NAME : DEEBAK KUMAR K

REG NO : 192324064

SUBECT CODE : CSA1622

DW & DM LAB PROGRAMS

DW - DM LAB PROGRAMS

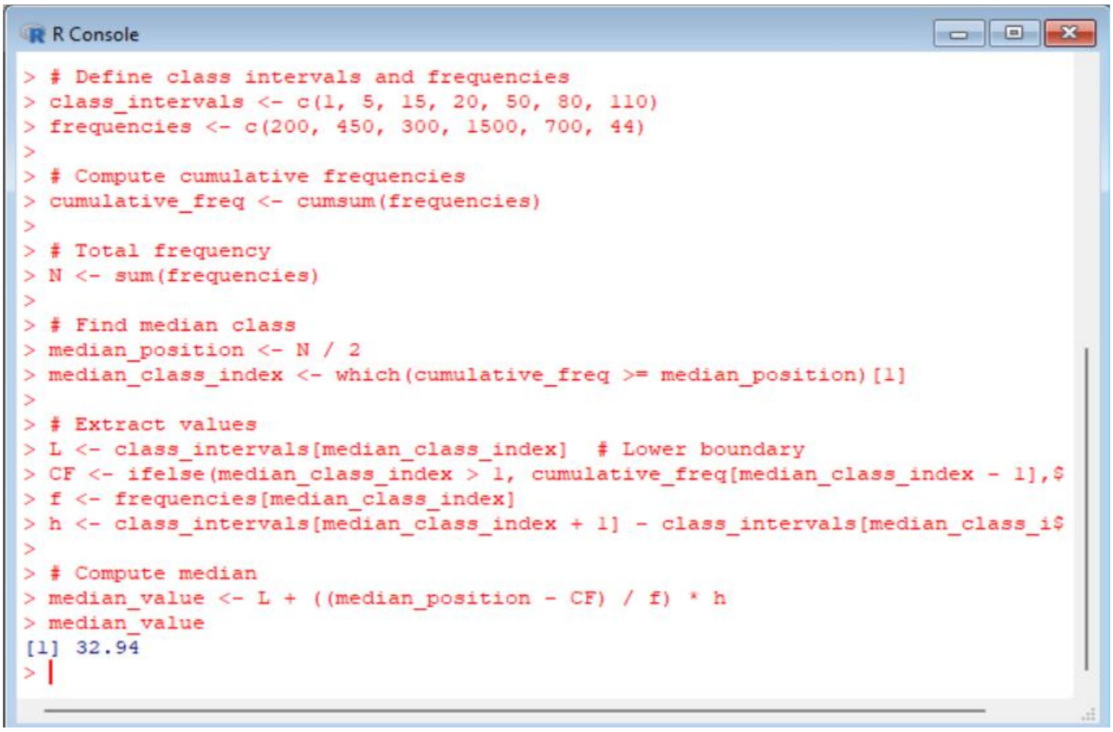
EXP NO : 1

The intervals and corresponding frequencies are as follows. age frequency
1-5. 200
5-15 450
15-20 300
20-50 1500
50-80 700
80-110 44
Compute an approximate median value for the data

CODE :

```
# Define class intervals and frequencies
class_intervals <- c(1, 5, 15, 20, 50, 80, 110)
frequencies <- c(200, 450, 300, 1500, 700, 44)

# Compute cumulative frequencies
cumulative_freq <- cumsum(frequencies)

# Total frequency
N <- sum(frequencies)

# Find median class
median_position <- N / 2
median_class_index <- which(cumulative_freq >= median_position)[1]

# Extract values
L <- class_intervals[median_class_index]  # Lower boundary
CF <- ifelse(median_class_index > 1, cumulative_freq[median_class_index - 1], 0)
f <- frequencies[median_class_index]
h <- class_intervals[median_class_index + 1] - class_intervals[median_class_index]

# Compute median
median_value <- L + ((median_position - CF) / f) * h
median_value
```

OUTPUT :



