**AIM: IMPLEMENTATION AND ANALYSIS OF LINEAR REGRESSION THROUGH GRAPHICAL METHODS.**

**THEORY:**

Linear regression is used for finding linear relationship between target and one or more predictors. There are two types of linear regression- Simple and Multiple.

Linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables. For example, relationship between height and weight.

The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

**Y(pred) = b0 + b1\*x**

The values b0 and b1 must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

**Error Calculation**



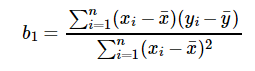
If we don’t square the error, then positive and negative point will cancel out each other.

For model with one predictor,

**Intercept Calculation:**



**Co-efficient Formula:**



1. **SIMPLE LINEAR REGRESSION:**

**SOURCE CODE:**

**LOADING LIBRARY & DATASET:**

#load library

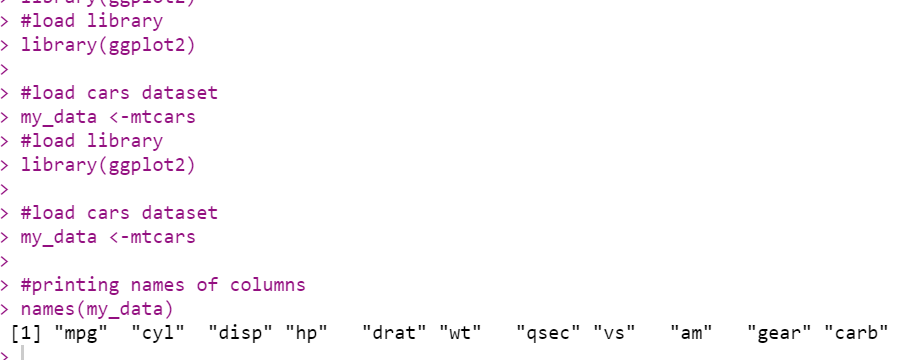
library(ggplot2)

#load cars dataset

my\_data <-mtcars

#printing names of columns

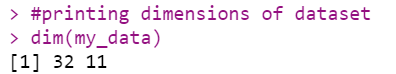
names(my\_data)



**PRINTING DIMENSIONS OF DATASET:**

#printing dimensions of dataset

dim(my\_data)

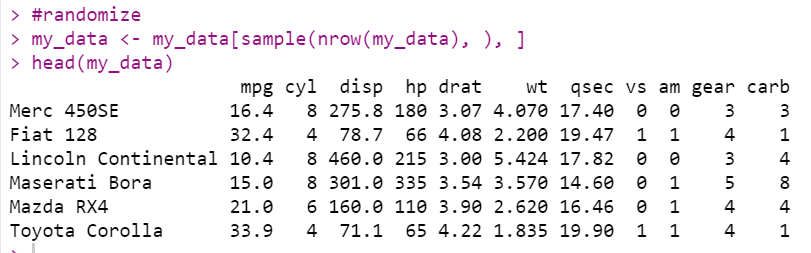


**CREATING RANDOM SAMPLE:**

#randomize

my\_data <- my\_data[sample(nrow(my\_data), ), ]

head(my\_data)



**CREATING TRAINING & TESTING DATASET:**

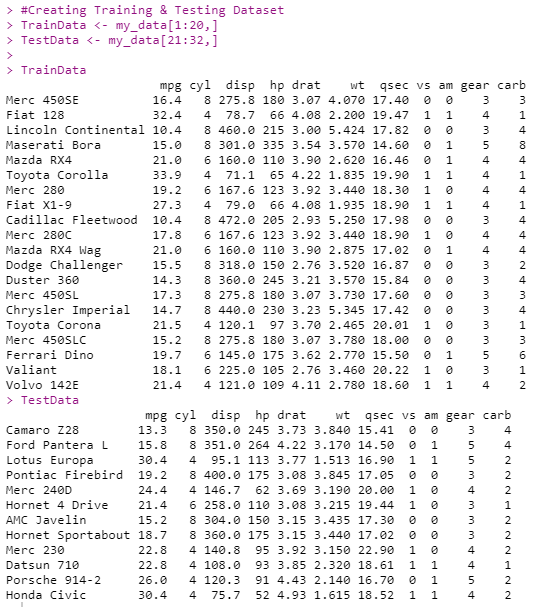
#Creating Training & Testing Dataset

TrainData <- my\_data[1:20,]

