DAY 1 LIST OF EXPERIMENTS

EXPERIMENT NO:01

Aim:

To compute the median of the given data using R programming .

Code:

# Provided data

intervals <- c("1-5", "5-15", "15-20", "20-50", "50-80", "80-110")

frequencies <- c(200, 450, 300, 1500, 700, 44)

# Calculate cumulative frequencies

cumulative\_frequencies <- cumsum(frequencies)

# Identify the median class

median\_class\_index <- which(cumulative\_frequencies >= sum(frequencies) / 2)[1]

median\_class <- intervals[median\_class\_index]

# Extract lower and upper bounds of the median class

lower\_bound <- as.numeric(strsplit(median\_class, "-")[[1]][1])

upper\_bound <- as.numeric(strsplit(median\_class, "-")[[1]][2])

# Compute the median

N <- sum(frequencies)

CF <- cumulative\_frequencies[median\_class\_index - 1]

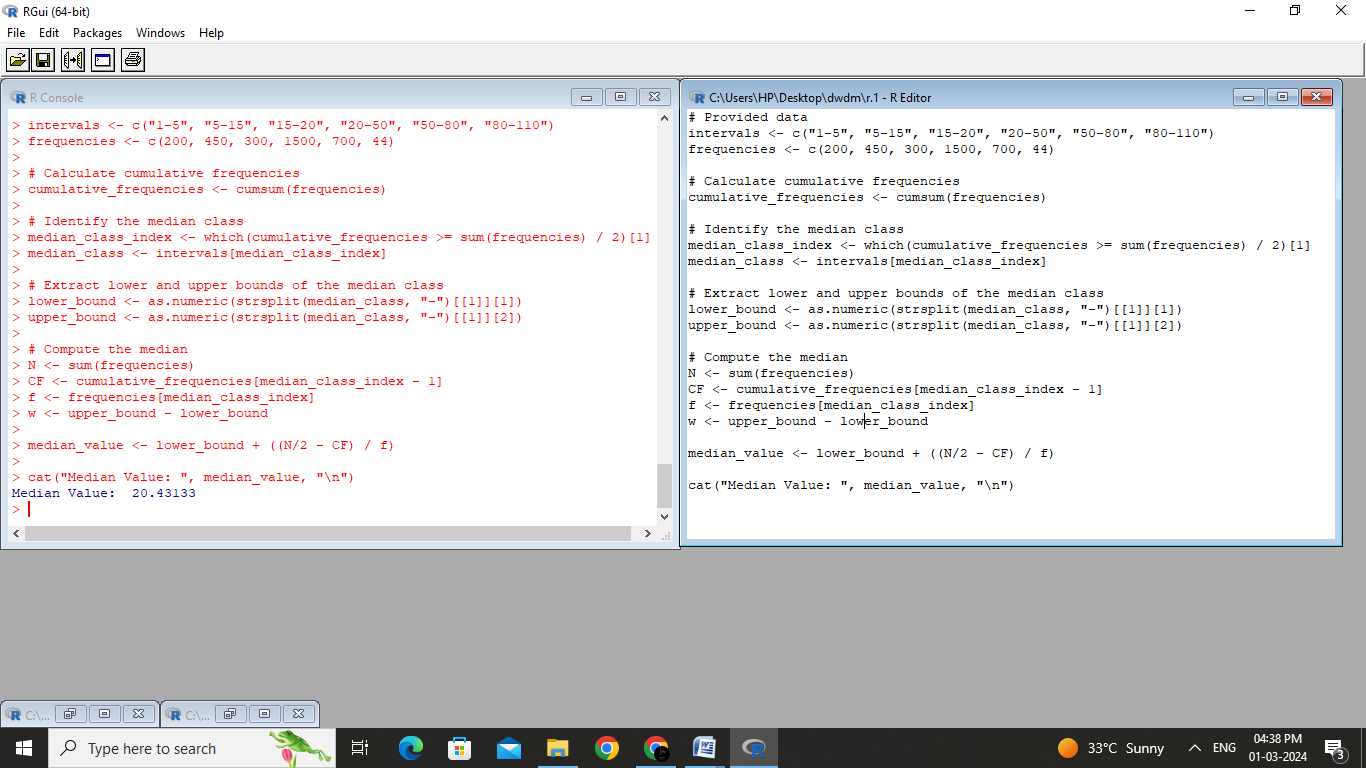
f <- frequencies[median\_class\_index]

w <- upper\_bound - lower\_bound

median\_value <- lower\_bound + ((N/2 - CF) / f)

cat("Median Value: ", median\_value, "\n")

OUTPUT:



Result:

The median is calculated successfully for the given data using the R programming.

EXPERIMENT NO :02

Aim:

To compute the median , mode , mean, midrange and quartiles(1,3) of the attribute age.