EXP NO : 2

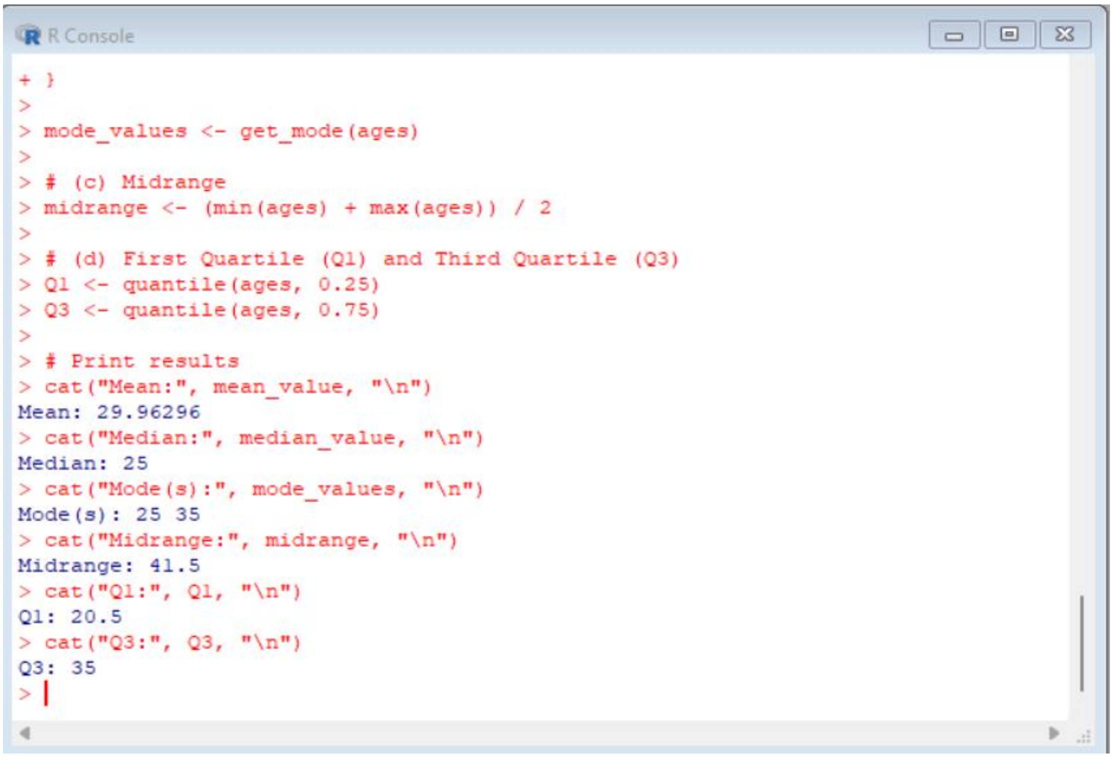
Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
(a) What is the mean of the data? What is the median?
(b) What is the mode of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).
(c) What is the midrange of the data?
(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

CODE:

```
# Given age data
ages <- 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)

# (a) Mean and Median
mean_value <- mean(ages)
median_value <- median(ages)

# (b) Mode and Modality
```

```
get_mode <- function(x) {
  ux <- unique(x)
  tab <- tabulate(match(x, ux))
  modes <- ux[tab == max(tab)]
  return(modes)
}

mode_values <- get_mode(ages)

# (c) Midrange
midrange <- (min(ages) + max(ages)) / 2

# (d) First Quartile (Q1) and Third Quartile (Q3)
Q1 <- quantile(ages, 0.25)
Q3 <- quantile(ages, 0.75)

# Print results
cat("Mean:", mean_value, "\n")
cat("Median:", median_value, "\n")
cat("Mode(s):", mode_values, "\n")
cat("Midrange:", midrange, "\n")
cat("Q1:", Q1, "\n")
cat("Q3:", Q3, "\n")
```

OUTPUT :



