PRACTICAL – 7

Aim: Implementation and analysis of Linear regression through graphical methods including Plots

Theory:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. There are many names for a regression's dependent variable. It may be called an outcome variable, criterion variable, endogenous variable, or regressand. The independent variables can be called exogenous variables, predictor variables, or regressors.

Linear regression is used in many different fields, including finance, economics, and psychology, to understand and predict the behaviour of a particular variable. For example, in finance, linear regression might be used to understand the relationship between a company's stock price and its earnings, or to predict the future value of a currency based on its past performance.

As the name suggests, linear regression in data mining functions by building a straight line between the target variable and one or more than one independent variable.

It is represented by the equation:

Y = a + b*X + e,

where a is the intercept, b is the slope of the regression line and e is the error. X and Y are the predictor and target variables respectively. When X is made up of more than one variable (or features) it is termed as multiple linear regression.

The best-fit line is achieved using the Least-Squared method. This method minimizes the sum of the squares of the deviations from each of the data points to the regression line. The negative and positive distances do not get cancelled out here as all the deviations are squared.

There are also divisions under linear regression in data mining named simple regression and multiple regression. Simple linear regression is where a singular predictor variable is known. However, in most real-world cases, the number of predictor variables is more than one, which is why multiple Regression data mining is used more than the simple one.

Setting working directory.

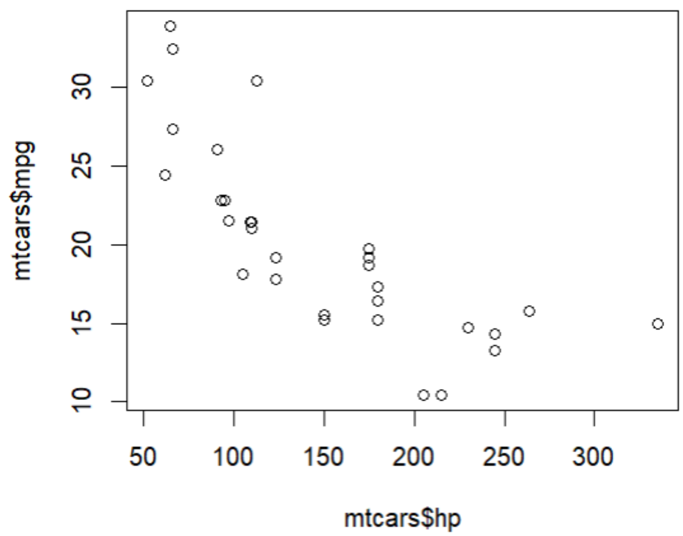
```
> getwd()
[1] "C:/Users/Suraj/Documents"
> 
```

```
> library(ggplot2)
> my_data <- mtcars
> names(my_data)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"  
[10] "gear" "carb"
> dim(my_data)
[1] 32 11
> my_data <- my_data[sample(nrow(my_data),),]
> head(my_data)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    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
Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0    3    4
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
> TrainData <-my_data[1:20,]
> TestData <- my_data[21:32,]
> 
```

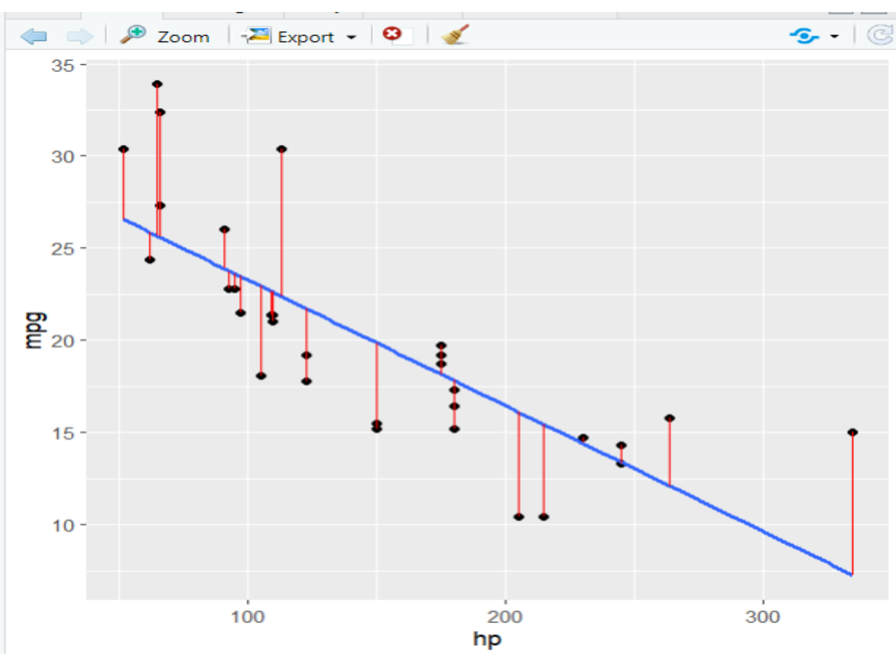
```
>
> fit = lm(mpg ~ hp, data=mtcars)
> summary(fit)

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$mpg)
> head(df1)
                      preds TestData.mpg
Fiat 128          25.595794         32.4
Datsun 710        23.753631         22.8
Maserati Bora      7.242387         15.0
Duster 360        13.382932         14.3
Chrysler Imperial 14.406357         14.7
Toyota Corona     23.480718         21.5
>
> cor(preds,TestData$mpg)
[1] 0.7335502
> plot(mtcars$hp, mtcars$mpg)
> 
```