Code

# Data

age\_data <- 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\_age <- mean(age\_data)

median\_age <- median(age\_data)

cat("Mean: ", mean\_age, "\n")

cat("Median: ", median\_age, "\n\n")

# (b) Mode and Modality

# Note: Mode may not exist or there can be multiple modes

mode\_result <- table(age\_data)

modes <- as.numeric(names(mode\_result[mode\_result == max(mode\_result)]))

cat("Mode(s): ", ifelse(length(modes) > 1, paste(modes, collapse = ", "), "None"), "\n")

#cat("Modality: ", ifelse(length(modes) > 1, paste("Multimodal (", length(modes), " modes)", sep = ""), "Unimodal"), "\n\n")

# (c) Midrange

midrange <- (min(age\_data) + max(age\_data)) / 2

cat("Midrange: ", midrange, "\n\n")

# (d) Quartiles

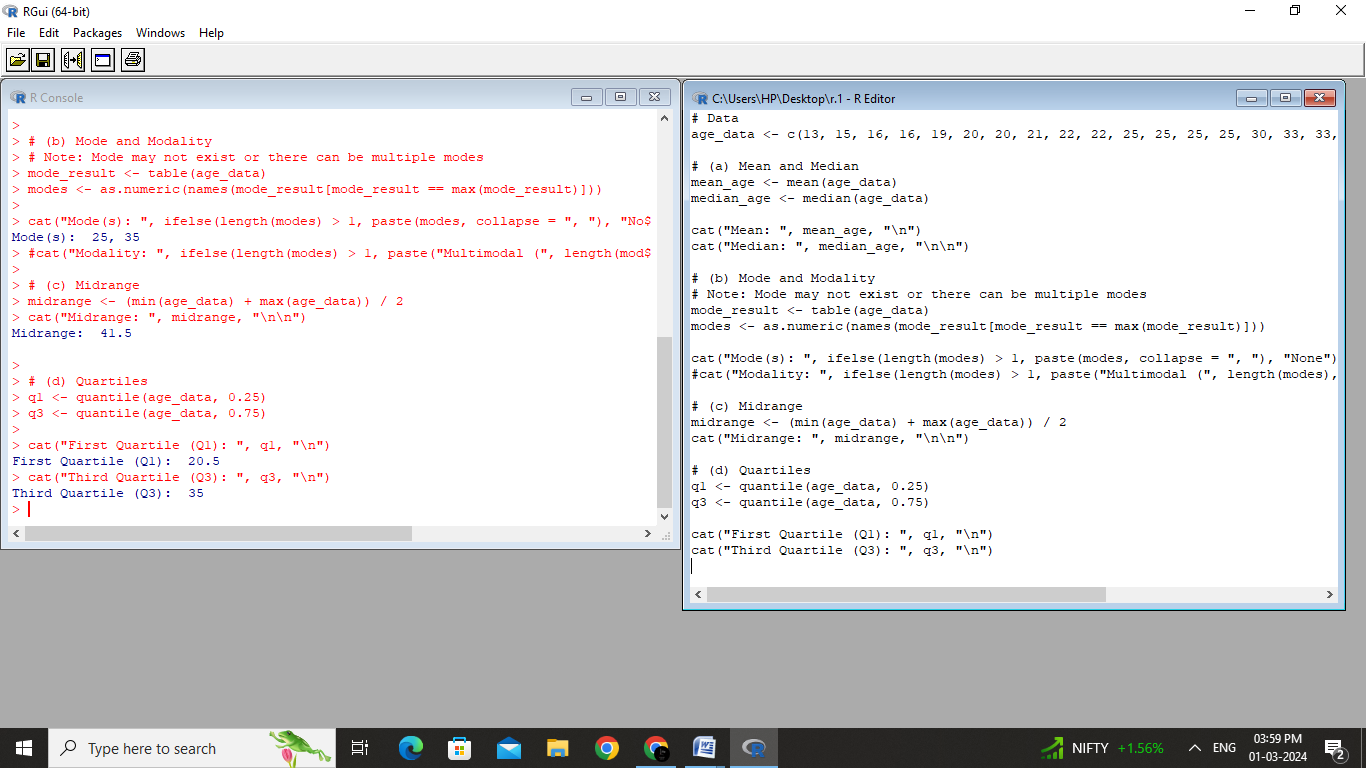
q1 <- quantile(age\_data, 0.25)

q3 <- quantile(age\_data, 0.75)

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

cat("Third Quartile (Q3): ", q3, "\n")

Output:



Result:

The mean , median , mode and midrange , quartiles of the given data are successfully calculated using R programming

EXPERIMENT NO:03

Aim:

To normalize the given data using the min max algorithm and z-score algorithm using R programming.

Code:

# Data

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

# (a) Min-Max Normalization

min\_max\_normalized <- (data - min(data)) / (max(data) - min(data))

cat("(a) Min-Max Normalization:\n")

cat("Original Data: ", data, "\n")

cat("Min-Max Normalized Data: ", min\_max\_normalized, "\n\n")

