Implement of the R script using a group of 12 sales price records has been sorted as follows: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215. Partition them into three bins by each of the following methods. (a) equal-frequency (equi depth) partitioning (b) equal-width partitioning (c) clustering

```
# Given sales price records (sorted)

sales_prices <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)


# (a) Equal-Frequency (Equi-Depth) Partitioning

equal_frequency_bins <- split(sales_prices, ceiling(seq_along(sales_prices) / (length(sales_prices) / 3)))

print("Equal-Frequency Bins:")

print(equal_frequency_bins)


# (b) Equal-Width Partitioning

min_val <- min(sales_prices)

max_val <- max(sales_prices)

bin_width <- (max_val - min_val) / 3  # Divide the range into 3 equal intervals

equal_width_bins <- list()


for (i in 1:3) {

  lower_bound <- min_val + (i - 1) * bin_width

  upper_bound <- min_val + i * bin_width

  equal_width_bins[[i]] <- sales_prices[sales_prices >= lower_bound & sales_prices < upper_bound]

}


# Ensure the last bin includes the max value

equal_width_bins[[3]] <- c(equal_width_bins[[3]], max_val)


print("Equal-Width Bins:")

print(equal_width_bins)


# (c) Clustering using K-means

set.seed(42)  # For reproducibility
```

```
kmeans_result <- kmeans(sales_prices, centers = 3)
clusters <- split(sales_prices, kmeans_result$cluster)

print("Clustering (K-Means) Bins:")
print(clusters)
```

A gadget factory has been quite successful for the past 10 years and Ms.Marry, the manager of the company wondering whether to expand the factory this year or not. The cost to expand factory is $2M. With no expansion, expected revenue is $4M if the economy stays good; while only $1.5M if the economy is bad. If manager expands the factory, expected to receive $7M. if economy is good and $3M if economy is bad. Assume that there is a 45% chance of a good economy and a 55% chance of a bad economy. Draw a Decision Tree showing these choices.

```
@relation factory_expansion

@attribute Economy {Good, Bad}
@attribute Revenue numeric
@attribute Decision {Expand, NoExpand}

@data
Good, 5, Expand
Bad, 1, Expand
Good, 4, NoExpand
Bad, 1.5, NoExpand
```

Z-Score Normalization using Mean Absolute Deviation (MAD)

Instead of standard deviation, use the **mean absolute deviation (MAD)**:

Z′=X−μMAD$Z' = \frac{X - \mu}{MAD}$

Where:

MAD=∑|Xi−μ|N$MAD = \frac{\sum |X_i - \mu|}{N}$

r

CopyEdit

```
mad_val <- mean(abs(data - mean_val)) z_score_mad <- (data - mean_val) / mad_val
print(z_score_mad)
```

(d) Decimal Scaling Normalization

Decimal scaling normalizes values based on the maximum absolute value:

X′=X10j$X' = \frac{X}{10^j}$

Where jj is the smallest integer such that **max(|X'|) < 1**.

r

CopyEdit

```
j <- ceiling(log10(max_val)) decimal_scaling_norm <- data / (10^j) print(decimal_scaling_norm)
```

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:

```
> source("~/.active-rstudio-document")
enter the first number:3
enter the second number:2
[1] 6
>
```

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:

```
R 4.2.2 · ~/
> source("~/.active-rstudio-document")
enter the first number:10
enter the second number:2
[1] 5
>
```

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-10

QUANTILE:

AIM:

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

PROGRAM:

```
names<-c("siri","mahi","chiru")

age<-c(23,24,25)

marks<-c(88,78,25)

df<-data.frame(names,age,marks)

quantile(df $age)

write.csv(df,"datafr.csv")
```

OUTPUT:

```
> quantile(df $age)
  0%  25%  50%  75% 100%
23.0 23.5 24.0 29.5 35.0
```

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:

```
names<-c("siri","mahi","chiru")

age<-c(23,24,25)

marks<-c(88,78,25)

df<-data.frame(names,age,marks)

mid range(df $age)

write.csv(df,"datafr.csv")
```

OUTPUT:

```
> range(df $age)
[1] 23 35
```

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:

```
diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")
A<-c(diabetest1$Age)
Mean<-mean(A)
Std<-sd(A)
Zscore<-(A-Mean)/Std
Zscore
```

OUTPUT:

```
> sd(A)
[1] 11.76023
>
```

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:

MEAN

diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")

A<-c(diabetest1$Age)

Mean<-mean(A)

OUTPUT:

```
> mean(A)
[1] 33.24089
>
```

MINIMUM

diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")

A<-c(diabetest1$Age)

Minimum<-Min(diabetest1$Age)

