# R PROGRAMMING

## EXPERIMENT-1

### ADDITION:

### AIM:

To prove the program for addition using R-tool.

### PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1+num2
print(num3)
```

### OUTPUT:

```
Enter a number1 : 2
Enter a number2 : 2
[1] 4
```

### RESULT:

Thus the basic program addition are executed successfully.

## EXPERIMENT-2

### SUBTRACTION:

### AIM:

To prove the program for subtraction using R-tool.

### PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1-num2
print(num3)
```

### OUTPUT:

```
Enter a number1 : 4
Enter a number2 : 2
[1] 2
```

RESULT:

Thus the basic program subtraction are executed successfully.

## EXPERIMENT-3

## MULTIPLICATION:

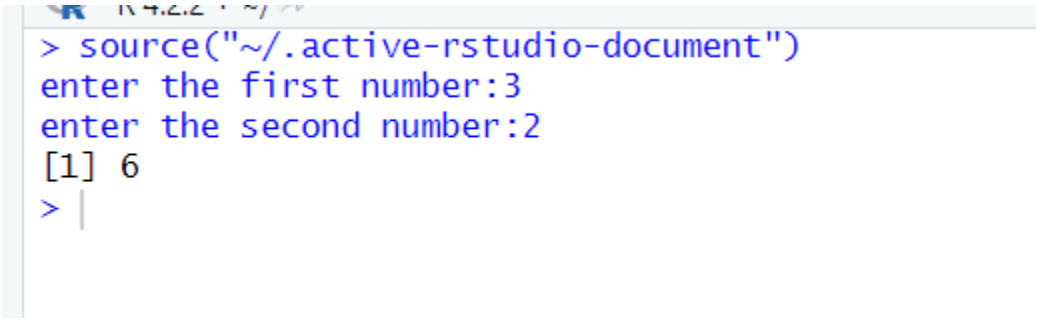
### AIM:

To prove the program for multiplication using R-tool.

### PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1*num2
print(num3)
```

### OUTPUT:




RESULT:

Thus the basic program multiplication are executed successfully.

## EXPERIMENT-4

## DIVISION:

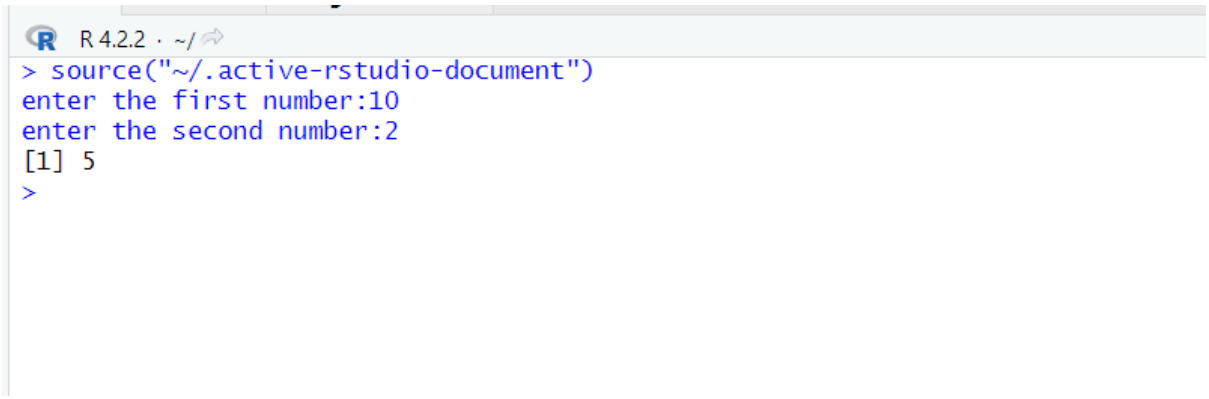
### AIM:

To prove the program for division using R-tool.

PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))

num2=as.integer(readline(prompt = "enter the second number:"))

num3=num1/num2

print(num3)
```

OUTPUT:




RESULT:

Thus the basic program division was executed successfully.

## EXPERIMENT-5

## ODD OR EVEN:

## AIM:

To write the program for odd or even using R-tool.

PROGRAM:

```
num=as.integer(readline(prompt="enter a number:"))

if((num%%2)==0)

{

  print("number is a even")

}else{

  print("number is odd")

}
```

OUTPUT:

```
> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number :      4
[1] "Number is even"
> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number : 5
[1] "Number is odd"
> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number : 1
[1] "Number is odd"
>
```

RESULT:

Thus the basic program odd or even was executed successfully.

## EXPERIMENT-6

## MEAN,MEDIAN,MODE:

### AIM:

To write the program for mean,median,mode.

### PROGRAM:

### MEAN

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
mean(df $age)
write.csv(df,"datafr.csv")
```

### MEDIAN

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
median(df $age)
write.csv(df,"datafr.csv")
```

## MODE

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
mode(df $age)
write.csv(df,"datafr.csv")
```

## OUTPUT:

```
> mode(df $age)
[1] "numeric"
```

```
> mean(df $age)
[1] 27.33333
```

```
> median(df $age)
[1] 24
```

```
> mode(df $age)
[1] "numeric"
```

## RESULT:

Thus the central tendency and measure of dispersion is executed successfully.

# EXPERIMENT-7

## SUMMARY:

## AIM:

To write the program for summary using R-tool.

## PROGRAM:

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
summary(df $age)
write.csv(df,"datafr.csv")
```

OUTPUT:

```
> summary(df $age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.00   23.50   24.00   27.33   29.50   35.00
```

RESULT:

Thus the central tendancy and measure of dispersion is executed successfully.

## EXPERIMENT-8

## GREATER AMONG THREE NUMBERS:

### AIM:

To write the program for the greatest among three numbers.

### PROGRAM:

```
x <- as.integer(readline(prompt = "Enter first number :"))

y <- as.integer(readline(prompt = "Enter second number :"))

z <- as.integer(readline(prompt = "Enter third number :"))


if (x > y && x > z) {

  print(paste("Greatest is :", x))

} else if (y > z) {

  print(paste("Greatest is :", y))

} else{

  print(paste("Greatest is :", z))

}
```

### OUTPUT:

```
R 4.2.2 · ~/
> source("~/.active-rstudio-document")
Enter first number :5
Enter second number :6
Enter third number :4
[1] "Greatest is : 6"
>
```

## RESULT:

Thus the greatest among the three numbers was executed successfully.

# EXPERIMENT-9

## IQR:

## AIM:

To write the program for central tendency and data dispersion measures using R tool.

## PROGRAM:

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
IQR(df $age)
write.csv(df,"datafr.csv")
```

## OUTPUT:

```
> IQR(df $age)
[1] 6
```

## RESULT:

Thus the program for central tendency and data dispersion measures was executed successfully.

# EXPERIMENT-11

## MID RANGE:

## AIM:

To write the program for central tendency and data dispersion measures.

## PROGRAM:

```
data <- c(10, 20, 30, 40, 50)
mid_range <- (min(data) + max(data)) / 2
print(paste("Mid-Range:", mid_range))
```

## OUTPUT:

```
> data <- c(10, 20, 30, 40, 50)

> mid_range <- (min(data) + max(data)) / 2

> print(paste("Mid-Range:", mid_range))
[1] "Mid-Range: 30"
> source("~/.active-rstudio-document", echo=TRUE)
```

RESULT:

Thus the program for central tendency and data dispersion measures was executed successfully

EXPERIMENT-12

Z-SCOORE NORMALIZATION:

AIM:

To write the program for Z-scoore normalization using R-tool.

PROGRAM:

```
data <- c(10, 20, 30, 40, 50)
z_scores <- scale(data)
print("Original Data:")
print(data)
print("Z-Score Normalized Data:")
print(z_scores)
```

OUTPUT:

```
> print(z_scores)
           [,1]
[1,] -1.2649111
[2,] -0.6324555
[3,]  0.0000000
[4,]  0.6324555
[5,]  1.2649111
attr(,"scaled:center")
[1] 30
attr(,"scaled:scale")
[1] 15.81139
> source("~/.active-rstudio-document", echo=TRUE)
```

RESULT:

Thus the Z-scoore normalization using R tool was executed successfully.

EXPERIMENT-13

MIN,MAX,MEAN,MINMAX:

## AIM:

To write the program for the minimum,maximum,mean and minmax using r-TOOL

## PROGRAM:

```
data <- c(10, 20, 30, 40, 50)
min_value <- min(data)
max_value <- max(data)
mean_value <- mean(data)
minmax_normalized <- (data - min_value) / (max_value - min_value)
print(paste("Min:", min_value))
print(paste("Max:", max_value))
print(paste("Mean:", mean_value))
print("Min-Max Normalized Data:")
print(minmax_normalized)
```

## OUTPUT:

```
> print(paste("Min:", min_value))
[1] "Min: 10"

> print(paste("Max:", max_value))
[1] "Max: 50"

> print(paste("Mean:", mean_value))
[1] "Mean: 30"

> print("Min-Max Normalized Data:")
[1] "Min-Max Normalized Data:"