# (b) Z-Score Normalization

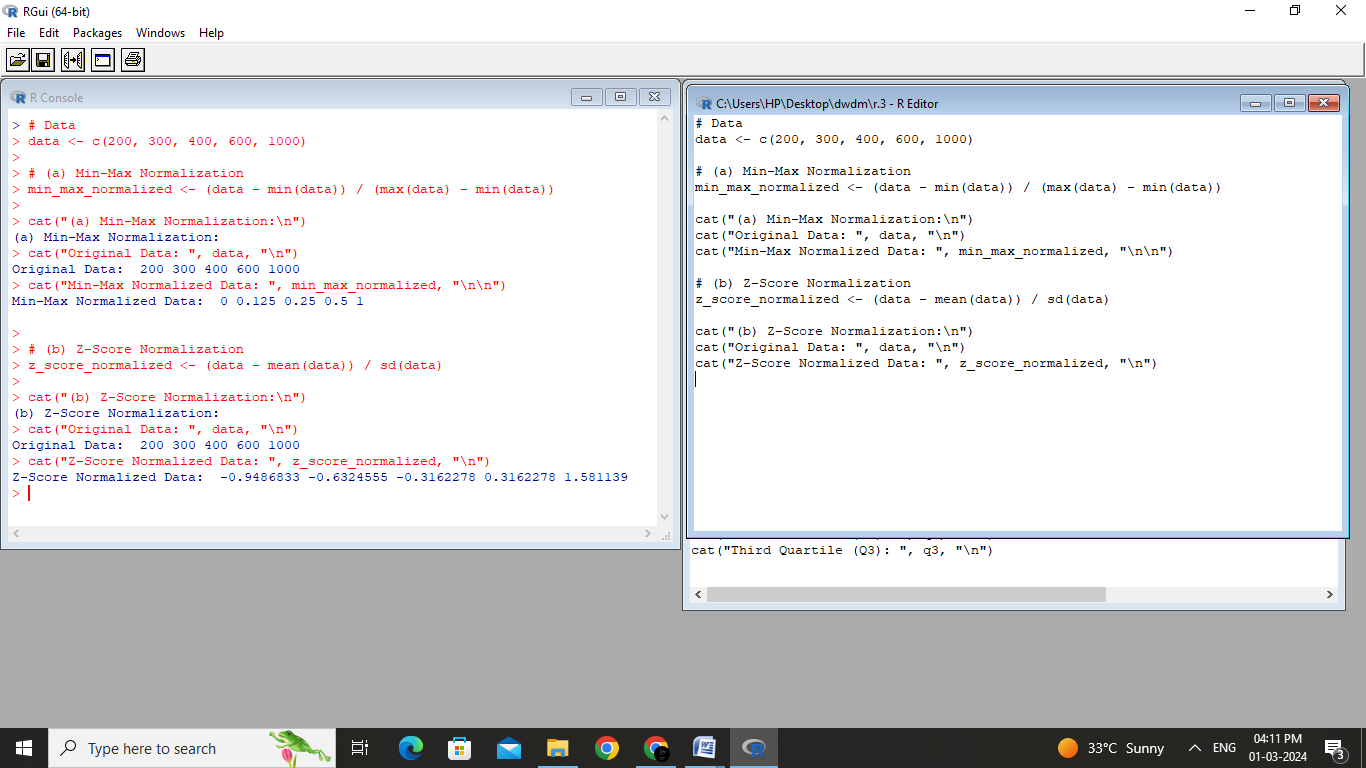
z\_score\_normalized <- (data - mean(data)) / sd(data)

cat("(b) Z-Score Normalization:\n")

cat("Original Data: ", data, "\n")

cat("Z-Score Normalized Data: ", z\_score\_normalized, "\n")

Output:



Result:

The given data is normalised using the min max algorithm and z-score algorithm by constructing executing R programming code.

Experiment: 04

Aim:

To clean the given data using smoothing by bin mean, bin median and 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)

# Bin width for smoothing

bin\_width <- 5

# Function to perform smoothing by bin mean

smooth\_by\_bin\_mean <- function(data, bin\_width) {

bin\_means <- tapply(data, (ceiling(data / bin\_width) - 1) \* bin\_width, mean)

smoothed\_data <- rep(bin\_means, each = table(ceiling(data / bin\_width)))

return(smoothed\_data)

}

# Function to perform smoothing by bin median

smooth\_by\_bin\_median <- function(data, bin\_width) {

bin\_medians <- tapply(data, (ceiling(data / bin\_width) - 1) \* bin\_width, median)

smoothed\_data <- rep(bin\_medians, each = table(ceiling(data / bin\_width)))

return(smoothed\_data)

}

# Function to perform smoothing by bin boundaries

smooth\_by\_bin\_boundaries <- function(data, bin\_width) {

bin\_boundaries <- cut(data, breaks = seq(0, max(data) + bin\_width, bin\_width), include.lowest = TRUE, right = FALSE)

smoothed\_data <- as.numeric(as.character(bin\_boundaries))

return(smoothed\_data)

}

# Apply smoothing methods

smoothed\_mean <- smooth\_by\_bin\_mean(data, bin\_width)

smoothed\_median <- smooth\_by\_bin\_median(data, bin\_width)

smoothed\_boundaries <- smooth\_by\_bin\_boundaries(data, bin\_width)