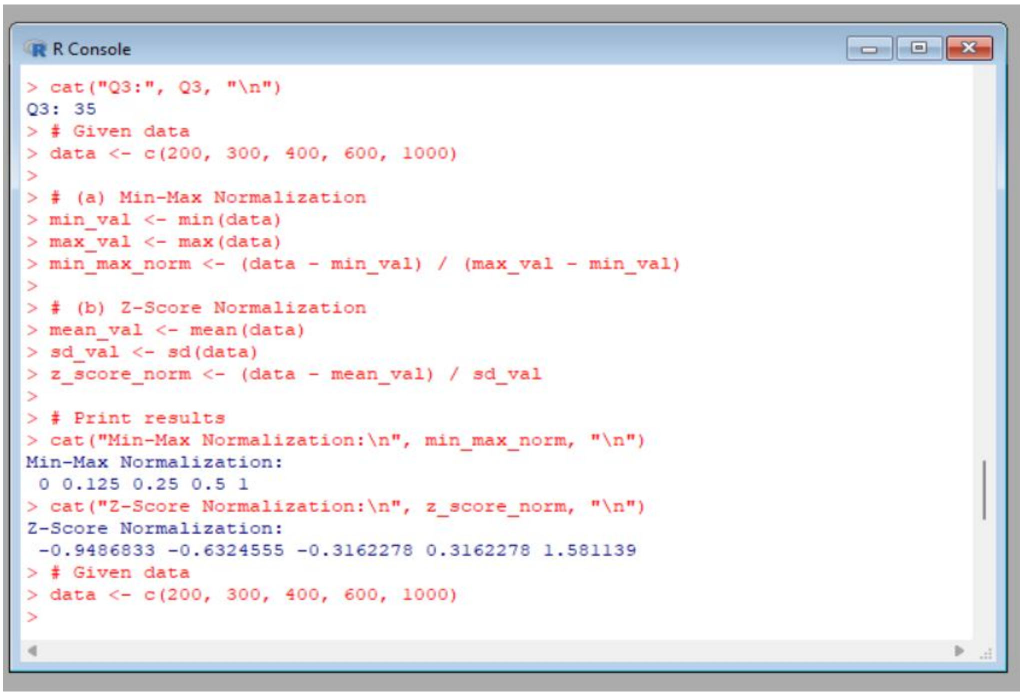
EXP NO : 3

Data Preprocessing : Reduction and Transformation
Use the two methods below to normalize the following group of data:
200, 300, 400, 600, 1000
(a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization

CODE:

```
# Given data
data <- c(200, 300, 400, 600, 1000)

# (a) Min-Max Normalization
min_val <- min(data)
max_val <- max(data)
min_max_norm <- (data - min_val) / (max_val - min_val)

# (b) Z-Score Normalization
mean_val <- mean(data)
sd_val <- sd(data)
z_score_norm <- (data - mean_val) / sd_val

# Print results
cat("Min-Max Normalization:\n", min_max_norm, "\n")
cat("Z-Score Normalization:\n", z_score_norm, "\n")
```

OUTPUT :



EXP NO : 4

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

a) Smoothing by bin mean

b) Smoothing by bin median

c) Smoothing by bin boundaries

CODE :

```
# Given data
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)

# Number of bins (choose k as needed, e.g., 4 bins)
k <- 4  # You can adjust the number of bins

# 1. Divide data into bins
bin_size <- ceiling(length(data) / k)
bins <- split(data, ceiling(seq_along(data) / bin_size))

# 2a. Smoothing by Bin Mean
bin_means <- lapply(bins, mean)
smoothed_mean <- unlist(lapply(seq_along(bins), function(i) rep(bin_means[[i]],
length(bins[[i]]))))
print("Smoothing by Bin Mean:")
print(smoothed_mean)

# 2b. Smoothing by Bin Median
bin_medians <- lapply(bins, median)
smoothed_median <- unlist(lapply(seq_along(bins), function(i) rep(bin_medians[[i]],
length(bins[[i]]))))
print("Smoothing by Bin Median:")
print(smoothed_median)

# 2c. Smoothing by Bin Boundaries
smoothed_boundary <- unlist(lapply(bins, function(bin) {
  min_val <- min(bin)
  max_val <- max(bin)
  sapply(bin, function(x) ifelse(abs(x - min_val) < abs(x - max_val), min_val, max_val))
}))
print("Smoothing by Bin Boundaries:")
print(smoothed_boundary)
```

OUTPUT :

R Console

```
> k <- 4  # You can adjust the number of bins
>
> # 1. Divide data into bins
> bin_size <- ceiling(length(data) / k)
> bins <- split(data, ceiling(seq_along(data) / bin_size))
>
> # 2a. Smoothing by Bin Mean
> bin_means <- lapply(bins, mean)
> smoothed_mean <- unlist(lapply(seq_along(bins), function(i) rep(bin_means[[i]], length(bins[[i]]))))
> print("Smoothing by Bin Mean:")
[1] "Smoothing by Bin Mean:"
> print(smoothed_mean)
 [1] 13.83333 13.83333 13.83333 13.83333 13.83333 13.83333 20.16667 20.16667
 [9] 20.16667 20.16667 20.16667 20.16667 30.66667 30.66667 30.66667 30.66667
[17] 30.66667 30.66667 63.50000 63.50000 63.50000 63.50000 63.50000 63.50000
>
> # 2b. Smoothing by Bin Median
> bin_medians <- lapply(bins, median)
> smoothed_median <- unlist(lapply(seq_along(bins), function(i) rep(bin_medians[[i]], length(bins[[i]]))))
> print("Smoothing by Bin Median:")
[1] "Smoothing by Bin Median:"
> print(smoothed_median)
 [1] 14.0 14.0 14.0 14.0 14.0 14.0 20.0 20.0 20.0 20.0 20.0 20.0 27.0 27.0 27.0
[16] 27.0 27.0 27.0 71.5 71.5 71.5 71.5 71.5 71.5
>
> # 2c. Smoothing by Bin Boundaries
> smoothed_boundary <- unlist(lapply(bins, function(bin) {
+   min_val <- min(bin)
+   max_val <- max(bin)
+   sapply(bin, function(x) ifelse(abs(x - min_val) < abs(x - max_val), min_val, max_val))
+ }))
> print("Smoothing by Bin Boundaries:")
[1] "Smoothing by Bin Boundaries:"
> print(smoothed_boundary)
11 12 13 14 15 16 21 22 23 24 25 26 31 32 33 34 35 36 41 42 43 44 45 46
11 11 11 16 16 16 19 21 21 21 21 21 22 22 22 22 45 45 45 45 75 75 75 75
>
```

