1.

AIM: To write the program for median for give age and frequency data.

Age	frequency
1-5	200
5-15	450
15-20	300
20-50	1500
50-80	700
80-110	44

PROGRAM:

```
#age, frequency
age<-c(5,15,20,50,80,110)
frequency<-c(200,450,300,1500,700,44)
median(age)
median(frequency)
```

OUTPUT:

```
> #age, frequency
> age<-c(5,15,20,50,80,110)
> frequency<-c(200,450,300,1500,700,44)
> median(age)
[1] 35
> median(frequency)
[1] 375
```

RESULT:

Thus, the program for median for given age and frequency data is executed successfully.

2.

AIM: To write the program for mean, median, mode and range.

The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

PROGRAM:

#mean, median, mode

age<c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)

mean(age)

median(age)

mode_age<-names(table(age))[table(age)==max(table(age))]

mode_age

range(age)

OUTPUT:

```
> #mean,median,mode,quatile
> age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
> mean(age)
[1] 29.96296
> median(age)
[1] 25
> mode_age<-names(table(age))[table(age)==max(table(age))]
> mode_age
[1] "25" "35"
> range(age)
[1] 13 70
```

RESULT:

Thus, the program for mean, median, mode and range for given data is executed successfully.

3.

AIM: To write the program for below using R tool

a) Smoothing by bin mean b) Smoothing by bin median c) Smoothing by bin boundaries

Data:11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,

72,73,75

PROGRAM:

```
data<-
c(11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75)

bins <- 5

bin_indices <- cut(data, bins)

mean_smooth <- tapply(data, bin_indices, mean)

print(mean_smooth)

median_smooth <- tapply(data, bin_indices, median)

median_smooth

min_max_smooth <- tapply(data, bin_indices, function(x) c(min(x), max(x)))

print(min_max_smooth)
```

OUTPUT:

```
> median_smooth <- tapply(data, bin_indices, median)
> median_smooth
(10.9,23.8] (23.8,36.6] (36.6,49.4] (49.4,62.2] (62.2,75.1]
       19.5        27.0        45.0          NA        72.5
> data <- c(11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75)
> bins <- 5
> bin_indices <- cut(data, bins)
> mean_smooth <- tapply(data, bin_indices, mean)
> print(mean_smooth)
(10.9,23.8] (23.8,36.6] (36.6,49.4] (49.4,62.2] (62.2,75.1]
   17.78571    27.00000    43.75000          NA    72.75000
> median_smooth <- tapply(data, bin_indices, median)
> median_smooth
(10.9,23.8] (23.8,36.6] (36.6,49.4] (49.4,62.2] (62.2,75.1]
       19.5        27.0        45.0          NA        72.5
> min_max_smooth <- tapply(data, bin_indices, function(x) c(min(x), max(x)))
> print(min_max_smooth)
$`(10.9,23.8]`
[1] 11 23

$`(23.8,36.6]`
[1] 24 30

$`(36.6,49.4]`
[1] 40 45

$`(49.4,62.2]`
NULL

$`(62.2,75.1]`
[1] 71 75
```

RESULT: Thus, the program for smoothing by bin mean, median and boundaries for given data is executed successfully.

4.

AIM: To write the program for min-max and Z-score normalization using R-tool.

Data: Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

(i) Use min-max normalization to transform the value 35 for age onto the range [0.0, 1.0].
(ii) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years

PROGRAM:

```
v<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)

min<-0

max<-1

#min_max

min_max=((35-min(v))/(max(v)-min(v)))

print(min_max)

#z-score

m=mean(v)

s<-12.94

z_score=(35-m)/s

print(z_score)
```

OUTPUT:

```
print(min_max)
] 0.3157895
#z-score
m=mean(v)
s<-12.94
z_score=(35-m)/s
print(z_score)
] -0.8844238
```

RESULT: Thus, the program for min-max normalization and z-score normalization for given data is executed successfully.

5.

