LAB

NAME:TELAPOLU MANI SAI LOKESH REGNO:192224105

1. Division:

Input:

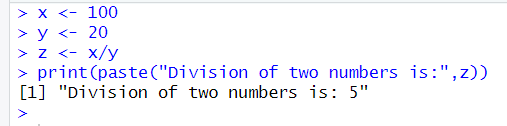
x <- 100

y <- 20

z <- x/y

print(paste("Division of two numbers is:",z))

Output:



1. Histogram:

Input:

# Create data for the graph.

v <- c(19, 23, 11, 5, 16, 21, 32,

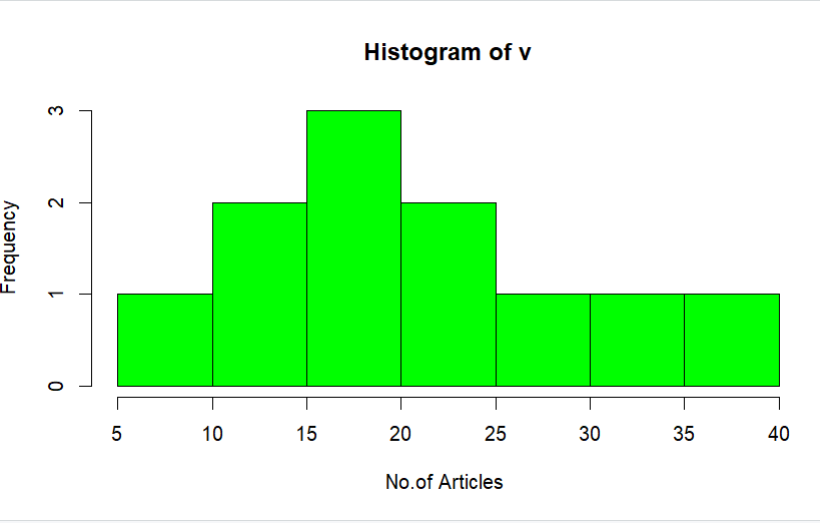
14, 19, 27, 39)

# Create the histogram.

hist(v, xlab = "No.of Articles ",

col = "green", border = "black")

Output:



3.Linear regression:

Input:

# Create the data frame

data <- data.frame(

Years\_Exp = c(1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.2, 3.7),

Salary = c(39343.00, 46205.00, 37731.00, 43525.00,

39891.00, 56642.00, 60150.00, 54445.00, 64445.00, 57189.00)

)

# Create the scatter plot

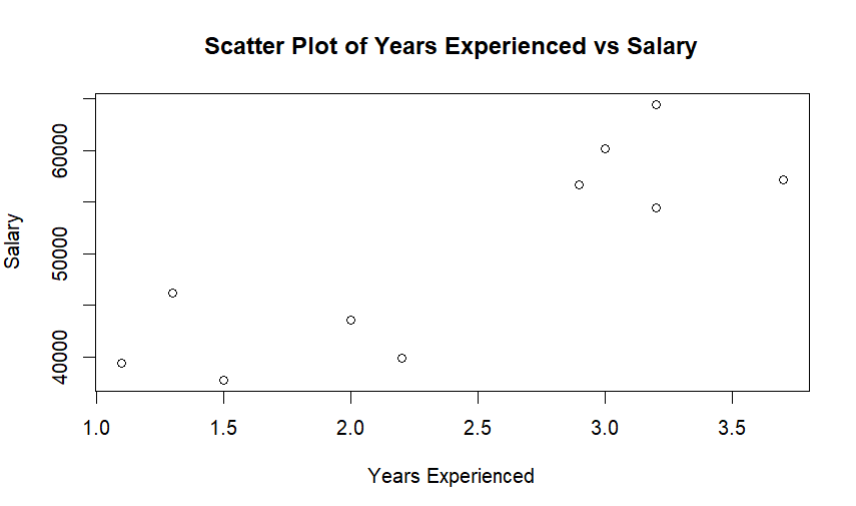
plot(data$Years\_Exp, data$Salary,

xlab = "Years Experienced",

ylab = "Salary",

main = "Scatter Plot of Years Experienced vs Salary")

Output:



4.Median

Input:

# vector of marks

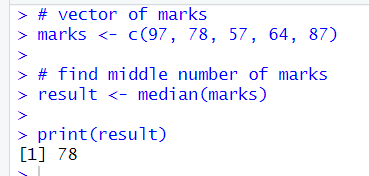
marks <- c(97, 78, 57, 64, 87)

# find middle number of marks

result <- median(marks)

print(result)

Output:



5. Min max normalization:

Input:

# create a vector

data = c(23, 4, 56, 21, 34, 56, 73)

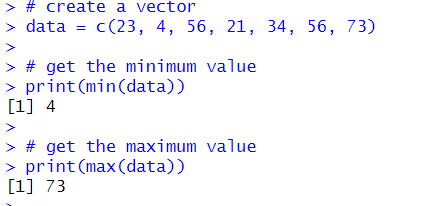
# get the minimum value

print(min(data))

# get the maximum value

print(max(data))

Output:



6.IQR in R :

Input:

# R program to calculate IQR value

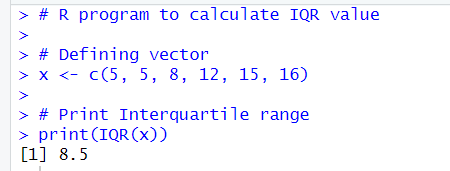
# Defining vector

x <- c(5, 5, 8, 12, 15, 16)

# Print Interquartile range

print(IQR(x))

Output:



7.Quartile in R

Input:

# Sample data

data <- c(12, 25, 30, 42, 50, 58, 65, 73, 80, 92)

# Calculate quartiles

q1 <- quantile(data, 0.25)

q2 <- quantile(data, 0.5) # This is the median

q3 <- quantile(data, 0.75)

# Print quartiles

print(q1)

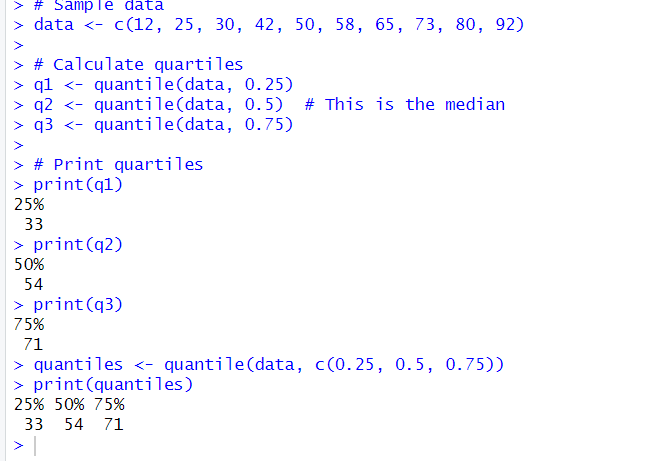
print(q2)

print(q3)

quantiles <- quantile(data, c(0.25, 0.5, 0.75))

print(quantiles)

Output:



8.Summary in R:

Input:

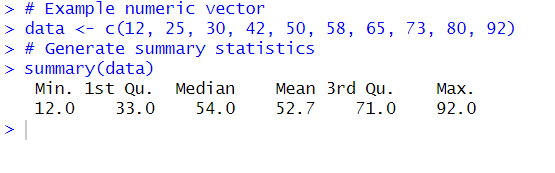
# Example numeric vector

data <- c(12, 25, 30, 42, 50, 58, 65, 73, 80, 92)

# Generate summary statistics

summary(data)

Output:



9.Barplot:

Input:

# Example data for grouped bar plot

categories <- c("A", "B", "C", "D")

values1 <- c(10, 20, 15, 30)

values2 <- c(5, 15, 10, 25)