EXP NO : 5

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

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

(a) Calculate the mean, median, and standard deviation of age and %fat.
(b) Draw the boxplots for age and %fat.
(c) Draw a scatter plot and a q-q plot based on these two variables.

CODE :

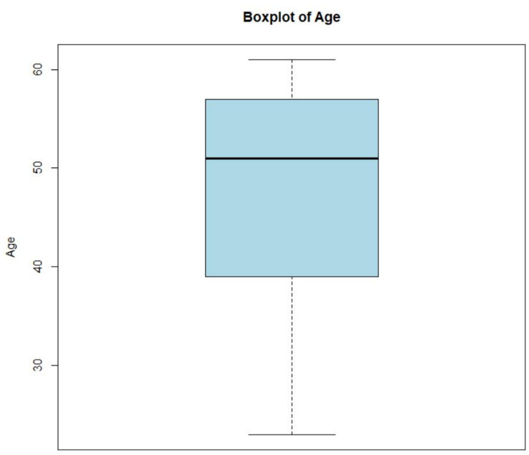
```
# Given data
age <- c(23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61)
fat_percent <- c(9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2,
            34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7)

# (a) Calculate Mean, Median, and Standard Deviation
mean_age <- mean(age)
median_age <- median(age)
sd_age <- sd(age)

mean_fat <- mean(fat_percent)
median_fat <- median(fat_percent)
sd_fat <- sd(fat_percent)

# Print results
cat("Age - Mean:", mean_age, "Median:", median_age, "Standard Deviation:",
sd_age, "\n")
cat("%Fat - Mean:", mean_fat, "Median:", median_fat, "Standard Deviation:", sd_fat,
"\n")

# (b) Boxplot for Age and %Fat
par(mfrow = c(1,2))  # Split plot into two columns
boxplot(age, main="Boxplot of Age", col="lightblue", ylab="Age")
boxplot(fat_percent, main="Boxplot of %Fat", col="lightgreen", ylab="% Body Fat")

# (c) Scatter Plot
plot(age, fat_percent, main="Scatter Plot of Age vs %Fat",
   xlab="Age", ylab="% Body Fat", col="blue", pch=16)

# (c) Q-Q Plot for Normality Check
par(mfrow = c(1,2))  # Split into two plots
```

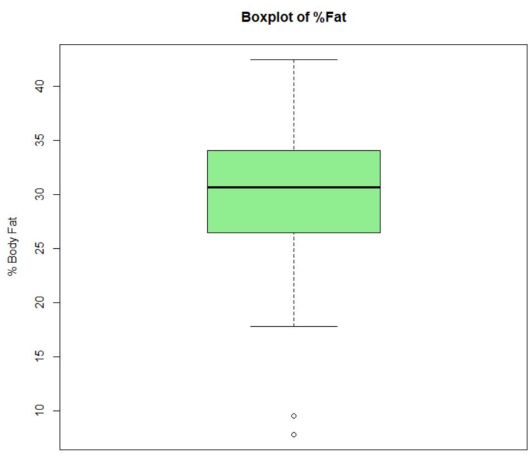
qqnorm(age, main="Q-Q Plot for Age")
qqline(age, col="red")

