

OUTPUT :

Enter a number1 : 2

Enter a number2 : 2

[1] 4

Exp.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)
```

1 : readLines(0) > 1357
2 : readline(0) > 1358
3 : [1]

RESULT:

Thus the basic program addition are executed successfully.

OUTPUT:

Enter a number: 4

Enter a number2: 2

[1] 2

Exp: 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)
```

1000000

8 1000 1000

2 1000 1000

6 [1]

RESULT:

Thus the basic program subtraction are executed successfully.

OUTPUT:

Enter num₁: 3

Enter num₂: 2

[1] 6

Expt 3

MULTIPLICATION

AIM: To prove the program for multiplication using R-tool.

PROGRAM:

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

RESULT:

Thus the basic program multiplication is executed successfully.

Conversion of binary to decimal & vice versa
Digital to analog

Conversion of digital to analog and vice versa
Digital to analog conversion
Digital to analog conversion
(analog to digital conversion)

OUTPUT:

Enter num1: 10

Enter num2: 2

[1] 5

DIVISION

AIM: To prove the program for division using R-tool.

PROGRAM:

```
num1 = as.integer(readline(prompt = "Enter num1:"))
num2 = as.integer(readline(prompt = "Enter num2:"))
num3 = num1 / num2
print(num3)
```

: ruruO

+ num3 <-

"num3 is result" [1]

RESULT:

Thus the basic program division was executed successfully.

Output:

Enter a num: 4

[1] "Number is even"

ODD OR EVEN

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

PROGRAM:

```
num = as.integer(readline(prompt = "enter a num:"))
if ((num %% 2) == 0) {
    print ("number is even")
} else {
    print ("number is odd")
}
```

3

RESULT:

Thus the basic program odd or even executed successfully.

now is how to manipulate the values of this
list. > print

library

(`mean` & `median` without `apply` function) ->
> (without `sum`)
(`max` & `min`)
> `ifelse`
(`length` function)
> `ifelse`

Output:

```
> mode (df $age)
[1] "numeric"
> mean (df $age)
[1] 27.3333
> median (df $age)
[1] 24
```

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 (name, 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 (name, 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")
```

RESULT:

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

(`min`, `sd`, `var`) \rightarrow mean

(`mean`, `sd`) \rightarrow ipo

(`mean`, `sd`) \rightarrow sd

(`mean`, `sd`, `var`) \rightarrow h

(`mean`, `sd`, `var`)

(`mean`, `sd`) \rightarrow alr

(`min`, `sd`, `var`) \rightarrow min

(`mean`, `sd`) \rightarrow ipo

(`mean`, `sd`) \rightarrow sd

OUTPUT: (`mean`, `sd`, `var`) \rightarrow h
(`mean`, `sd`) \rightarrow ipo

> summary (df\$age)

Min	1 st Qu	Median	Mean	3 rd Qu	Max
23.00	23.50	24.00	27.33	29.50	35.00

SUMMARY

AIM: To write the program for summary using R-tool.

PROGRAM:

```
names <- c ("siri", "mahi", "chien")
age <- c (23, 24, 25)
marks <- c (88, 78, 65)
df <- data.frame (name, age, marks)
summary (df $age)
write.csv (df, "databr.csv")
```

RESULT:

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

Ans. 11

ANSWER

(Greatest Common Factor) \Rightarrow 600

(80, 120, 80) \Rightarrow 80

(20, 80, 80) \Rightarrow 80

(120, 80, 300) \Rightarrow 80

(40, 80) greatest.

(Greatest Common Factor) \Rightarrow 600

OUTPUT:

Enter num1: 5

Enter num2: 6

Enter num3: 4

[1] "Greatest is: 6"

Exp: 8

GREATER AMONG THREE NUMBERS

AIM: To write the program for the greatest among three numbers.

PROGRAM:

```
x <- as.integer(readline(prompt = "Enter num1:"))
y <- as.integer(readline(prompt = "Enter num2:"))
z <- as.integer(readline(prompt = "Enter num3:"))

