Progressive Education Society's

**Modern College of Engineering**

**MCA Department**

**A.Y. 2024-25**

**Subject Code: 410908: Data Science Laboratory**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class: SY MCA Div: A Batch: S3 Roll Number:52062

Name: Laxman Shinde Assignment No: 05 Date of Implementation: 14/09/2024

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1 Find the mean, median, Mode, Range, Interquartile Range IQR and normal distribution of the physical-fitness scores. Third graders at Roth Elementary School were given a physicalfitness test. Their scores were:

1. 12 22 6 9 2 9 5 9 3 5 16 1 22 18
2. 6 12 21 23 9 10 24 21 17 11 18 19 17 5
3. 14 16 19 19 18 3 4 21 16 20 15 14 17 4
4. 5 22 12 15 18 20 8 10 13 20 6 9 2 17
5. 15 9 4 15 14 19 3 24

Code :

# Step 1: Combine all data into a single vector scores <- c(12, 22, 6, 9, 2, 9, 5, 9, 3, 5, 16, 1, 22, 18, 6, 12, 21, 23, 9, 10, 24, 21, 17, 11, 18, 19, 17, 5,

14, 16, 19, 19, 18, 3, 4, 21, 16, 20, 15, 14, 17, 4,

5, 22, 12, 15, 18, 20, 8, 10, 13, 20, 6, 9, 2, 17,

15, 9, 4, 15, 14, 19, 3, 24)

# Step 2: Calculate Mean mean\_score <- mean(scores) cat("Mean:", mean\_score, "\n")

# Step 3: Calculate Median median\_score <- median(scores) cat("Median:", median\_score, "\n")

# Step 4: Calculate Mode (Mode is not a built-in function in R, so we define it) get\_mode <- function(v) { uniqv <- unique(v) uniqv[which.max(tabulate(match(v, uniqv)))]

} mode\_score <- get\_mode(scores) cat("Mode:", mode\_score, "\n")

# Step 5: Calculate Range range\_score <- range(scores)

cat("Range: ", range\_score[1], "-", range\_score[2], "\n")

# Step 6: Calculate Interquartile Range (IQR) iqr\_score <- IQR(scores)

cat("Interquartile Range (IQR):", iqr\_score, "\n")

# Step 7: Check if the data is normally distributed

# Plot histogram and add normal curve

hist(scores, breaks=10, col="lightblue", main="Histogram of Physical Fitness Scores", xlab="Scores", freq=FALSE) curve(dnorm(x, mean=mean(scores), sd=sd(scores)), add=TRUE, col="red", lwd=2)