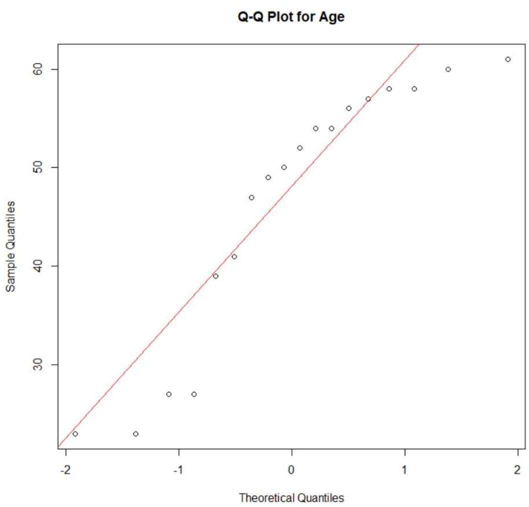
qqnorm(fat_percent, main="Q-Q Plot for %Fat")
qqline(fat_percent, col="red")

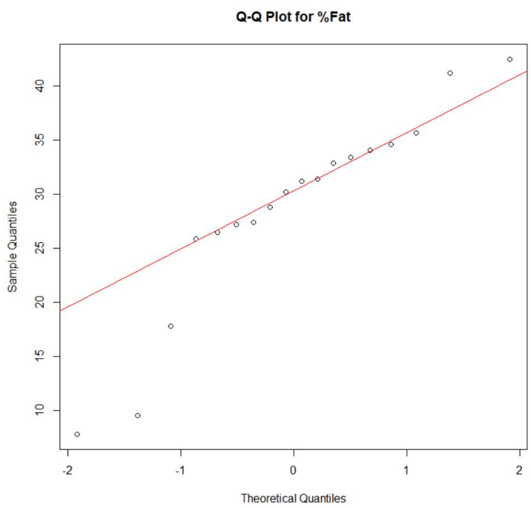
OUTPUT :

```
> # Print results
> cat("Age - Mean:", mean_age, "Median:", median_age, "Standard Deviation:", sd_age, "\n")
Age - Mean: 46.44444 Median: 51 Standard Deviation: 13.21862
> cat("%Fat - Mean:", mean_fat, "Median:", median_fat, "Standard Deviation:", sd_fat, "\n$
%Fat - Mean: 28.78333 Median: 30.7 Standard Deviation: 9.254395
>
```









EXP NO : 6

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.
(iii) Use normalization by decimal scaling to transform the value 35 for age. Perform the above functions using R – tool

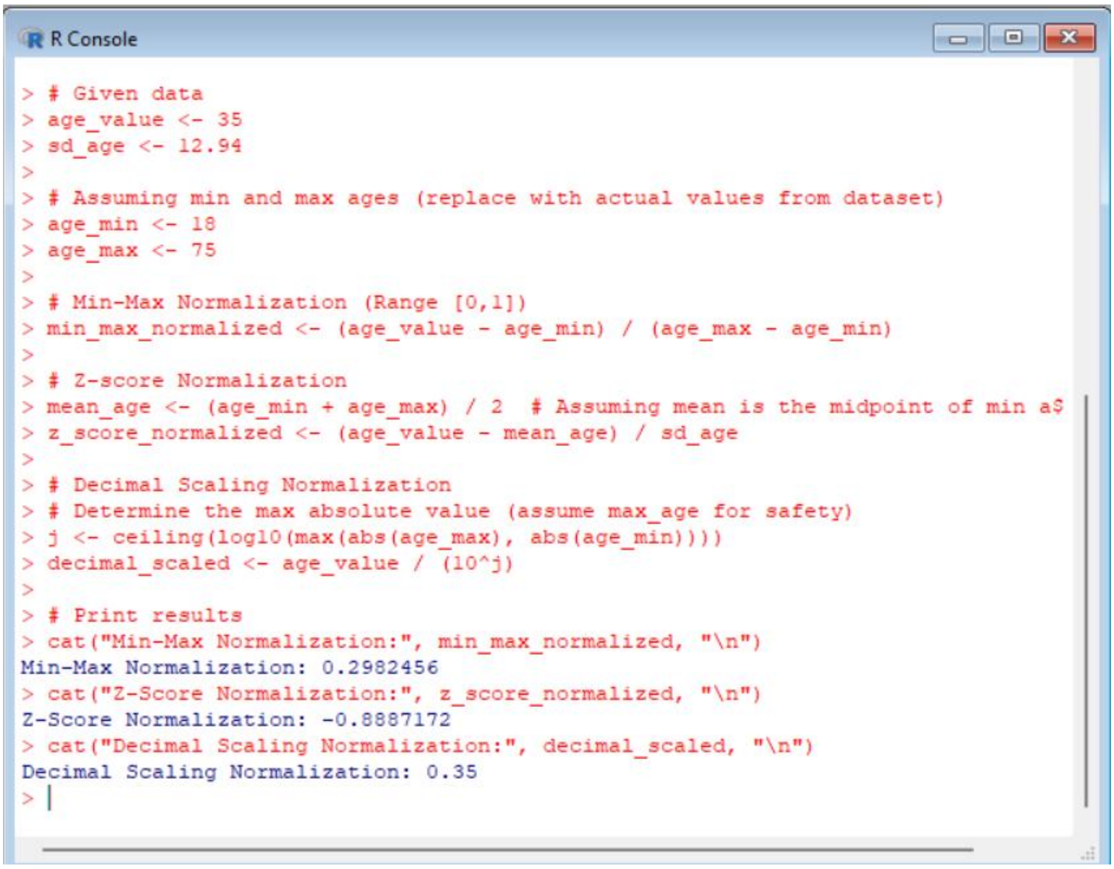
CODE :

```
# Given data

age_value <- 35

sd_age <- 12.94

# Assuming min and max ages (replace with actual values from dataset)

age_min <- 18

age_max <- 75

# Min-Max Normalization (Range [0,1])

min_max_normalized <- (age_value - age_min) / (age_max - age_min)

# Z-score Normalization

mean_age <- (age_min + age_max) / 2  # Assuming mean is the midpoint of min and max

z_score_normalized <- (age_value - mean_age) / sd_age

# Decimal Scaling Normalization

# Determine the max absolute value (assume max_age for safety)

j <- ceiling(log10(max(abs(age_max), abs(age_min))))

decimal_scaled <- age_value / (10^j)

# Print results

cat("Min-Max Normalization:", min_max_normalized, "\n")
```

