Progressive Education Society's

**Modern College of Engineering, Pune**

**MCA Department**

**A.Y.2023-24**

**(410908) : Data Science Laboratory**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Class: SY-MCA Shift / Div : B (S1) Roll Number : 52119

Name: Harsh Ghodke Assignment No: 5 Date of Implementation: 09/10/2024 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Q.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 physical-fitness test. Their scores were:**

**a. 12 22 6 9 2 9 5 9 3 5 16 1 22 18**

**b. 6 12 21 23 9 10 24 21 17 11 18 19 17 5**

**c. 14 16 19 19 18 3 4 21 16 20 15 14 17 4**

**d. 5 22 12 15 18 20 8 10 13 20 6 9 2 17**

**e. 15 9 4 15 14 19 3 24**

**Answer:-**

# Defining the scores for each group

group\_a <- c(12, 22, 6, 9, 2, 9, 5, 9, 3, 5, 16, 1, 22, 18)

group\_b <- c(6, 12, 21, 23, 9, 10, 24, 21, 17, 11, 18, 19, 17, 5)

group\_c <- c(14, 16, 19, 19, 18, 3, 4, 21, 16, 20, 15, 14, 17, 4)

group\_d <- c(5, 22, 12, 15, 18, 20, 8, 10, 13, 20, 6, 9, 2, 17)

group\_e <- c(15, 9, 4, 15, 14, 19, 3, 24)

# Function to calculate mean, median, mode, range, IQR

calculate\_stats <- function(scores) {

mean\_value <- mean(scores)

median\_value <- median(scores)

mode\_value <- as.numeric(names(sort(table(scores), decreasing = TRUE))[1])

range\_value <- range(scores)

iqr\_value <- IQR(scores)

return(list(

Mean = mean\_value,

Median = median\_value,

Mode = mode\_value,

Range = range\_value,

IQR = iqr\_value

))

}

# Calculating stats for each group

stats\_a <- calculate\_stats(group\_a)

stats\_b <- calculate\_stats(group\_b)

stats\_c <- calculate\_stats(group\_c)

stats\_d <- calculate\_stats(group\_d)

stats\_e <- calculate\_stats(group\_e)

# Displaying the results

print("Group A stats:")

print(stats\_a)

print("Group B stats:")

print(stats\_b)

print("Group C stats:")

print(stats\_c)

print("Group D stats:")

print(stats\_d)

print("Group E stats:")

print(stats\_e)

# Plotting normal distribution for one group (for example, group A)

group\_a\_mean <- mean(group\_a)

group\_a\_sd <- sd(group\_a)

x <- seq(min(group\_a), max(group\_a), length=100)

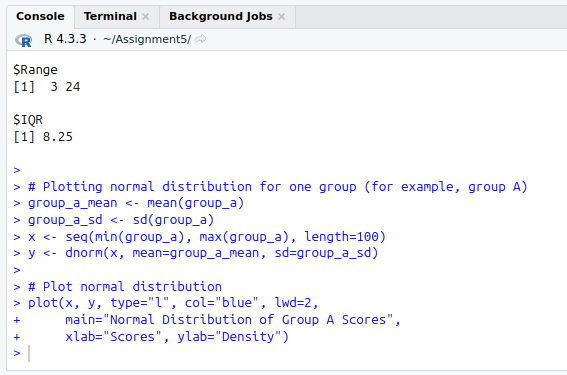
y <- dnorm(x, mean=group\_a\_mean, sd=group\_a\_sd)

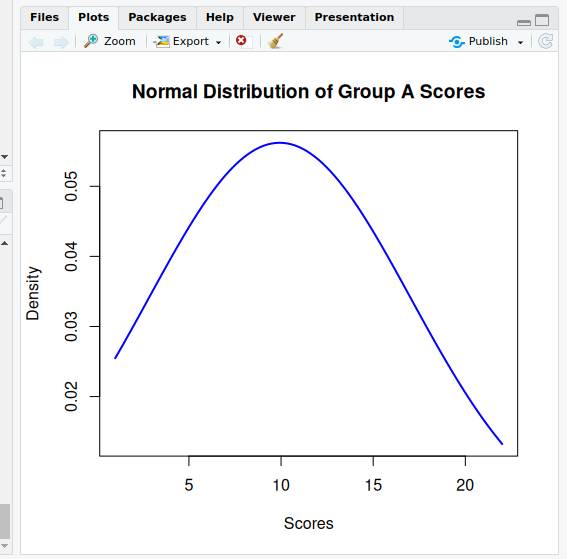
# Plot normal distribution

plot(x, y, type="l", col="blue", lwd=2,

main="Normal Distribution of Group A Scores",

xlab="Scores", ylab="Density")