AIM: To write the program for below using R tool

a) Mean b) Median c) Standard deviation d) Box plot
e) Scatter plot e) q-q plot

Data:

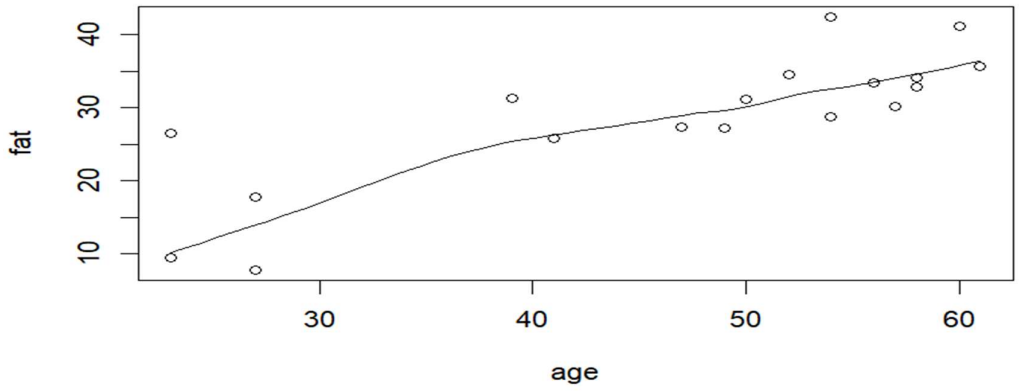
age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

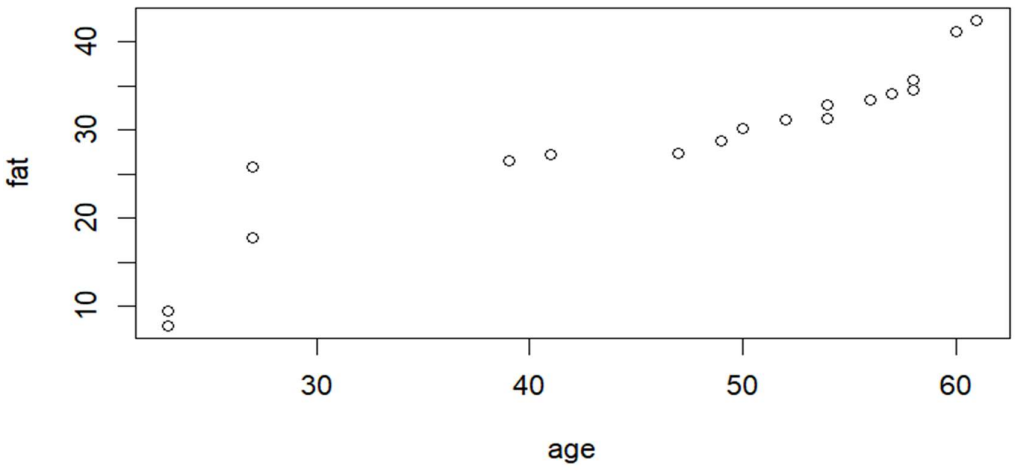
PROGRAM:

```
age<-c(23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61)

fat<-
c(9.5,26.5,7.8,17.8,31.4,25.9,27.4,27.2,31.2,34.6,42.5,28.8,33.4,30.2,34.1,32.9,
41.2,35.7)

mean(age)

median(age)

sd(age)

mean(fat)

median(fat)

sd(fat)

#boxplot

boxplot(age,fat)

#scatter plot

scatter.smooth(age,fat)

#qplot

qqplot(age,fat)
```

OUTPUT:

```
> mean(age)
[1] 46.44444
> median(age)
[1] 51
> sd(age)
[1] 13.21862
> mean(fat)
[1] 28.78333
> median(fat)
[1] 30.7
> sd(fat)
[1] 9.254395
> #boxplot
> boxplot(age,fat)
> #scatter plot
> scatter.smooth(age,fat)
> #qplot
> qqplot(age,fat)
```





RESULT: Thus, the program for mean, median, sd, boxplot and scatterplot for given data is executed successfully.

