Programs List:

1) Write a R-Program for to compute mean, median, minimum, maximum, variance, standard deviation, skewness and quantities (Q1, Q2, Q3)
2) Write an R Program that includes variables, constants, and data types.
3) Write an R Program that include different operators, control structures, default values for arguments, returning complex objects.
4) Write a R Program for calculating cumulative sums and products, minima, maxima and calculus.
5) Write a R-Program for finding stationary distribution of markov chains.
6) Write a R Program that include linear algebra operations on vectors and matrices.
7) Write a R Program for any visual representation of an object with creating graphs using graphic functions: Hist(), Linechart(), Pie()
8) Write a R Program for any visual representation of an object with creating graphs using graphic functions: Plot(), Box plot(), Scatter plots()
9) Write R Program for any dataset containing data frame objects, indexing and subs setting data frames and employ manipulating and analyzing data.
10) Write a Program to create any application of linear regression in multivariate context for predictive purpose.

Lab-1:

Write a R-Program for to compute mean, median, minimum, maximum, variance, standard deviation, skewness and quantities (Q1, Q2, Q3)

```
data<-c(12,25,36,45,21,67,43,18,50,30)   #sample data
mvalue<-mean(data)                       #Computemean
mevalue<-median(data)                    #Computemedian
minvalue<-min(data)                      #Computeminimumandmaximum
maxvalue<-max(data)
#Compute variance and standard deviation
varvalue<-var(data)
sdvalue<-sd(data)
#Compute skewness and kurtosis
skvalue<-skewness(data)                  #install moments package for this
kurtvalue<-kurtosis(data)
#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("StandardDeviation:",sdvalue,"\n")
cat("Skewness:",skvalue,"\n")
cat("Kurtosis:",kurtvalue,"\n")
cat("Q1:",q1,"\n")
cat("Q2(Median):",q2,"\n")
cat("Q3:",q3,"\n")
```

Lab-2:

Write an R Program that includes variables, constants, and 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 datatypes
cat("\n Data 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")
```

Lab-3:

Write an 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)
}
calculate_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")
#Controlstructure-if-else
if(area_result>10)
{
  cat("Thisisalargearea.\n")
}else
{
  cat("Thisisasmallarea.\n")
}
#Controlstructure-forloop
cat("\nCountingfrom 1to 5:\n")
for(i in 1:5)
{
  cat(i,"")
}
cat("\n\n")
#Using the function to create a person object
person1 <- create_person("Virat", 30, "India")
person2 <- create_person("Max Well",36,"Australia")
```

```
cat("Person1:\n")
cat("Name:",person1$Name, "\n")
cat("Age:",person1$Age,"\n")
cat("City:",person1$City,"\n")
cat("\nPerson2:\n")
cat("Name:",person2$Name, "\n")
cat("Age:",person2$Age,"\n")
cat("City:",person2$City,"\n")
```

Lab-4:

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

```
#Sample data set
data<-c(3,1,4,1,5,9,2,6,5,3)
#Cumulative sum
cumulative_sum <- cumsum(data)
cat("CumulativeSum:\n")
cat(cumulative_sum,"\n\n")
#Cumulative product
cumulative_product <- cumprod(data)
cat("CumulativeProduct:\n")
cat(cumulative_product,"\n\n")
#Minimum and Maximum
min_value<- min(data)
max_value<- max(data)
```

```
cat("MinimumValue:",min_value,"\n")
cat("MaximumValue:",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")
```

Lab-5:

Write a R-Program for finding stationary distribution of markov chains

```
library (Matrix)
# Transition probability matrix of the Markov chain
P <- matrix(c(0.4, 0.6, 0.2, 0.8), nrow = 2, byrow = TRUE)
print("probability matrix P is")
print(P)
# Check if the matrix P is stochastic (i.e., rows sum to 1)
if (all(rowSums(P) == 1)) {
  eigen_result <- eigen(t(P)) # Transpose P and find eigenvalues/vectors
  eigenvalues <- eigen_result$values
  eigenvectors <- eigen_result$vectors
```

```
  # Find the index of the eigenvalue that is closest to 1 (within a tolerance)
  tol <- 1e-6
  stationary_index <- which(abs(eigenvalues - 1) < tol)
  if (length(stationary_index) == 0) {
    cat("No unique stationary distribution found.\n")
  } else {
    stationary_distribution <- eigenvectors[, stationary_index] / sum(eigenvectors[,
stationary_index])
    cat("Stationary Distribution:", stationary_distribution, "\n")
  }
} else {
  cat("The transition matrix is not stochastic (rows should sum to 1).\n")
}
```

Lab-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)
#Vector algebra
sum<- vec1 + vec2
diff <- vec1 - vec2
prod<- sum(vec1 * vec2)
```

```
#print result of vector algebra
cat("Vector Addition(vec1 + vec2):\n")
cat(sum, "\n\n")
cat("Vector Sub traction (vec1 - vec2):\n")
cat(diff , "\n\n")
cat("Vector Dot Product:\n")
cat(prod, "\n\n")
#Create matrices
mat1 <- matrix(1:6, nrow = 2)
mat2 <- matrix(7:12, nrow = 2)
#Matrix algebra
msum<- mat1 + mat2
msub<- mat1-mat2
mmul<- mat1 * mat2
mdiv<- mat1 / mat2
#print result of Matrix algebra
cat("Matrix Addition (mat1 + mat2):\ n")
print(msum)
cat("Matrix Subtraction (mat1 - mat2):\ n")
print(msub)
cat("Matrix multiplication (mat1 * mat2):\ n")
print(mmul)
cat("Matrix division (mat1 / mat2):\ n")
print(mdiv)
```

Lab-7:

Write a R Program for any visual representation of an object with creating graphs using graphic 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")
```

Lab-8:

Write a R Program for any visual representation of an object with creating graphs using graphic functions: Plot(), Box plot(), Scatter plots()

```
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-AxisLabel",ylab="Y-AxisLabel",col="blue",pch=16)
# Create a box plot
#seed(arguments) is a random number generator
set.seed(123)
db<-list(A=rnorm(50),B=rnorm(50),C=rnorm(50))
boxplot(db,col=rainbow(length(db)),main="BoxPlot")
#Create scatter plots
set.seed(456)
x<-rnorm(50)
y<-2*x+rnorm(50)
plot(x,y,col="blue",xlab="XValues",ylab="YValues",main="Scatterplot")
```

Lab-9:

Write R Program for any dataset containing data frame objects, indexing and sub setting data frames and employ manipulating and analyzing data.

```
#Create a sample data set as a data frame
data <-data.frame(Stdid=c(1,2,3,4,5),
Name= c("aaa","bbb","ccc", "ddd","eee"), 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 sub setting data frames
#Select rows where Age is greater than 25
subset<-data[data$Age>25, ]
cat("\n Subset of Data Frame(Age>25):\n")
print(subset)
#Select specific columns (Name and Score)
selectcol<-data[ ,  c("Name", "Score")]
cat("\n Selected Columns (Name and Score):\n")
print(selectcol)
#Calculate summary statistics
sumstat<-summary(data$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("\n Mean Age:",meanage,"\n")
cat("Standard Deviation of Age:",devage,"\n")
#Calculate the correlation between Age and Score
corre<-cor(data$Age,data$Score)
cat("\n Correlation between Age and Score:",corre,"\n")
```

Lab-10:

Write a Program to create any application of linear regression in multivariate context for predictive purpose.

```
# Sample data for multivariate linear regression

# seed() is able to reproduce a particular sequence of random numbers

set.seed(123)

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)

data

# Perform multivariate linear regression

    model <- lm(Price ~ SquareFootage + Bedrooms + Bathrooms, data = data)

# Summary of the reg

    summary(model)
```

****