1.Write a R program for to compute mean, median, minimum, maximum, variance, standard deviation, skewness, kurtosis and quantities (Q1, Q2, Q3)

```
# Sample dataset (replace this with your own data)
data <- c(12, 25, 36, 45, 21, 67, 43, 18, 50, 30)
# Compute mean
mvalue <- mean(data)
# Compute median
mevalue <- median(data)
# Compute minimum and maximum
minvalue <- min(data)
maxvalue <- max(data)
# Compute variance and standard deviation
varvalue <- var(data)
sdvalue <- sd(data)
# Compute skewness and kurtosis
skvalue <- skewness(data)
kurtvalue <- kurtosis(data)
```

```r
# Compute quantiles (Q1, Q2, Q3)
q1 <- quantile(data, 0.25)
q2 <- quantile(data, 0.50) # Same as median
q3 <- quantile(data, 0.75)
# Print the results
cat("Mean:", mvalue, "\n")
cat("Median:", mevalue, "\n")
cat("Minimum:", minvalue, "\n")
cat("Maximum:", maxvalue, "\n")
cat("Variance:", varvalue, "\n")
cat("Standard Deviation:", sdvalue, "\n")
cat("Skewness:",skvalue, "\n")
cat("Kurtosis:",kurtvalue, "\n")
cat("Q1:", q1, "\n")
cat("Q2 (Median):", q2, "\n")
cat("Q3:", q3, "\n")
```

OUTPUT

```
Mean: 34.7
Median: 33
Minimum: 12
Maximum: 67
Variance: 283.5667
Standard Deviation: 16.83944
Q1: 22
Q2 (Median): 33
Q3: 44.5
```

2. Write a R Program that include variables, Constants, data types


```
# Constants
PI <- 3.14159
GREETING <- "Hello, World!"

# Variables
age <- 30
name <- "Alice"
height <- 165.5
is_student <- TRUE

# Printing constants and variables
cat("Constants:\n")
cat("PI:", PI, "\n")
cat("GREETING:", GREETING, "\n\n")

cat("Variables:\n")
cat("Name:", name, "\n")
cat("Age:", age, "\n")
cat("Height:", height, "cm\n")
cat("Is Student:", is_student, "\n")

# Checking data types
```

```
cat("\nData Types:\n")
cat("Name is of type:", class(name), "\n")
cat("Age is of type:", class(age), "\n")
cat("Height is of type:", class(height), "\n")
cat("Is Student is of type:", class(is_student),"\n")
```

OUTPUT

```
Constants:
PI: 3.14159
GREETING: Hello, World!

Variables:
Name: Alice
Age: 30
Height: 165.5 cm
Is Student: TRUE


Data Types:

Name is of type:   character
Age is of type:    numeric
Height is of type: numeric
Is Student is of type: logical
```

3.write a R program that include different operators, Control structures, default values for arguments, returning complex objects

```
# Function with default argument values
calculate_area<- function(length = 1, width = 1)
 {
  area <- length * width
  return(area)
}

# Function that returns a complex object
create_person<- function(name, age, city)
{
```

```
  person <- list(
    Name = name,
    Age = age,
    City = city
  )
  return(person)
}


# Using different operators and control structures
length_value<- 5
width_value<- 3
area_result<- calculate_area(length_value, width_value)
cat("Area:", area_result, "\n")

# Control structure - if-else
if (area_result> 10)
 {
cat("This is a large area.\n")
} else
{
cat("This is a small area.\n")
}

# Control structure - for loop
cat("\nCounting from 1 to 5:\n")
for (i in 1:5)
 {
cat(i, " ")
}
cat("\n\n")

# Control structure - while loop
count <- 1
cat("Counting while less than or equal to 5:\n")
while (count <= 5)
```

```
{
cat(count, " ")
 count <- count + 1
}
cat("\n\n")

# Using the function to create a person object
person1 <- create_person("Alice", 30, "New York")
person2 <- create_person("Bob", 25, "Los Angeles")

cat("Person 1:\n")
cat("Name:", person1$Name, "\n")
cat("Age:", person1$Age, "\n")
cat("City:", person1$City, "\n")

cat("\nPerson 2:\n")
cat("Name:", person2$Name, "\n")
cat("Age:", person2$Age, "\n")
cat("City:", person2$City, "\n")
```