# Perform Shapiro-Wilk test for normality shapiro\_test <- shapiro.test(scores)

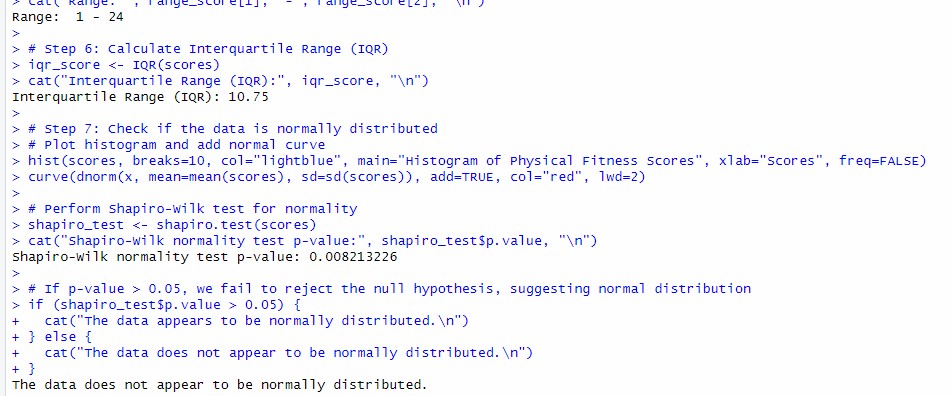
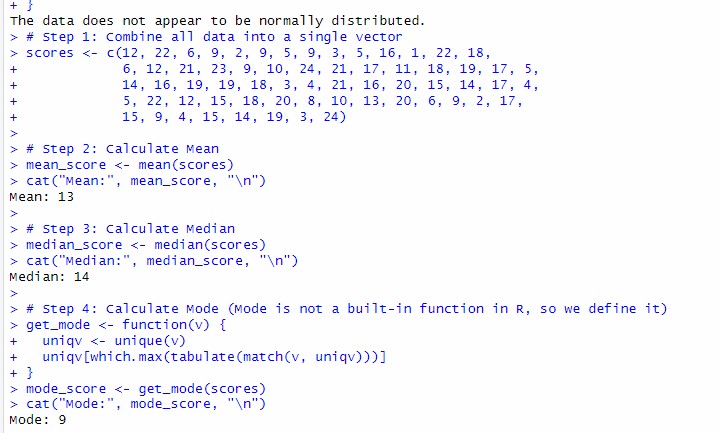
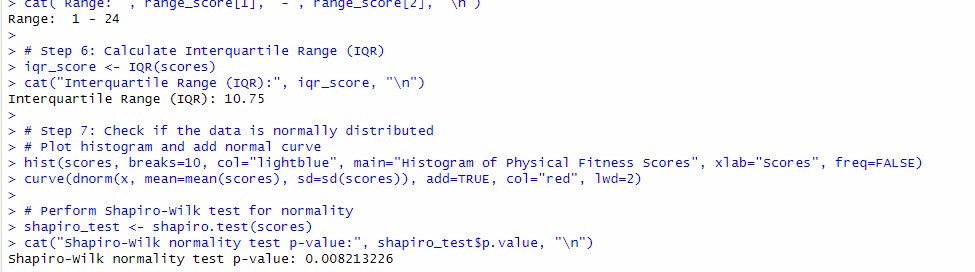
cat("Shapiro-Wilk normality test p-value:", shapiro\_test$p.value, "\n")

# If p-value > 0.05, we fail to reject the null hypothesis, suggesting normal distribution if (shapiro\_test$p.value > 0.05) {

cat("The data appears to be normally distributed.\n")

} else {

cat("The data does not appear to be normally distributed.\n") } Output :



**Q.2)Plot the line graph using v&lt;- c(7,12,28,3,41) and save the plot.**

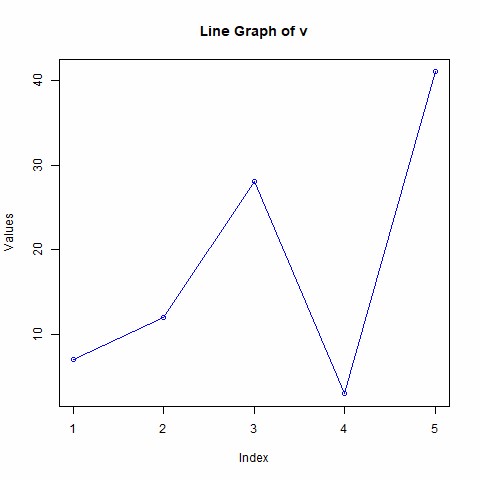
**Code :**

# Step 1: Create the vector v <- c(7, 12, 28, 3, 41)

# Step 2: Plot the line graph

plot(v, type="o", col="blue", xlab="Index", ylab="Values", main="Line Graph of v") # Step 3: Save the plot as a PNG file

png("line\_graph.png") # Open PNG device to save the plot plot(v, type="o", col="blue", xlab="Index", ylab="Values", main="Line Graph of v") dev.off() # Close the device to save the file Output :



**Q.3)Read the file moviesData.csv create a bar chart of critics\_score for the first 10 movies. Save the plot.**

**Code :**

# Step 1: Load necessary libraries

# You can install the library if not already installed using install.packages("ggplot2") library(ggplot2)

# Step 2: Read the CSV file

movies\_data <- read.csv("moviesData.csv")

# Step 3: Extract the first 10 movies and their critics' scores first\_10\_movies <- head(movies\_data, 10) # Extract the first 10 rows