**Output:-** 



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

**Answer:-**

# Define the vector v

v <- c(7, 12, 28, 3, 41)

# Display the line graph in the IDE

plot(v, type = "o", col = "blue", xlab = "Index", ylab = "Values",

main = "Line Graph of v")

# Save the line graph as a PNG file

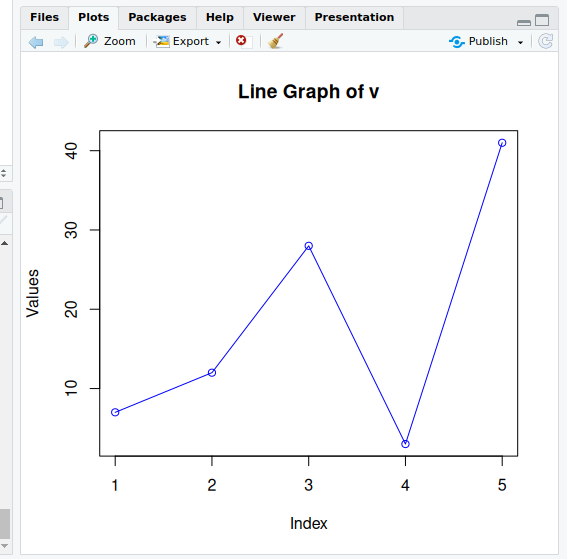
png("line\_graph\_v.png") # Open a PNG device to save the plot

plot(v, type = "o", col = "blue", xlab = "Index", ylab = "Values",

main = "Line Graph of v")

dev.off() # Close the PNG 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.**

**Answer:-**

# Load necessary libraries

library(ggplot2)

# Read the CSV file

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

# Check the structure of the data to ensure it contains 'critics\_score'

str(movies\_data)

# Extract the first 10 rows and relevant columns

first\_10\_movies <- movies\_data[1:10, c("title", "critics\_score")]

# Create a bar chart using ggplot2

bar\_chart <- ggplot(first\_10\_movies, aes(x = reorder(title, -critics\_score), y = critics\_score)) +

geom\_bar(stat = "identity", fill = "blue") +

labs(title = "Critics Score for the First 10 Movies",

x = "Movie Title",

y = "Critics Score") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

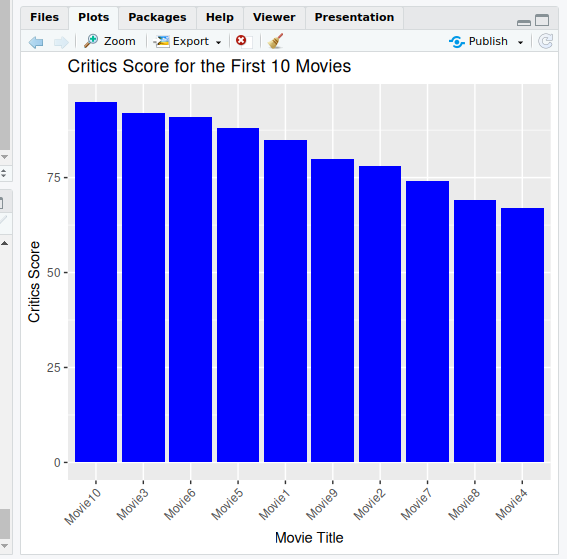
# Display the bar chart

print(bar\_chart)

# Save the plot as a PNG file

ggsave("critics\_score\_bar\_chart.png", plot = bar\_chart)

**Output:-**

****

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

**the plot.**

**Answer:-**

# Load the necessary library

library(ggplot2)

# Read the CSV file (ensure moviesData.csv is in your working directory)

movies\_data <- read.csv("moviesData1.csv")

# Create a scatterplot of imdb\_rating vs imdb\_num\_votes

scatter\_plot <- ggplot(movies\_data, aes(x = imdb\_num\_votes, y = imdb\_rating)) +