6

AIM: To write the program for plotting histogram for below using R tool

a) equal-frequency (equi-depth) partitioning b)equal-width partitioning

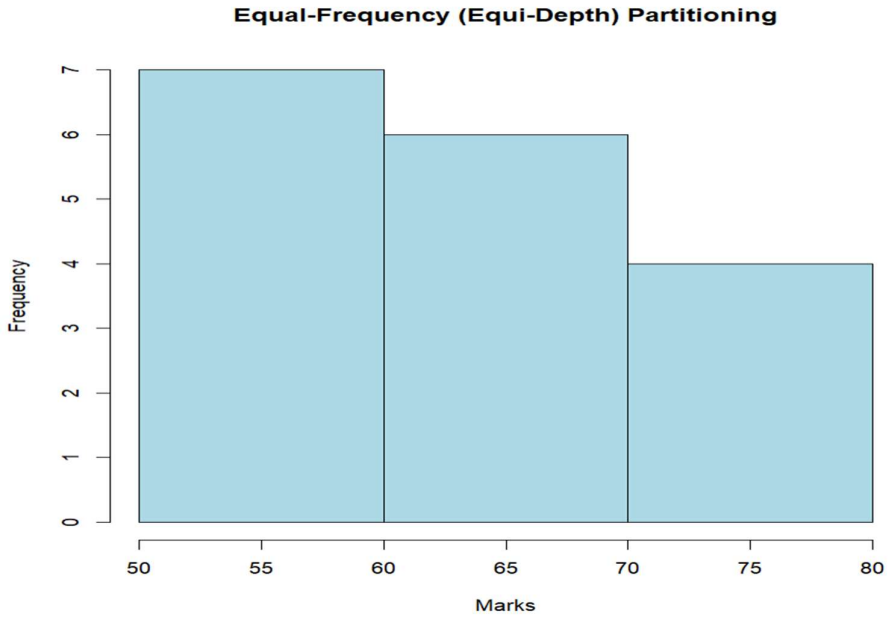
Data: Marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55,58,59,61,63,65,67,71,72,75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

PROGRAM:

```
marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)

num_bins <- 3

bins_eq_frequency <- cut(marks, breaks = num_bins, labels = FALSE)

hist(marks, breaks = num_bins, col = "lightblue", xlab = "Marks", main = "Equal-Frequency (Equi-Depth) Partitioning")

marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)

bin_mean <- tapply(data, cut(data, num_bins), mean)

smoothed_data_by_mean <- unname(bin_mean[as.character(cut(data, num_bins))])

bin_median <- tapply(data, cut(data, num_bins), median)

smoothed_data_by_median <- unname(bin_median[as.character(cut(data, num_bins))])

bin_boundaries <- tapply(data, cut(data, num_bins), function(x) c(min(x), max(x)))

smoothed_data_by_boundaries <- unlist(bin_boundaries[as.character(cut(data, num_bins))])

print("Original data:")

print(data)

print("Smoothed data by bin mean:")

print(smoothed_data_by_mean)

print("Smoothed data by bin median:")

print(smoothed_data_by_median)

print("Smoothed data by bin boundaries:")

print(smoothed_data_by_boundaries)
```

OUTPUT:

```
> print(smoothed_data_by_mean)
 [1] 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375
 [9] 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375 18.9375
[17] 43.7500 43.7500 43.7500 43.7500 72.7500 72.7500 72.7500 72.7500
> print("Smoothed data by bin median:")
[1] "Smoothed data by bin median:"
> print(smoothed_data_by_median)
 [1] 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
[14] 20.0 20.0 20.0 45.0 45.0 45.0 45.0 72.5 72.5 72.5 72.5
> print("Smoothed data by bin boundaries:")
[1] "Smoothed data by bin boundaries:"
> print(smoothed_data_by_boundaries)
(10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1
          11           30           11           30           11
(10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2
          30           11           30           11           30
(10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1
          11           30           11           30           11
(10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2 (10.9,32.3]1 (10.9,32.3]2
          30           11           30           11           30
```



RESULT:

Thus, the program for plotting histogram for equal-frequency (equi-depth) partitioning and equal-width partitioning for given data is executed successfully.