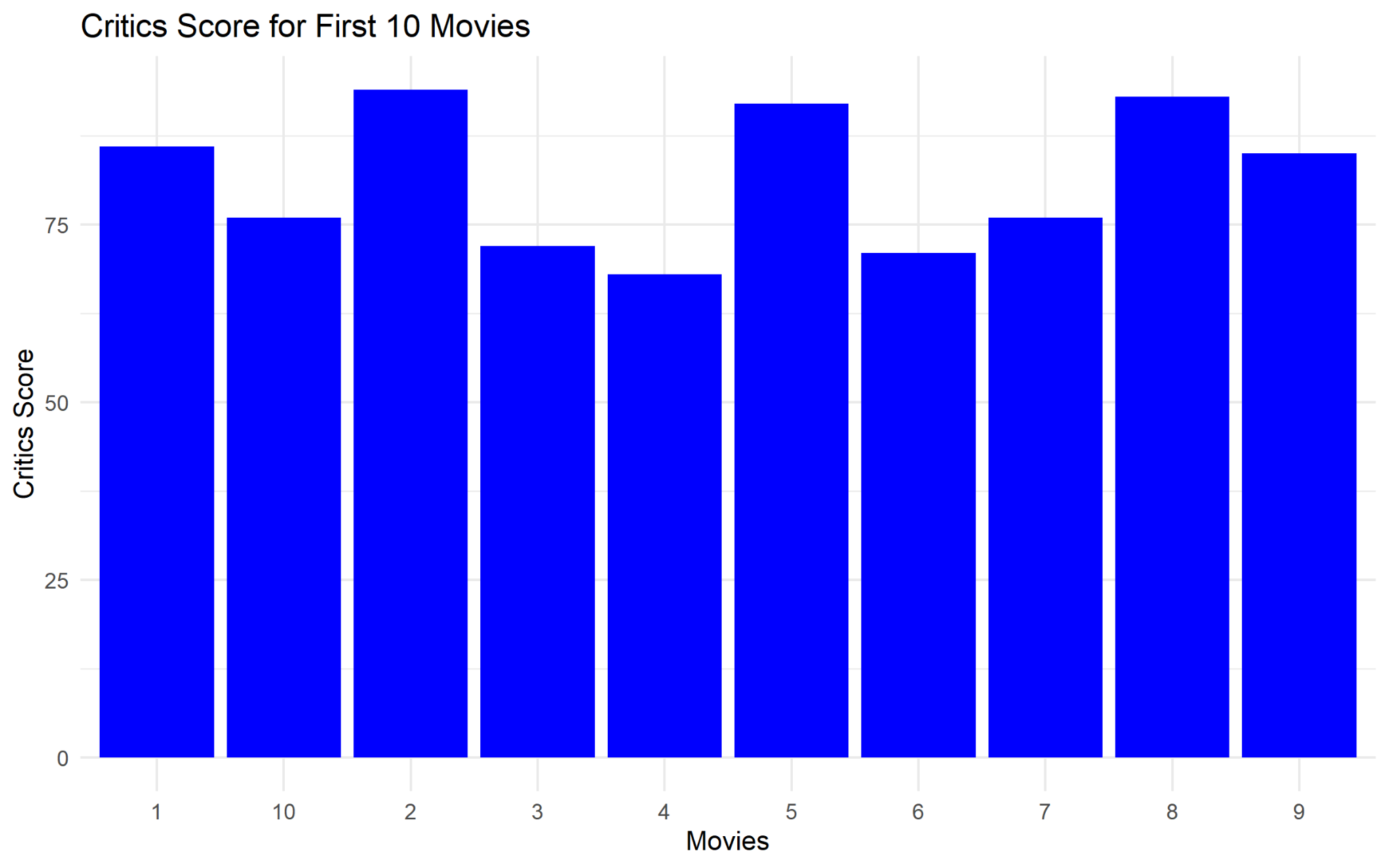
# Check the structure of the data to confirm column names # str(first\_10\_movies)

# Step 4: Create a bar chart using ggplot2

bar\_chart <- ggplot(data=first\_10\_movies, aes(x=as.factor(rownames(first\_10\_movies)), y=critics\_score)) + geom\_bar(stat="identity", fill="blue") + labs(title="Critics Score for First 10 Movies", x="Movies", y="Critics Score") + theme\_minimal()

