1. 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

   PROGRAM

```
# Implementing equal-frequency partitioning in R
library(dplyr)
sales_price <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)
records_per_group <- length(sales_price) / 3
bins <- cut(sales_price, breaks = quantile(sales_price, probs = seq(0, 1, by =
1/records_per_group)))
table(bins)

# Implementing equal-width partitioning in R
library(dplyr)
sales_price <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)
range_of_data <- range(sales_price)
width_per_group <- (range_of_data[2] - range_of_data[1]) / 3
bins <- cut(sales_price, breaks = seq(range_of_data[1], range_of_data[2], by =
width_per_group))
table(bins)

# Implementing clustering in R
library(stats)
sales_price <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)
sales_price_matrix <- matrix(sales_price, ncol = 1)
cluster_result <- kmeans(sales_price_matrix, centers = 3)
table(cluster_result$cluster)
```

   2)Use following group of data: 200, 300, 400, 600, 1000 (a) min-max normalization by setting min = 0 and max = 1 (b) (b) z-score normalization (c) (c) z-score normalization using the mean absolute deviation instead of standard deviation (d) normalization by decimal scaling R program

   PROGRAM

```
# Given data

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


# Min-Max normalization

min_max_norm <- (data - min(data)) / (max(data) - min(data))


# Z-score normalization
```

```
z_score_norm <- (data - mean(data)) / sd(data)


# Z-score normalization using mean absolute deviation

mad <- mean(abs(data - mean(data)))

z_score_norm_mad <- (data - mean(data)) / mad


# Decimal scaling normalization

dec_scale_norm <- data / 1000


# Output the results

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

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

cat("Z-Score Normalization using MAD: ", z_score_norm_mad, "\n")

cat("Decimal Scaling Normalization: ", dec_scale_norm, "\n")
```

3) Implement using R language in which age group of people are affected byblood pressure based on the diabetes dataset show it using scatterplot and bar chart (that is BloodPressure vs Age using dataset “diabetes.csv”)

PROGRAM

```
# Load the diabetes dataset

diabetes <- read.csv("diabetes.csv")


# Create a scatter plot of BloodPressure vs Age

plot(diabetes$Age, diabetes$BloodPressure,

xlab = "Age", ylab = "Blood Pressure",

main = "Blood Pressure vs Age Scatterplot")
```

```
# Create a bar chart to show the distribution of age groups affected by high blood pressure

age_groups <- cut(diabetes$Age, breaks = seq(20, 80, by = 10), right = FALSE)

high_bp_counts <- table(age_groups, diabetes$BloodPressure >= 140)

barplot(high_bp_counts, beside = TRUE,

col = c("red", "green"),

xlab = "Age Group", ylab = "Count",

main = "Age Groups affected by High Blood Pressure")

legend("topright", legend = c("Normal BP", "High BP"), fill = c("green", "red"))
```

4) Analysis the dataset “diabetes. csv” how the diabetes trend is for different age people, using linear regression and multiple regression.

PROGRAM

```
# Load the diabetes dataset

diabetes <- read.csv("diabetes.csv")


# Simple linear regression

lm_age_diabetes <- lm(diabetes$DiabetesPedigreeFunction ~ diabetes$Age)

summary(lm_age_diabetes)


# Multiple regression

lm_age_bmi_pedigree <- lm(DiabetesPedigreeFunction ~ Age + BMI + Glucose, data =
diabetes)

summary(lm_age_bmi_pedigree)
```

5) R PROGRAM 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?

PROGRAM

```
# Define the age values
```

```
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)


# Find the Q1 and Q3 values

q1 <- quantile(age, 0.25)

q3 <- quantile(age, 0.75)


# Print the results

cat("Q1:", q1, "\n")

cat("Q3:", q3, "\n")
```

6) Download the Dataset "water" From R dataset Link.Find out whether there is a linear relation between attributes"mortality" and"hardness" by plot function.Fit the Data into the Linear Regression model. Predict the mortality for the hardness=88.