> print(minmax_normalized)
[1] 0.00 0.25 0.50 0.75 1.00
>
```

## RESULT:

Thus the program for min,max,minmax,mean was executed successfully.

## EXPERIMENT-14, 15

## BOX PLOT, BAR PLOT AND HORIZONTAL BAR:

## AIM:

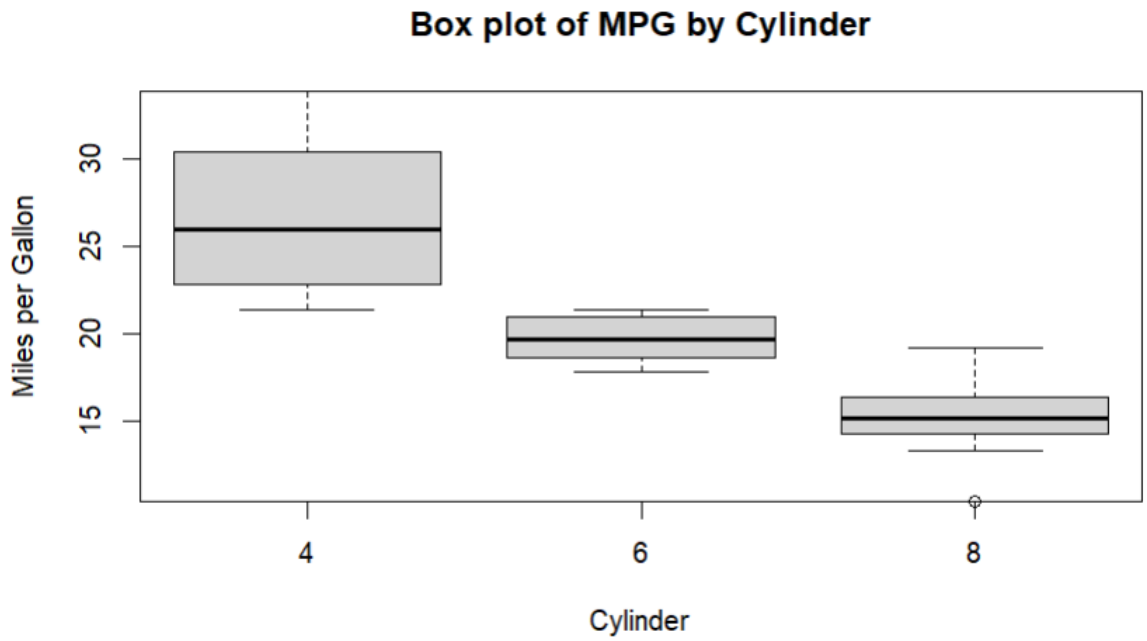
To draw the bar plot and horizontal bar using R-tool.

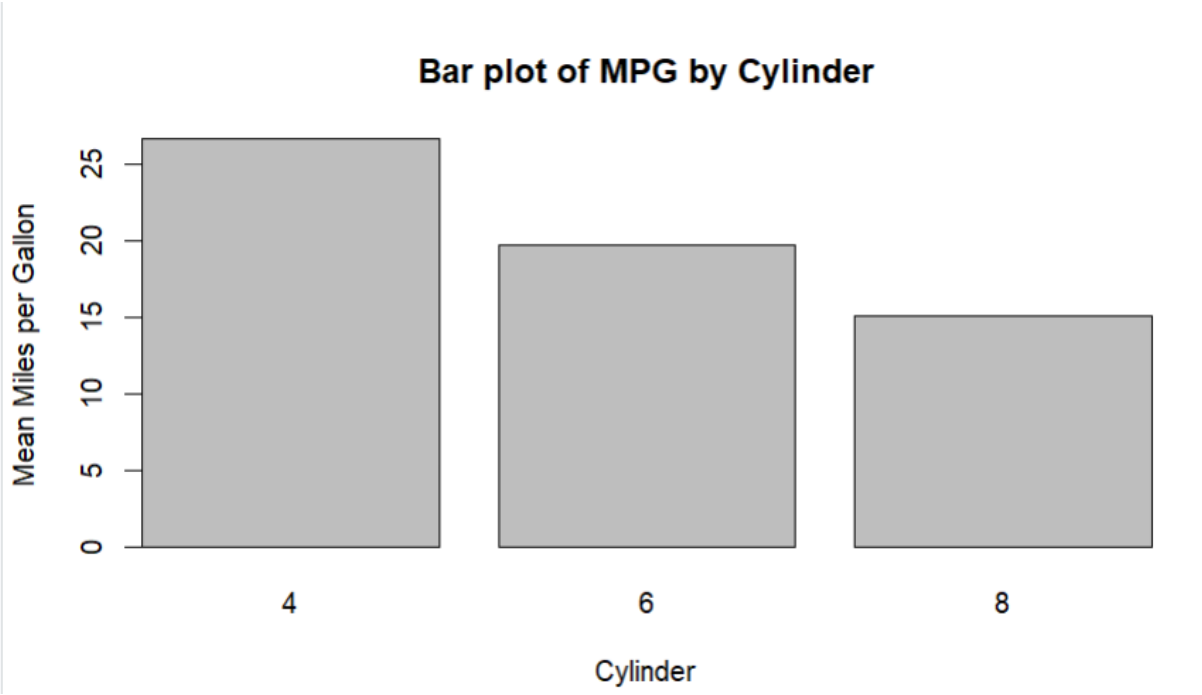
## PROGRAM:

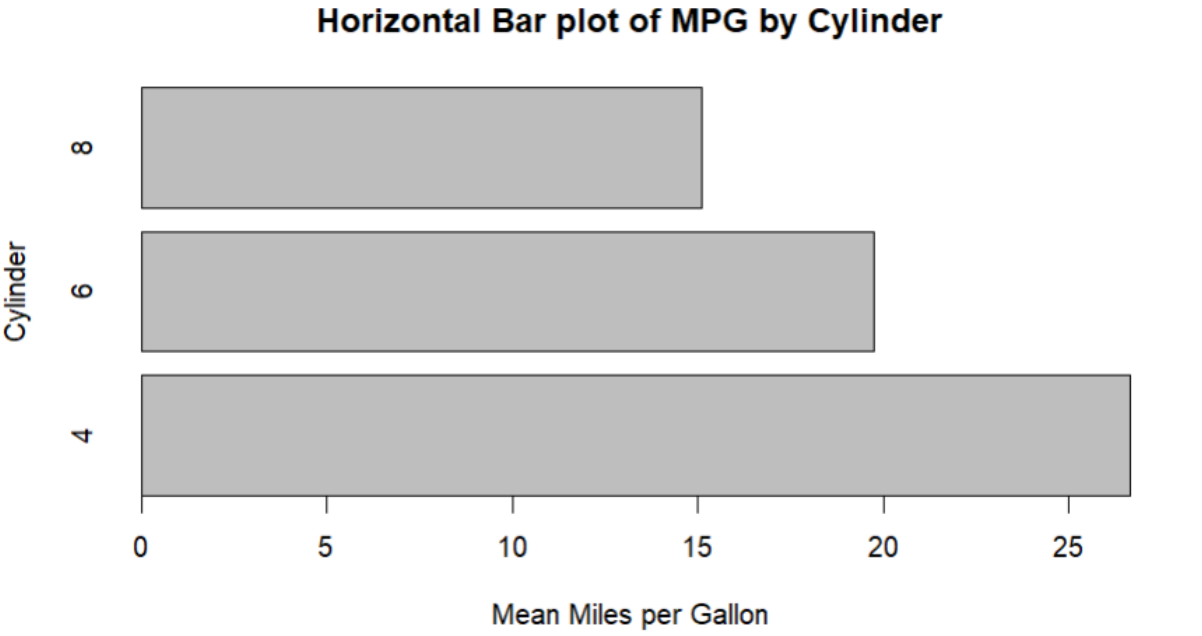
```
data(mtcars)
boxplot(mtcars$mpg ~ mtcars$cyl,
```

