**Calories Consumed *and* Weight gained**

**Calories\_consumed-> predict weight gained using calories consumed**

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

**Inferences from the Data Set:**

Data Set talks about the Calories Consumedwith respect to Weight gained with 14

Observations

**Columns:**

|  |  |
| --- | --- |
| Weight gained (grams) | |
| Calories Consumed |

**Data Set Size:** 14

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

**Weight gained :**

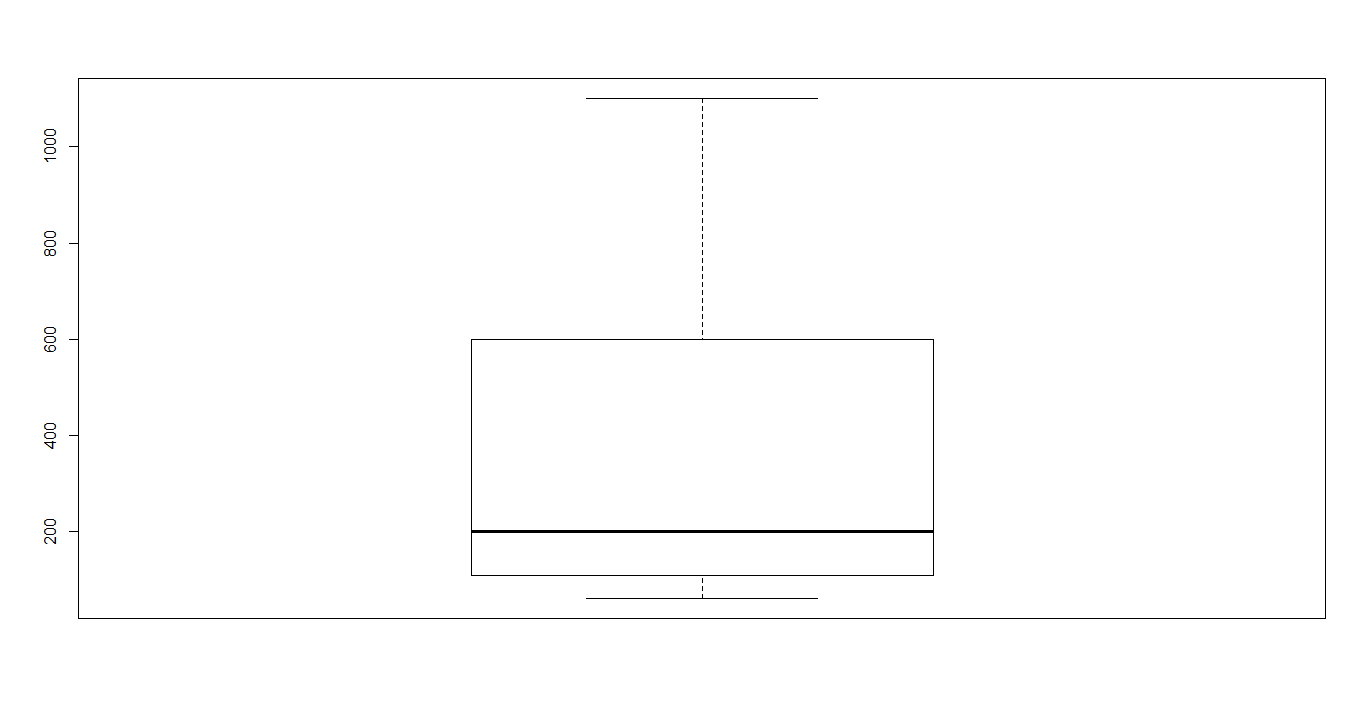
Ranges between 62-1100

For this Weight gained the mean is 357.7, it is just the average of the Weight gained data

The median for the given data is 200, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=200>mean-357.7), 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= 1.11



