

**EX.NO: 01**

**Date:**

## **ODD\_OR\_EVEN, ADDITION AND SUBTRACTION PROGRAM USING R TOOLS**

### **PROBLEM STATEMENT:**

Download the dataset from the UCI repository (or) any other appropriate website and perform (or) implement the central tendency measures.(mean, median, mode and midrange) and Data dispersion technique including summary.

### **DESCRIPTION:**

This data comes from the 2010 census profile of general population and housing characteristics. Zip codes are limited to those that fall at least partially within LA city boundaries. The dataset will be updated after the next census in 2020.

### **CENTRAL TENDENCY:**

- i. **Odd or Even :** Odd numbers are those numbers that cannot be divided into two equal parts, whereas even numbers are those numbers that can be divided into two equal parts.
- ii. **Addition :** adding something, especially two or more numbers.
- iii. **Subtraction :** subtracting something, especially two or more numbers.

### **INPUTS AND OUTPUTS OF BASIC PROGRAM :**

#### **ODD OR EVEN :**

#### **INPUT :**

```
num=as.integer(readline(prompt = "Enter a number : "))
if((num%%2) == 0){
  print('Number is 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"
>
```

## ADDITION :

### INPUT :

```
num1 = as.integer(readline(prompt= "Enter a number1 : "))
num2 = as.integer(readline(prompt= "Enter a number2 : "))
num3 = num1 + num2
print(num3)|
```

### OUTPUT :

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

## SUBTRACTION :

### INPUT :

```
num1 = as.integer(readline(prompt= "Enter a number1 : "))
num2 = as.integer(readline(prompt= "Enter a number2 : "))
num3 = num1 - num2
print(num3)|
```

### OUTPUT :

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

## RESULT:

Thus the basic programs like odd or even, addition and subtraction are executed successfully.

**EX.NO: 02**

**Date:**

## **CENTRAL TENDENCY AND DATA DISPERSION MEASURES USING R-TOOL**

### **PROBLEM STATEMENT:**

Download the dataset from the UCI repository (or) any other appropriate website and perform (or) implement the central tendency measures.(mean, median, mode and midrange) and Data dispersion technique including summary.

### **DESCRIPTION:**

This data comes from the 2010 census profile of general population and housing characteristics. Zip codes are limited to those that fall at least partially within LA city boundaries. The dataset will be updated after the next census in 2020.

### **CENTRAL TENDENCY:**

- i. **Mean** :The mean is the average of the numbers: a calculated "central" value of a set of numbers.
- ii. **Median** :The median is a statistical term that is one way of finding the 'average' of a set of data points.
- iii. **Mode** :The mode of a set of data values is the value that appears most often.
- iv. **Summary** :A summary table stores data that has been aggregated in a way that answers a meancommon (or resource-intensive) business query.

### **INPUTS AND OUTPUTS OF CENTRAL TENDENCY AND DATA DISPERSION:**

#### **MEAN:**

#### **INPUT :**

```
names<-c("Ram", "Shyam", "Kumar")
age<-c(23,24,35)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
mean(df $age)
write.csv(df,"datafr.csv")
```

#### **OUTPUT :**

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

## MEDIAN :

### INPUT :

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
median(df $age)
write.csv(df,"datafr.csv")|
```

### OUTPUT :

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

## MODE :

### INPUT :

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
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"
```

## SUMMARY :

### INPUT :

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
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 tendency and measures of dispersion have been executed successfully. The outlier values are from more than upper fence there are no lower fence values.

**EX.NO: 03**

**Date:**

## **CENTRAL TENDENCY AND DATA DISPERSION MEASURES USING R-TOOL**

### **PROBLEM STATEMENT:**

Download the dataset from the UCI repository (or) any other appropriate website and perform (or) implement the central tendency measures.(mean, median, mode and midrange) and Data dispersion technique including summary.