# Create a grouped bar plot

barplot(rbind(values1, values2),

beside = TRUE,

names.arg = categories,

legend.text = c("Group 1", "Group 2"),

args.legend = list(x = "topright"),

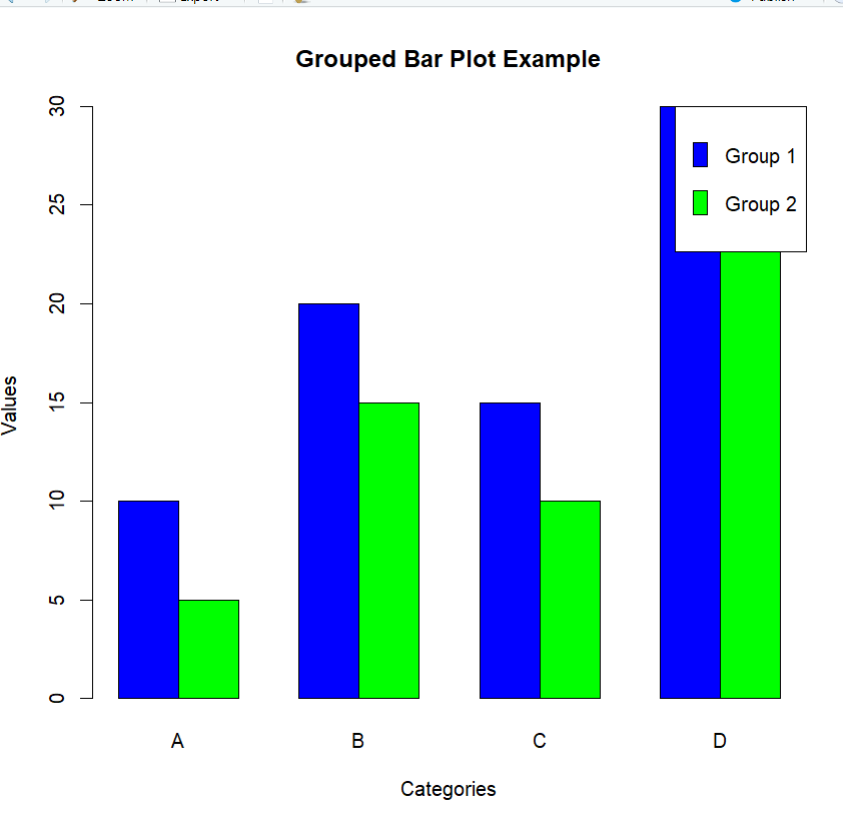
main = "Grouped Bar Plot Example",

xlab = "Categories",

ylab = "Values",

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

Output:



Horizontal Barplot:

Input:

# Example data

categories <- c("A", "B", "C", "D")

values <- c(10, 20, 15, 30)

# Create a horizontal bar plot

barplot(values, names.arg = categories,

main = "Horizontal Bar Plot Example",

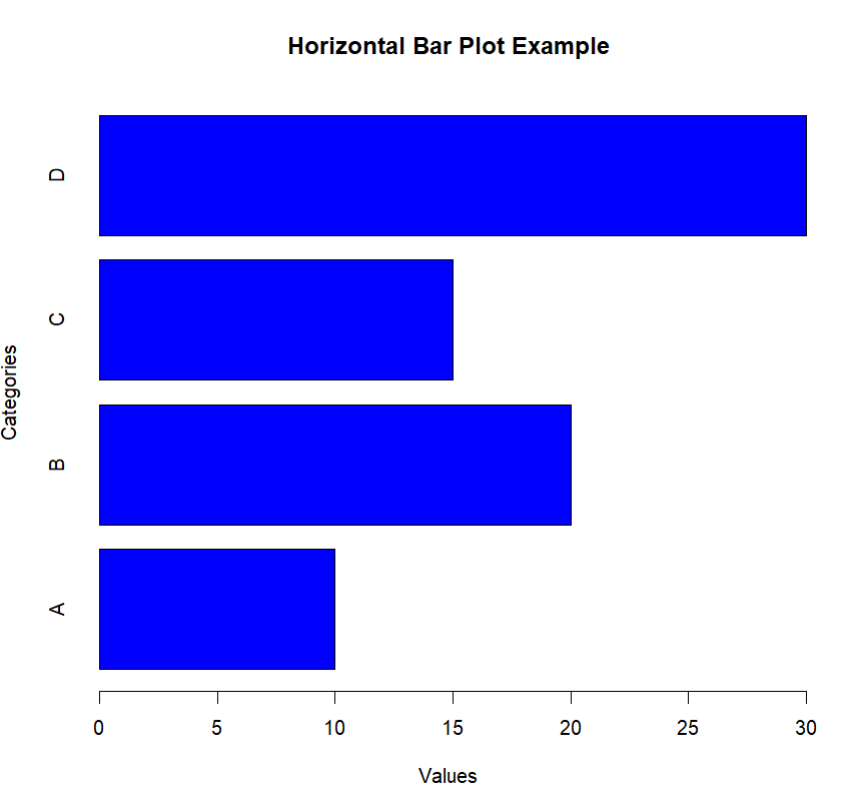
xlab = "Values",

ylab = "Categories",

horiz = TRUE,

col = "blue")

Output:



10.Boxplot in R:

Input:

# Example data for side-by-side boxplot

data1 <- c(25, 30, 40, 50, 55, 60, 70)

data2 <- c(15, 20, 30, 35, 50, 65, 75)