**Calories Consumed :**

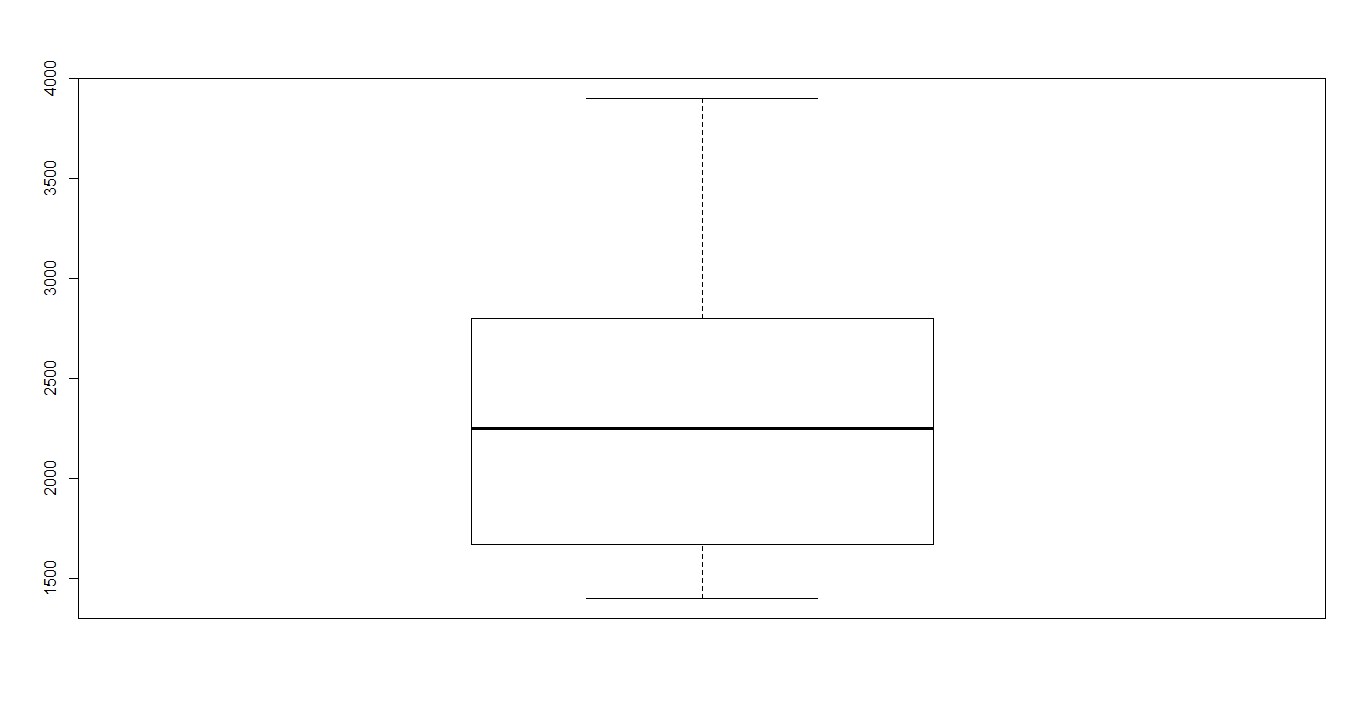
Ranges between 1400-3900

For this Calories Consumed the mean is 2341 , it is just the average of the Calories Consumed data

