EXPERIMENT-8

Problem Statement: To implement linear and multiple regression on 'mtcars' dataset to estimate the value of 'mpg' variable, with best R2 and plot the original values in 'green' and predicted values in 'red'

Theoretical Background:

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

lm() function:

This function creates the relationship model between the predictor and the response variable.

Syntax

The basic syntax for **lm()** function in linear regression is −

```
lm(formula,data)
```

Following is the description of the parameters used −

- **formula** is a symbol presenting the relation between x and y.
- **data** is the vector on which the formula will be applied.

abline() function:

Itis used to add one or more straight lines to a graph. The **abline()** function can be used to add vertical, horizontal or regression lines to plot.

abline(a, b, h, v, col, lwd)

Parameters:
a, b: It specifies the intercept and the slope of the line

h: specifies y-value for horizontal line(s)
v: specifies x-value(s) for vertical line(s)

col: optional color parameter for the line
lwd: optional line width parameter for the line

paste0() function:

The paste0() is a built-in R function used to concatenate all elements without a separator. The paste0() function accepts one or more R objects as arguments and optional collapse argument.

summary() function:

The summary() function is a generic function used to produce result summaries of the results of various model fitting functions.

mtext() function :

The mtext() function is used to write text in one of the four margins of the current figure region or one of the outer margins of the device region.

mtext(text, side, line)

- **text :** A character or an expression vector specifying the text to be written.
- **side :** on which side of the plot (1 = bottom, 2 = left, 3 = top, 4 = right)
- **line :** on which margin line starting at 0 counting outwards.

Simple Linear Regression on mtcars dataset

```
View(mtcars)

plot(mpg ~ wt, data= mtcars, col=2)
fit <-lm(mpg ~ wt, data= mtcars)
summary(fit)

##A residual is a measure of how far away a point is vertically
from the regression line.
##Simply, it is the error between a predicted value and the
observed actual value
##The intercept (often labeled the constant) is the expected mean
value of Y
##when all X=0. Start with a regression equation with one
predictor, X.
##If X sometimes equals 0, the intercept is simply the expected
mean value of Y at that value. ...
##It's the mean value of Y at the chosen value of X.

abline(fit,col=3,lwd=2)  #We could add a line to our first plot to
show the predicted mpg:
bs<-round(coef(fit), 3)
print(bs)
lmlab <- paste0("mpg = ", bs[1],ifelse(sign(bs[2])==1, " + ", "
-"), abs(bs[2]), " wt ")
mtext(lmlab, 3, line=-2)  #we add it to the plot using mtext
```

Output

```
Call:

lm(formula = mpg ~ wt, data = mtcars)


Residuals:
```

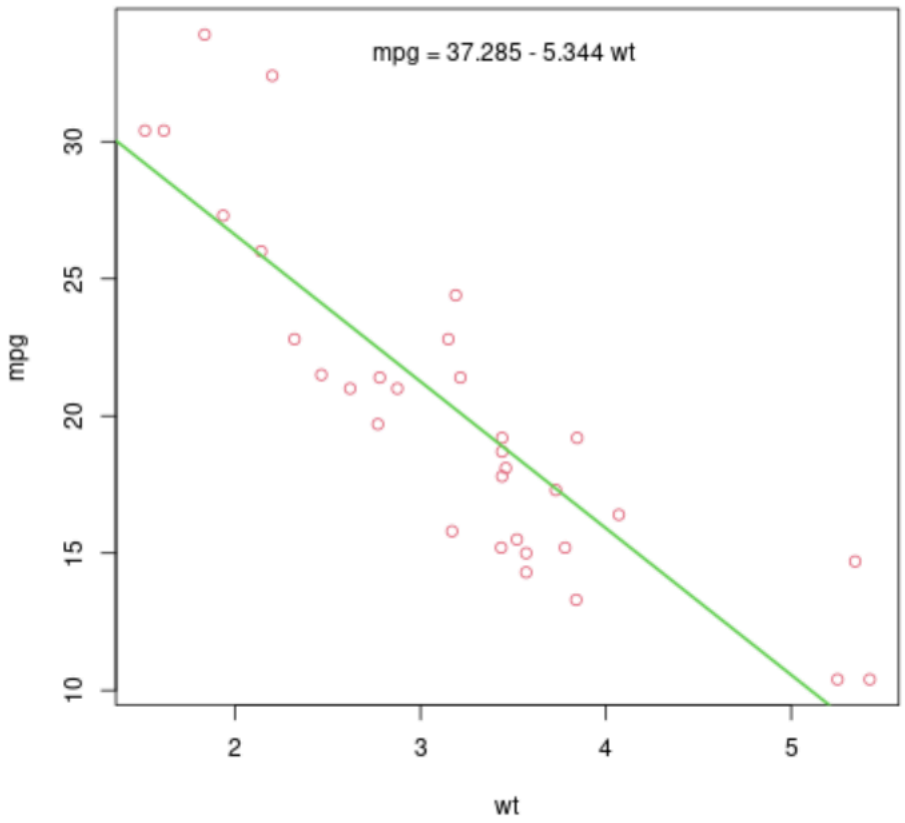
```
    Min      1Q  Median      3Q     Max 
-4.5432 -2.3647 -0.1252  1.4096  6.8727 


Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446 
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10


(Intercept)          wt 
     37.285      -5.344 


      37.285      -5.344
```


mpg = 37.285 - 5.344 wt
mpg
wt
10
15
20
25
30
2
3
4
5

Multiple Linear Regression on mtcars dataset

```
input <- mtcars[,c("mpg","disp","hp","wt")]
print(head(input)

model <-lm(mpg~disp+hp+wt, data= input)

# Show the model.
print(model)

# Get the Intercept and coefficients as vector elements.
cat("# # # # The Coefficient Values # # # ","\n")

a <-coef(model)[1]
print(a)

Xdisp <-coef(model)[2]
Xhp <-coef(model)[3]
Xwt <-coef(model)[4]

print(Xdisp)
print(Xhp)
print(Xwt)
```

Output

```
                   mpg disp  hp    wt

Mazda RX4         21.0  160 110 2.620

Mazda RX4 Wag     21.0  160 110 2.875

Datsun 710        22.8  108  93 2.320

Hornet 4 Drive    21.4  258 110 3.215

Hornet Sportabout 18.7  360 175 3.440

Valiant           18.1  225 105 3.460


Call:

lm(formula = mpg ~ disp + hp + wt, data = input)


Coefficients:

(Intercept)         disp           hp           wt

  37.105505    -0.000937    -0.031157    -3.800891


# # # # The Coefficient Values # # #

(Intercept)
```

```
   37.10551

         disp

-0.0009370091

         hp

-0.03115655

       wt

-3.800891
```