Experiment # 8

Aim: Fit simple linear regression models using built-in functions.

8.1. Definition:-

Linear Regression:-

A linear regression is a statistical model that analyses the relationship between a response variable/dependent variable (often called y) and one predictor variables (often called x or explanatory variables) and their interactions using a regression line.

Linear Regression Equation is y=ax+b

where a is slope and b is intercept.

Example: when we calculate the age of a child based on their height, we assumed how older they are, the taller they will be.

In this particular example, you can calculate the height of a child if you know her/his age:

$$\boldsymbol{Height = a + Age \times b}$$

In this case, a and b are called the intercept and the slope, respectively. The slope measures the change in height with respect to the age in months (or years). In general, for every month older the child is, their height will increase with b.

R - Multiple Regression:-

Multiple regression is an extension of linear regression into relationship between more than two variables. In simple linear relation we have one predictor and one response variable, but in multiple regression we have more than one predictor variable and one response variable.

The general mathematical equation for multiple regression is –

$$y = a + b_1x_1 + b_2x_2 + ... + b_nx_n$$

where

- y is the response variable.
- a, b_1, b_2,...,b_n are the coefficients.
- x_1, x_2, ..., x_n are the predictor variables.

Real world Applications: (Why are you studying this model)

1. Predicting house prices based on features like size, location, etc.
2. Estimating sales revenue based on advertising spend.
3. Analyzing relationships between biological or environmental variables.
4. **Analyzing relationships between Mid-term marks and End-term marks.**

8.2. Commands and calculation of R: Basic steps to perform Linear Regression in R

1. Use the **lm()** function to fit a linear model.

   The model **lm()** determines the value of the coefficients using the input data. Next we can predict the value of the response variable for a given set of predictor variables using these coefficients.

   The syntax is:

```
model <- lm(Y ~ X, data = dataset)
```

Here, Y is the dependent variable, X is the independent variable, and data is the data frame containing the variable you want to study.

2. **Check the Model Summary:** The **summary()** function provides detailed information on the model, including coefficients, R-squared, and p-values.

   summary(model)

   - **Coefficients**: The estimated values for the intercept and the slope.
   - **R-squared**: A measure of how well the model explains the variance in the data.
   - **p-values**: To test the significance of the coefficients.
   - **Residual standard error**: A measure of the typical size of the residuals.

3. **Plot the Model :** A quick visualization of the model fit can be achieved using **plot().**

   plot(dataset$X, dataset$Y)

   abline(model, col = "blue")

   abline() - Adds the regression

   line to the plot.

8.3. Example of Simple Linear Regression in R:

Example no.1:-

Using an **in-built dataset** like **mtcars** in R is quite simple. Here's a step-by-step guide on how to use an in-built dataset in R: **(Instead of using pre-loaded dataset we can also use our own file, such as CSV file, dataframe etc.)**

Step 1: Load the Dataset

For most in-built datasets, you don't need to explicitly load them; they are pre-loaded with the datasets package, which comes with base R. Simply type the dataset name to view it:

```
data(mtcars)
View(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4

Step 2: Explore the Dataset

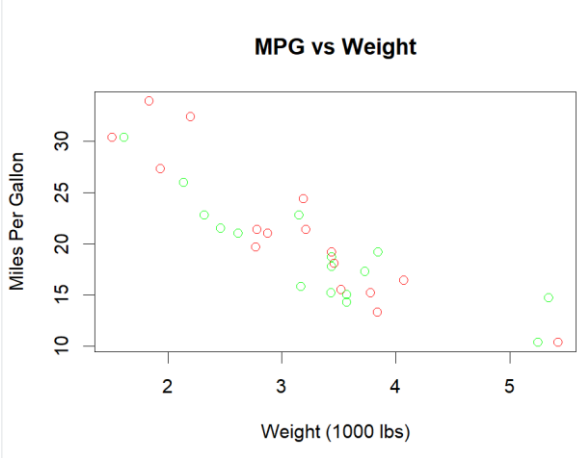
```
  str(mtcars)
'data.frame':  32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
```

$ carb: num 4 4 1 1 2 1 4 2 2 4 ...

Step 3: Visualize the Data: Basic plots are useful for understanding the relationships in the dataset. For example, with **mtcars**

Plotting example: scatter plot of mpg vs wt (weight)

```
plot(mtcars$wt, mtcars$mpg, main = "MPG vs Weight", xlab = "Weight (1000 lbs)",
ylab = "Miles Per Gallon",col=c("green","red"))
```



Step 4: Analyze the Data

Now that the dataset is loaded and explored, you can apply various statistical models or functions. For example, performing a linear regression:

```
model <- lm(mpg ~ wt, data = mtcars)
summary(model)
```

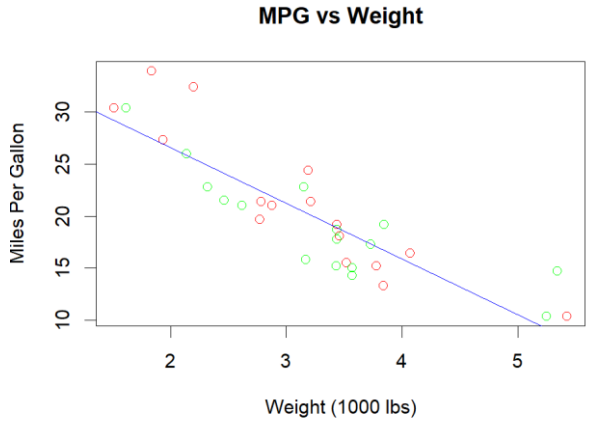
Call:
lm(formula = mpg ~ wt, data = mtcars)

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**

Add the regression line to the plot

```
> abline(model, col = "blue")
```



Example No. 2:-

x <- c(151,174,138,186,128,136,179,163,152,131)
y <- c(63,81,56,91,47,57,76,72,62,48)
relation=lm(x~y)
print(relation)

```
Call:
lm(formula = x ~ y)

Coefficients:
(Intercept)            y
     61.380        1.415
```

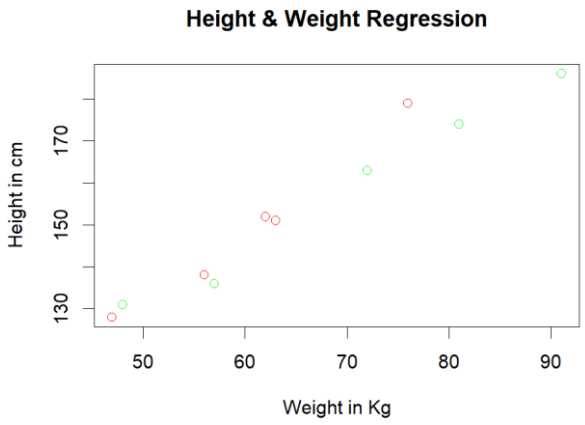
>print(summary(relation))

```
Call:
lm(formula = x ~ y)

Residuals:
    Min      1Q  Median      3Q     Max
-6.0529 -2.4833 -0.0912  1.3774 10.0562

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  61.3803     7.2653   8.448 2.94e-05 ***
y             1.4153     0.1089  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.712 on 8 degrees of freedom
Multiple R-squared:  0.9548,  Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

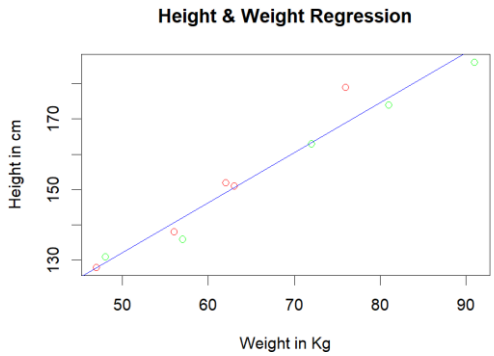
> cor(x,y)
[1] 0.9771296

Therefore the regression line is y=61.3803+1.4153x where slope is 1.4153 and intercept is 61.38 03

plot(y,x,main ="Height & Weight Regression",xlab ="Weight in Kg",ylab ="Height in cm",col=c("red","green"))



abline(lm(x~y),col="blue")



8.4 Practice Questions

Predicting Fuel Efficiency with the mtcars Dataset

1.The mtcars dataset in R contains various attributes of different car models, such as miles per gallon (mpg), horsepower (hp), weight (wt), and more. Your task is to predict the fuel efficiency (mpg) of cars based on their weight (wt) and horsepower (hp).

2. Fit a linear model for inbuilt data-**women** like mtcars.

Note:-

- **Standard Error**: Measures the precision of the coefficient estimates. Smaller values suggest more precise estimates.
- **t value**: A measure of how many standard errors the estimated coefficient is away from 0. Larger values indicate that the predictor is more significant.
- **Pr(>|t|)**: The p-value for testing the null hypothesis that the coefficient is zero. A small p-value (typically < 0.05) indicates that the predictor is statistically significant.
- **R-squared**: A measure of how well the model fits the data. It indicates the proportion of variance in the dependent variable explained by the independent variable(s).
- **Residual Standard Error**: The standard deviation of the residuals. Smaller values indicate better fit.

- **F-statistic and p-value**: Tests the overall significance of the model. A significant F-statistic (p-value < 0.05) indicates that at least one of the predictors is significantly related to the dependent variable.