### **DESCRIPTION:**

This data comes from the 2010 census profile of general population and housing characteristics. Zip codes are limited to those that fall at least partially within LA city boundaries. The dataset will be updated after the next census in 2020.

### **MEASURES OF DISPERSION:**

- i. **Inter Quartile Range :**The interquartile range (IQR) is a measure of variability, based on dividing a data set into quartiles. Quartiles divide a rank-ordered data set into four equal parts.
- ii. **Quartiles :**A quartile is a statistical term describing a division of observations into four defined intervals based upon the values of the data and how they compare to the entire set of observations.
- iii. **Mid Range :**The arithmetic mean of the largest and the smallest values in a sample or other group.

### **INPUTS AND OUTPUTS OF CENTRAL TENDENCY AND DATA DISPERSION:**

**IQR :**

**INPUT :**

```
names<-c("Ram", "Shyam", "Kumar")
age<-c(23,24,35)
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
```

## QUANTILE :

### INPUT :

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
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
```

## RANGE :

### INPUT :

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
range(df $age)
write.csv(df,"datafr.csv")
```

### OUTPUT :

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

## RESULT:

Thus the central tendency and measures of dispersion have been executed successfully. The outlier values are from more than upper fence there are no lower fence values.

**EX.NO: 04**

**Date :**

## **PLOTTING GRAPHS USING R-TOOL**

### **PROBLEM STATEMENT:**

Plot the boxplot, barplot and horizontal barplot for the dataset which was taken in the previous exercise.

### **DESCRIPTION:**

Consider a dataset `diabetes.csv`, where it contains the attributes are Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcomes.

### **IMPLEMENTATION:**

- i. BoxPlot
- ii. BarPlot

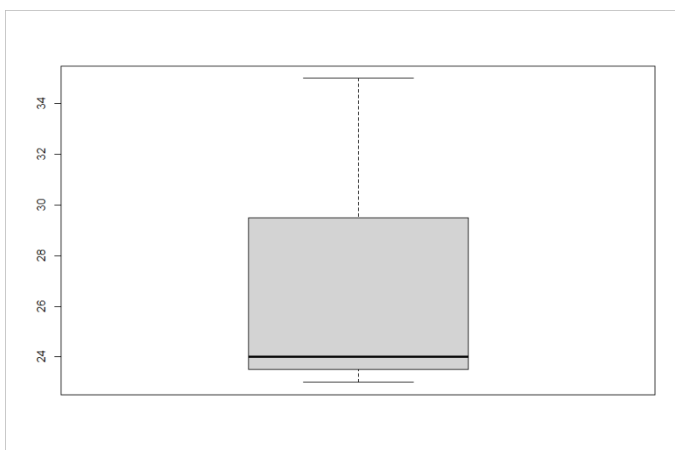
### **BOXPLOT :**

A box plot is a graphical rendition of statistical data based on the minimum, first quartile, median, third quartile, and maximum. The term "box plot" comes from the fact that the graph looks like a rectangle with lines extending from the top and bottom. Because of the extending lines, this type of graph is sometimes called a box-and-whisker plot.

### **INPUT :**

```
names<-c("Ram","Shyam","Kumar")
age<-c(23,24,35)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
hist(df$age)
boxplot(df$age)
```

### **OUTPUT:**





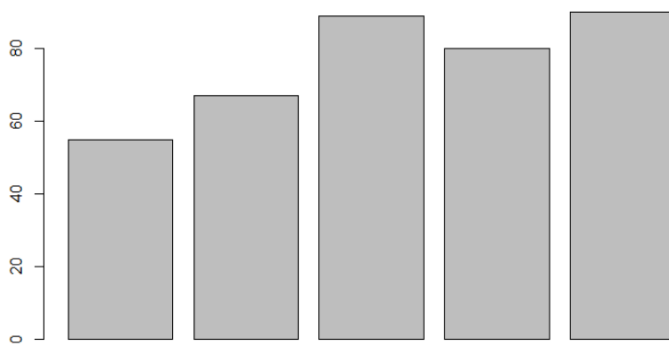
## BARPLOT :

A barplot (or barchart) is one of the most common types of graphic. It shows the relationship between a numeric and a categoric variable. Each entity of the categoric variable is represented as a bar. The size of the bar represents its numeric value.