OUTPUT

```
Area: 15
This is a large area.

Counting from 1 to 5:
1  2  3  4  5

Counting while less than or equal to 5:
1  2  3  4  5


Person 1:

Name: Alice
Age: 30
City: New York

Person 2:

Name: Bob
```

```
Age: 25
City: Los Angeles
```

4.Write a R program for calculating cumulative sums, and products minima maxima and calculus.

```
# Sample dataset
data <- c(3, 1, 4, 1, 5, 9, 2, 6, 5, 3)

# Cumulative sum
cumulative_sum<- cumsum(data)
cat("Cumulative Sum:\n")
cat(cumulative_sum, "\n\n")

# Cumulative product
cumulative_product<- cumprod(data)
cat("Cumulative Product:\n")
cat(cumulative_product, "\n\n")

# Minimum and Maximum
min_value<- min(data)
max_value<- max(data)
cat("Minimum Value:", min_value, "\n")
cat("Maximum Value:", max_value, "\n\n")

# Calculus
# Calculate the derivative of the data
derivative <- diff(data)
cat("Derivative of the Data:\n")
cat(derivative, "\n\n")
```

```
# Integrate the data
integral <- cumsum(derivative)
cat("Integral of the Data (Cumulative Sum of Derivative):\n")
cat(integral, "\n")
```

OUTPUT

```
Cumulative Sum:
3 4 8 9 14 23 25 31 36 39

Cumulative Product:
3 3 12 12 60 540 1080 6480 32400 97200

Minimum Value: 1
Maximum Value: 9

Derivative of the Data:
-2 3 -3 4 4 -7 4 -1 -2

Integral of the Data (Cumulative Sum of Derivative):
-2 1 -2 2 6 -1 3 2 0
```

5.Write a R Program for finding the stationary distribution of markanov chains.

```
# Define the transition matrix for your Markov chain

tmatrix <- matrix(c(0.7, 0.2, 0.1, 0.3, 0.6, 0.1, 0.1, 0.3, 0.6), nrow = 3, byrow = TRUE)

# Function to compute stationary distribution using iterative method

computest <- function(tmatrix, tol = 1e-6, miter = 1000)

{

  n <- nrow(tmatrix)
```

```
  pi <- rep(1/n, n)  # Initial guess for stationary distribution
  for (iter in 1:miter)
  {
    new_pi <- pi %*% tmatrix
    if (sum(abs(new_pi - pi)) < tol)
    {
      break
    }
    pi <- new_pi
  }
  return(new_pi)
}
# Compute stationary distribution
st <- computest(tmatrix)
# Print the stationary distribution
cat("Stationary Distribution:")
print(st)
```

OUTPUT

```
Stationary Distribution:
          [,1]      [,2]      [,3]
[1,] 0.4333331 0.3666668 0.2000001
```

6.Write a R program that include linear algebra operations on vectors and matrices.

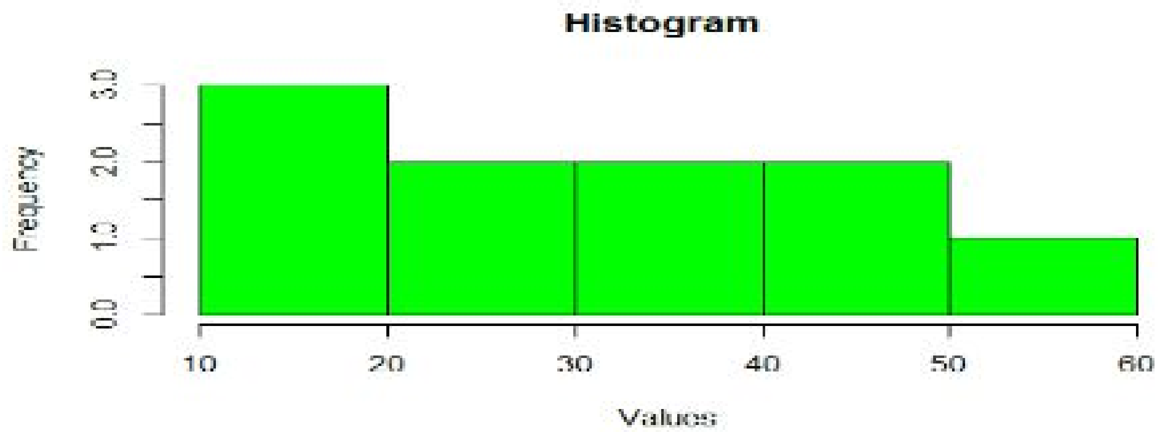
```
# Create vectors
vec1 <- c(1, 2, 3)
vec2 <- c(4, 5, 6)

# Create matrices
mat1 <- matrix(1:6, nrow = 2)
mat2 <- matrix(7:12, nrow = 2)

# Vector addition
sum<- vec1 + vec2
cat("Vector Addition (vec1 + vec2):\n")
cat(sum, "\n\n")

# Vector subtraction
diff<- vec1 - vec2
cat("Vector Subtraction (vec1 - vec2):\n")
cat(diff, "\n\n")

# Vector dot product
prod<- sum(vec1 * vec2)
cat("Vector Dot Product:\n")
cat(prod, "\n\n")

# Matrix addition
msum<- mat1 + mat2
cat("Matrix Addition (mat1 + mat2):\n")
print(msum)
cat("\n")
```