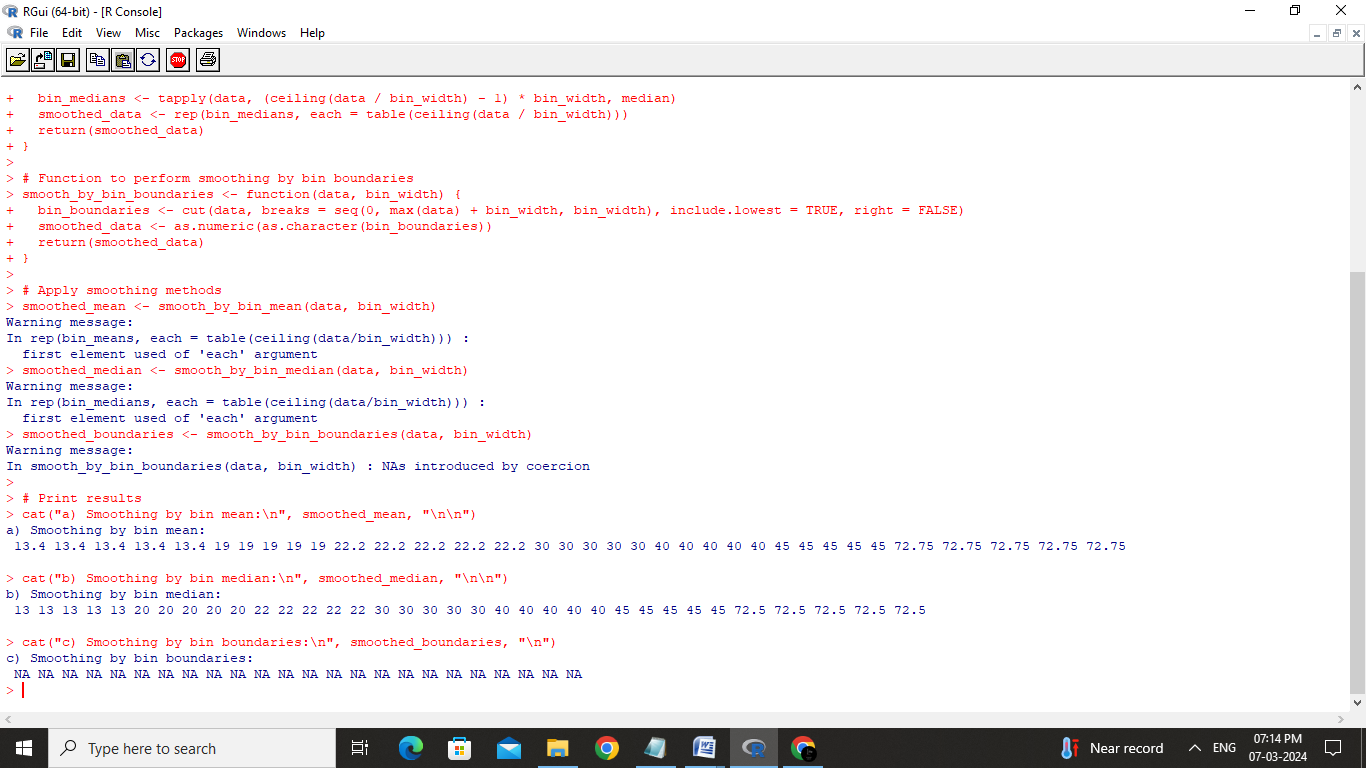
# Print results

cat("a) Smoothing by bin mean:\n", smoothed\_mean, "\n\n")

cat("b) Smoothing by bin median:\n", smoothed\_median, "\n\n")

cat("c) Smoothing by bin boundaries:\n", smoothed\_boundaries, "\n")

OUTPUT:



Result:

The given data is smoothed using the bin mean , bin median and bin boundaries by constructing and executing a R programming code.

Eperiment:05

Aim:

To calculate mean, median, std of the given age and fat data ,to plot box plots and scatter plots , q-q plots.

Code:

# Age and %fat data

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

percent\_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)

# (a) Calculate mean, median, and standard deviation

mean\_age <- mean(age)

median\_age <- median(age)

sd\_age <- sd(age)

mean\_percent\_fat <- mean(percent\_fat)

median\_percent\_fat <- median(percent\_fat)

sd\_percent\_fat <- sd(percent\_fat)

cat("Age:\n")

cat("Mean:", mean\_age, "\n")

cat("Median:", median\_age, "\n")

cat("Standard Deviation:", sd\_age, "\n\n")

cat("%fat:\n")

cat("Mean:", mean\_percent\_fat, "\n")

cat("Median:", median\_percent\_fat, "\n")

cat("Standard Deviation:", sd\_percent\_fat, "\n\n")

# (b) Draw boxplots for age and %fat

par(mfrow = c(1, 2)) # Set up a 1x2 grid for side-by-side plots

boxplot(age, main = "Boxplot of Age", ylab = "Age", col = "lightblue")

boxplot(percent\_fat, main = "Boxplot of %fat", ylab = "%fat", col = "lightgreen")

# (c) Draw a scatter plot

par(mfrow = c(1, 1)) # Reset the plotting layout

plot(age, percent\_fat, main = "Scatter Plot of Age vs %fat", xlab = "Age", ylab = "%fat", pch = 16, col = "blue")

# Interpretation

cat("Interpretation of Boxplots:\n")