```
      main = "Box plot of MPG by Cylinder",
      xlab = "Cylinder",
      ylab = "Miles per Gallon")
mean_mpg <- tapply(mtcars$mpg, mtcars$cyl, mean)
barplot(mean_mpg,
      main = "Bar plot of MPG by Cylinder",
      xlab = "Cylinder",
      ylab = "Mean Miles per Gallon")
barplot(mean_mpg,
      main = "Horizontal Bar plot of MPG by Cylinder",
      xlab = "Mean Miles per Gallon",
      ylab = "Cylinder",
      horiz = TRUE)
```

OUTPUT:






RESULT:

Thus the bar and horizontal bar plot was executed successfully.

## EXPERIMENT-16

HISTOGRAM:

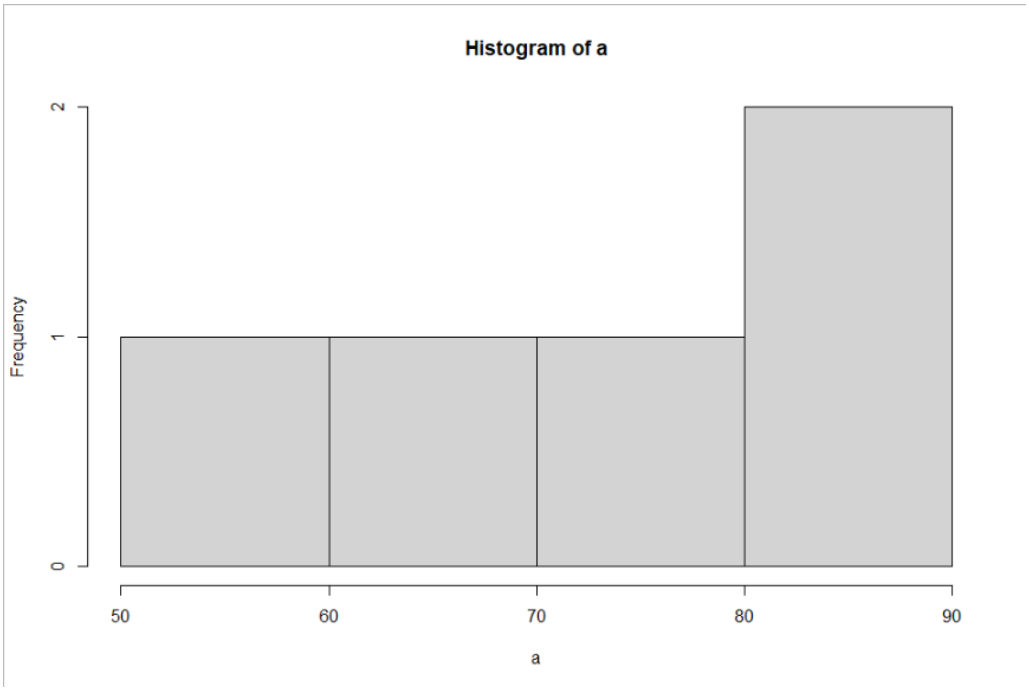
AIM:

To draw the histogram plot using R-tooll.

## PROGRAM:

```
a<-c(55,67,89,80,90)
hist(a)
```

## OUTPUT:




## RESULT:

Thus the histogram plot was executed successfully.

# EXPERIMENT-17

## CORRELATION ANALYSIS:

## AIM:

To write the program for correlation analysis using R-tool.

## PROGRAM:

```
x <- c(10, 20, 30, 40, 50)
y <- c(15, 25, 35, 45, 55)
correlation <- cor(x, y)
print(paste("Correlation Coefficient:", correlation))
```

OUTPUT:

```
> source("~/.active-rstudio-document", echo=TRUE)

> x <- c(10, 20, 30, 40, 50)

> y <- c(15, 25, 35, 45, 55)

> correlation <- cor(x, y)

> print(paste("Correlation Coefficient:", correlation))
[1] "Correlation Coefficient: 1"
>
```

RESULT:

Thus the correlation analysis was executed successfully.

## EXPERIMENT-18

## SCATTER PLOT:

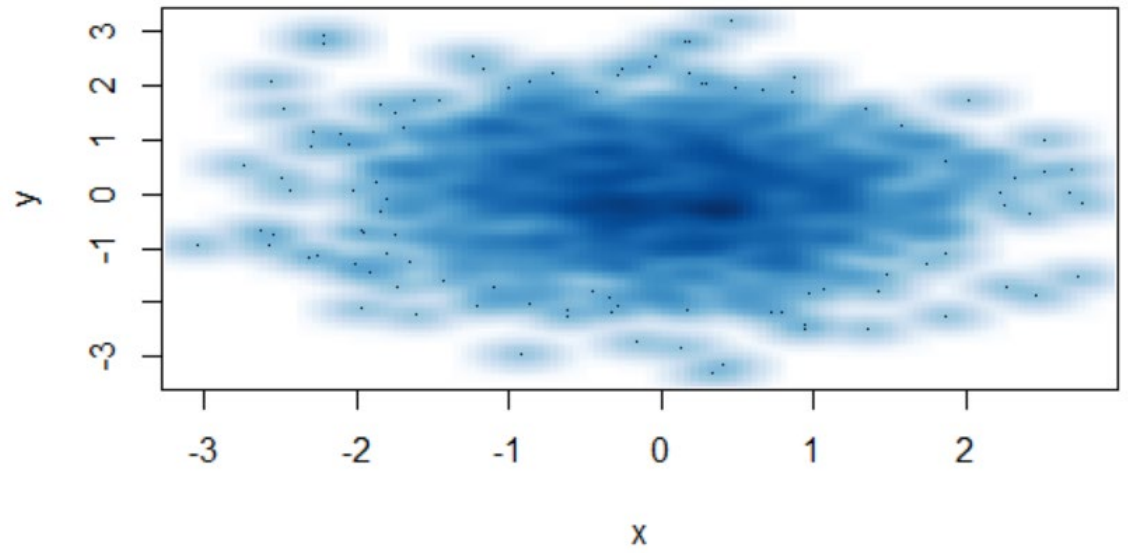
AIM:

To draw the scatter plot using R-tool

PROGRAM:

```
set.seed(9)

x <- rnorm(1000)

y <- rnorm(1000)

smoothScatter(y - x)

smoothScatter(x,y)
```

OUTPUT:




RESULT:

Thus the scatter plot was executed successfully.

## EXPERIMENT-19

## LINEAR REGRESSION:

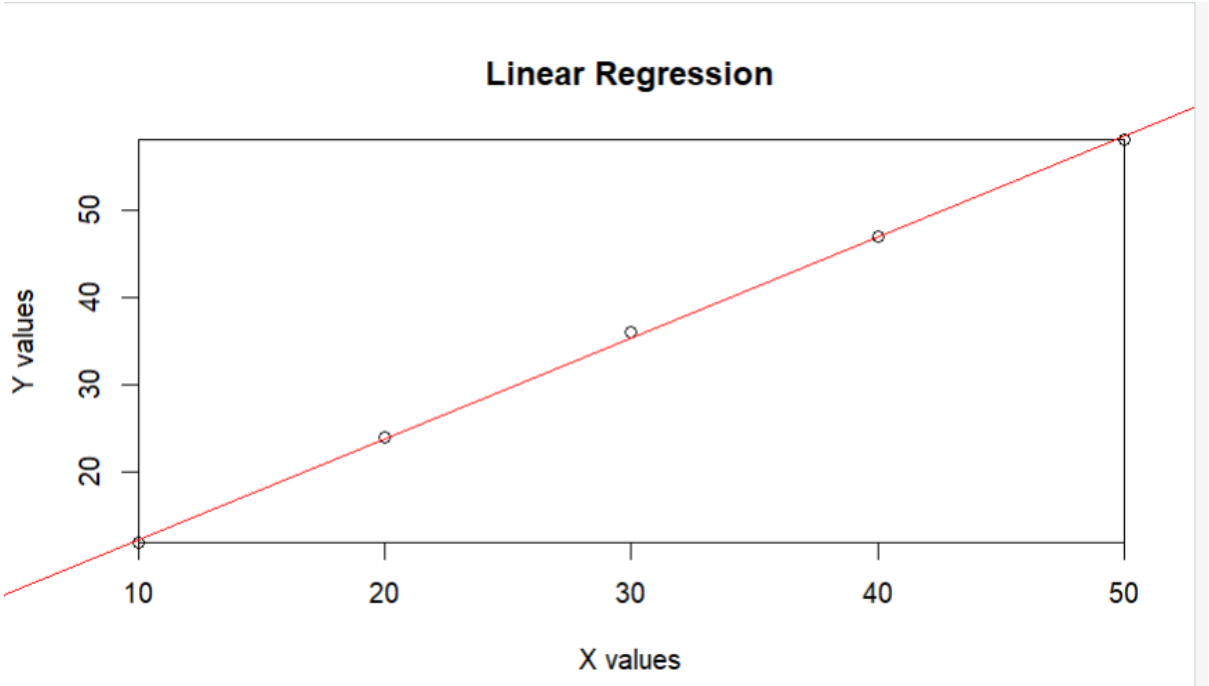
## AIM:

To write thr program for the linear regression using R-tool.

## PROGRAM:

```
x <- c(10, 20, 30, 40, 50)
y <- c(12, 24, 36, 47, 58)
model <- lm(y ~ x)
summary(model)
predicted_values <- predict(model)
plot(x, y, main = "Linear Regression", xlab = "X values", ylab = "Y values")
abline(model, col="red")
```

## OUTPUT:




**RESULT:**

Thus the linear regression program was executed successfully.

## EXPERIMENT-20

## MULTIPLE REGRESSION:

## AIM:

To write the program for the multiple regression.

## PROGRAM:

Input <- diabetes[,c("Age", "BloodPressure", "Glucose")]

Model <- lm(Age~ BloodPressure+Glucose,data=input)

Print(model)

## OUTPUT:

```
> print(diabetes)

Call:
lm(formula = Age ~ BloodPressure + Glucose, data = input)

Coefficients:
  (Intercept)  BloodPressure        Glucose
     14.33937        0.12399        0.08547

> |
```

A<- coef(model)[1]

Print(A)

OUTPUT:

```
> print(A)
(Intercept)
   14.33937
> |
```

xBloodPressure<- coef(model)[2]

yGlucose<- coef(model)[3]

print(xBloodPressure)

print(yGlucose)

**OUTPUT**:

```
> print(yGlucose)
   Glucose
0.08547277
>
```

y = A+xBloodPressure + yGlucose

print(y)

**OUTPUT:**

```
>
> print(y)
(Intercept)
   14.54883
>
```