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))
}
```

RESULT:

Thus the greatest among the three numbers was executed successfully.

Output:

[1] b

Exp 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)
db <- data.frame (names, age, marks)
IQR (db $age)
write.csv (db, "datafe.csv")
```

Age
23 24 25
Q1 Q3 Median Mean Std. Dev.
23.0 25.0 24.0 24.0 2.00

RESULT:

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

OUTPUT:

0%	25%	50%	75%	100%
23.0	23.5	24.0	29.5	35.0

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(name, age, marks)
quantile(df$age)
write.csv(df, "datafr.csv")
```

RESULT:

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

OUTPUT:

[1] 23 35

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)
midrange (df $age)
write.csv (df, "datafr.csv")
```

RESULT:

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

and letters ref. mapping with some of the
original letters becoming stuck with

: 14039

(mists "inert" "sin") and some

(25, 35, 28) > → 32

(25, 35, 36) > → 3123

(dismay, vision) many. stick → 1

(gated (b) question

(veto, sketch*, (b) visualise

OUTPUT:

[1] 11.76028

Z- SCORE NORMALIZATION

Aim: To write the program for z-score normalization using R tool

PROGRAM:

```
diabetes1 <- read_excel ("C:/Users/Jayan/Downloads/  
NARA.xlsx")  
A <- c (diabetes1$Age)  
Mean <- mean (A)  
Std <- sd (A)  
Zscore <- (A-Mean)/Std  
Zscore
```

RESULT:

Thus the z-score normalization using R tool was executed successfully.

OUTPUT:

MEAN

[1] 33.24089

MINIMUM

[1] 21

MAXIMUM

[1] 81

MIN MAX

[1]	0.48333	0.1667	0.1833	0.000	0.200
[6]	0.15000	0.8333	0.1333	0.533	0.550
[11]	0.1500	0.2166	0.600	0.633	0.500
[16]	0.1833	0.1667	0.1667	0.200	0.183
[21]	0.1000	0.4833	0.333	0.133	0.500
[26]	0.333	0.366	0.0166	0.600	0.283
[31]	0.650	0.116	0.0166	0.116	0.400
[36]	0.200	0.233	0.9166	0.100	0.583
[41]	0.833	0.266	0.4500	0.550	0.316
[46]	0.066	0.133	0.0166	0.0166	0.0500

Exp: 13

MIN, MAX, MEAN

Aim: To write the program for the minimum, maximum, mean and minmax using R-Tool.

PROGRAM:

MEAN:

```
diabetes1 <- read-excel ("C:/Users/Jayan/Downloads/NARA.xlsx")
A <- c (diabetes1 $ Age)
Mean <- mean (A)
```

MINIMUM:

```
diabetes1 <- read-excel ("C:/Users/Jayan/Downloads/NARA.xlsx")
A <- c (diabetes1 $ Age)
Minimum <- Min (diabetes1 $ Age)
```

MAXIMUM

```
diabetes1 <- read-excel ("C:/Users/Jayan/Downloads/NARA.xlsx")
A <- c (diabetes1 $ Age)
Maximum <- Max (diabetes1 $ Age)
```