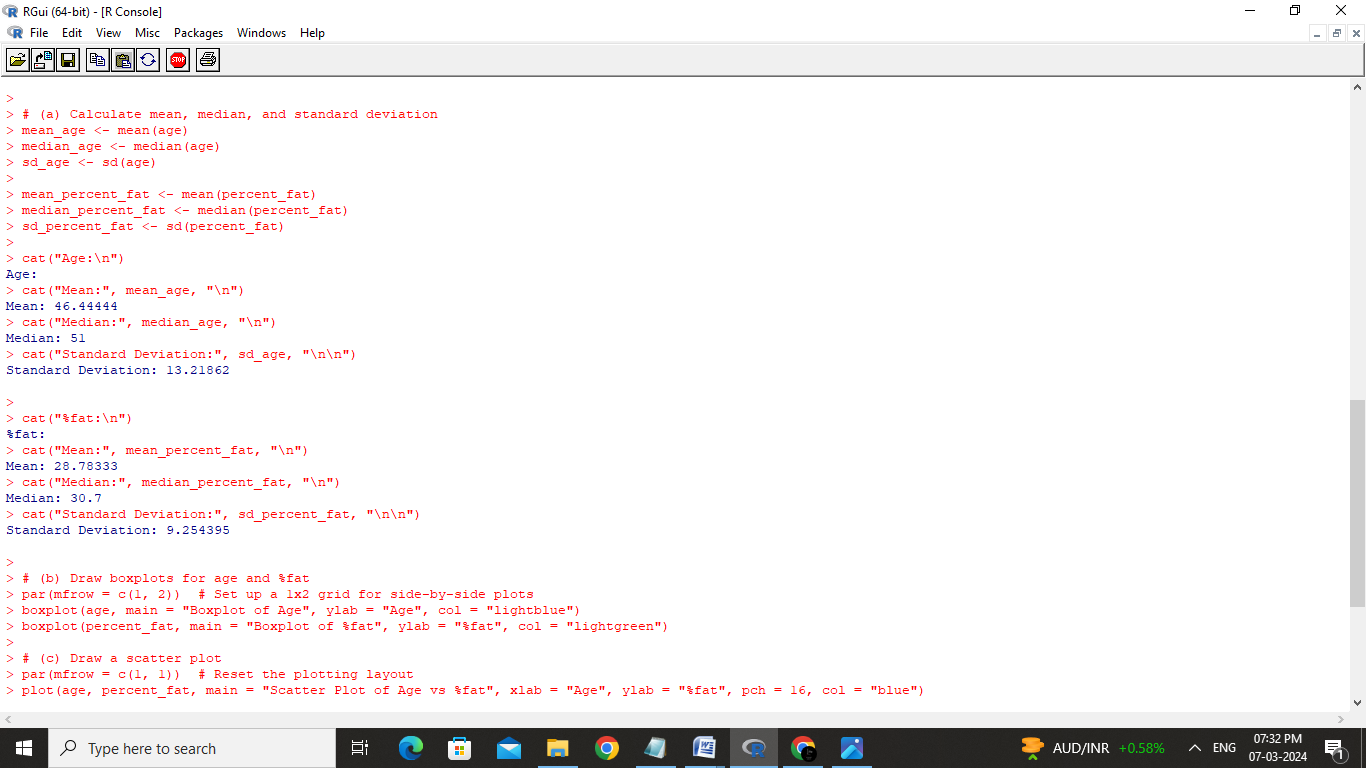
cat("The boxplot for Age shows a relatively symmetric distribution with no outliers.\n")

cat("The boxplot for %fat shows variability, and there is one potential outlier on the higher end.\n\n")

cat("Interpretation of Scatter Plot:\n")

cat("There seems to be a positive correlation between Age and %fat, as older individuals generally have higher body fat percentages.\n")

OUTPUT:



Result:

The given data are plotted on the box and scatter plots successfully by finding their mean and median.

Experiment:06

Aim:

To normalize the data using min-max algorithm ,z-score algorithm , and decimal scalling algorithm using R –tool

Code:

# Given age data

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

# Given age value

age\_value <- 35

# (i) Min-max normalization

min\_age <- min(age)

max\_age <- max(age)

min\_max\_normalized <- (age\_value - min\_age) / (max\_age - min\_age)

cat("Min-Max Normalized Age:", min\_max\_normalized, "\n")

# (ii) Z-score normalization

sd\_age <- sd(age)

z\_score\_normalized <- (age\_value - mean(age)) / sd\_age

cat("Z-Score Normalized Age:", z\_score\_normalized, "\n")