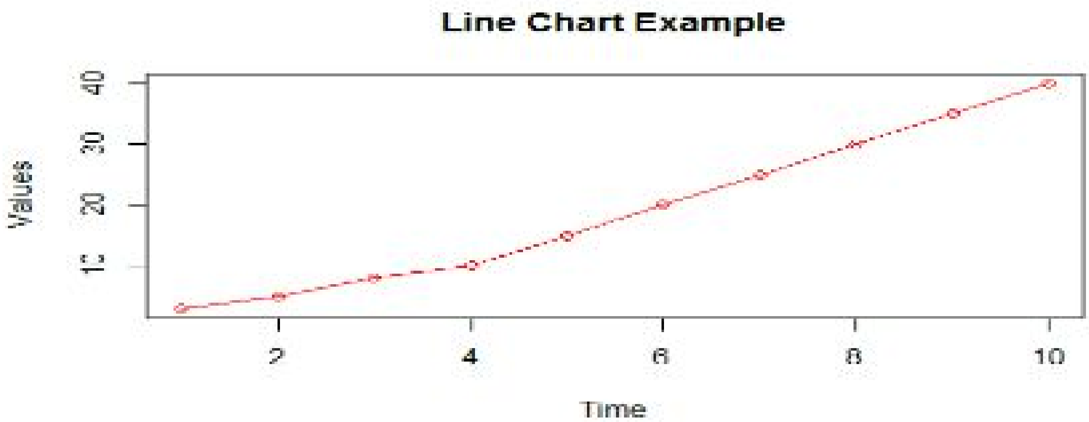
7.Write a R program for any visual representation of an object with creating graphs usinggraphic functions:Hist(),Linechart(),Pie().

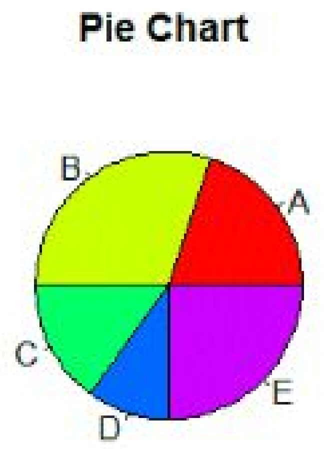
```
# Sample data

data <- c(10, 15, 20, 25, 30, 35, 40, 45, 50, 55)

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


# Create a histogram

hist(data, col = "green", main = "Histogram ", xlab = "Values", ylab =
"Frequency")


# Create a line chart

time <- 1:10

values <- c(3, 5, 8, 10, 15, 20, 25, 30, 35, 40)
```

```
plot(time, values, type = "o", col = "red", xlab = "Time", ylab = "Values",
main = "Line Chart ")


# Create a pie chart

percentages <- c(20, 30, 15, 10, 25)

pie(percentages, labels = categories, col = rainbow(length(categories)),
main = "Pie Chart ")
```

OUTPUT:







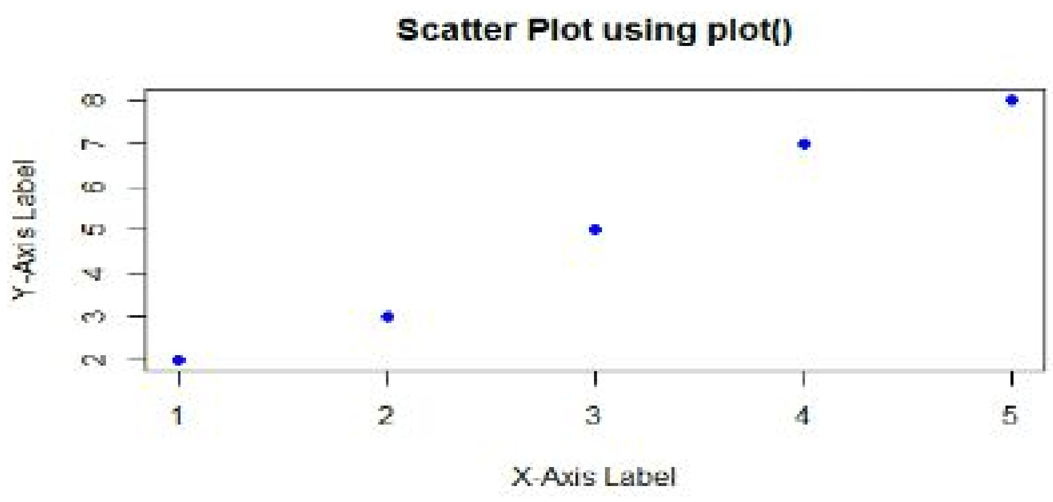
8.Write a R program for any visual representation of an object with creating graphs using graphic functions: Plot(),Boxplot(),Scatterplots().

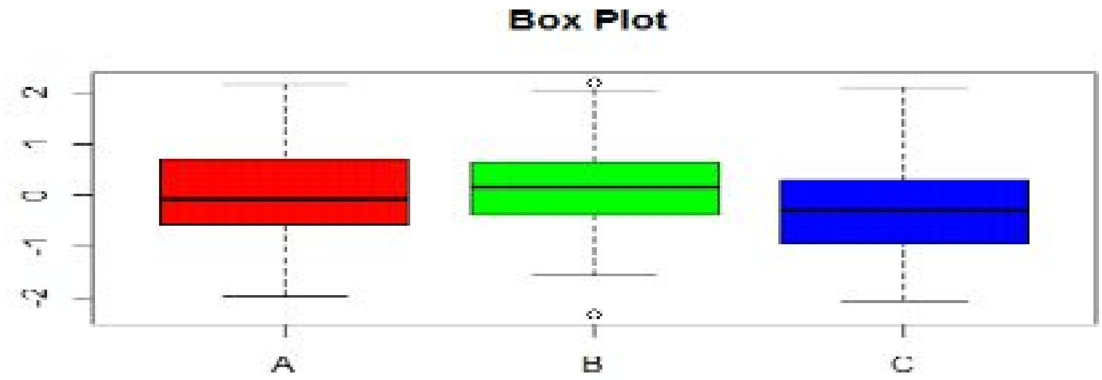
```
x <- c(1, 2, 3, 4, 5)

y <- c(2, 3, 5, 7, 8)

# Create a scatter plot using plot()

plot(x, y, main = "Scatter Plot using plot()", xlab = "X-Axis Label", ylab =
"Y-Axis Label", col = "blue", pch = 16)

# Create a boxplot

set.seed(123)

db<- list(A = rnorm(50), B = rnorm(50), C = rnorm(50))

boxplot(db, col = rainbow(length(db)), main = "Box Plot ")
```