PROGRAM

```
# Load the water dataset

data(water)


# Plot mortality vs. hardness

plot(water$hardness, water$mortality, xlab = "Hardness", ylab = "Mortality")


# Fit linear regression model

lm_water <- lm(mortality ~ hardness, data = water)

summary(lm_water)


# Predict mortality for hardness = 88

hardness_new <- data.frame(hardness = 88)

mortality_pred <- predict(lm_water, newdata = hardness_new)
```

```
cat("Predicted mortality for hardness = 88:", mortality_pred)
```

7)R PROGRAM Imagine that you have selected data from the All Electronics data warehouse for analysis. The data set will be huge! The following data are a list of All Electronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30. (i) Partition the dataset using an equal-frequency partitioning method with bin equal to 3 (ii) apply data smoothing using bin means and bin boundary. (iii) Plot Histogram for the above frequency division

PROGRAM

```
# Define the data

prices <- c(1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30)



# Calculate the equal-frequency partitioning

partitions <- cut(prices, breaks = 3, labels = FALSE)



# Calculate the bin means and bin boundaries

bin_means <- tapply(prices, partitions, mean)

bin_boundaries <- tapply(prices, partitions, function(x) c(min(x), max(x)))



# Apply data smoothing using bin means and bin boundaries

smoothed_prices <- bin_means[partitions]

for (i in seq_along(partitions)) {

if (prices[i] < bin_boundaries[1, partitions[i]]) {

smoothed_prices[i] <- bin_means[partitions[i]]

} else if (prices[i] > bin_boundaries[2, partitions[i]]) {

smoothed_prices[i] <- bin_means[partitions[i]]
```

```
}

}


# Plot a histogram of the frequency division

hist(prices, breaks = 3, main = "Histogram of Prices")
```

8)R PROGRAM Two Maths teachers are comparing how their Year 9 classes performed in the end of year exams. Their results are as follows: Class A: 76, 35, 47, 64, 95, 66, 89, 36, 8476,35,47,64,95,66,89,36,84 Class B: 51, 56, 84, 60, 59, 70, 63, 66, 5051,56,84,60,59,70,63,66,50 (i) Find which class had scored higher mean, median and range. (ii) Plot above in boxplot and give the inferences

PROGRAM

```
# Define the data for the two classes

class_A <- c(76, 35, 47, 64, 95, 66, 89, 36, 84)

class_B <- c(51, 56, 84, 60, 59, 70, 63, 66, 50)


# Calculate the mean, median, and range for each class

mean_A <- mean(class_A)

mean_B <- mean(class_B)


median_A <- median(class_A)

median_B <- median(class_B)


range_A <- range(class_A)[2] - range(class_A)[1]

range_B <- range(class_B)[2] - range(class_B)[1]


# Print the results

cat("Class A mean:", mean_A, " median:", median_A, " range:", range_A, "\n")
```

```
cat("Class B mean:", mean_B, " median:", median_B, " range:", range_B, "\n")


# Plot the boxplot

boxplot(class_A, class_B, names = c("Class A", "Class B"), col = c("blue", "red"), ylab =
"Scores")


# Add a legend

legend("topleft", legend = c("Class A", "Class B"), fill = c("blue", "red"))
```

9) R 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. Perform the above functions using R – tool

PROGRAM

```
# Define the data

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

fat_pct <- 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)


# Calculate mean, median, and standard deviation of age and %fat

cat("Age: Mean =", mean(age), "Median =", median(age), "Standard Deviation =", sd(age),
"\n")

cat("%fat: Mean =", mean(fat_pct), "Median =", median(fat_pct), "Standard Deviation =",
sd(fat_pct), "\n")


# Draw boxplots for age and %fat
```

boxplot(age, main="Age Boxplot")

boxplot(fat_pct, main="%fat Boxplot")

Draw a scatter plot and a q-q plot based on these two variables

plot(age, fat_pct, main="Scatterplot of Age vs. %fat", xlab="Age", ylab="%fat")

qqplot(age, fat_pct, main="Q-Q Plot of Age vs. %fat")

10) 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

(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

PROGRAM

Define function for min-max normalization

minmax_norm <- function(x) {

(x - min(x)) / (max(x) - min(x))

}

Apply min-max normalization to age

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

```
minmax_norm(age)

# Define function for z-score normalization

zscore_norm <- function(x) {

(x - mean(x)) / sd(x)

}


# Apply z-score normalization to age

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

zscore_norm(age)

# Define function for decimal scaling normalization

decimal_norm <- function(x) {

x / 10^ceiling(log10(max(abs(x))))

}

# Apply decimal scaling normalization to age

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

decimal_norm(age)
```

11) 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.R PROGRAMING

Box1 Box2 Box3 Box4 Box5 Box6 Box7 Box8 Box9

25 23 12 11 6 7 8 9 10

PROGRAM

```
 # create a vector of pencil counts

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


# calculate mean
```

```
mean_pencils <- mean(pencils)

print(mean_pencils)


# calculate median

median_pencils <- median(pencils)

print(median_pencils)


# calculate mode

mode_pencils <- names(table(pencils))[table(pencils) == max(table(pencils))]

print(mode_pencils)
```

12) R PROGRAM Assume the Tennis coach wants to determine if any of his team players are scoring

outliers. To visualize the distribution of points scored by his players, then how can he

decide to develop the box plot? Give suitable example using Boxplot visualization

technique.