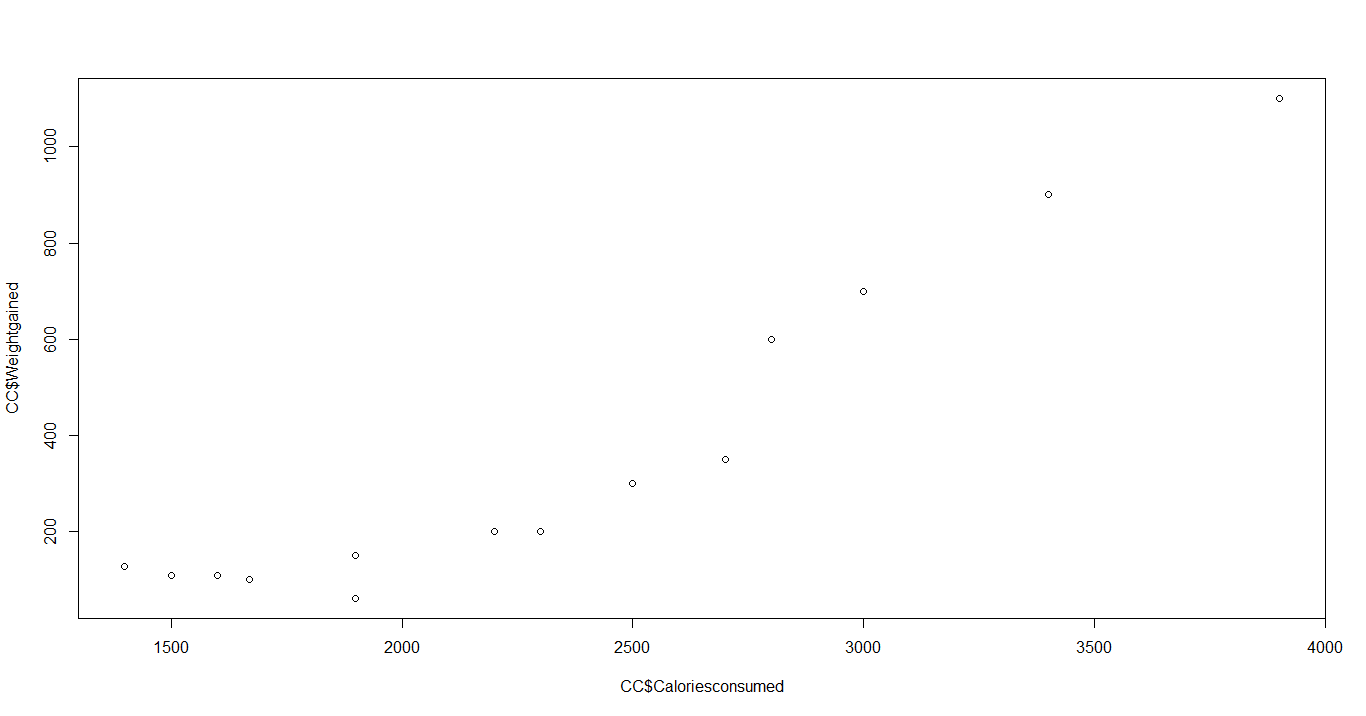
The median for the given data is 2250, it speaks about the center of data

A comparison between mean and median tell us that data is skewed (median=2250<mean-2341), if data was not skewed, we would have considered mean but hear it is skewed so we take Median to talk about data.

Skewness =0.58



**Plot for Calories Consumed *vs* Weight gained:**

****

The above scatter diagram infer that the Weight gained and Calories Consumedare moderately positive correlated.

**Correlation Coefficient:**

Let’s see the relationship between the Weight gained and Calories Consumed

**cor(CC$Caloriesconsumed,CC$Weightgained)**

**0.946991**

Based on the correlation value obtained which is 0.94 also tells that it is Positive correlation

We use **lm() function from Base Package in R-Studio** to estimate the Weight gained using the other variable Calories Consumedwhereas in **python LinearRegression() is used from the sklearn package**

> reg <- lm(CC$Weightgained~CC$Caloriesconsumed,data = CC)

> summary(reg)

Call:

lm(formula = CC$Weightgained ~ CC$Caloriesconsumed, data = CC)

Residuals:

Min 1Q Median 3Q Max

-158.67 -107.56 36.70 81.68 165.53

Coefficients:

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

(Intercept) -625.75236 100.82293 -6.206 4.54e-05 \*\*\*

CC$Caloriesconsumed 0.42016 0.04115 10.211 2.86e-07 \*\*\*

---

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

Residual standard error: 111.6 on 12 degrees of freedom

Multiple R-squared: 0.8968, Adjusted R-squared: 0.8882

F-statistic: 104.3 on 1 and 12 DF, p-value: 2.856e-07

**P-values:**

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

Look at the r-squared values are 0.89

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** |
| WG | CC | 0.946 | 0.896 | 103.3 | SLR | C:\Users\RAVI\Desktop\SR |
| WG | log(CC) | 0.898 | 0.807 | 141 | LT | C:\Users\RAVI\Desktop\G |
| WG | CC\*CC | 0.97 | 0.95 | 70.4 | QM-2D | C:\Users\RAVI\Desktop\Q2 |
| WG | CC\*CC\*CC | 0.99 | 0.98 | 44.15 | QM-3D | C:\Users\RAVI\Desktop\Q3 |

|  |  |
| --- | --- |
| WG=weight gained | CC=Caloriesconsumed |
| SLR=Simple linear regression | LT=log transformation |
| QM2D=Quadratic model 2Degree | QM3D=Quadratic model 3Degree |