# Step 5: Display the plot in RStudio or R print(bar\_chart)

# Step 6: Save the plot as a PNG file ggsave("critics\_score\_bar\_chart.png", plot=bar\_chart, width=8, height=5) Output :



**Q.4)Create a scatterplot of imdb\_rating and imdb\_num\_votes to see their relation and save the plot.**

**Code :**

# Load necessary libraries library(ggplot2)

# Read the demo CSV file

file\_path <- "C:/path/to/your/imdb\_data.csv" df <- read.csv(file\_path)

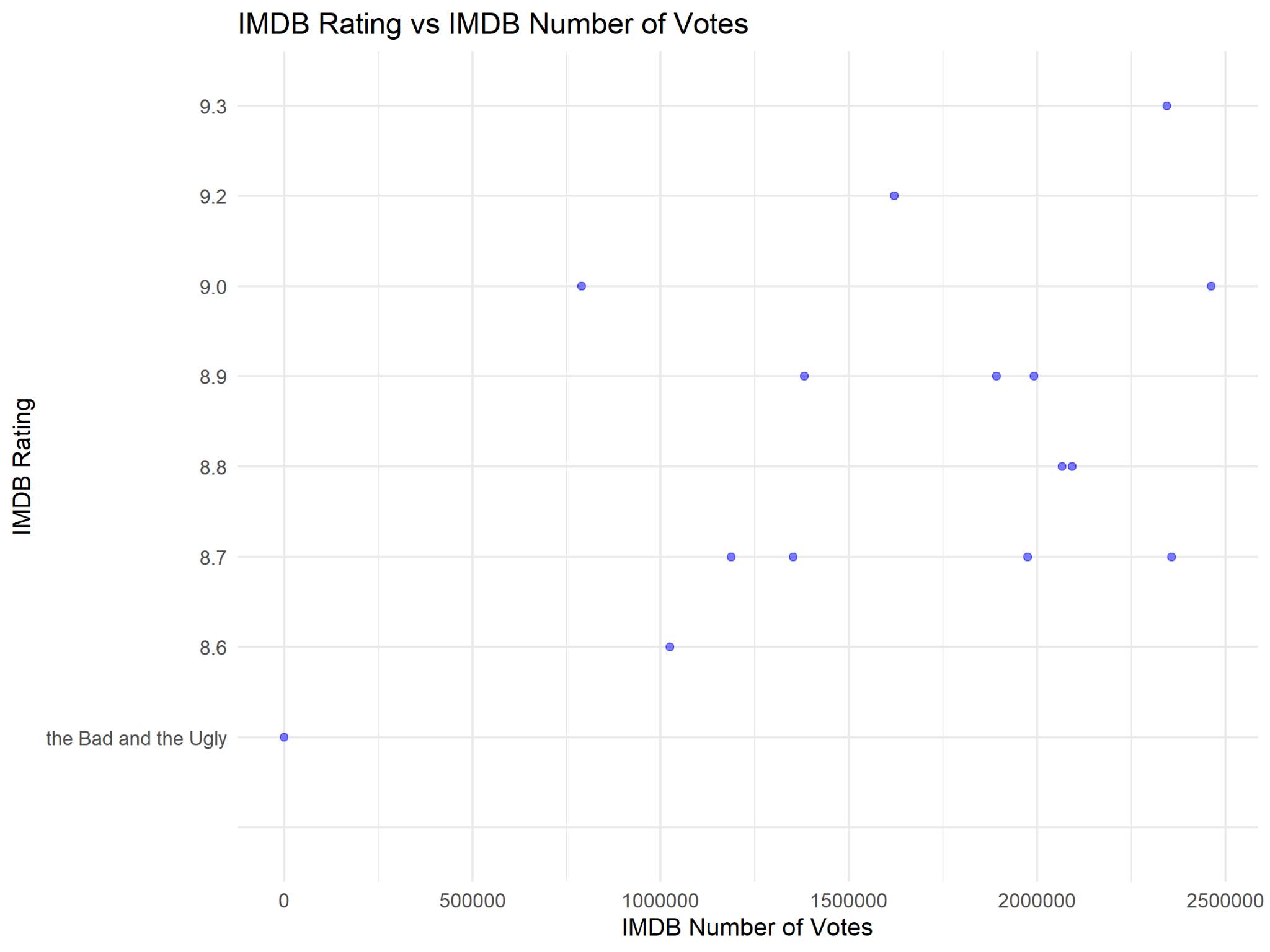
# Create a scatter plot of imdb\_rating vs imdb\_num\_votes scatter\_plot <- ggplot(df, aes(x = imdb\_num\_votes, y = imdb\_rating)) + geom\_point(color = "blue", alpha = 0.5) + # Create points labs(title = "IMDB Rating vs IMDB Number of Votes", x = "IMDB Number of Votes",