cat("Z-Score Normalization:", z_score_normalized, "\n")

cat("Decimal Scaling Normalization:", decimal_scaled, "\n")

OUTPUT:



```
R Console
> # Given data
> age_value <- 35
> sd_age <- 12.94
>
> # Assuming min and max ages (replace with actual values from dataset)
> age_min <- 18
> age_max <- 75
>
> # Min-Max Normalization (Range [0,1])
> min_max_normalized <- (age_value - age_min) / (age_max - age_min)
>
> # Z-score Normalization
> mean_age <- (age_min + age_max) / 2  # Assuming mean is the midpoint of min a$
> z_score_normalized <- (age_value - mean_age) / sd_age
>
> # Decimal Scaling Normalization
> # Determine the max absolute value (assume max_age for safety)
> j <- ceiling(log10(max(abs(age_max), abs(age_min))))
> decimal_scaled <- age_value / (10^j)
>
> # Print results
> cat("Min-Max Normalization:", min_max_normalized, "\n")
Min-Max Normalization: 0.2982456
> cat("Z-Score Normalization:", z_score_normalized, "\n")
Z-Score Normalization: -0.8887172
> cat("Decimal Scaling Normalization:", decimal_scaled, "\n")
Decimal Scaling Normalization: 0.35
>
```

EXP NO : 7

The following values are the number of pencils available in the different boxes. Create a vector and find out the mean, median and mode values of set of pencils in the given data.

Box1 Box2 Box3 Box4 Box5 Box6 Box7 Box8 Box9 Box 10

9 25 23 12 11 6 7 8 9 10

CODE :

```
# Creating a vector with the number of pencils in each box

pencils <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)

# Calculating Mean

mean_pencils <- mean(pencils)

# Calculating Median

median_pencils <- median(pencils)

# Function to calculate Mode

get_mode <- function(v) {

  uniq_vals <- unique(v)

  freq <- tabulate(match(v, uniq_vals))

  mode_val <- uniq_vals[freq == max(freq)]

  return(mode_val)

}

# Calculating Mode

mode_pencils <- get_mode(pencils)

# Print Results

cat("Mean:", mean_pencils, "\n")

cat("Median:", median_pencils, "\n")

cat("Mode:", mode_pencils, "\n")
```

OUTPUT :

```
R Console

> # Creating a vector with the number of pencils in each box
> pencils <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)
>
> # Calculating Mean
> mean_pencils <- mean(pencils)
>
> # Calculating Median
> median_pencils <- median(pencils)
>
> # Function to calculate Mode
> get_mode <- function(v) {
+   uniq_vals <- unique(v)
+   freq <- tabulate(match(v, uniq_vals))
+   mode_val <- uniq_vals[freq == max(freq)]
+   return(mode_val)
+ }
>
> # Calculating Mode
> mode_pencils <- get_mode(pencils)
>
> # Print Results
> cat("Mean:", mean_pencils, "\n")
Mean: 12
> cat("Median:", median_pencils, "\n")
Median: 9.5
> cat("Mode:", mode_pencils, "\n")
Mode: 9
>
```

EXP NO : 8

The following table would be plotted as (x,y) points, with the first column being the x values as number of mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones sold.