Based on obtained R-squared values, RMSE and the plot the best transformation technique is Polynomial 3Degree with 0.98 R-squared value and RMSE 44.15

**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

**# Simple Linear Regression Assignment #**

**# 1) Calories\_consumed-> predict weight gained using calories consumed**

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

library(readr)

library(ggplot2)

library(moments)

CC <- read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/calories\_consumed.csv")

View(CC)

attach(CC)

colnames(CC) <- c("Weightgained","Caloriesconsumed")

View(CC)

summary(CC)

range(CC$Caloriesconsumed)

range(CC$Weightgained)

skewness(CC$Weightgained)

skewness(CC$Caloriesconsumed)

**#Exploratory Data Analysis**

boxplot(CC$Weightgained)

boxplot(CC$Caloriesconsumed)

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

plot(CC$Caloriesconsumed,CC$Weightgained)

**#calculate correlation coefficient**

cor(CC$Caloriesconsumed,CC$Weightgained)

**#Simple Regression model**

reg <- lm(CC$Weightgained~CC$Caloriesconsumed,data = CC)

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 =CC$Caloriesconsumed , y =CC$Weightgained ),

colour='red') +

geom\_line(aes(x = CC$Caloriesconsumed, y = predict(reg, newdata=CC)),

colour='blue') +

ggtitle('Caloriesconsumed vs Weightgained') +

xlab('Caloriesconsumed') +

ylab('Weightgained')

cor(pred$fit,CC$Weightgained)

**#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(Caloriesconsumed); y = Weightgained ############**

plot(log(CC$Caloriesconsumed),CC$Weightgained)

cor(log(CC$Caloriesconsumed),CC$Weightgained)

log\_reg <- lm(CC$Weightgained ~ log(CC$Caloriesconsumed),data = CC)

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

cor(pred\_log$fit,CC$Weightgained)

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 =CC$Caloriesconsumed , y =CC$Weightgained ),

colour='red') +

geom\_line(aes(x = CC$Caloriesconsumed, y = predict(log\_reg, newdata=CC)),

colour='blue') +

ggtitle('Caloriesconsumed vs Weightgained') +

xlab('Caloriesconsumed') +

ylab('Weightgained')

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

**############Polynomial model with 2 degree (quadratic model) ;x =Caloriesconsumed^2 ; y = Weightgained ############**

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

reg\_quad2<- lm(CC$Weightgained ~ CC$Caloriesconsumed+I(CC$Caloriesconsumed\*CC$Caloriesconsumed),data =CC)

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=CC$Weightgained-pred\_quad2$fit

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

rmse\_quad

cor(pred\_quad2$fit,CC$Weightgained)

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

ggplot() +

geom\_point(aes(x =CC$Caloriesconsumed , y =CC$Weightgained ),

colour='red') +

geom\_line(aes(x = CC$Caloriesconsumed, y = predict(reg\_quad2,data=CC)),

colour='blue') +

ggtitle('Caloriesconsumed vs Weightgained') +

xlab('Caloriesconsumed') +

ylab('Weightgained')

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

**############Polynomial model with 3 degree (quadratic model) ;x = Caloriesconsumed^3; y = Weightgained ############**

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

reg\_quad3<- lm(CC$Weightgained ~ CC$Caloriesconsumed+I(CC$Caloriesconsumed\*CC$Caloriesconsumed)+I(CC$Caloriesconsumed\*CC$Caloriesconsumed\*CC$Caloriesconsumed),data =CC)