### INPUT :

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

### OUTPUT :

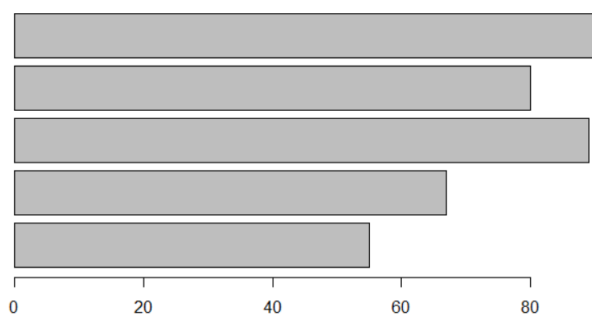


## HORIZONTAL BARPLOT :

### INPUT :

```
a<-c(55,67,89,80,90)
barplot(a)
barplot(a,hORIZ = TRUE)
```

### OUTPUT :



**RESULT:**

Thus, the plotting of graphs like boxplot, barplot and horizontal barplot for the given dataset has been successfully completed.

**EX.NO: 05**

**Date :**

## **PLOTTING GRAPHS USING R-TOOL**

### **PROBLEM STATEMENT:**

Plot the histogram and scatterplot for the dataset which was taken in the previous exercise.

### **DESCRIPTION:**

Consider a dataset `diabetes.csv`, where it contains the attributes are Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcomes

### **IMPLEMENTATION:**

- i. Histogram
- ii. Scatterplot (Scatter Smooth)

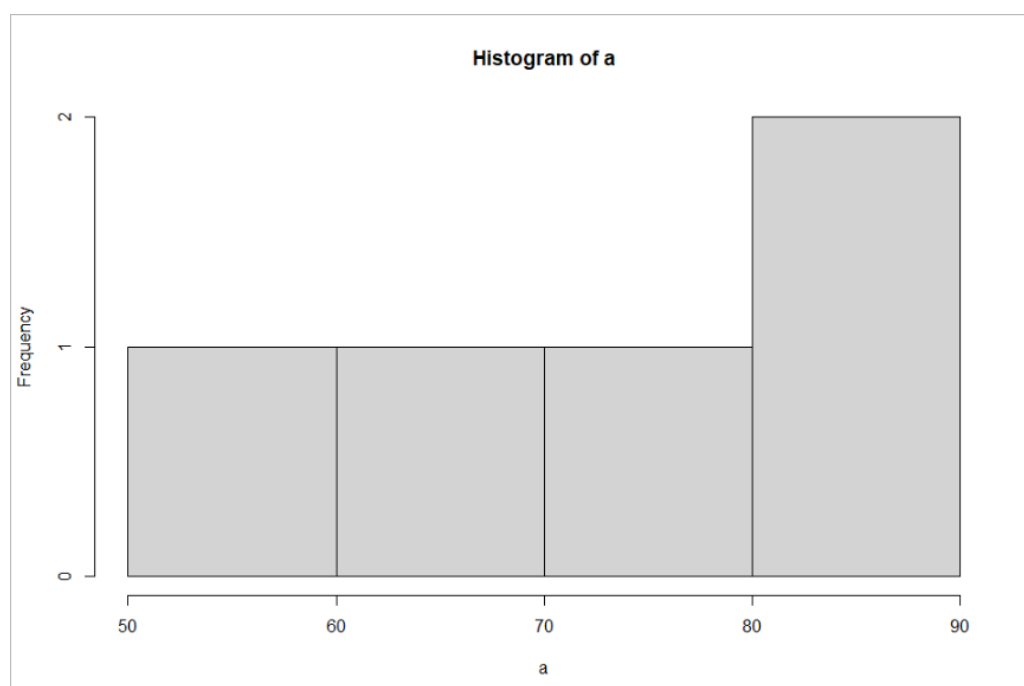
### **HISTOGRAM :**

A diagram consisting of rectangles whose area is proportional to the frequency of a variable and whose width is equal to the class interval.

