_exp$Salary ~ Salary\_exp$YearsExperience+I(Salary\_exp$YearsExperience\*Salary\_exp$YearsExperience)+I(Salary\_exp$YearsExperience\*Salary\_exp$YearsExperience\*Salary\_exp$YearsExperience),data =Salary\_exp)

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

cor(pred\_quad3$fit,Salary\_exp$Salary)

resq3=Salary\_exp$Salary-pred\_quad3$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\_exp$YearsExperience, y = Salary\_exp$Salary),

colour='red') +

geom\_line(aes(x = Salary\_exp$YearsExperience, y = predict(reg\_quad3, newdata=Salary\_exp)),

colour='blue') +

ggtitle('Salary vs Experience (Salary\_exp)') +

xlab('Years of experience') +

ylab('Salary')

**############ END OF Polynomial model with 3 degree (quadratic model) ############**

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

**#not random sampling**

**train <- Salary\_exp[1:24,]**

**test <- Salary\_exp[25:30,]**

**#random sampling**

**#n <- nrow(Salary\_exp)**

**#n1 <- n\*0.8**

**#n1**

**#n2 <- n-n1**

**#n2**

**#train\_ind <-sample(1:n,n1)**

**#train <- Salary\_exp[train\_ind,]**

**#test <-Salary\_exp[-train\_ind,]**

**model<- lm(Salary~YearsExperience+I(YearsExperience\*YearsExperience)+I(YearsExperience\*YearsExperience\*YearsExperience),data = train)**

**summary(model)**

**#prediction**

**confint(model,level = 0.95)**

**#test data**

**res<-predict(model,interval = "confidence",newdata = test)**

**predict\_original <- as.data.frame(res)**

**predict\_original**

**test\_error <- test$Salary-predict\_original$fit #calculate error/residual**

**test\_error**

**test\_rmse <- sqrt(mean(test\_error^2))**

**test\_rmse**

**#training data**

**res\_train <-predict(model,interval = "confidence",newdata = train)**

**predict\_original\_train <- as.data.frame(res\_train)**

**train\_error <- train$Salary - predict\_original\_train$fit #calculate error/residual**

**train\_error**

**train\_rmse <- sqrt(mean(train\_error^2))**

**train\_rmse**

**##########################################################################**

**PYTHON CODE:**

**# For reading data set**

**# importing necessary libraries**

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

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

Salary\_exp = pd.read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/Salary\_Data.csv")

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

plt.scatter(x=Salary\_exp['YearsExperience'], y=Salary\_exp['Salary'],color='green'**)# Scatter plot**

np.corrcoef(Salary\_exp.YearsExperience, Salary\_exp.Salary) **#correlation**

help(np.corrcoef)

import statsmodels.formula.api as smf

plt.hist(Salary\_exp["YearsExperience"])

model = smf.ols('Salary ~ YearsExperience', data=Salary\_exp).fit()

model.summary()

**#values prediction**

**#Confidence interval Calculation**

pred1 = model.predict(pd.DataFrame(Salary\_exp['YearsExperience']))

pred1

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

res = Salary\_exp.Salary - pred1

sqres = res\*res

mse = np.mean(sqres)

rmse = np.sqrt(mse)

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

**# Log Transformation**

**# x = log(YearsExperience); y = Salary**

plt.scatter(x=np.log(Salary\_exp['YearsExperience']),y=Salary\_exp['Salary'],color='brown')

np.corrcoef(np.log(Salary\_exp.YearsExperience), Salary\_exp.Salary) **#correlation**

model2 = smf.ols('Salary ~ np.log(YearsExperience)',data=Salary\_exp).fit()

model2.summary()

pred2 = model2.predict(pd.DataFrame(Salary\_exp['YearsExperience']))

pred2

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

res2 = Salary\_exp.Salary - pred2

sqres2 = res2\*res2

mse2 = np.mean(sqres2)

rmse2 = np.sqrt(mse2)

**# Exponential transformation**

plt.scatter(x=Salary\_exp['YearsExperience'], y=np.log(Salary\_exp['Salary']),color='orange')

np.corrcoef(Salary\_exp.YearsExperience, np.log(Salary\_exp.Salary)) **#correlation**

model3 = smf.ols('np.log(Salary) ~ YearsExperience',data=Salary\_exp).fit()

model3.summary()

pred\_log = model3.predict(pd.DataFrame(Salary\_exp['YearsExperience']))

pred\_log

pred3 = np.exp(pred\_log)

pred3

print(model3.conf\_int(0.95)) **# 95% confidence level**

res3 = Salary\_exp.Salary - pred3

sqres3 = res3\*res3

mse3 = np.mean(sqres3)

rmse3 = np.sqrt(mse3)

**############Polynomial model with 2 degree (quadratic model) ;x = YearsExperience\*YearsExperience; y = Salary############**

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

model4 = smf.ols('Salary ~ YearsExperience+I(YearsExperience\*YearsExperience)', data=Salary\_exp).fit()

model4.summary()

pred\_p2 = model4.predict(pd.DataFrame(Salary\_exp['YearsExperience']))

pred\_p2

print(model3.conf\_int(0.95)) **# 95% confidence level**

res4 = Salary\_exp.Salary - pred\_p2

sqres4 = res4\*res4

mse4 = np.mean(sqres4)

rmse4 = np.sqrt(mse4)

**###########Polynomial model with 3 degree (quadratic model) ;x = YearsExperience\*YearsExperience\*YearsExperience; y = Salary############**

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

model5 = smf.ols('Salary ~ YearsExperience+I(YearsExperience\*YearsExperience)+I(YearsExperience\*YearsExperience\*YearsExperience)', data=Salary\_exp).fit()

model5.summary()

pred\_p3 = model5.predict(pd.DataFrame(Salary\_exp['YearsExperience']))

pred\_p3

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

res5 = Salary\_exp.Salary - pred\_p3

sqres5 = res5\*res5

mse5 = np.mean(sqres5)

rmse5 = np.sqrt(mse5)