TestData <- my\_data[21:32,]

TrainData

TestData



**CREATING LINEAR REGRESSION MODEL AND PRINTING RESULTS:**

## Linear Model

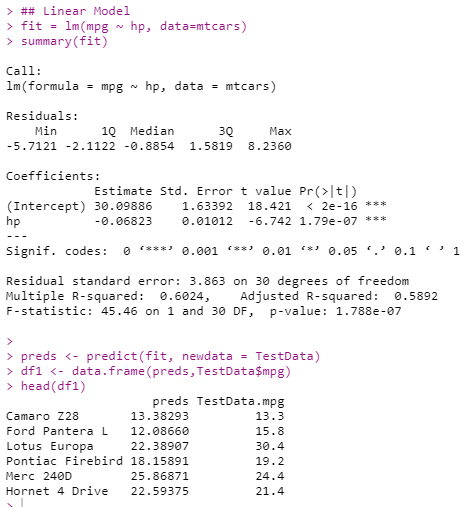
fit = lm(mpg ~ hp, data=mtcars)

summary(fit)

preds <- predict(fit, newdata = TestData)

df1 <- data.frame(preds,TestData$mpg)

head(df1)



**CALCULATE CORRELATION:**

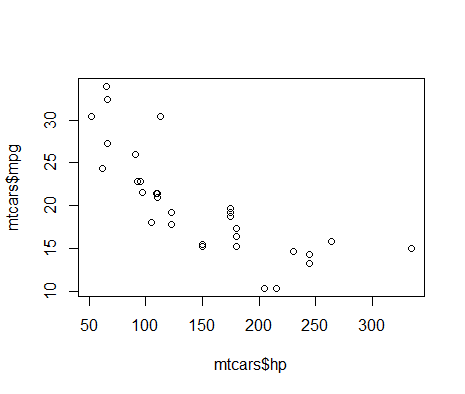
#correlation

cor(preds,TestData$mpg)



**PLOTTING POINTS:**

plot(mtcars$hp, mtcars$mpg)



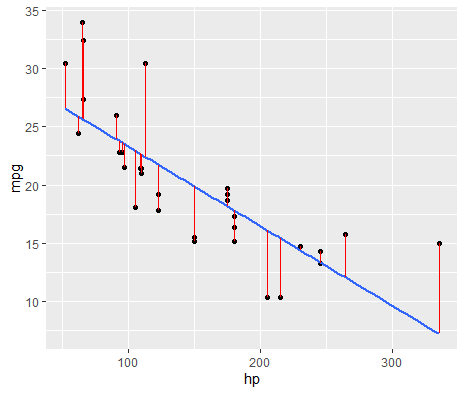
**PLOTTING LINEAR REGRESSION GRAPH:**

ggplot(fit, aes(hp, mpg)) +

geom\_point() +

stat\_smooth(method = lm, se = FALSE) +

geom\_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)



1. **MULTI LINEAR REGRESSION:**

**SOURCE CODE:**

**CREATING MULTI LINEAR REGRESSION MODEL & PRINTING SUMMARY:**

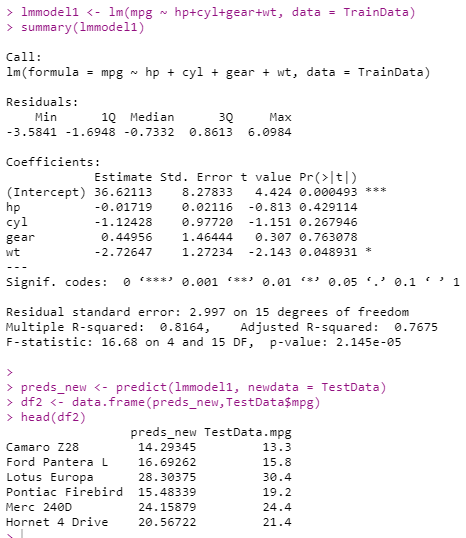
fit = lm(mpg ~ hp, data=mtcars)

summary(fit)

preds <- predict(fit, newdata = TestData)

df1 <- data.frame(preds,TestData$mpg)

head(df1)