OUTPUT:

```
> Minimum
[1] 21
>
```

MAXIMUM

diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")

A<-c(diabetest1$Age)

Maximum<-Max(diabetest1$Age)

OUTPUT:

```
> Maximum
[1] 81
```

MINMAX

diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")

A<-c(diabetest1$Age)

Maximum<-Max(diabetest1$Age)

Minimum<-Min(diabetest1$Age)

MinMax<-(A-Minimum)/(Maximum-Minimum)

MinMax

OUTPUT:

```
>
> MinMax
  [1] 0.48333333 0.16666667 0.18333333 0.00000000 0.20000000
  [6] 0.15000000 0.08333333 0.13333333 0.53333333 0.55000000
 [11] 0.15000000 0.21666667 0.60000000 0.63333333 0.50000000
 [16] 0.18333333 0.16666667 0.16666667 0.20000000 0.18333333
 [21] 0.10000000 0.48333333 0.33333333 0.13333333 0.50000000
 [26] 0.33333333 0.36666667 0.01666667 0.60000000 0.28333333
 [31] 0.65000000 0.11666667 0.01666667 0.11666667 0.40000000
 [36] 0.20000000 0.23333333 0.41666667 0.10000000 0.58333333
 [41] 0.08333333 0.26666667 0.45000000 0.55000000 0.31666667
 [46] 0.06666667 0.13333333 0.01666667 0.16666667 0.05000000
```

RESULT:

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

EXPERIMENT-14

BAR PLOT AND HORIZONTAL BAR:

AIM:

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

PROGRAM:

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

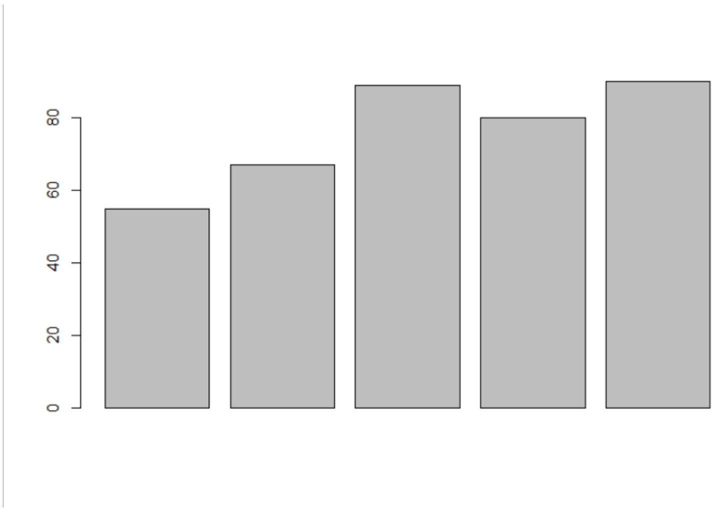
barplot(a)

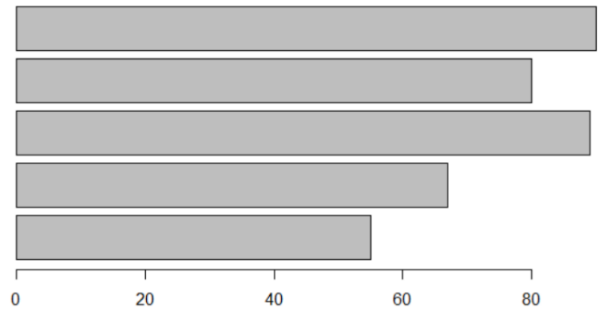
a<-c(55,67,89,80,90)

barplot(a)

barplot(a,horiz=TRUE)

OUTPUT:





RESULT:

Thus the bar and horizontal bar plot was executed successfully.

EXPERIMENT-15

BOX PLOT:

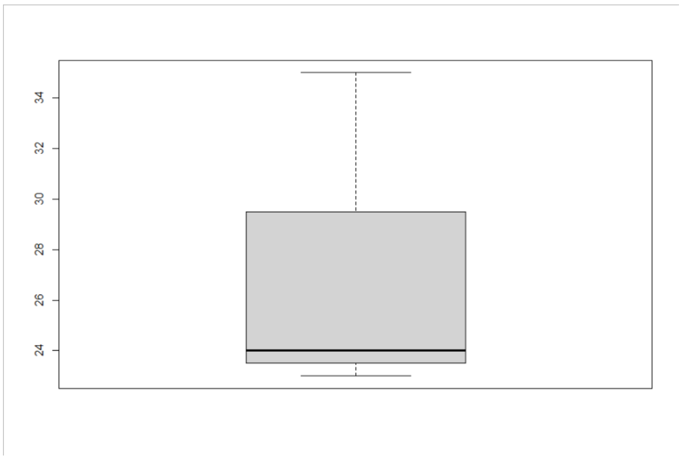
AIM:

To draw the box plot using R-tool.

PROGRAM:

```
names<-c("siri","chru","loki")

age<-c(23,24,25)

marks<-c(88,78,25)

df<-data.frame(names,age,marks)

hist(df$age)

boxplot(df$age)
```

OUTPUT:



RESULT:

Thus the box plot was executed successfully.

EXPERIMENT-16

HISTOGRAM:

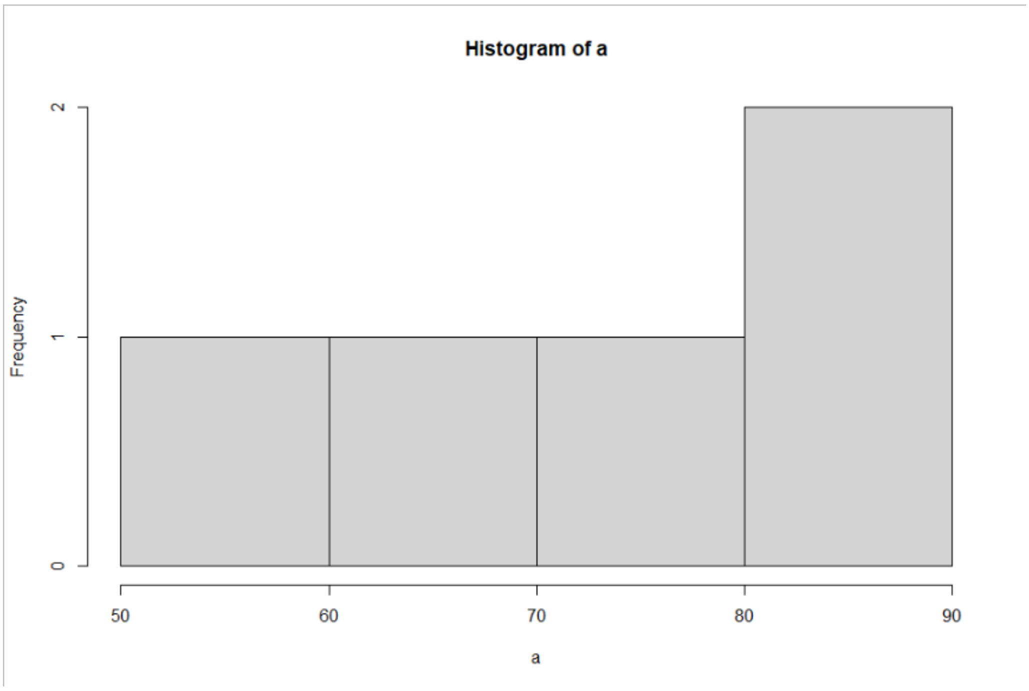
AIM:

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:

```
diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")

diabetest1<-table(diabetest1 $Age,diabetest1 $Insulin)

diabetest1

chisq.test(diabetest1)
```

OUTPUT:

```
> diabetes1

     0 14 15 16 18 22 23 25 29 32 36 37 38 40 41 42 43 44 45 46 48 49 50 51
  21 28  0  0  0  1  0  1  1  0  0  0  0  0  1  0  0  0  0  1  0  0  0  1  1
  22 29  0  0  1  0  0  0  0  0  1  1  1  0  0  0  0  0  1  1  0  0  0  1  0
  23 10  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  1  0  2  0  0  0
  24 15  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  1  0  0  0  0  0  1  0
  25 18  1  0  0  1  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0

    52 53 54 55 56 57 58 59 60 61 63 64 65 66 67 68 70 71 72 73 74 75 76 77
  21  0  0  0  0  1  0  0  0  0  1  0  1  0  1  0  0  0  1  0  0  0  0  2  0
  22  0  1  1  0  0  0  1  0  1  0  1  1  0  1  0  0  0  0  0  0  0  1  1  0
```

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)

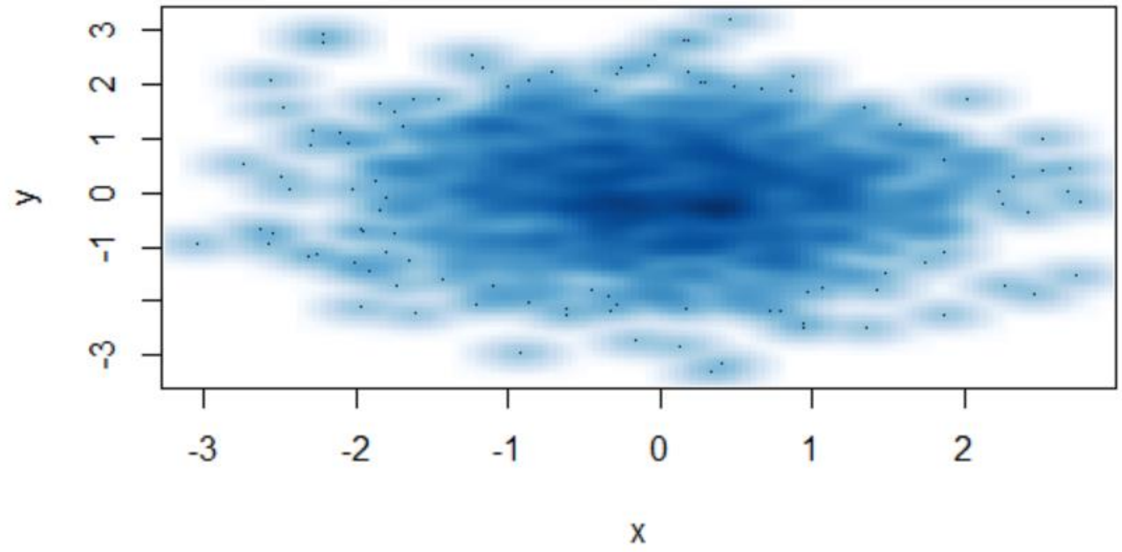
x <- rnorm(1000)

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.

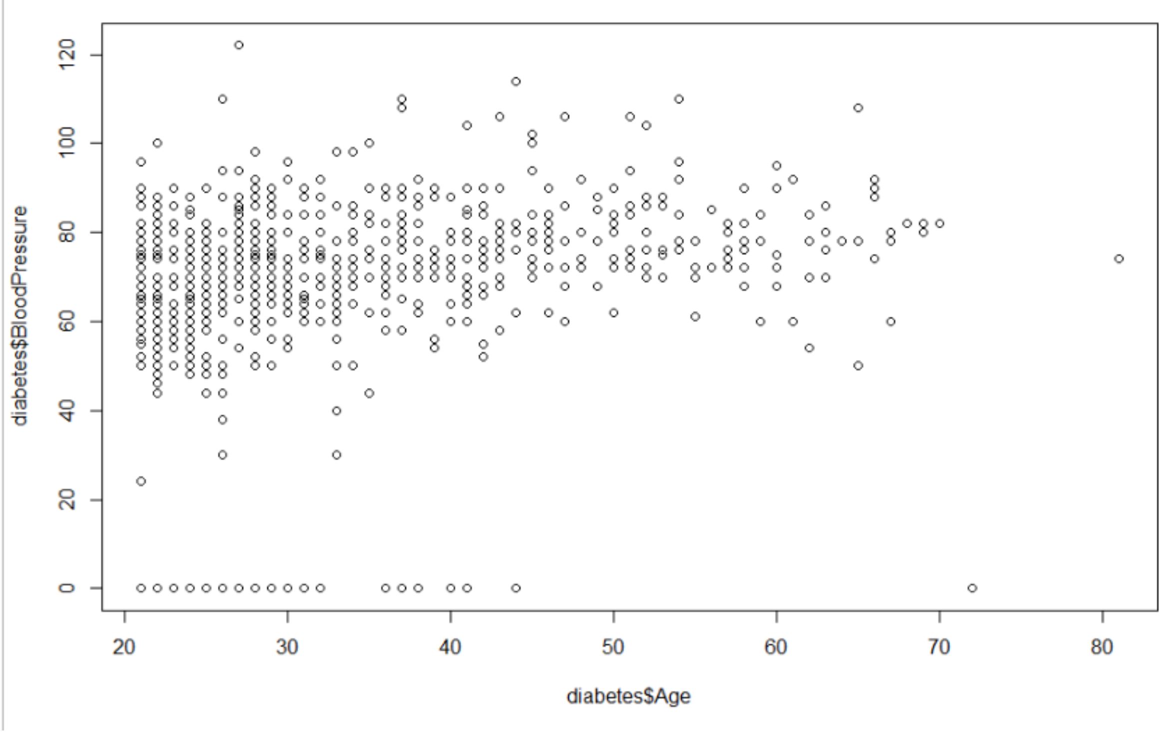
PROGRAM:

Relation <- lm(diabetes$BloodPressure~diabetes$Age)

Png<- (file=”linear regression.png”)

Plot(diabetes$Age, diabetes$BloodPressure, col=”green”, main= “ Linear Regression Analysis” , abline= (lm(diabetes$BloodPressure~ diabetes$Age)), xlab = “BloodPressure”, ylanb= “Age”)

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
>
```