MINMAX

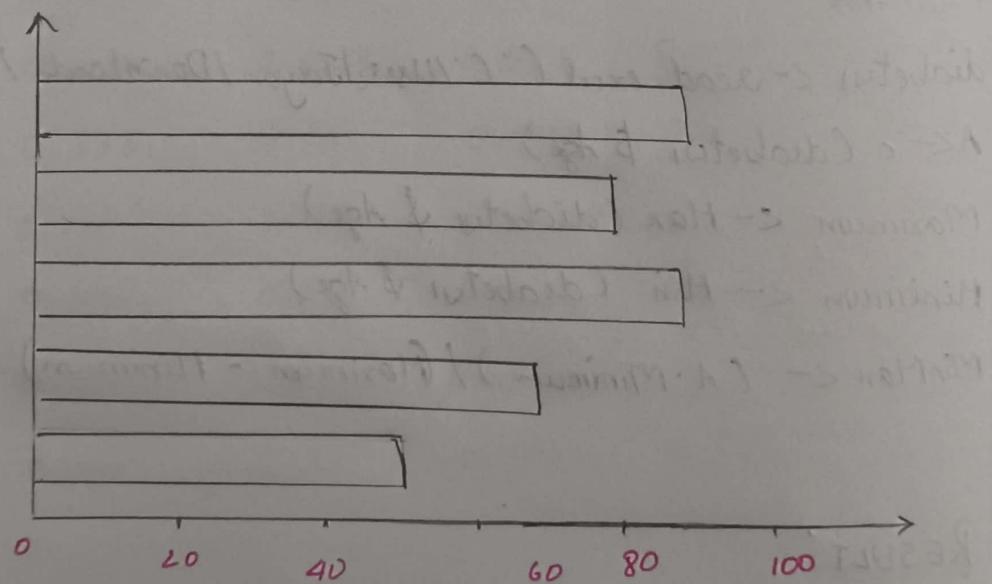
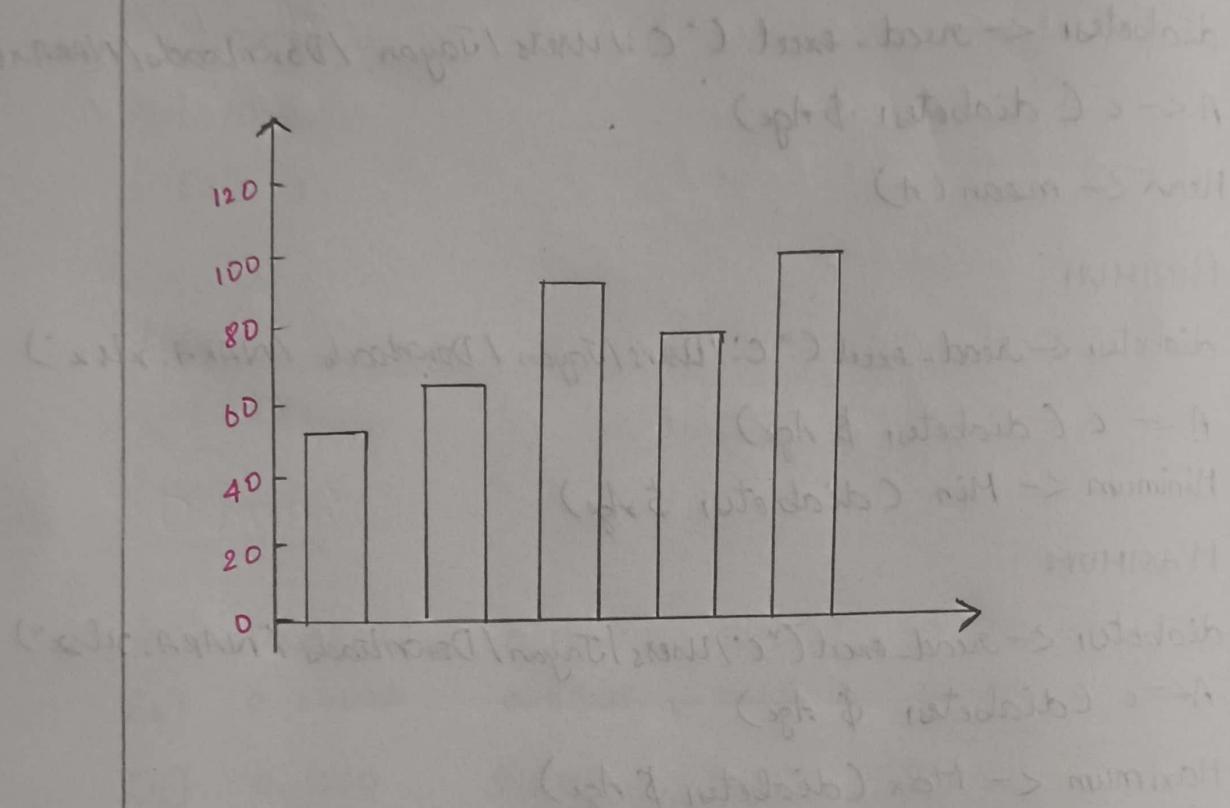
```
diabetes1 <- read-excel ("C:/Users/Jayan/Downloads/NARA.xlsx")
A <- c (diabetes1 $ Age)
Maximum <- Max (diabetes1 $ Age)
Minimum <- Min (diabetes1 $ Age)
MinMax <- (A - Minimum) / (Maximum - Minimum)
```