PROGRAM

```
# create a vector of player scores

scores <- c(25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110)


# create a box plot

boxplot(scores, main = "Player Scores", ylab = "Points", boxwex = 0.5)
```

13)R PROGRAM The following list of persons with vegetarian or not details given in the table. How will you find out how many of them are vegetarian and how many of them are non-vegetarian? Which type of the person total count is greater value?

Person	Gopu	Babu	Baby	Gopal	Krishna	Jai	Dev	Malini	Hema	Anu
Vegetarian	yes	yes	yes	no	yes	no	no	yes	yes	yes

PROGRAM

```
# Create a vector with the Vegetarian column

vegetarian <- c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes")


# Use the table() function to count the number of "yes" and "no" values in the vector

table(vegetarian)


# Create a matrix with the Vegetarian column

person_matrix <- matrix(c("yes", "yes", "yes", "no", "yes", "no", "no", "yes", "yes", "yes"),
nrow = 1)


# Use rowSums() to calculate the sum of "yes" values in each row

row_sums <- rowSums(person_matrix == "yes")


# Get the total count of rows

total_rows <- nrow(person_matrix)


# Determine which type of person has a greater total count

if (sum(row_sums) > (total_rows - sum(row_sums))) {

print("There are more vegetarians than non-vegetarians.")

} else if (sum(row_sums) < (total_rows - sum(row_sums))) {

print("There are more non-vegetarians than vegetarians.")

} else {

print("There are an equal number of vegetarians and non-vegetarians.")

}
```

14) 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

x 4 1 5 7 10 2 50 25 90 36

y 12 5 13 19 31 7 153 72 275 110

PROGRAM

```
# Create a vector for the number of mobile phones sold

x <- c(4, 1, 5, 7, 10, 2, 50, 25, 90, 36)



# Create a vector for the amount of money

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



# Plot the data as a scatter plot

plot(x, y, main = "Mobile Phones Sold vs. Amount of Money", xlab = "Mobile Phones Sold",
ylab = "Amount of Money")

# Customize the scatter plot

plot(x, y, main = "Mobile Phones Sold vs. Amount of Money", xlab = "Mobile Phones Sold",
ylab = "Amount of Money", col = "blue", pch = 16, xlim = c(0, 100), ylim = c(0, 300))

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

15) 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 (c) clustering

PROGRAM

```
# create a vector of marks

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



# (a) equal-frequency partitioning (3 bins)

eq_freq_bins <- cut(marks, breaks = 3, labels = FALSE)
```

```
# (b) equal-width partitioning (3 bins)
eq_width_bins <- cut(marks, breaks = 3, labels = FALSE,
include.lowest = TRUE, dig.lab = 2)

# (c) clustering (3 bins)
library(cluster)
kmeans_bins <- kmeans(marks, centers = 3)$cluster

# plot histogram for each partitioning method
par(mfrow=c(1,3))
hist(marks[eq_freq_bins == 1], main="Equal-Frequency Partitioning (Bin 1)")
hist(marks[eq_freq_bins == 2], main="Equal-Frequency Partitioning (Bin 2)")
hist(marks[eq_freq_bins == 3], main="Equal-Frequency Partitioning (Bin 3)")
hist(marks[eq_width_bins == 1], main="Equal-Width Partitioning (Bin 1)")
hist(marks[eq_width_bins == 2], main="Equal-Width Partitioning (Bin 2)")
hist(marks[eq_width_bins == 3], main="Equal-Width Partitioning (Bin 3)")
hist(marks[kmeans_bins == 1], main="Clustering (Bin 1)")
hist(marks[kmeans_bins == 2], main="Clustering (Bin 2)")
hist(marks[kmeans_bins == 3], main="Clustering (Bin 3)")
```

16)R The given are the strike-rates scored by a batsman in season 1 in different tournaments. 100, 70, 60, 90, 90 (a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization (c) z-score normalization using the mean absolute deviation instead of standard deviation (d) normalization by decimal scaling

PROGRAM

```
# Create a vector of strike-rates

strike_rates <- c(100, 70, 60, 90, 90)