7.

AIM: To write the program for the first quartile (Q1) and the third quartile (Q3) of the data.

Data: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

PROGRAM:

```
#Q1, Q2

age<-
c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,
52,70)

quantile(age,.25)

quantile(age,.75)
```

OUTPUT:

```
> #Q1, Q2
> age<-c(13,15,16,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35,35,35,36,40,45,46,52,70)
> quantile(age,.25)
 25%
20.5
> quantile(age,.75)
75%
 35
```

RESULT:

Thus, the program for first quartile (Q1) and the third quartile (Q3) of the given data is executed successfully.

8.

AIM: To write the program for the Inter quantile and standard deviation of the given data.

PROGRAM:

#IQR, SD

v<-c(78.3,81.8,82,74.2,83.4,84.5,82.9,77.5,80.9,70.6)

IQR(v)

sd(v)

OUTPUT:

```
v<-c(78.3,81.8,82,74.2,83.4,84.5,82.9,77.5,80.9,70.6)
IQR(v)
.] 4.975
sd(v)
.] 4.445835
```

RESULT:

Thus, the program for the Inter quantile and standard deviation of the given data.is executed successfully.

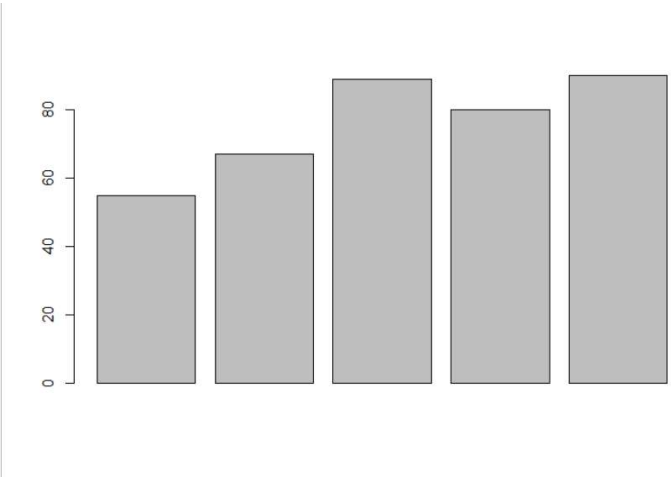
9.

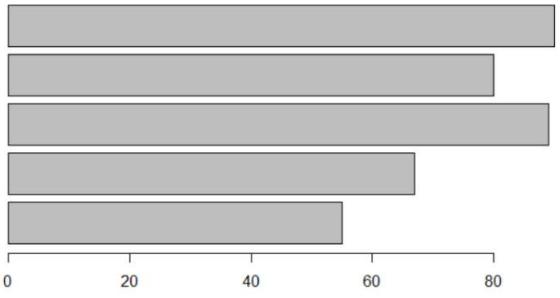
AIM: To draw the bar plot and horizontal bar using R-tool.

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.

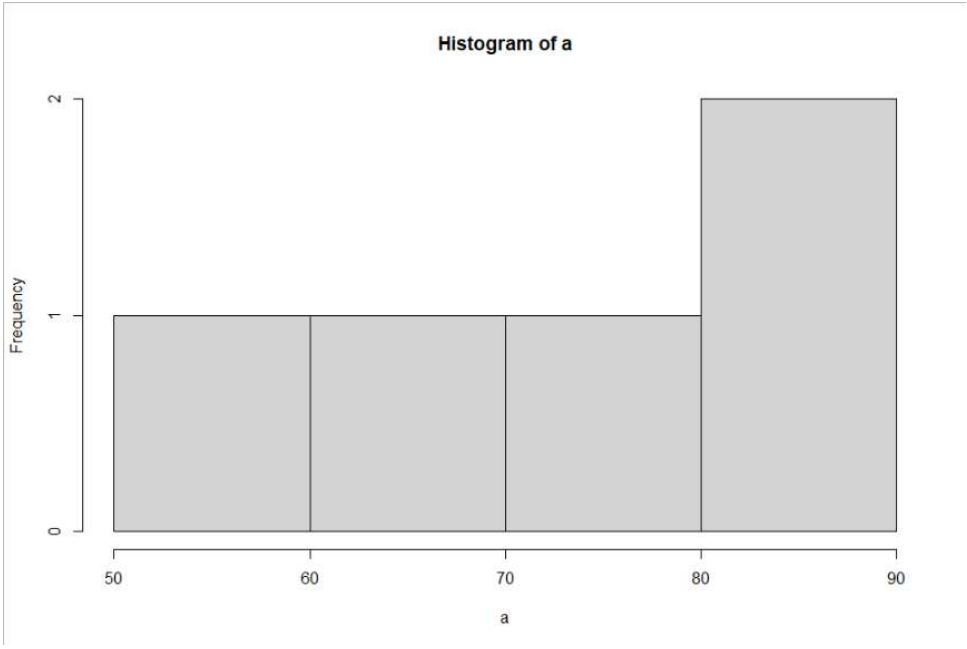
10.