geom\_point(color = "blue") + # Scatter plot points in blue

labs(title = "Scatterplot of IMDb Rating vs Number of IMDb Votes",

x = "IMDb Number of Votes",

y = "IMDb Rating") +

theme\_minimal() # Applying a clean theme

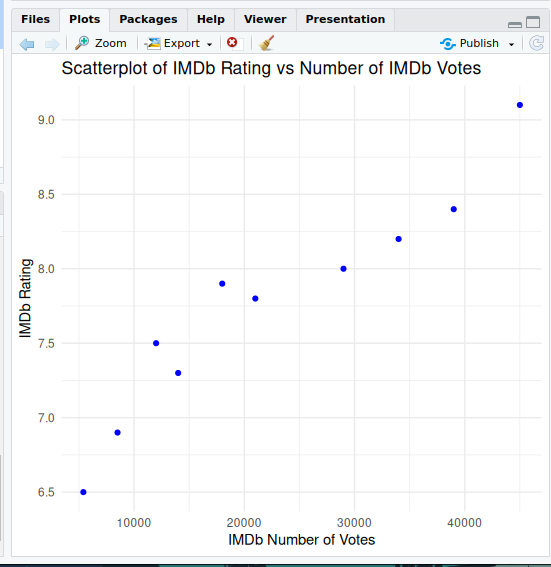
# Display the scatterplot

print(scatter\_plot)

# Save the scatterplot as a PNG file

ggsave("imdb\_scatterplot.png", plot = scatter\_plot)

**Output:-**

****

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

**Answer:-**

# Load the necessary library

library(ggplot2)

# Use the built-in mtcars dataset

data("mtcars")

# Create a boxplot of mpg (Miles per Gallon) grouped by cyl (Number of Cylinders)

boxplot <- ggplot(mtcars, aes(x = factor(cyl), y = mpg)) +

geom\_boxplot(fill = "lightblue") +

labs(title = "Boxplot of MPG by Number of Cylinders",

x = "Number of Cylinders",

y = "Miles per Gallon (MPG)") +

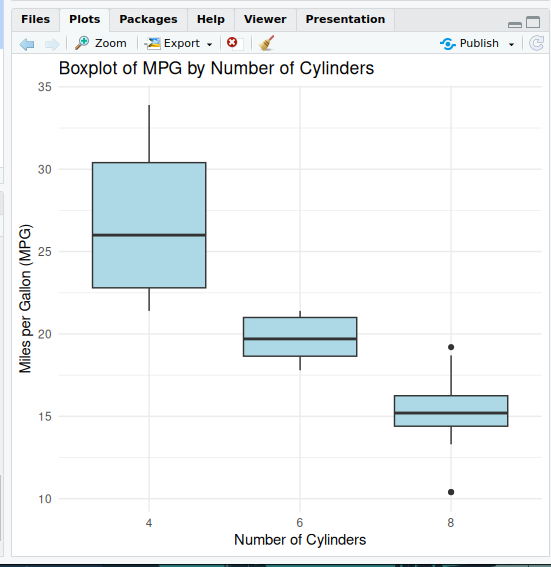
theme\_minimal()

# Display the boxplot

print(boxplot)

# Save the boxplot as a PNG file

ggsave("mpg\_cyl\_boxplot.png", plot = boxplot)



**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.**

**Answer:-**

# Load the necessary library

library(ggplot2)

# Read the CSV file (make sure the file is in your working directory)

movies\_data <- read.csv("moviesData1.csv")

# Check the structure to ensure imdb\_num\_votes is present

str(movies\_data)

# Create a histogram of imdb\_num\_votes

histogram <- ggplot(movies\_data, aes(x = imdb\_num\_votes)) +

geom\_histogram(binwidth = 1000, fill = "lightblue", color = "black") +

labs(title = "Histogram of IMDb Number of Votes",

x = "Number of IMDb Votes",

y = "Frequency") +

theme\_minimal()

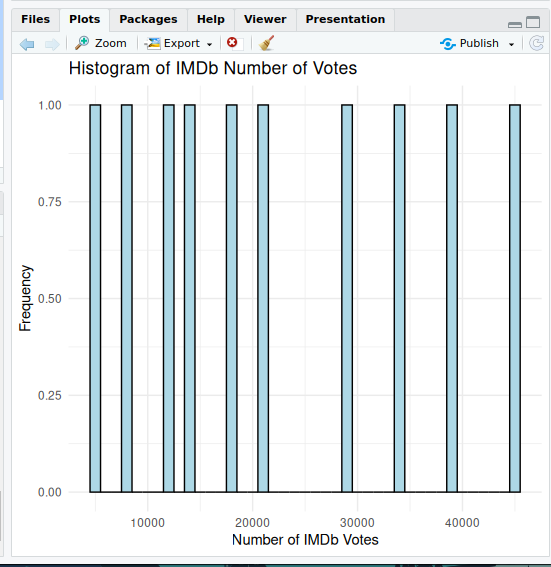
# Display the histogram

print(histogram)

# Save the histogram plot as a PNG file

ggsave("imdb\_num\_votes\_histogram.png", plot = histogram)

**Output:-**

****