**CALCULATING CORREALATION:**

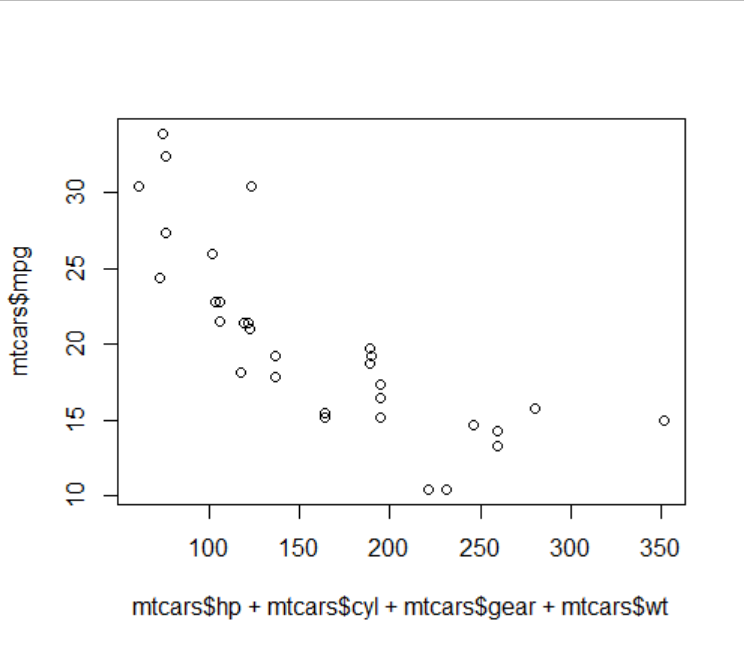
#correlation

cor(preds\_new,TestData$mpg)



**PLOTTING POINTS:**

plot(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mtcars$mpg)



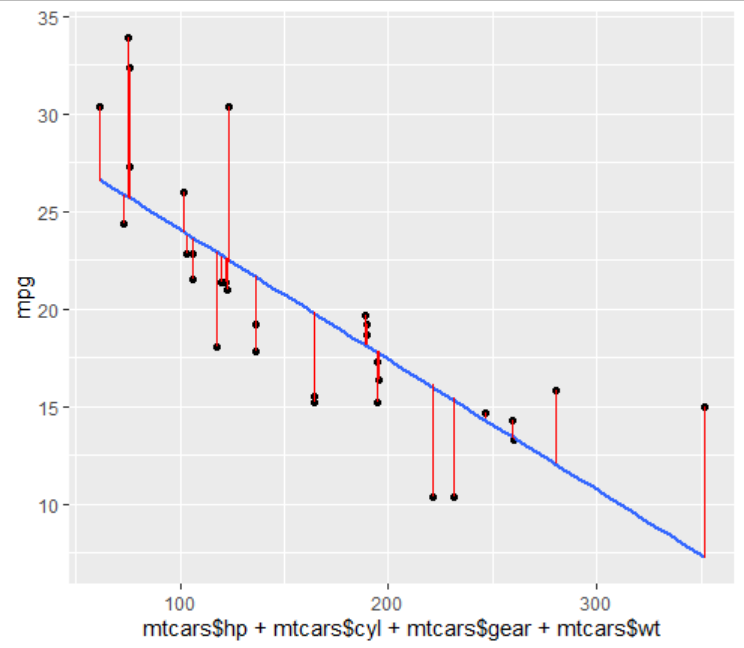
**PLOTTING MULTI LINEAR REGRESSION GRAPH:**

ggplot(fit, aes(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mpg)) +

geom\_point() +

stat\_smooth(method = lm, se = FALSE) +

geom\_segment(aes(xend = mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, yend = .fitted), color = "red", size = 0.3)



**CONCLUSION:**

From this practical, I have learned how to implement linear regression in r programming.