# Create a side-by-side boxplot

boxplot(data1, data2,

names = c("Data 1", "Data 2"),

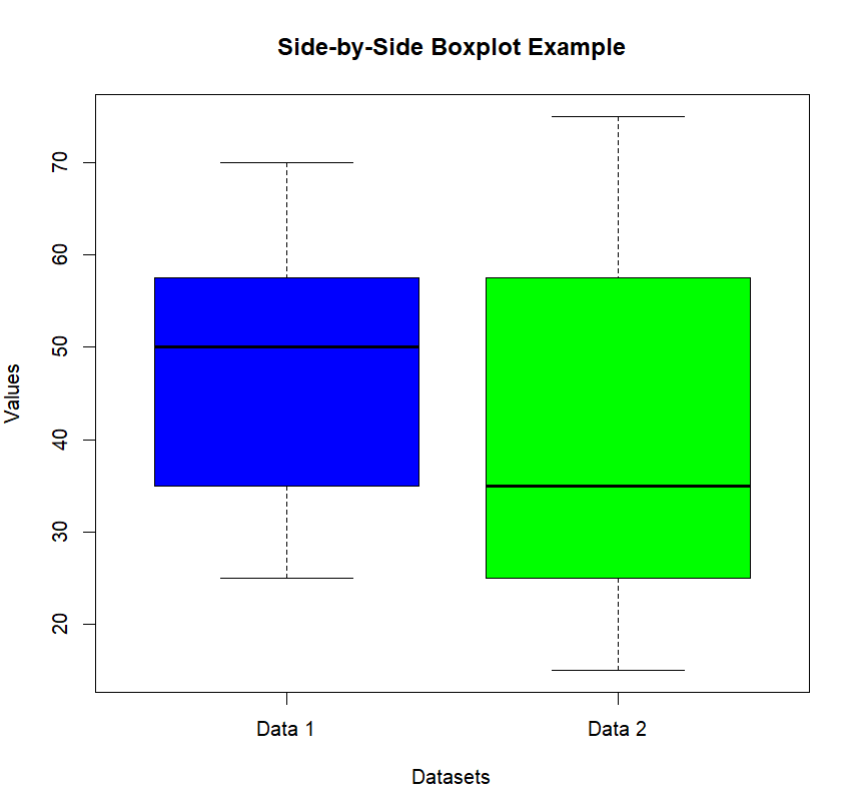
main = "Side-by-Side Boxplot Example",

xlab = "Datasets",

ylab = "Values",

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

Output:



11.Multiplication:

Input:

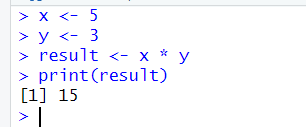
x <- 5

y <- 3

result <- x \* y

print(result)

Output:



12.Even or odd:

Input:

# Check if a number is even or odd

number <- 7

if (number %% 2 == 0) {

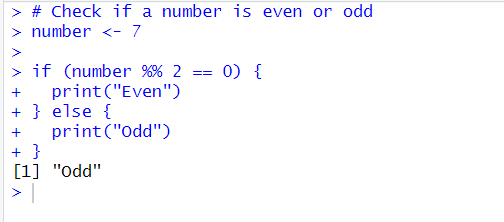
print("Even")

} else {

print("Odd")

}

Output:



13.Greater than three numbers:

Input:

# Example numbers

numbers <- c(2, 5, 1, 8, 4, 6)

# Find and print numbers greater than 3 using if-else

for (num in numbers) {

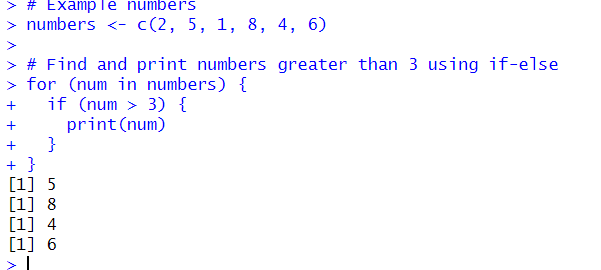
if (num > 3) {

print(num)

}

}

Output:



14.Midrange in R:

Input:

# Example data

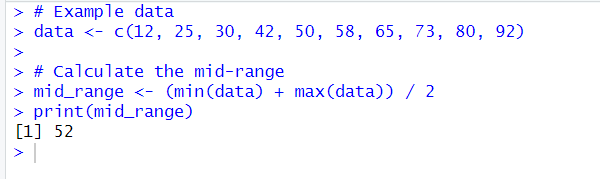
data <- c(12, 25, 30, 42, 50, 58, 65, 73, 80, 92)

# Calculate the mid-range

mid\_range <- (min(data) + max(data)) / 2

print(mid\_range)

Output:



15.Z-score Normalization:

Input:

# Example data

data <- c(12, 25, 30, 42, 50, 58, 65, 73, 80, 92)

# Calculate mean and standard deviation

mean\_data <- mean(data)

sd\_data <- sd(data)

