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

$$\text{Error} = \sum_{i=1}^{n} (actual\_output - predicted\_output) ** 2$$

If we don't square the error, then positive and negative point will cancel out each other.
For model with one predictor,

Intercept Calculation:

$$b_0 = \bar{y} - b_1 \bar{x}$$

Co-efficient Formula:

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

A) 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)
```

```
> #load library
> library(ggplot2)
>
> #load cars dataset
> my_data <-mtcars
> #load library
> library(ggplot2)
>
> #load cars dataset
> my_data <-mtcars
>
> #printing names of columns
> names(my_data)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
```

PRINTING DIMENSIONS OF DATASET:

#printing dimensions of dataset

```
dim(my_data)
```

```
> #printing dimensions of dataset
> dim(my_data)
[1] 32 11
```

CREATING RANDOM SAMPLE:

#randomize

```
my_data <- my_data[sample(nrow(my_data), ), ]

head(my_data)
```

```
> #randomize
> my_data <- my_data[sample(nrow(my_data), ), ]
> head(my_data)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

CREATING TRAINING & TESTING DATASET:

#Creating Training & Testing Dataset
TrainData <- my_data[1:20,]
TestData <- my_data[21:32,]

TrainData
TestData

```
> #Creating Training & Testing Dataset
> TrainData <- my_data[1:20,]
> TestData <- my_data[21:32,]
>
> TrainData
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Dodge Challenger  15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
> TestData
                  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Camaro Z28       13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Ford Pantera L   15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Lotus Europa     30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Merc 240D        24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Hornet 4 Drive   21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
AMC Javelin      15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Merc 230         22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Datsun 710       22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Porsche 914-2    26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Honda Civic      30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```

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)
```

```
> ## Linear Model
> fit = lm(mpg ~ hp, data=mtcars)
> summary(fit)

Call:
lm(formula = mpg ~ hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7121 -2.1122 -0.8854  1.5819  8.2360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.09886    1.63392  18.421  < 2e-16 ***
hp          -0.06823    0.01012  -6.742 1.79e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.863 on 30 degrees of freedom
Multiple R-squared:  0.6024,    Adjusted R-squared:  0.5892
F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07

>
> preds <- predict(fit, newdata = TestData)
> df1 <- data.frame(preds,TestData$mpg)
> head(df1)
                   preds TestData.mpg
Camaro Z28       13.38293         13.3
Ford Pantera L   12.08660         15.8
Lotus Europa     22.38907         30.4
Pontiac Firebird 18.15891         19.2
Merc 240D        25.86871         24.4
Hornet 4 Drive   22.59375         21.4
```

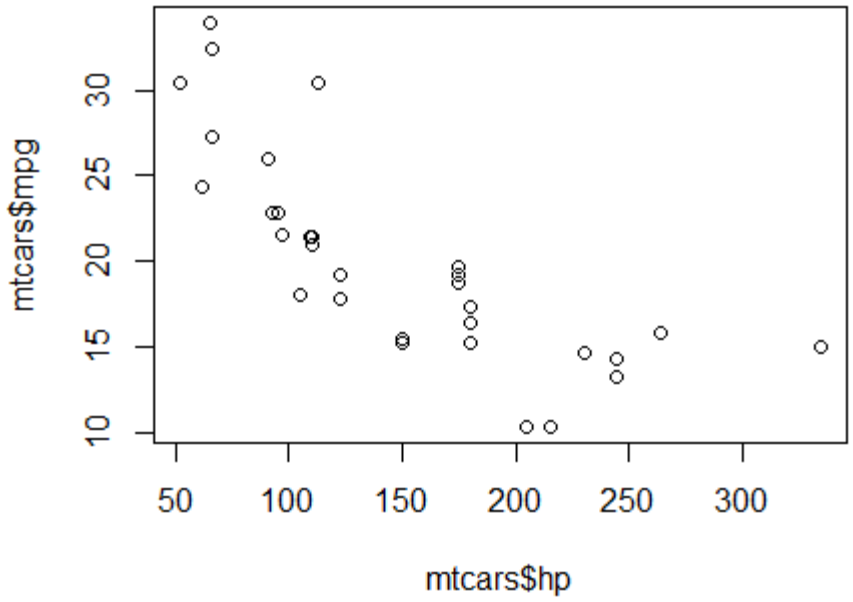
CALCULATE CORRELATION:

```
#correlation
cor(preds,TestData$mpg)
```

```
> cor(preds,TestData$mpg)
[1] 0.8115641
```