**INPUT :**

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

**OUTPUT :**



## SCATTERPLOT :

### INPUT :

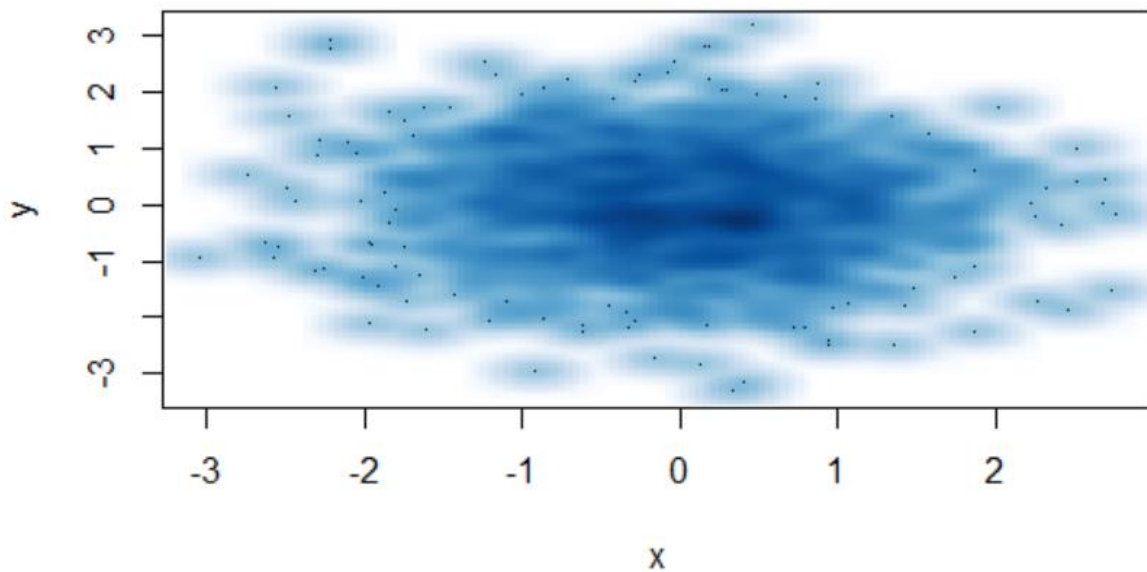
```
# Data
set.seed(9)
x <- rnorm(1000)
y <- rnorm(1000)

# Smooth scatter plot
smoothScatter(y ~ x)

# Equivalent to:
smoothScatter(x, y)
```

### OUTPUT :

---



### RESULT:

Thus, the plotting of graphs like histogram and scatterplot for the given dataset has been successfully completed.

**EX.NO : 06**

**Date :**

## **PERFORM CORRELATION ANALYSIS AND NORMALIZATION USING R-TOOL**

### **PROBLEM STATEMENT :**

Perform the correlation analysis for the numerical attribute using pearson coefficient and for categorical attribute using chi-square and also, perform the normalization technique using z score for the given data frames of particular dataset.

### **DESCRIPTION :**

A dataset of name diabetes.csv is given for the correlation analysis, to calculate or to correlate between Age and Insulin and the same dataset for the performance of normalization technique.

- CORRELATION ANALYSIS:**

#### **STEPS INVOLVED:**

- Create a new table with required dataframes.
- After that apply the formula or query for the chi-square test.