y = "IMDB Rating") + theme\_minimal()

# Display the plot print(scatter\_plot)

# Save the plot as a PNG file output\_image\_path <- "imdb\_rating\_vs\_num\_votes.png" ggsave(output\_image\_path, plot = scatter\_plot, width = 8, height = 6)

Output :



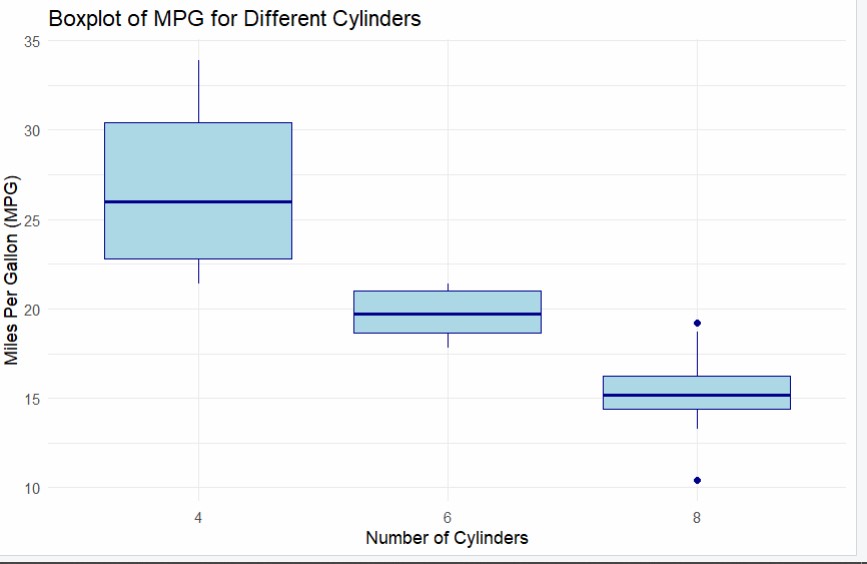
**Q.5)Use the data set “mtcars”and create boxplot for “mpg” and “cyl” columns.**

**Code :**

# Load necessary library library(ggplot2)

# Use the built-in mtcars dataset data("mtcars")

# Create a boxplot for mpg (miles per gallon) across different cylinder counts (cyl) ggplot(mtcars, aes(x = factor(cyl), y = mpg)) + geom\_boxplot(fill = "lightblue", color = "darkblue") + # Create the boxplot labs(title = "Boxplot of MPG for Different Cylinders", x = "Number of Cylinders", y = "Miles Per Gallon (MPG)") + theme\_minimal() Output :



**Q.6)Read the file movies Data.csv, create a histogram of the object named imdb\_num\_votes in this file. Save the plot.**

**Code :**

# Load necessary library library(ggplot2)

# Read the CSV file

file\_path <- "C:/Users/shant/Documents/imdb\_data.csv" # Replace with the correct path movies\_data <- read.csv(file\_path)

# Create a histogram with the exact column name found

histogram\_plot <- ggplot(movies\_data, aes(x = `imdb\_num\_votes`)) + # Use the correct column name here geom\_histogram(binwidth = 50000, fill = "blue", color = "black", alpha = 0.7) + labs(title = "Histogram of IMDB Number of Votes", x = "IMDB Number of Votes",

y = "Frequency") + theme\_minimal()

# Display the histogram print(histogram\_plot)

# Save the plot as a PNG file

output\_image\_path <- "imdb\_num\_votes\_histogram.png" ggsave(output\_image\_path, plot = histogram\_plot, width = 8, height = 6)

Output :