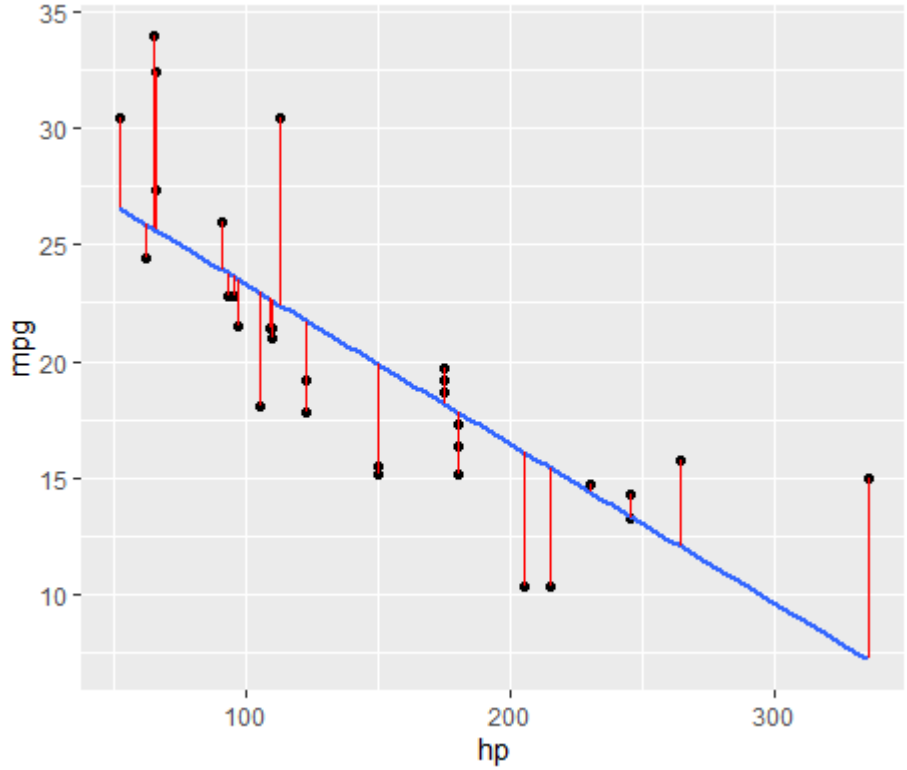
PLOTTING POINTS:

```
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)
```



B) 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)
```

```
> lmmodel1 <- lm(mpg ~ hp+cyl+gear+wt, data = TrainData)
> summary(lmmodel1)

Call:
lm(formula = mpg ~ hp + cyl + gear + wt, data = TrainData)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5841 -1.6948 -0.7332  0.8613  6.0984

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.62113    8.27833   4.424 0.000493 ***
hp          -0.01719    0.02116  -0.813 0.429114
cyl         -1.12428    0.97720  -1.151 0.267946
gear         0.44956    1.46444   0.307 0.763078
wt          -2.72647    1.27234  -2.143 0.048931 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.997 on 15 degrees of freedom
Multiple R-squared:  0.8164,    Adjusted R-squared:  0.7675
F-statistic: 16.68 on 4 and 15 DF,  p-value: 2.145e-05

>
> preds_new <- predict(lmmodel1, newdata = TestData)
> df2 <- data.frame(preds_new,TestData$mpg)
> head(df2)
                 preds_new TestData.mpg
Camaro Z28        14.29345         13.3
Ford Pantera L    16.69262         15.8
Lotus Europa      28.30375         30.4
Pontiac Firebird  15.48339         19.2
Merc 240D         24.15879         24.4
Hornet 4 Drive    20.56722         21.4
```

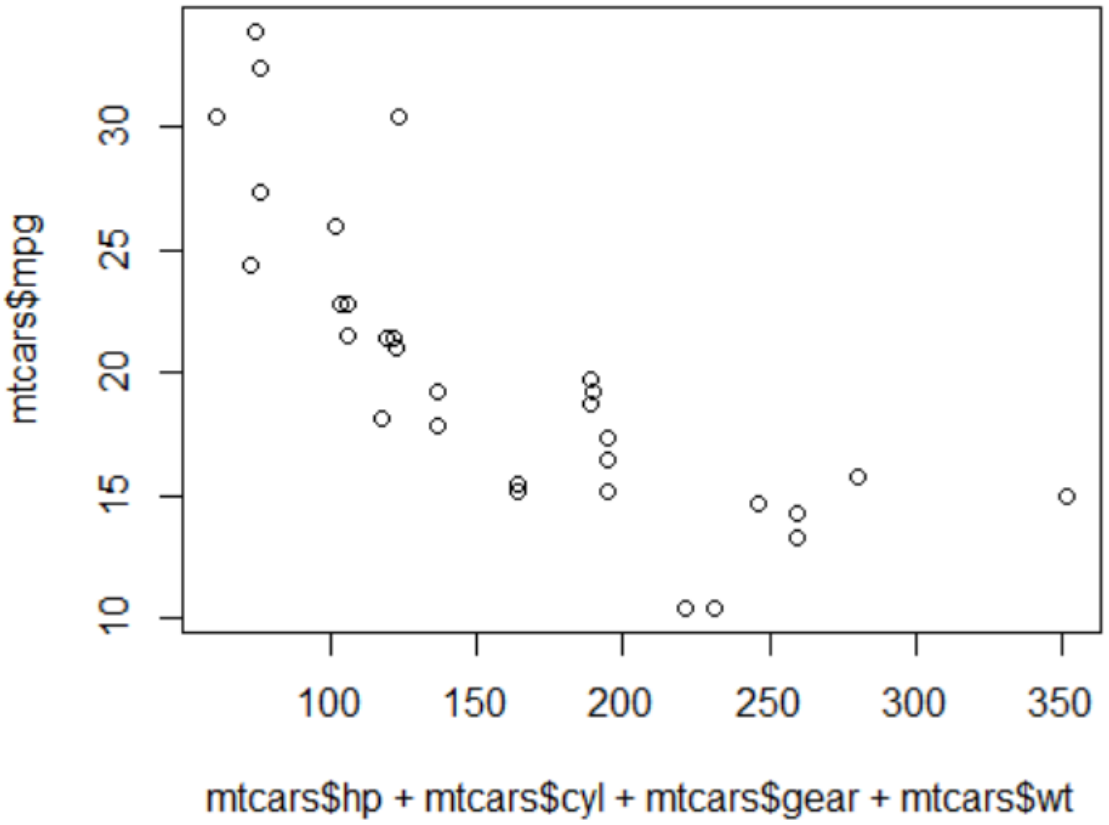
CALCULATING CORREALATION:

#correlation
cor(preds_new,TestData$mpg)