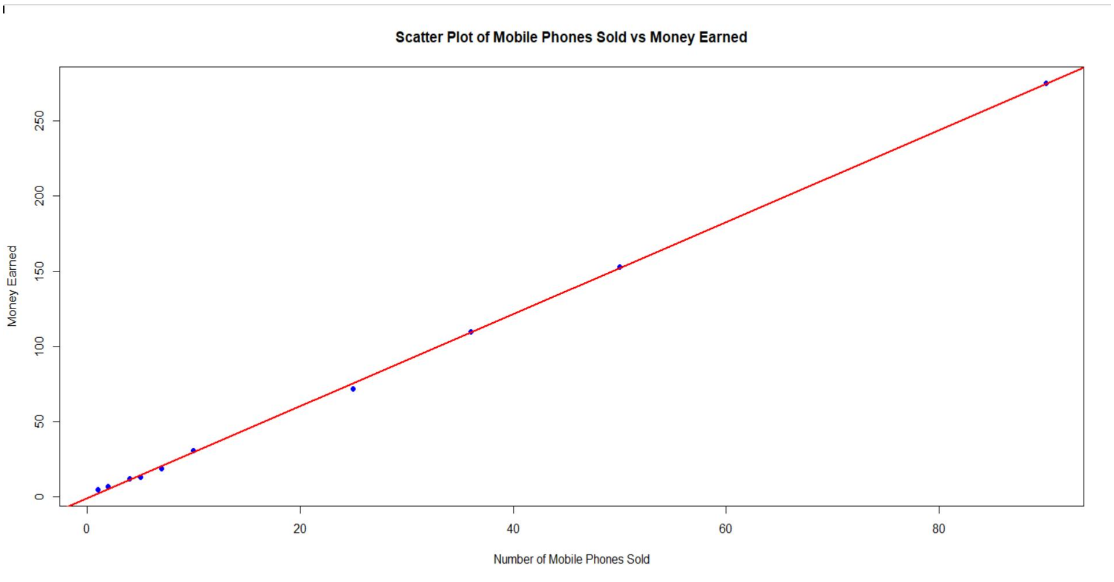
x :4 1 5 7 10 2 50 25 90 36

y :12 5 13 19 31 7 153 72 275 110

CODE:

```
# Given data

x <- c(4, 1, 5, 7, 10, 2, 50, 25, 90, 36)  # Number of mobile phones sold

y <- c(12, 5, 13, 19, 31, 7, 153, 72, 275, 110)  # Money earned

# Create a scatter plot

plot(x, y, main = "Scatter Plot of Mobile Phones Sold vs Money Earned",

    xlab = "Number of Mobile Phones Sold", ylab = "Money Earned",

    col = "blue", pch = 16)

# Add a trend line

abline(lm(y ~ x), col = "red", lwd = 2)
```

OUTPUT :



EXP NO : 9

Implement of the R script using 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.

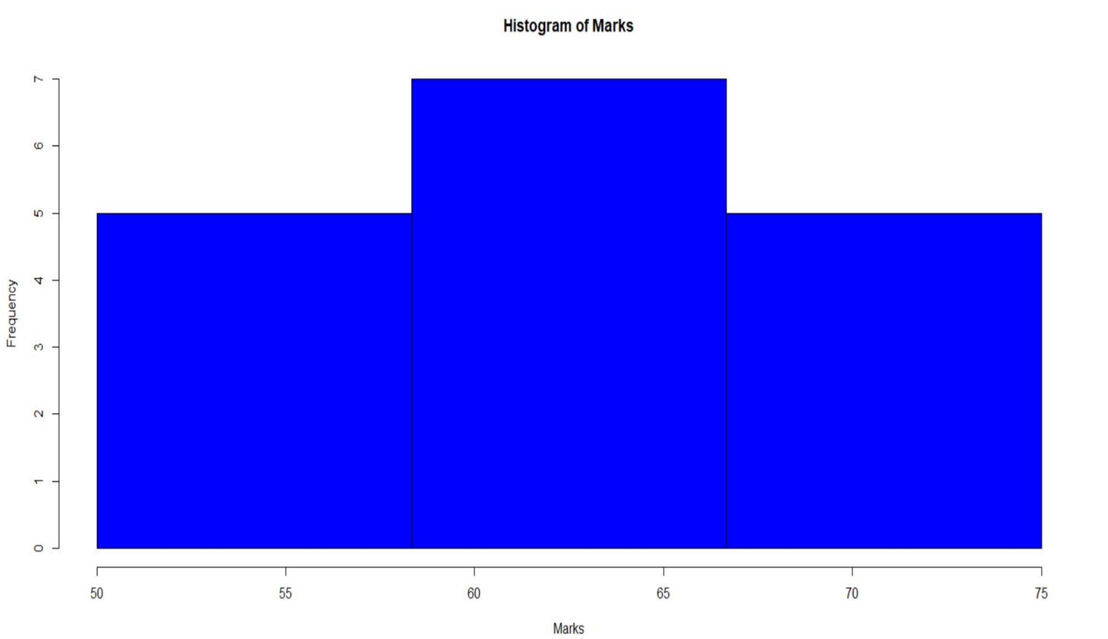
(a) equal-frequency (equi-depth) partitioning (b) equal-width partitioning