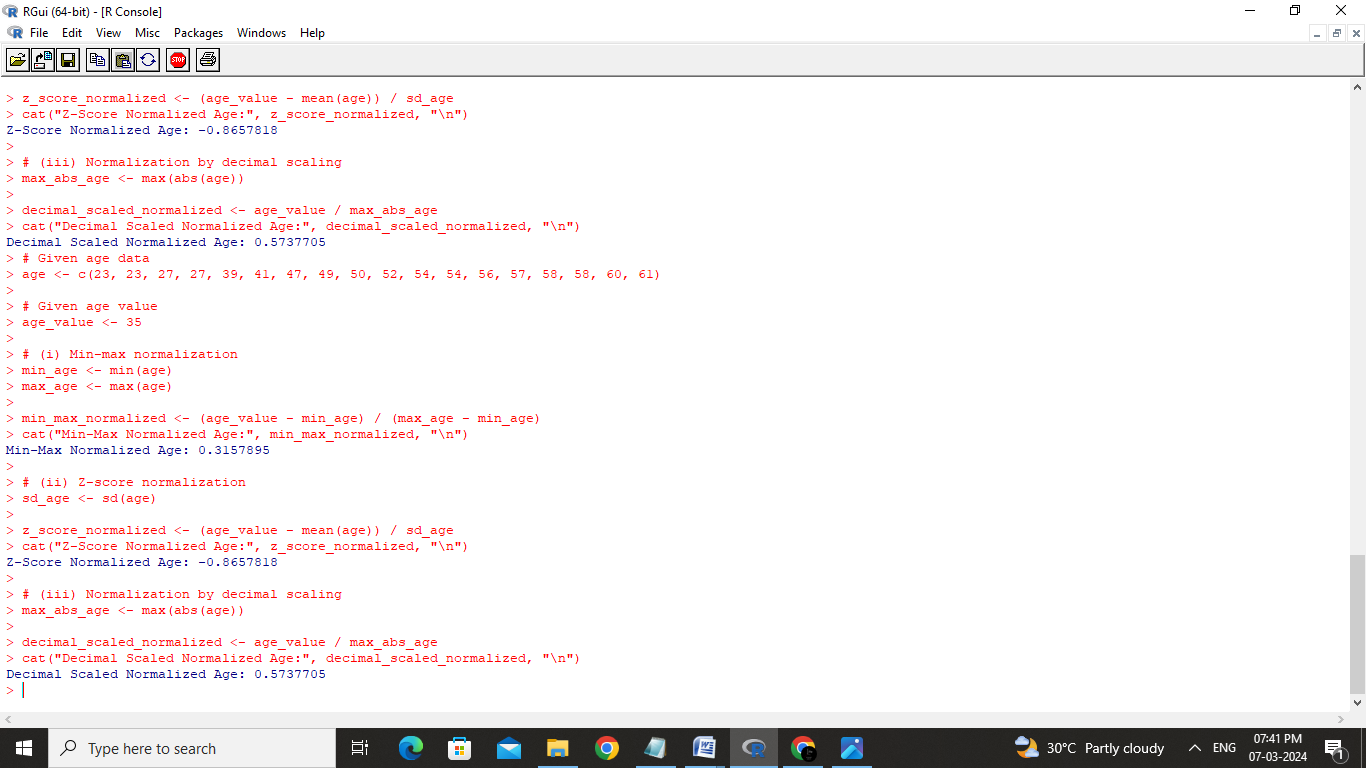
# (iii) Normalization by decimal scaling

max\_abs\_age <- max(abs(age))

decimal\_scaled\_normalized <- age\_value / max\_abs\_age

cat("Decimal Scaled Normalized Age:", decimal\_scaled\_normalized, "\n")

Output:



Result:

The given data is normalized using the min-max algorithm and z-score algorithm ,decimal scalling algorithms successfully.

Experiment:07

Aim :

To fin the bin mean ,bin median and bin mode for the given data

Code:

# Given data

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

# Calculate mean, median, and mode

mean\_pencils <- mean(pencils)

median\_pencils <- median(pencils)

# Mode function (creating a custom function)

mode <- function(x) {

unique\_x <- unique(x)

counts <- table(match(x, unique\_x))

mode\_value <- unique\_x[which.max(counts)]

return(mode\_value)

}

mode\_pencils <- mode(pencils)