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$fit,CC$Weightgained)

resq3=CC$Weightgained-pred\_quad3$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 =CC$Caloriesconsumed , y =CC$Weightgained ),

colour='red') +

geom\_line(aes(x = CC$Caloriesconsumed, y = predict(reg\_quad3,data=CC)),

colour='blue') +

ggtitle('Caloriesconsumed vs Weightgained') +

xlab('Caloriesconsumed') +

ylab('Weightgained')

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

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

**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

CC = pd.read\_csv("C:/RAVI/Data science/Assignments/Module 6 Simple linear regression/DataSets/calories\_consumed.csv")

CC.columns="Weightgained","Caloriesconsumed"

CC

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

plt.scatter(x=CC['Caloriesconsumed'], y=CC['Weightgained'],color='green'**)# Scatter plot**

np.corrcoef(CC.Caloriesconsumed, CC.Weightgained) **#correlation**

help(np.corrcoef)

import statsmodels.formula.api as smf

plt.hist(CC["Weightgained"])

plt.hist(CC["Caloriesconsumed"])

model = smf.ols('Weightgained ~ Caloriesconsumed', data=CC).fit()

model.summary()

**#values prediction**

**#Confidence interval Calculation**

pred1 = model.predict(pd.DataFrame(CC['Caloriesconsumed']))

pred1

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

res = CC.Weightgained - pred1

sqres = res\*res

mse = np.mean(sqres)

rmse = np.sqrt(mse)

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

**# Log Transformation**

**# x = log(Caloriesconsumed); y = Weightgained**

plt.scatter(x=np.log(CC['Caloriesconsumed']),y=CC['Weightgained'],color='brown')

np.corrcoef(np.log(CC.Caloriesconsumed), CC.Weightgained**) #correlation**

model2 = smf.ols('Weightgained ~ np.log(Caloriesconsumed)',data=CC).fit()

model2.summary()

pred2 = model2.predict(pd.DataFrame(CC['Caloriesconsumed']))

pred2

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

res2 = CC.Weightgained - pred2

sqres2 = res2\*res2

mse2 = np.mean(sqres2)

rmse2 = np.sqrt(mse2)

**# Exponential transformation**

plt.scatter(x=CC['Caloriesconsumed'], y=np.log(CC['Weightgained']),color='orange')

np.corrcoef(CC.Caloriesconsumed, np.log(CC.Weightgained)) #correlation

model3 = smf.ols('np.log(Weightgained) ~ Caloriesconsumed',data=CC).fit()

model3.summary()

pred\_log = model3.predict(pd.DataFrame(CC['Caloriesconsumed']))

pred\_log

pred3 = np.exp(pred\_log)

pred3

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

res3 = CC.Weightgained - pred3

sqres3 = res3\*res3

mse3 = np.mean(sqres3)

rmse3 = np.sqrt(mse3)

**############Polynomial model with 2 degree (quadratic model) ;x = Caloriesconsumed\*Caloriesconsumed; y = Weightgained############**

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

model4 = smf.ols('Weightgained ~ Caloriesconsumed+I(Caloriesconsumed\*Caloriesconsumed)', data=CC).fit()

model4.summary()

pred\_p2 = model4.predict(pd.DataFrame(CC['Caloriesconsumed']))

pred\_p2

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

res4 = CC.Weightgained - pred\_p2

sqres4 = res4\*res4

mse4 = np.mean(sqres4)

rmse4 = np.sqrt(mse4)

**###########Polynomial model with 3 degree (quadratic model) ;x = Caloriesconsumed\*Caloriesconsumed\*Caloriesconsumed; y = Weightgained############**

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

model5 = smf.ols('Weightgained ~ Caloriesconsumed+I(Caloriesconsumed\*Caloriesconsumed)+I(Caloriesconsumed\*Caloriesconsumed\*Caloriesconsumed)', data=CC).fit()

model5.summary()

pred\_p3 = model5.predict(pd.DataFrame(CC['Caloriesconsumed']))

pred\_p3

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

res5 = CC.Weightgained - pred\_p3

sqres5 = res5\*res5

mse5 = np.mean(sqres5)

rmse5 = np.sqrt(mse5)