# Perform min-max normalization

min_strike_rates <- (strike_rates - min(strike_rates)) / (max(strike_rates) - min(strike_rates))

min_strike_rates

# Perform z-score normalization

z_strike_rates <- (strike_rates - mean(strike_rates)) / sd(strike_rates)

z_strike_rates

# Define a function to calculate the mean absolute deviation

mad <- function(x) {

  mean(abs(x - mean(x)))

}


# Perform z-score normalization using mean absolute deviation

mad_strike_rates <- (strike_rates - mean(strike_rates)) / mad(strike_rates)

mad_strike_rates

# Find the decimal factor

decimal_factor <- max(abs(strike_rates))


# Perform decimal scaling normalization

decimal_strike_rates <- strike_rates / decimal_factor

decimal_strike_rates
```

17)R Suppose some car is tested for the AvgSpeed and TotalTime data for 9 randomly selected car with the following result AvgSpeed (in kph) 78 81 82 74 83 82 77 80 70 TotalTime (in mins) 39 37 36 42 35 36 40 38 46 a) Calculate the standard deviation of

AvgSpeed and TotalTime. b) Calculate the Variance of AvgSpeed and TotalTime for the above dataset.

PROGRAM

```
# Create vectors for AvgSpeed and TotalTime

AvgSpeed <- c(78, 81, 82, 74, 83, 82, 77, 80, 70)

TotalTime <- c(39, 37, 36, 42, 35, 36, 40, 38, 46)



# Calculate standard deviation of AvgSpeed and TotalTime

sd(AvgSpeed)

sd(TotalTime)

# Calculate variance of AvgSpeed and TotalTime

var(AvgSpeed)

var(TotalTime)
```

18) .a) Suppose that the "Diabetes data set " data for analysis includes the attribute age. The age values for the data are (in increasing order) 30, 57, 68, 96, 39, 40, 20, 19, 42, 12, 25, 25, 65, 35, 30, 23, 23, 35, 45, 85. What is the mean?

b) Suppose that the speed car is mentioned in different driving style. Regular Speed 78.3 81.8 82 74.2 83.4 84.5 82.9 77.5 80.9 70.6 Calculate the Inter quantile and standard deviation of the given data

PROGRAM

```
age <- c(30, 57, 68, 96, 39, 40, 20, 19, 42, 12, 25, 25, 65, 35, 30, 23, 23, 35, 45, 85)

mean(age)

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

IQR(speed)

sd(speed)
```

19) R PROGRAM .Consider a person want to take a censes / plot for the breast-cancer affected people

through the years.Create a own dataset with this parameters age, tumorsize,inv-nodes

[example between age 1-5 = no.of.count, 6-10=no.of.count,etc]

Draw the Histogram, scatterplot,boxplot.

PROGRAM

```
set.seed(123)

age <- c(rnorm(20, 3, 1), rnorm(20, 8, 1), rnorm(20, 13, 1), rnorm(20, 18, 1), rnorm(20, 23, 1))

tumorsize <- rnorm(100, 5, 1)

invnodes <- rnorm(100, 3, 1)

# Histogram

hist(tumorsize)



# Scatterplot

plot(tumorsize, invnodes)



# Boxplot

boxplot(tumorsize ~ age)
```

20) R TOOL a)Let us consider one example to make the calculation method clear. Assume that the

minimum and maximum values for the feature F are $50,000 and $100,000 correspondingly.

It needs to range F from 0 to 1. In accordance with min-max normalization, v = $80, R PROGRAM

b) Use the two methods below to normalize the following group of data: 200, 300, 400, 600,

1000 R PROGRAM

(a) min-max normalization by setting min = 0 and max = 1

(b) z-score normalization

PROGRAM

```
# Define min and max values
```

```
min_val <- 50000

max_val <- 100000


# Perform min-max normalization

normalized_data <- (80 - min_val) / (max_val - min_val)


# Scale to desired range (0 to 1)

final_normalized_data <- normalized_data * (1 - 0) + 0


# Print the final normalized data

final_normalized_data


# Define min and max values

min_val <- min(c(200, 300, 400, 600, 1000))

max_val <- max(c(200, 300, 400, 600, 1000))


# Perform min-max normalization

normalized_data <- (c(200, 300, 400, 600, 1000) - min_val) / (max_val - min_val)


# Scale to desired range (0 to 1)

final_normalized_data <- normalized_data * (1 - 0) + 0


# Print the final normalized data

final_normalized_data

# Perform z-score normalization
```

```
normalized_data <- (c(200, 300, 400, 600, 1000) - mean(c(200, 300, 400, 600, 1000))) /
sd(c(200, 300, 400, 600, 1000))



# Print the normalized data

normalized_data
```