# Perform Z-score normalization

z\_scores <- (data - mean\_data) / sd\_data

# Print the original data and the normalized data

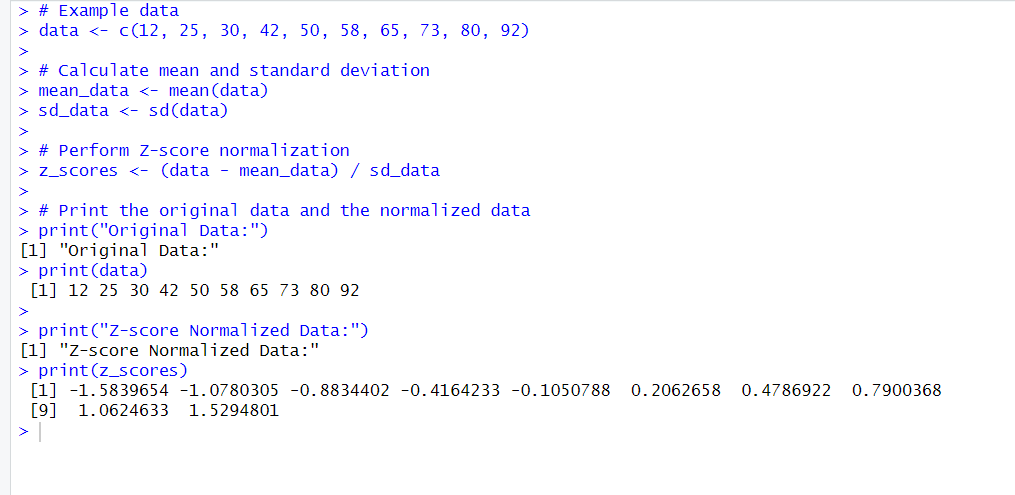
print("Original Data:")

print(data)

print("Z-score Normalized Data:")

print(z\_scores)

Output:



16.Correlation in R:

Input:

# Example data

variable1 <- c(12, 25, 30, 42, 50)

variable2 <- c(15, 28, 35, 45, 60)

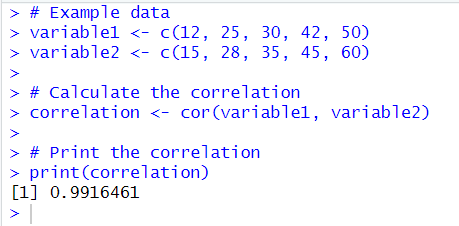
# Calculate the correlation

correlation <- cor(variable1, variable2)

# Print the correlation

print(correlation)

Output:



17.Scatterplot:

Input:

# Example data

variable1 <- c(12, 25, 30, 42, 50)

variable2 <- c(15, 28, 35, 45, 60)

# Create a scatterplot

plot(variable1, variable2,

main = "Scatterplot Example",

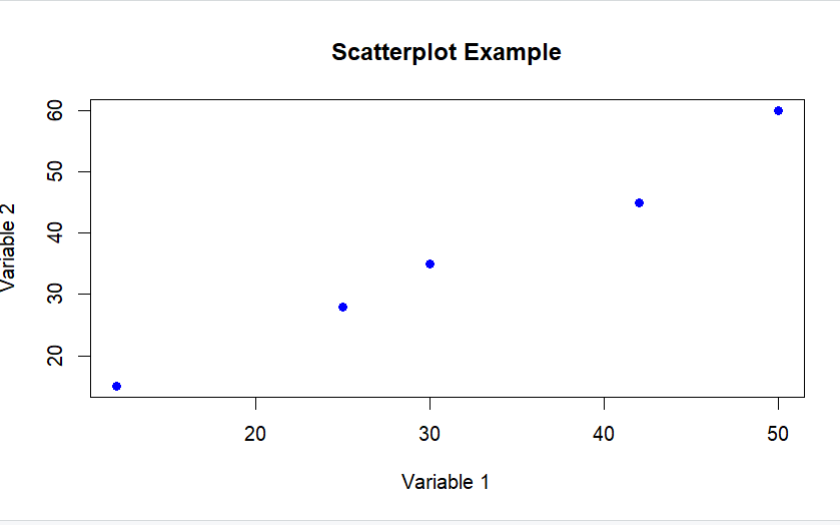
xlab = "Variable 1",

ylab = "Variable 2",

col = "blue",

pch = 16) # Use filled circles for points

Output:



18.Multiple regression in R:

Input:

# Load the dataset

data(mtcars)

# Perform multiple regression

model <- lm(mpg ~ wt + hp + qsec, data = mtcars)

# Print summary of the regression model

summary(model)

Output:

