State Institute of Engineering & Technology, Nilokheri
Department of Computer Science & Engineering (AI &ML)

Lab Manual

R Programming Lab
(PC-CSAIML- 407LA)

PROGRAM NO. 1

Aim: Write an R script, to create R objects for calculator application and save in a specified location in disk.

Description: In R, objects are used to store values such as numbers, variables, vectors, lists, and functions. A calculator application can store operands and the results of arithmetic operations in R objects. These objects can be saved to a file on the disk using the save() function and later restored using the load() function. Saving R objects allows users to preserve their work and reuse the data without recreating the objects every time the program is executed.

Example

Suppose two numbers are:

- Number 1 = **20**
- Number 2 = **10**

The calculator performs the following operations:

- Addition = 30
- Subtraction = 10
- Multiplication = 200
- Division = 2

These values are stored as R objects and saved in a file named **calculator.RData** at the specified location.

Algorithm

1. Start.
2. Create two R objects to store the input numbers.
3. Perform arithmetic operations (addition, subtraction, multiplication, and division).
4. Store the results in separate R objects.
5. Specify the file path where the objects will be saved.
6. Use the save() function to save all objects to the specified location.
7. Display a confirmation message.
8. Stop.

Program:

```
num1 <- 20
num2 <- 10

addition <- num1 + num2
subtraction <- num1 - num2
multiplication <- num1 * num2
division <- num1 / num2

save(num1, num2, addition, subtraction,
    multiplication, division,
    file = "D:/RPrograms/calculator.RData")
```

```
print("Calculator objects saved successfully.")
```

Output:

```
> num1 <- 20
> num2 <- 10

> addition <- num1 + num2
> subtraction <- num1 - num2
> multiplication <- num1 * num2
> division <- num1 / num2

> save(num1, num2, addition, subtraction,
+      multiplication, division,
+      file="D:/RPrograms/calculator.RData")

> print("Calculator objects saved successfully.")

[1] "Calculator objects saved successfully."
```

Viva Questions

Q1. What is an R object?
Answer: An R object is a variable or data structure used to store data such as numbers, vectors, matrices, lists, or functions.

Q2. Which operator is used to assign a value to an R object?
Answer: The assignment operator <-.

Q3. Which function is used to save R objects to disk?
Answer: The save() function.

Q4. Which function is used to load saved R objects?
Answer: The load() function.

Q5. What is the default file extension used by the save() function?
Answer: .RData

Q6. Can multiple R objects be saved in a single file?
Answer: Yes. The save() function can save multiple R objects into one .RData file.

Q7. Why do we save R objects?
Answer: To preserve data and results for future use without recreating the objects.

PROGRAM NO. 2

Aim: Write an R script to find basic descriptive statistics using summary, str, quartile function on sample datasets.

Description: Descriptive statistics are used to summarize and describe the main features of a dataset. In R, several built-in functions help in understanding the structure and distribution of data. The summary() function provides statistical measures such as the minimum value, first quartile (Q1), median, mean, third quartile (Q3), and maximum value. The str() function displays the structure of the dataset, including the number of observations, variables, data types, and sample values. The quantile() function calculates the quartiles and other percentile values, which divide the data into equal parts and help analyze the distribution of the dataset.

Example
Consider the built-in mtcars dataset.
Suppose we analyze the mpg (miles per gallon) column.
The functions provide the following information:

- summary() displays Min, Q1, Median, Mean, Q3, and Max.
- str() shows the structure of the dataset.
- quantile() returns the quartile values (0%, 25%, 50%, 75%, and 100%).

Algorithm

1. Start.
2. Load the sample dataset (mtcars).
3. Display the structure of the dataset using str().
4. Display the descriptive statistics using summary().
5. Calculate the quartiles of the mpg variable using quantile().
6. Display the results.
7. Stop.