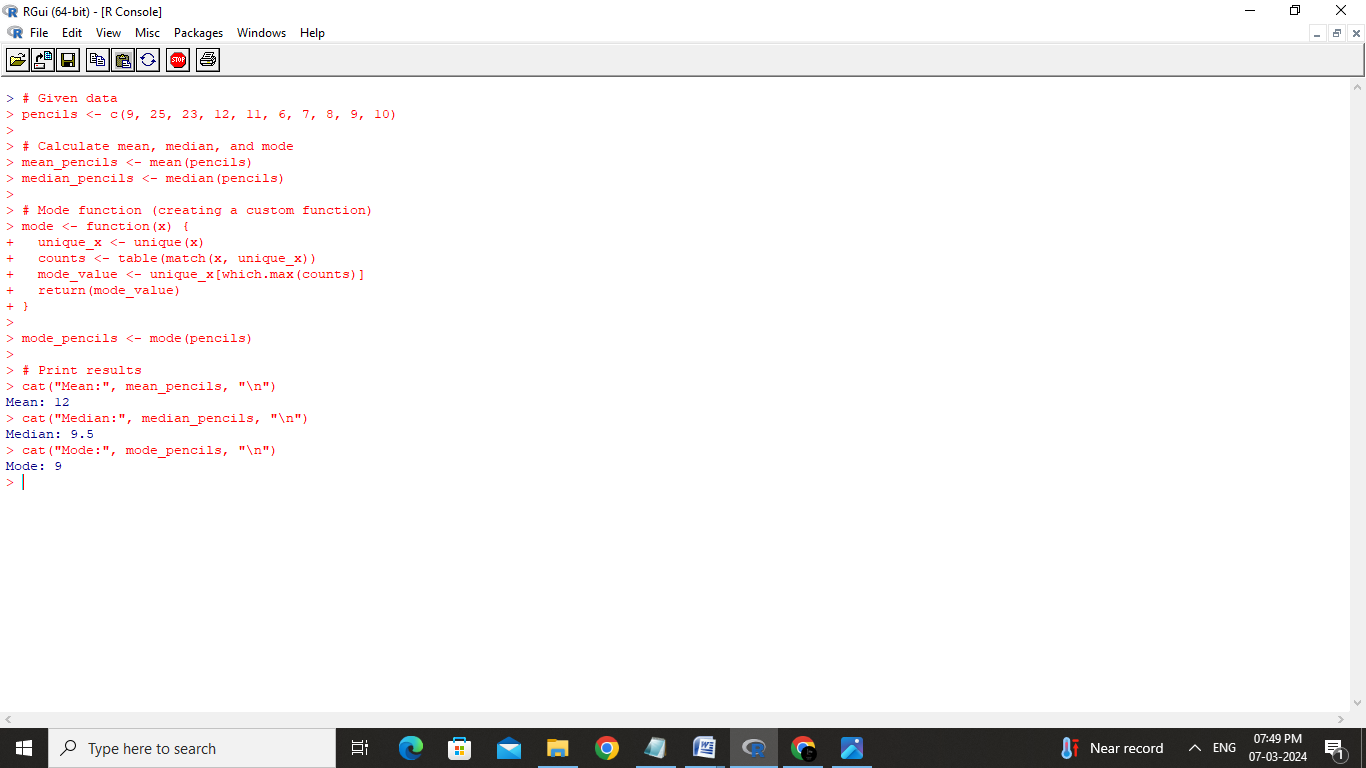
# Print results

cat("Mean:", mean\_pencils, "\n")

cat("Median:", median\_pencils, "\n")

cat("Mode:", mode\_pencils, "\n")

Output:



Result:

The mean,median and mode for the given data is calculated successfully using R-tool

Experiment:08

Aim:

To plot a scatter plot on given data

Code:

# Given data

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

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

# Create a scatter plot

plot(phones\_sold, money, main = "Scatter Plot: Phones Sold vs Money",

xlab = "Number of Mobile Phones Sold", ylab = "Money", pch = 16, col = "blue")

# Add labels to each point for clarity (optional)

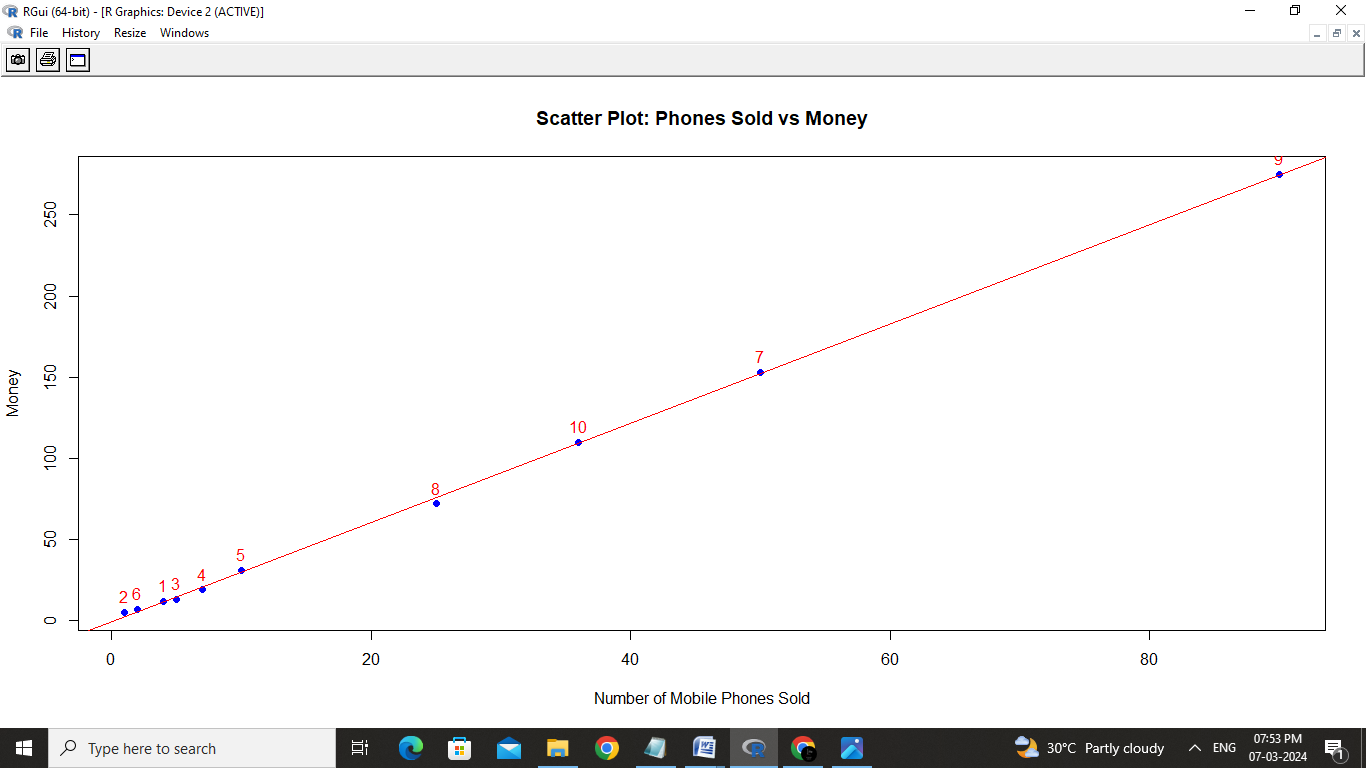
text(phones\_sold, money, labels = seq\_along(phones\_sold), pos = 3, col = "red")