#### **QUERIES:**

- `diabetes1<-table(diabetes$Age,diabetes$Insulin)`
- `diabetes1`
- `chi sq.test(diabetes1)`

### **INPUT :**

```
diabetes1<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")

#step 1
diabetes1<-table(diabetes1 $Age,diabetes1 $Insulin)
diabetes1

#step 2
chisq.test(diabetes1)|
```

### **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 1 0 0 1 0 0 1 0 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
```

```
> chisq.test(diabetes1)

Pearson's Chi-squared test

data:  diabetes1
X-squared = 7561.7, df = 9435, p-value = 1
```

- **Z SCORE NORMALIZATION :**

- `A<- c(diabetes$Age)`
- `Mean<- mean(A)`
- `Std<- sd(A)`
- `Zscore<- (A-Mean)/Std`
- `Zscore`

**INPUT :**

```
diabetes<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A<-c(diabetes$Age)
Mean<-mean(A)
std<-sd(A)
Zscore<-(A-Mean)/Std
Zscore
```

**OUTPUT :**

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

**RESULT:**

Thus, the correlation analysis and normalization for the given dataset has been successfully executed and observed.

**EX.NO : 07**

**Date :**

## **PERFORM CORRELATION ANALYSIS AND NORMALIZATION USING R-TOOL**

### **PROBLEM STATEMENT :**

Perform the correlation analysis for perform the normalization technique for the given data frames of particular dataset.

### **DESCRIPTION :**

A dataset of name diabetes.csv is given for the correlation analysis, to calculate or to correlate between Age and Insulin and the same dataset for the performance of normalization technique.

- **NORMALIZATION :**

- i. **MEAN NORMALIZATION**

- `A<-c(diabetes$Age)`
    - `Mean<-mean(A)`

**INPUT :**

```
diabetes<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A<-c(diabetes$Age)
#step 1
Mean<-mean(A)
```

**OUTPUT :**

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

- ii. **MINIMUM NORMALIZATION**

- `A<-c(diabetes$Age)`
    - `Minimum<-min(diabetes$Age)`

**INPUT :**

```
diabetes<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A<-c(diabetes$Age)
#step 2
Minimum<-min(diabetes$Age)
```

**OUTPUT :**

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

### iii. MAXIMUM NORMALIZATION

- `A<-c(diabetes$Age)`
- `Maximum<-max(diabetes$Age)`

INPUT :

```
diabetes<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A<-c(diabetes$Age)
|
#step 3
Maximum<-max(diabetes$Age)
```

OUTPUT :

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

### iv. MINMAX NORMALIZATION

- `A<-c(diabetes$Age)`
- `MinMax<-(A-Minimum)/(Maximum-Minimum)`
- `MinMax`

INPUT :

```
diabetes<-read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A<-c(diabetes$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
[51] 0.01666667 0.08333333 0.15000000 0.61666667 0.35000000
```



v. **DECIMAL SCALING NORMALIZATION**

- $A=c(\text{diabetes\$Age})$
- $\text{decimalscaling}=(A/100)$
- `decimalscaling`

**INPUT :**

```
diabetes=read.csv("D:\\folders\\DWHDM\\diabetes.csv")
A=c(diabetes$Age)
decimalscaling=(A/100)
decimalscaling
```

**OUTPUT :**

```
> decimalscaling
[1] 0.50 0.31 0.32 0.21 0.33 0.30 0.26 0.29 0.53 0.54 0.30 0.34 0.57 0.59
[15] 0.51 0.32 0.31 0.31 0.33 0.32 0.27 0.50 0.41 0.29 0.51 0.41 0.43 0.22
[29] 0.57 0.38 0.60 0.28 0.22 0.28 0.45 0.33 0.35 0.46 0.27 0.56 0.26 0.37
[43] 0.48 0.54 0.40 0.25 0.29 0.22 0.31 0.24 0.22 0.26 0.30 0.58 0.42 0.21
[57] 0.41 0.31 0.44 0.22 0.21 0.39 0.36 0.24 0.42 0.32 0.38 0.54 0.25 0.27
[71] 0.28 0.26 0.42 0.23 0.22 0.22 0.41 0.27 0.26 0.24 0.22 0.22 0.36 0.22
[85] 0.37 0.27 0.45 0.26 0.43 0.24 0.21 0.34 0.42 0.60 0.21 0.40 0.24 0.22
[99] 0.23 0.31 0.33 0.22 0.21 0.24 0.27 0.21 0.27 0.37 0.25 0.24 0.24 0.46
[113] 0.23 0.25 0.39 0.61 0.38 0.25 0.22 0.21 0.25 0.24 0.23 0.69 0.23 0.26
[127] 0.30 0.23 0.40 0.62 0.33 0.33 0.30 0.39 0.26 0.31 0.21 0.22 0.29 0.28
[141] 0.55 0.38 0.22 0.42 0.23 0.21 0.41 0.34 0.65 0.22 0.24 0.37 0.42 0.23
```

**RESULT:**

Thus, the correlation analysis and normalization for the given dataset has been successfully executed and observed.

**EX.No: 08**

**Date :**

## **REGRESSION ANALYSIS USING R TOOL**

### **PROBLEM STATEMENT :**

Perform the linear regression and multiple regression for the given dataset.

### **DESCRIPTION :**

Consider a dataset of diabetes.csv with the attributes pregnancies, Glucose, BloodPressure, SkinThickness, BMI, Diabetes, Age, Outcome for the analysis. There will be linear regression analysis between Age and BloodPressure. Where, for the multiple regression, the analysis is between Age, BloodPressure, Glucose from the dataset.

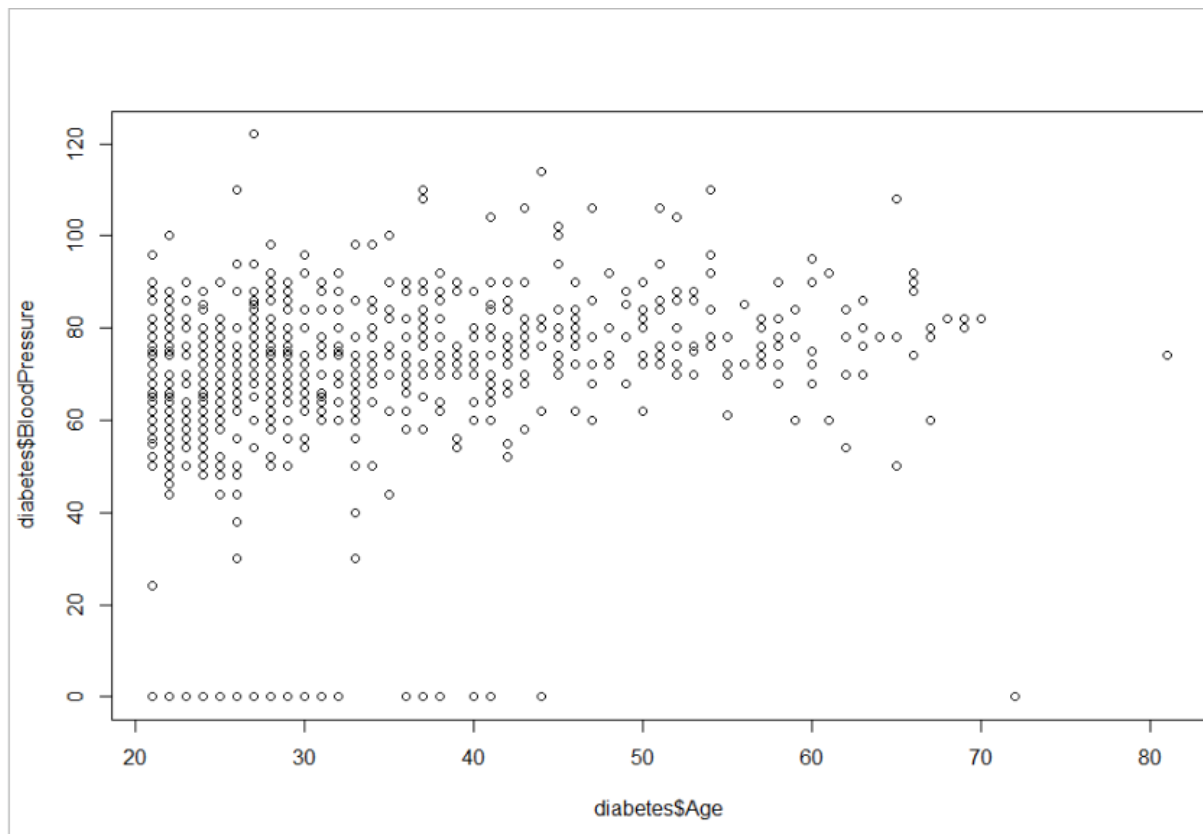
#### **❖ LINEAR REGRESSION :**

Linear regression is a kind of statistical analysis that attempts to show a relationship between two variables. Linear regression looks at various data points and plots a trend line. Linear regression can create a predictive model on apparently random data, showing trends in data, such as in cancer diagnoses or in stock prices.

#### **INPUT :**

- `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", ylab= "Age")`

#### **OUTPUT:**



### INPUT :

- `A<- data.frame(diabetes$Age)`
- `Result<- predict(relation, A)`
- `Print(Result)`

### OUTPUT :

```
> A<- data.frame(diabetes$Age)
> Result<- predict(relation, A)
> print(Result)
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
75.71244	68.22204	68.61627	64.27972	69.01050	67.82781	66.25088	67.43358	76.89514	77.28937	67.82781	69.40474	78.47207	79.26053	76.10668
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
68.61627	68.22204	68.22204	69.01050	68.61627	66.64511	75.71244	72.16436	67.43358	76.10668	72.16436	72.95282	64.67395	78.47207	70.98166
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
79.65476	67.03935	64.67395	67.03935	73.74129	69.01050	69.79897	74.13552	66.64511	78.07783	66.25088	70.58743	74.92398	77.28937	71.77013
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
65.85665	67.43358	64.67395	68.22204	65.46242	64.67395	66.25088	67.82781	78.86630	72.55859	64.27972	72.16436	68.22204	73.34705	64.67395
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
64.27972	71.37589	70.19320	65.46242	72.55859	68.61627	70.98166	77.28937	65.85665	66.64511	67.03935	66.25088	72.55859	65.06819	64.67395
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
64.67395	72.16436	66.64511	66.25088	65.46242	64.67395	64.67395	70.19320	64.67395	70.58743	66.64511	73.74129	66.25088	72.95282	65.46242
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
64.27972	69.40474	72.55859	79.65476	64.27972	71.77013	65.46242	64.67395	65.06819	68.22204	69.01050	64.67395	64.27972	65.46242	66.64511
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
64.27972	66.64511	70.58743	65.85665	65.46242	65.46242	74.13552	65.06819	65.85665	71.37589	80.04899	70.98166	65.85665	64.67395	64.27972
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
65.85665	65.46242	65.06819	83.20285	65.06819	66.25088	67.82781	65.06819	71.77013	80.44323	69.01050	69.01050	67.82781	71.37589	66.25088
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150

### ❖ MULTIPLE REGRESSION :

Multiple regression is a statistical tool used to derive the value of a criterion from several other independent, or predictor, variables. It is the simultaneous combination of multiple factors to assess how and to what extent they affect a certain outcome.

### INPUT :

- `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

> |
```

### INPUT :

- `A<- coef(model)[1]`
- `Print(A)`

### OUTPUT:

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

> |
```

**INPUT :**

- `xBloodPressure<- coef(model)[2]`
- `yGlucose<- coef(model)[3]`
- `print(xBloodPressure)`
- `print(yGlucose)`

**OUTPUT:**

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

**INPUT :**

- $y = A + x\text{BloodPressure} + y\text{Glucose}$
- `print(y)`

**OUTPUT:**

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

**RESULT :**

Thus, the linear regression and the multiple regression analysis for the given dataset has been successfully completed.