RESULT:

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

Output will be mapped with values of input
but S operation has been mentioned

Output:



Ex: 14

BAR PLOT AND HORIZONTAL PLOT

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.

we have to find all such 3D trihedra
that it gives

163027

(0, 0, 0), (1, 1, 0) \Rightarrow 1

(1) left

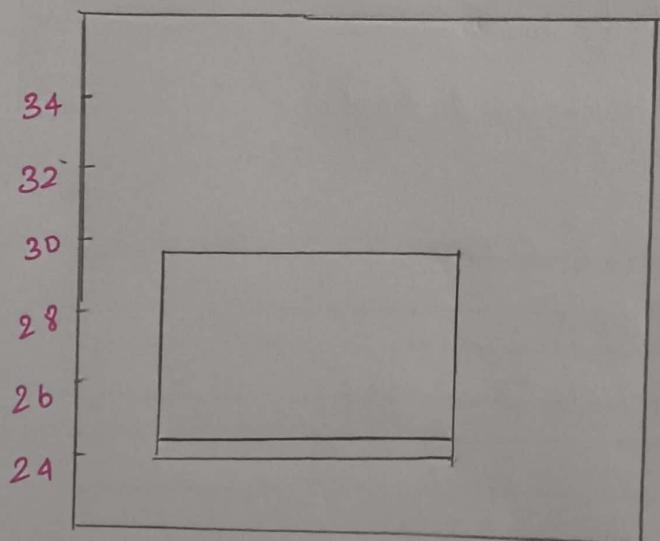
(0, 0, 0), (1, 1, 1) \Rightarrow 1

(2) right

(0, 0, 1), (1, 1, 1) \Rightarrow 1

(3) top

OUTPUT:



Exp: 15

Box Plot

Aim: To draw the box plot using R tool

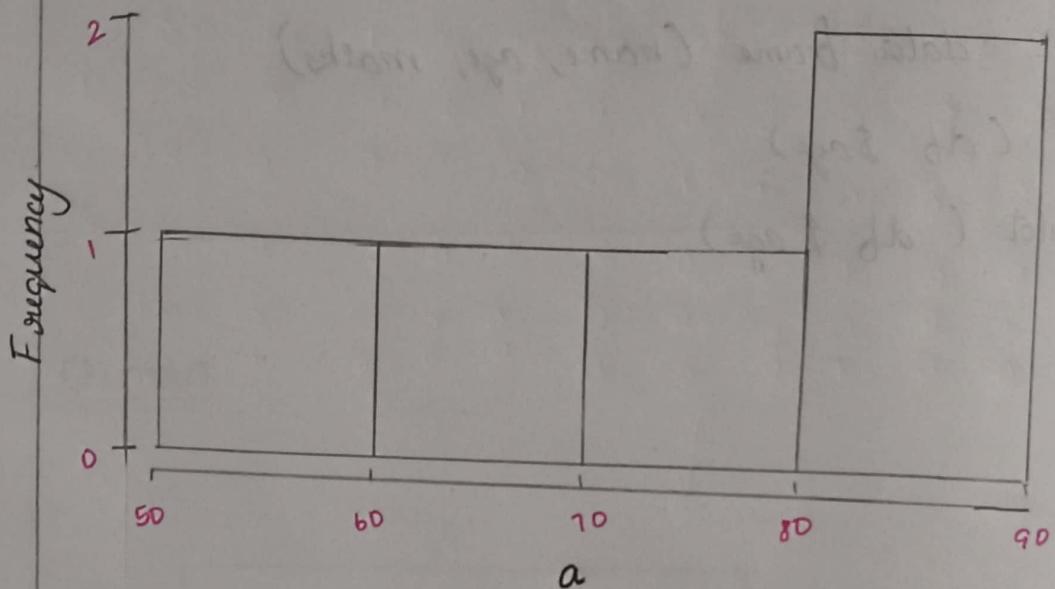
PROGRAM:

```
names <- c ("Siri", "Chiru", "Loki")  
age <- c (23, 24, 25)  
marks <- c (88, 78, 25)  
df <- data.frame (name, age, marks)  
hist (df $age)  
boxplot ( df $age)
```