AIM: To draw the histogram plot using R-tool.

PROGRAM:

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

OUTPUT:



RESULT:

Thus, the histogram plot was executed successfully.

CREATE OWN DATASET PROGRAMS:

11.

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

Dataset: [Create your own dataset in excel]

Age Insulin

20-29 Normal

20-29 High

30-39 Low

30-39 Normal

40-49 High

PROGRAM:

```
diabetest1 <- read_excel("C:/Users/harsh/OneDrive/Desktop/Book.xlsx")
diabetest1_table <- table(diabetest1$Age, diabetest1$Insulin)
print(diabetest1_table)t
chisq_result <- chisq.test(diabetest1_table)
print(chisq_result)
```

OUTPUT:

```
[1] "Chi-Square Test Result:"
> print(chisq_result)

        Pearson's Chi-squared test

data:  diabetest1_table
X-squared = 3.75, df = 4, p-value = 0.4409

>
> # Check expected frequencies
> print("Expected Frequencies:")
[1] "Expected Frequencies:"
> print(chisq_result$expected)

        High Low Normal
  20-29  0.8 0.4    0.8
  30-39  0.8 0.4    0.8
  40-49  0.4 0.2    0.4
```

RESULT:

Thus, the correlation analysis was executed successfully.

12. AIM:

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

Dataset: [Create your own dataset in excel]

x	BloodPressure	Glucose	Outcome
25	120	85	0
30	130	90	0
35	135	95	0
40	140	100	1
45	145	105	1
50	150	110	1
55	160	115	1
60	170	120	1
65	180	125	1
70	190	130	1

PROGRAM:

```
diabetes_data <- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/diabetes.xlsx")

colnames(diabetes_data) <- c("Age", "BloodPressure", "Glucose", "Outcome")

head(diabetes_data)

linear_model <- lm(Outcome ~ Age, data = diabetes_data)

summary(linear_model)
```

OUTPUT:

```
# A tibble: 6 × 4
    Age BloodPressure Glucose Outcome
  <db1>         <db1>   <db1>   <db1>
1    25           120      85       0
2    30           130      90       0
3    35           135      95       0
4    40           140     100       1
5    45           145     105       1
6    50           150     110       1
> # Linear regression: Predict Outcome based on Age
> linear_model <- lm(Outcome ~ Age, data = diabetes_data)
> # Display linear model summary
> summary(linear_model)

Call:
lm(formula = Outcome ~ Age, data = diabetes_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.38182 -0.22727 -0.07273  0.20455  0.49091

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.509091   0.337590  -1.508  0.16998
Age          0.025455   0.006803   3.742  0.00569 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.309 on 8 degrees of freedom
Multiple R-squared:  0.6364,    Adjusted R-squared:  0.5909
F-statistic:    14 on 1 and 8 DF,  p-value: 0.005692
```

RESULT:

Thus, the linear regression was executed successfully.

13.AIM: To write the program for the multiple regression using R -tool

Dataset: [Create your own dataset in excel]

x	BloodPressure	Glucose	Outcome
25	120	85	0
30	130	90	0
35	135	95	0
40	140	100	1
45	145	105	1
50	150	110	1
55	160	115	1
60	170	120	1
65	180	125	1
70	190	130	1

PROGRAM:

```
diabetes_data <- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/diabetes.xlsx")

colnames(diabetes_data) <- c("Age", "BloodPressure", "Glucose", "Outcome")

head(diabetes_data)

linear_model_multiple <- lm(Outcome ~ Age + BloodPressure + Glucose, data = diabetes_data)

summary(linear_model_multiple)
```

OUTPUT:

```
    Age BloodPressure Glucose Outcome
  <db7>       <db7>   <db7>   <db7>
1    25         120      85       0
2    30         130      90       0
3    35         135      95       0
4    40         140     100       1
5    45         145     105       1
6    50         150     110       1
>
> # Step 4: Fit the multiple regression model
> linear_model_multiple <- lm(Outcome ~ Age + BloodPressure + Glucose, data = diabetes_data)
>
> # Step 5: Display the summary of the multiple regression model
> summary(linear_model_multiple)

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

Residuals:
      Min        1Q    Median        3Q       Max
-0.28298 -0.08511 -0.02340  0.03237  0.43040

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.93313    1.77142   2.785  0.02711 *
Age           0.12389    0.03211   3.859  0.00622 **
BloodPressure -0.06657    0.02148  -3.099  0.01734 *
Glucose             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2144 on 7 degrees of freedom
Multiple R-squared:  0.8467,    Adjusted R-squared:  0.8029
F-statistic: 19.33 on 2 and 7 DF,  p-value: 0.00141
```

RESULT:

Thus, the multiple regression is executed successfully.

14. AIM: To write the program for creating own dataset for Box plot.

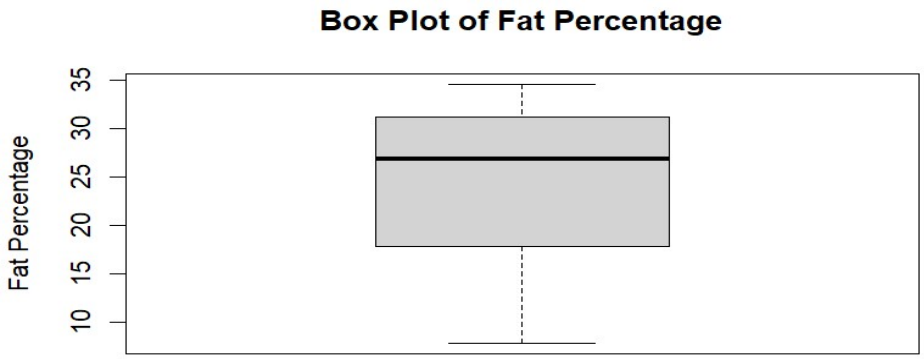
Dataset: [Create your own dataset in excel]

Age	Fat
23	9.5
23	26.5
27	7.8
27	17.8
39	31.4
41	25.9
47	27.4
49	27.2
50	31.2
52	34.6

PROGRAM:

```
library(readxl)
data<- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/box.xlsx")
str(data)
boxplot(data$Fat,
    main = "Box Plot of Fat Percentage",
    ylab = "Fat Percentage")
```

OUTPUT:



RESULT:

Thus write the program for creating own dataset for Box plot is executed successfully.