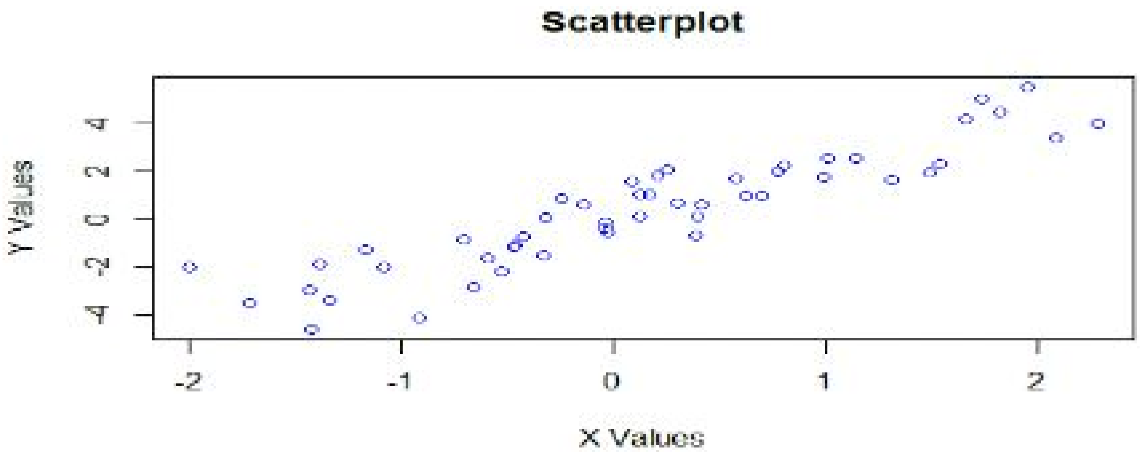
Create scatterplots

```
set.seed(456)

x <- rnorm(50)

y <- 2 * x + rnorm(50)

plot(x, y, col = "blue", xlab = "X Values", ylab = "Y Values", main = "Scatterplot")
```


OUTPUT:







9.Write a R program for with any dataset containing data frame objects, indexing and subsetting data frames, and employ manipulating and analyzing data.

```
# Create a sample dataset as a data frame

data <- data.frame(

  Stdid = c(1, 2, 3, 4, 5),

  Name = c("Alice","Bob", "Charlie", "David", "Eve"),

  Age = c(25, 30, 22, 28, 24),

  Score = c(95, 87, 75, 92, 88)
```

```
)

# Print the entire data frame

cat("Original Data Frame:₩n")

print(data)

# Indexing and subsetting data frames

# Select rows where Age is greater than 25

subset<- data[data$Age> 25, ]

cat("₩nSubset of Data Frame (Age > 25):₩n")

print(subset)

# Select specific columns (Name and Score)

selectcol<- data[, c("Name", "Score")]

cat("₩nSelectedColumns (Name and Score):₩n")

print(selectcol)

# Calculate summary statistics

sumstat<- summary(data$Score)

cat("₩nSummary Statistics for Score:₩n")

cat(sumstat, "₩n")

# Calculate the mean and standard deviation of Age

meanage<- mean(data$Age)

devage<- sd(data$Age)

cat("₩nMean Age:", meanage, "₩n")

cat("Standard Deviation of Age:", devage, "₩n")
```

```
# Calculate the correlation between Age and Score

corre <- cor(data$Age, data$Score)

cat("₩nCorrelation between Age and Score:", corre, "₩n")
```

OUTPUT:

```
Original Data Frame:

  Stdid    Name Age  Score
    1   Alice  25     95
    2     Bob  30     87
    3  Charlie  22     75
    4   David  28     92
    5     Eve  24     88

Subset of Data Frame (Age > 25):

  Stdid   Name  Age  Score
    2    Bob    30    87
    4   David   28    92
```

```
Selected Columns (Name and Score):

     Name  Score
1   Alice    95
2     Bob    87
3 Charlie    75
4   David    92
5     Eve    88


Summary Statistics for Score:

75 87 88 87.4 92 95


Mean Age: 25.8

Standard Deviation of Age: 3.193744
```

```
Correlation between Age and Score: 0.4961934
```

10.Write a program to create an any application of Linear Regression in multivariate context forpredictive purpose.

```
# Sample data for multivariate linear regression
set.seed(123) # Setting seed for reproducibility
num_samples <- 100
square_footage <- runif(num_samples,min = 800, max = 3000)
num_bedrooms <- sample(2:5, num_samples, replace = TRUE)
num_bathrooms <- sample(1:3, num_samples,replace = TRUE)
house_prices <- 50000 + 250 * square_footage + 15000 *
num_bedrooms + 10000 * num_bathrooms + rnorm(num_samples,
mean = 0, sd = 20000)
# Creating a data frame with the sample data
data <- data.frame(SquareFootage = square_footage, Bedrooms =
num_bedrooms, Bathrooms = num_bathrooms, Price = house_prices)
# Perform multivariate linear regression
model <- lm(Price ~ SquareFootage + Bedrooms + Bathrooms, data =
data)

# Summary of the regression model
summary(model)
```