RESULT:

Thus the box plot was executed successfully.

OUTPUT:



Exp.16

HISTOGRAM

Aim: To draw the histogram plot using R tool.

PROGRAM:

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

```
hist(a)
```

78 62 68 48 38 28 18 28 31 41 41 0
0 6 0 0 0 0 0 1 0 0 0 0 65 12
0 1 0 0 0 1 1 0 0 1 0 0 83 18
0 1 0 0 0 1 0 0 0 0 1 0 0 1 88
0 0 1 0 0 0 0 0 0 0 0 0 21 45
0 0 0 1 0 0 0 0 0 0 0 1 32 88

RESULT:

Thus the histogram plot was executed successfully.

OUTPUT:

	0	14	15	16	18	22	23	25	29	32	36	37	38
21	28	0	0	0	1	0	0	0	0	0	0	1	0
22	29	0	0	1	0	0	1	1	0	0	0	1	0
23	10	0	1	0	0	0	0	1	0	0	0	1	0
24	15	0	0	0	0	0	0	0	0	0	1	0	0
25	18	1	0	0	0	0	0	0	1	0	0	0	0

Exp: 17

CORRELATION ANALYSIS

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

PROGRAM:

```
diabetes1 <- read_excel ("C:/Users/Jayan/Downloads/  
NARA.xlsx")  
diabetes1 <- table(diabetes1 $ Age, diabetes1 $ Insulin)  
diabetes1  
chi.sq.test (diabetes1)
```

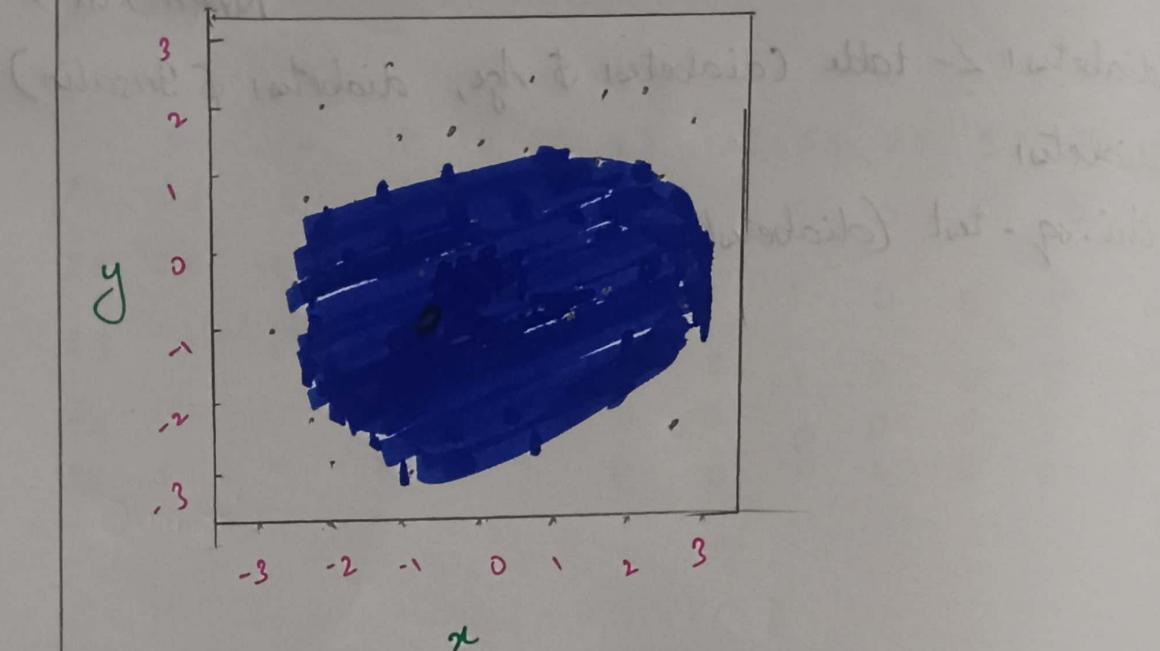
RESULT:

Thus the correlation analysis was executed successfully.

System stabilisieren (durch geleg. Zeit- oder Phasenverschiebung)

OUTPUT:

• Zeitverlauf



Lösung
bekannt war die jahres periodischen mit und
• Phasenverschiebung

SCATTER PLOT

Aim: To draw the scatter plot using R-tool.

PROGRAM:

set.seed(9)

n <- rnorm(1000)

y <- rnorm(1000)

smoothScatter(y~n)

smoothScatter(n,y)

RESULT:

Thus the scatter plot was executed successfully.

(e) loss due

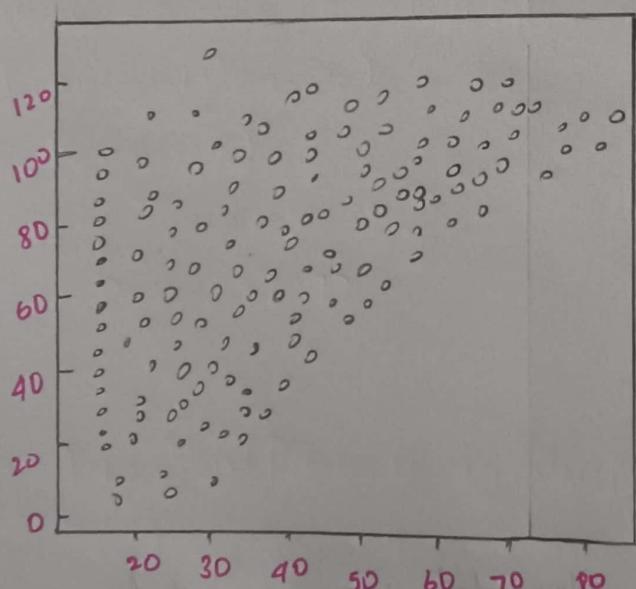
(loss) margin = 0

(loss) output = 0

(loss) attachment

(loss) return steady

OUTPUT:



LINEAR REGRESSION

Aim: To write the program for the linear regression using R tool.

PROGRAM:

```
Relation <- lm (diabetes $ Blood Pressure ~ diabetes $ Age)
```

```
Png <- (file = "linear regression.png")
```

```
Plot (diabetes $ Age, diabetes $ Blood Pressure, col = "green",
      main = "Linear Regression Analysis",
      abline = (lm (diabetes $ Blood Pressure ~ diabetes $ Age)),
      xlab = "Blood Pressure", ylab = "Age")
```

RESULT:

Thus the linear regression program was executed successfully.

OUTPUT:

call:

lm(formula = Age ~ BloodPressure + Glucose, data = ^{input}inpt)

Coefficients:

(Intercept)	Blood Pressure	Glucose
14.33937	0.12399	0.08547

=>

A <- coef(model)[1]

Print(A)

(Intercept)

14.33937

=>

xBloodPressure <- coef(model)[2]

yGlucose <- coef(model)[3]

print(xBloodPressure)

print(yGlucose)

Glucose

0.08547277

=>

y = A + xBloodPressure + yGlucose

print(y)

(Intercept)

14.5883

MULTIPLE REGRESSION

Aim: To write the program for multiple regression.

PROGRAM:

```
Input <- diabetes[,c("Age", "BloodPressure", "Glucose")]
Model <- lm (Age ~ BloodPressure + Glucose, data = input)
Print (model)
```

RESULT:

Thus the multiple regression was executed successfully.