# Optionally, you can add a trendline

fit <- lm(money ~ phones\_sold)

abline(fit, col = "red")

Output:



Result:

The scatter plot for the given data is calculated successfully

Experiment: 09

Aim:

To implement the equip-width and depth partitioning using R-tool

Code:

# Load necessary packages

install.packages("ggplot2") # Install if not already installed

library(ggplot2)

# Example data: Points scored by players

points <- c(50, 60, 65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 125, 130, 140, 150)

# Create a box plot

boxplot(points,

main="Boxplot of Points Scored by Tennis Team Players",

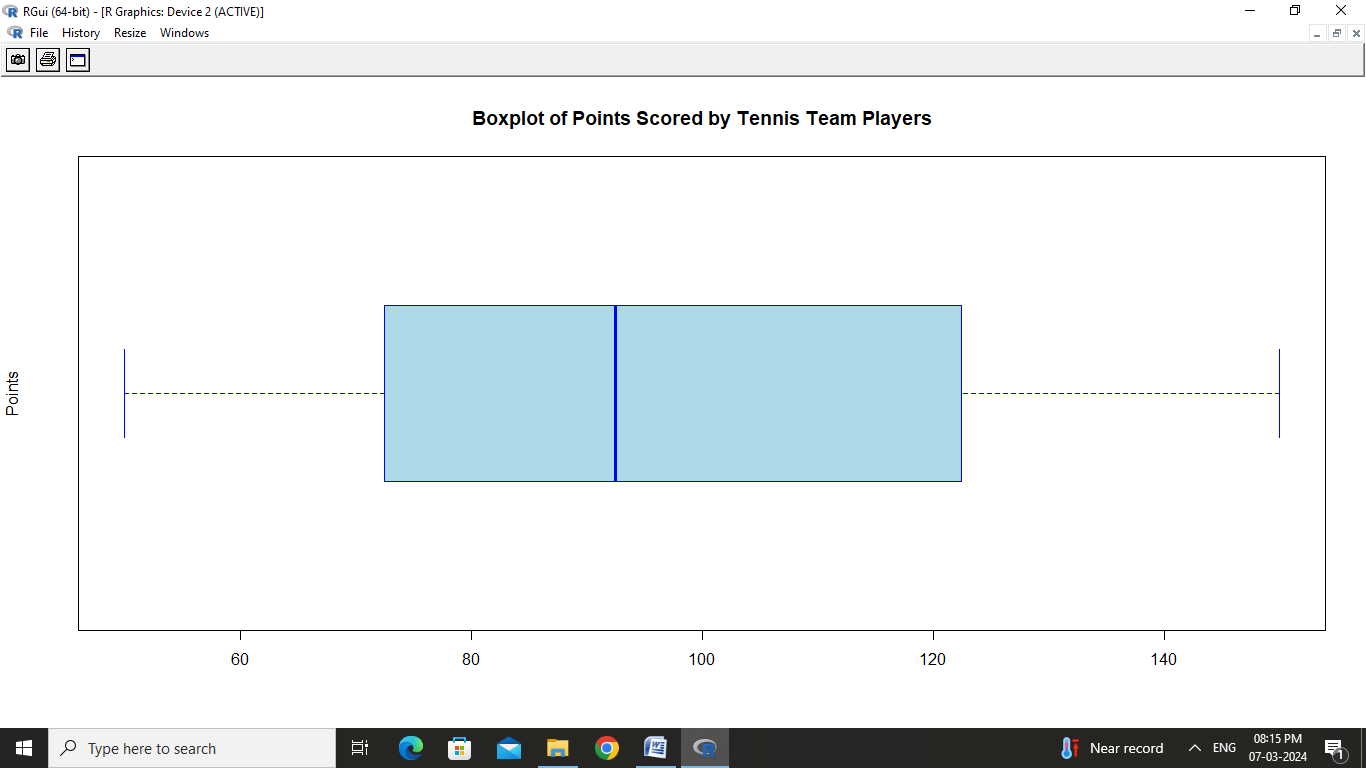
ylab="Points",

col="lightblue",

border="blue",

horizontal=TRUE) # Horizontal orientation for better visualization if many players

Ouptut:



Result:

The box plots for the given data has been plotted successfully