15. AIM: To write the program for creating own dataset for Histogram.

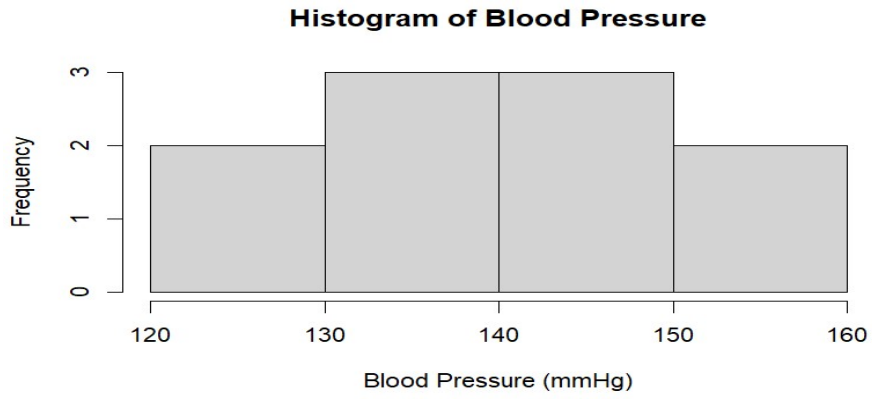
Dataset: [Create your own dataset in excel]

Age	BloodPressure
25	120
30	130
35	140
40	150
45	135
50	145
55	155
60	160
65	150
70	140

PROGRAM:

```
library(readxl)
data<- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/hist.xlsx")
head(data)
hist(data$BloodPressure,
    main = "Histogram of Blood Pressure",
    xlab = "Blood Pressure (mmHg
```

OUTPUT:



RESULT

Thus write the program for creating own dataset for Histogram is executed successfully.

16. AIM: To write the program for creating own dataset for Histogram.

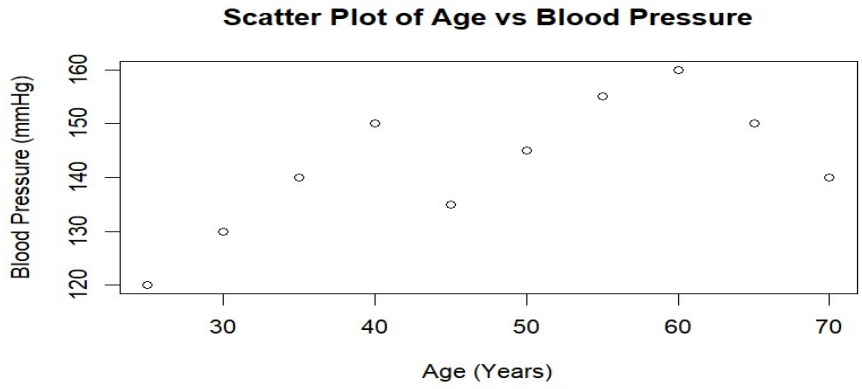
Dataset: : [Create your own dataset in excel]

Age	BloodPressure
25	120
30	130
35	140
40	150
45	135
50	145
55	155
60	160
65	150
70	140

PROGRAM:

```
library(readxl)

data<- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/hist.xlsx")

head(data)

plot(data$Age, data$BloodPressure,

    main = "Scatter Plot of Age vs Blood Pressure",

    xlab = "Age (Years)",

    ylab = "Blood Pressure (mmHg)")
```

OUTPUT:



RESULT:

Thus write the program for creating own dataset for scatter plot is executed successfully.

16. AIM:

To write the program for creating own dataset for Z-Score.

Dataset: [Create your own dataset in excel]

Value
50
60
70
80
90
100
110
120
130
140

PROGRAM:

```
library(readxl)

data<- read_excel("C:/Users/harsh/OneDrive/Desktop/Weka codes/zscore.xlsx")

head(data)

mean_data <- mean(data$Value)

sd_data <- sd(data$Value)

z_scores <- (data$Value - mean_data) / sd_data

z_scores
```

OUTPUT:

```
> head(data)
# A tibble: 6 × 1
  Value
  <dbl>
1    50
2    60
3    70
4    80
5    90
6   100
> # Calculate the mean and standard deviation
> mean_data <- mean(data$Value)
> sd_data <- sd(data$Value)
> # Calculate the Z-Score for each value in the dataset
> z_scores <- (data$Value - mean_data) / sd_data
> # Display the Z-Scores
> z_scores
 [1] -1.4863011 -1.1560120 -0.8257228 -0.4954337 -0.1651446  0.1651446  0.4954337  0.8257228
 [9]  1.1560120  1.4863011
>
```

RESULT:

Thus write the program for creating own dataset for Z-Score is executed successfully.