```
> cor(preds_new,TestData$mpg)
[1] 0.934887
```

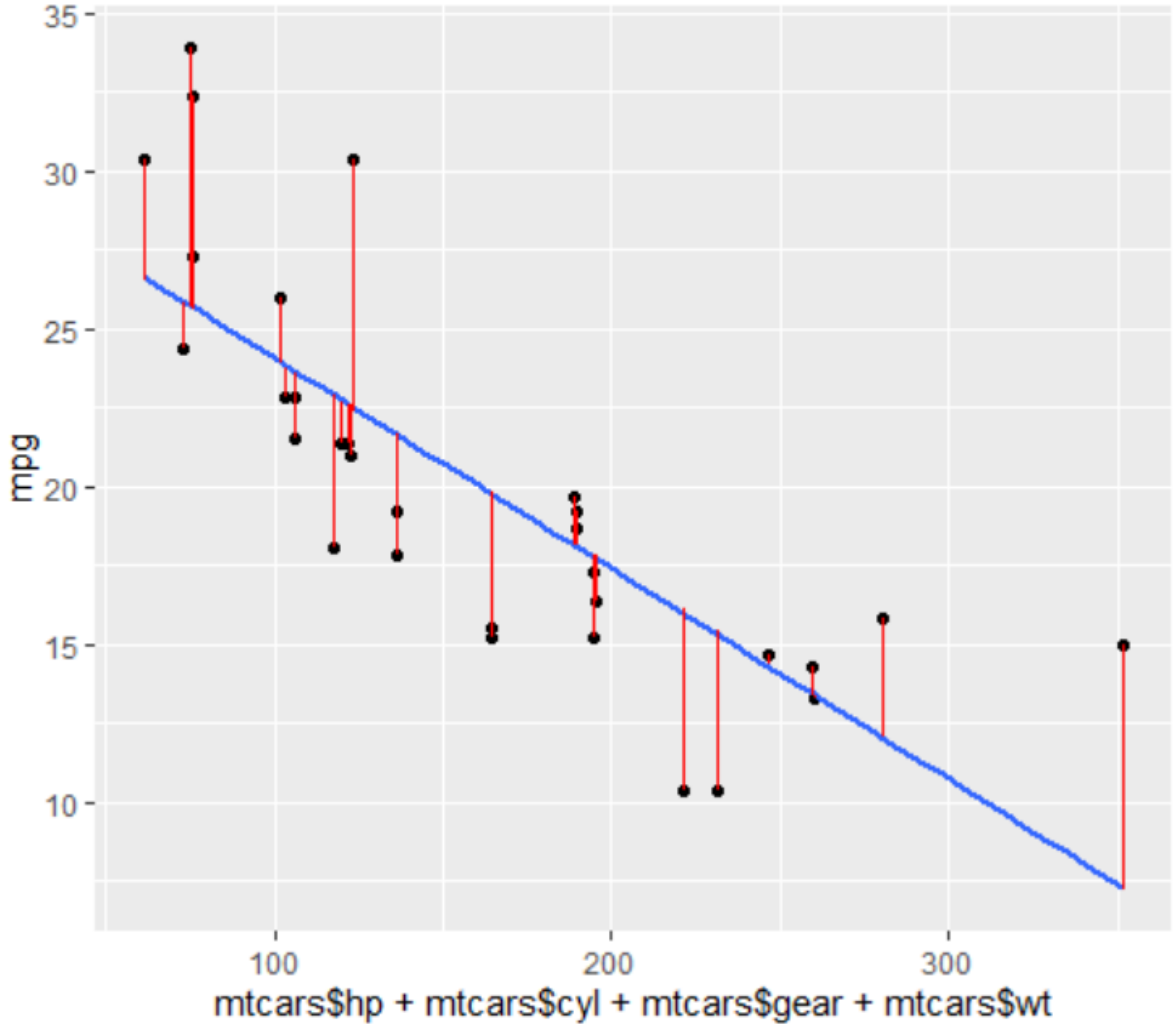
PLOTTING POINTS:

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.