Plotting

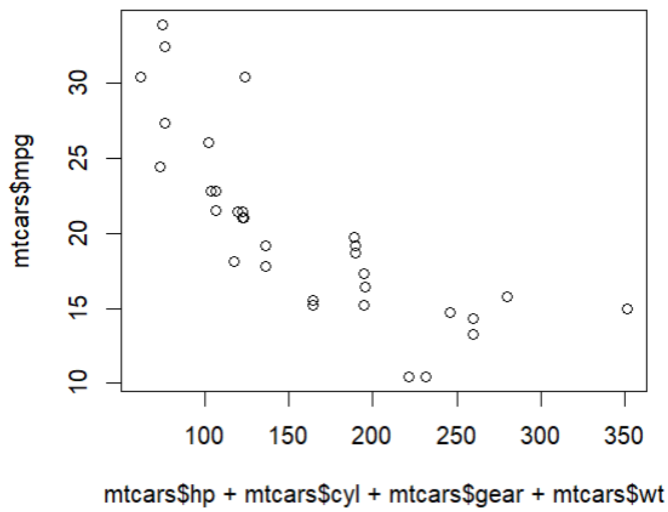
```
> plot(mtcars$hp, mtcars$mpg)
> 
```



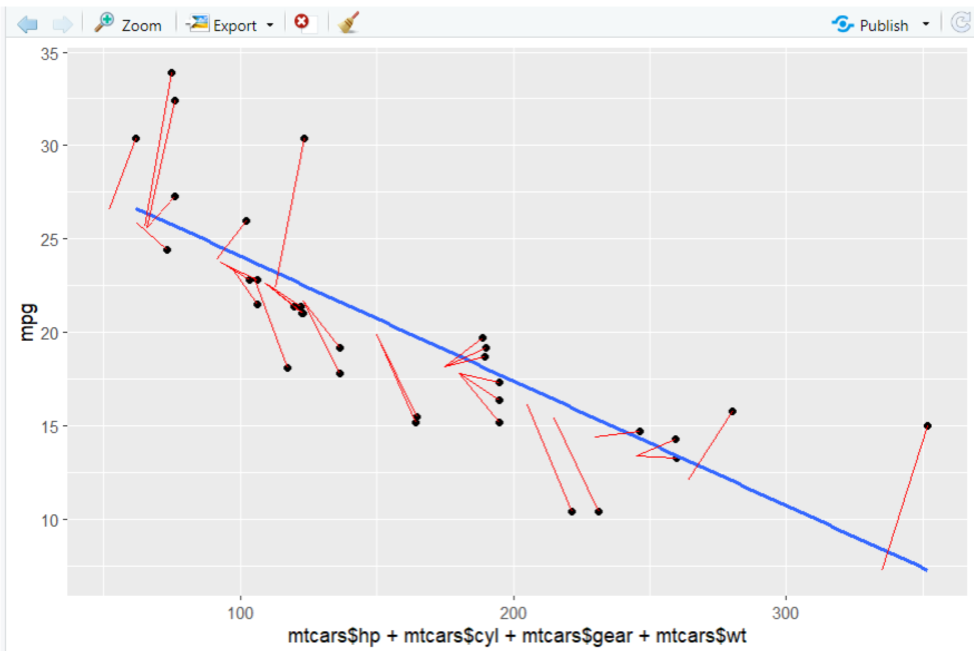

```
> plot(mtcars$hp, mtcars$mpg)
> ggplot(fit, aes(hp, mpg)) +
+     geom_point() +
+     stat_smooth(method = lm, se = FALSE) +
+     geom_segment(aes(xend = hp, yend = .fitted), color = "red",
+
+ size=0.3)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
>
```



```
> 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 
-3.0432 -1.1317 -0.2652  0.6732  5.0793 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 47.67466    7.97803   5.976 2.54e-05 ***
hp          -0.01966    0.02436  -0.807  0.43236    
cyl         -1.21598    0.78130  -1.556  0.14047    
gear        -1.47291    1.31247  -1.122  0.27940    
wt          -3.71693    1.03278  -3.599  0.00263 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.154 on 15 degrees of freedom
Multiple R-squared:  0.902,	Adjusted R-squared:  0.8759 
F-statistic: 34.51 on 4 and 15 DF,  p-value: 2.111e-07

> preds_new <- predict(lmmodel1, newdata = TestData)
> df2 <- data.frame(preds_new,TestData$mpg)
> head(df2)
                  preds_new TestData.mpg
Fiat 128          27.444354         32.4
Datsun 710        26.467533         22.8
Maserati Bora     10.727094         15.0
Duster 360        15.442218         14.3
Chrysler Imperial  9.139549         14.7
Toyota Corona     27.322856         21.5
> cor(preds_new,TestData$mpg)
[1] 0.8496726
> plot(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mtcars$mpg)
> 
```




```
> ggplot(fit, aes(mtcars$hp+mtcars$cyl+mtcars$gear+mtcars$wt, mpg)) +
+   geom_point() +
+   stat_smooth(method = lm, se = FALSE) +
+   geom_segment(aes(xend = hp, yend = .fitted), color = "red", size = 0.3)
`geom_smooth()` using formula = 'y ~ x'
> |
```



Conclusion:

I have successfully implemented and analysed linear regression through graphical method including plots.