CODE :

```
# Given marks

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



# Sort the marks

marks <- sort(marks)



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

num_bins <- 3

bin_size <- length(marks) / num_bins  # Items per bin



# Splitting data into bins (equal frequency)

equal_freq_bins <- split(marks, ceiling(seq_along(marks) / bin_size))



# Print the bins

cat("Equal-Frequency Bins:\n")

print(equal_freq_bins)



# (b) Equal-Width Partitioning

min_mark <- min(marks)
```

```
max_mark <- max(marks)


# Compute bin width

bin_width <- (max_mark - min_mark) / num_bins


# Assign bins based on equal width

equal_width_bins <- cut(marks, breaks=seq(min_mark, max_mark, by=bin_width),
include.lowest=TRUE, right=TRUE)


# Print the bins

cat("Equal-Width Bins:\n")

print(table(equal_width_bins))

# Plot histogram

hist(marks, breaks=seq(min_mark, max_mark, by=bin_width),

    main="Histogram of Marks", xlab="Marks", col="blue", border="black")
```

OUTPUT :



EXP NO : 10

Suppose that the data for analysis includes the attribute age. 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.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

CODE :

```
# Given data

ages <- 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)



# Compute Q1 and Q3

Q1 <- quantile(ages, 0.25)

Q3 <- quantile(ages, 0.75)



# Print results

cat("First Quartile (Q1):", Q1, "\n")

cat("Third Quartile (Q3):", Q3, "\n")
```

OUTPUT :

```
> # Given data
> ages <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 3$
>
> # Compute Q1 and Q3
> Q1 <- quantile(ages, 0.25)
> Q3 <- quantile(ages, 0.75)
>
> # Print results
> cat("First Quartile (Q1):", Q1, "\n")
First Quartile (Q1): 20.5
> cat("Third Quartile (Q3):", Q3, "\n")
Third Quartile (Q3): 35
> |
```

EXP NO : 11

Suppose that the speed car is mentioned in different driving style.

Regular 78.3 81.8 82 74.2 83.4 84.5 82.9 77.5 80.9 70.6 Speed

Calculate the Inter quantile and standard deviation of the given data.

CODE :

```
# Given data (Speed of the car in Regular driving style)
speed <- c(78.3, 81.8, 82, 74.2, 83.4, 84.5, 82.9, 77.5, 80.9, 70.6)

# Calculate Quartiles
Q1 <- quantile(speed, 0.25)  # First quartile (25th percentile)
Q3 <- quantile(speed, 0.75)  # Third quartile (75th percentile)

# Calculate Interquartile Range (IQR)
IQR_value <- Q3 - Q1

# Calculate Standard Deviation
SD_value <- sd(speed)

# Print results
cat("First Quartile (Q1):", Q1, "\n")
cat("Third Quartile (Q3):", Q3, "\n")
cat("Interquartile Range (IQR):", IQR_value, "\n")
cat("Standard Deviation (SD):", SD_value, "\n")
```

Output :

```
> # Print results
> cat("First Quartile (Q1):", Q1, "\n")
First Quartile (Q1): 77.7
> cat("Third Quartile (Q3):", Q3, "\n")
Third Quartile (Q3): 82.675
> cat("Interquartile Range (IQR):", IQR_value, "\n")
Interquartile Range (IQR): 4.975
> cat("Standard Deviation (SD):", SD_value, "\n")
Standard Deviation (SD): 4.445835
>
```