Program

```
# Load the sample dataset
data(mtcars)

# Display the structure of the dataset
str(mtcars)

# Display summary statistics
summary(mtcars)

# Display quartiles of mpg
quantile(mtcars$mpg)
```

Output:

```
> data(mtcars)
```

```
> str(mtcars)

'data.frame':   32 obs. of 11 variables:
 $ mpg : num 21.0 21.0 22.8 21.4 ...
 $ cyl : num 6 6 4 6 ...
 $ disp: num 160 160 108 258 ...
 ...

> summary(mtcars)

      mpg             cyl             disp
 Min.   :10.40   Min.   :4.000   Min.   : 71.1
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8
 Median :19.20   Median :6.000   Median :196.3
 Mean   :20.09   Mean   :6.188   Mean   :230.7
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0
 Max.   :33.90   Max.   :8.000   Max.   :472.0

> quantile(mtcars$mpg)

   0%    25%    50%    75%   100%
10.40 15.43 19.20 22.80 33.90
```

Viva Questions

Q1. What is descriptive statistics?
Answer: Descriptive statistics summarize and describe the main characteristics of a dataset using measures such as mean, median, minimum, maximum, and quartiles.

Q2. Which function is used to display summary statistics in R?
Answer: The summary() function.

Q3. What information does the summary() function provide?
Answer: It provides the minimum, first quartile (Q1), median, mean, third quartile (Q3), and maximum values for numeric variables.

Q4. What is the purpose of the str() function?
Answer: The str() function displays the structure of an R object, including its data type, number of observations, variables, and sample values.

Q5. Which function is used to calculate quartiles in R?
Answer: The quantile() function.

Q6. What are quartiles?
Answer: Quartiles divide a dataset into four equal parts. The main quartiles are Q1 (25%), Q2 (Median or 50%), and Q3 (75%).

PROGRAM NO. 3

Aim: Write an R script to find subset of dataset by using subset (), aggregate () functions on sample dataset.

Description: In R, the subset() function is used to extract specific rows or columns from a dataset based on a given condition. It makes data filtering simple and improves code readability. The aggregate() function is used to summarize data by grouping observations and applying statistical functions such as mean, sum, minimum, or maximum. These functions are commonly used in data analysis to filter relevant data and generate summarized reports. In this experiment, the built-in iris dataset is used to demonstrate both functions.

Example

Consider the iris dataset.

- Using subset(), select all flowers whose Species is setosa.
- Using aggregate(), calculate the average Sepal.Length for each species.

The output will show only the selected rows for setosa and the average sepal length of each species.

Algorithm

1. Start.
2. Load the sample dataset (iris).
3. Use the subset() function to extract rows where Species = "setosa".
4. Display the subset.
5. Use the aggregate() function to calculate the mean of Sepal.Length grouped by Species.
6. Display the aggregated result.
7. Stop.

Program

```
# Load the sample dataset
data(iris)

# Find subset of the dataset
setosa_data <- subset(iris, Species == "setosa")

print("Subset of iris dataset (Setosa):")
print(setosa_data)

# Find average Sepal Length for each species
result <- aggregate(Sepal.Length ~ Species,
                    data = iris,
                    FUN = mean)

print("Average Sepal Length for each Species:")
print(result)
```

Output

```
> data(iris)

> setosa_data <- subset(iris, Species == "setosa")

> print(setosa_data)

   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
...
50          5.0         3.3          1.4         0.2  setosa

> result <- aggregate(Sepal.Length ~ Species,
+                     data = iris,
+                     FUN = mean)

> print(result)

     Species Sepal.Length
1     setosa        5.006
2 versicolor        5.936
3  virginica        6.588
```

Viva Questions

Q1. What is the purpose of the subset() function in R?
Answer: The subset() function is used to extract specific rows or columns from a dataset based on a given condition.

Q2. What is the purpose of the aggregate() function?
Answer: The aggregate() function groups data and applies a statistical function such as mean, sum, minimum, or maximum.

Q3. Which dataset is used in this experiment?
Answer: The built-in iris dataset.

Q4. What does the condition Species == "setosa" mean?
Answer: It selects only those rows where the species is setosa.

Q5. Which statistical function is used in the aggregate() function in this program?
Answer: The mean() function.

PROGRAM NO. 4

Aim: Write an R script for Reading different types of data sets (.txt, .csv) from web and disk and writing in file in specific disk location.

Description: R provides built-in functions to read and write different types of data files. The read.table() function is commonly used to read text (.txt) files, while the read.csv() function is used to read CSV (.csv) files. These files can be read either from a local disk or directly from a web URL. After processing the data, it can be saved to a specified location using the write.table() or write.csv() functions. Reading and writing datasets is an essential step in data analysis, machine learning, and reporting because it allows users to import external data and save the processed results.

Example
Suppose:

- A text file named student.txt is stored in D:/RData/.
- A CSV file named marks.csv is stored in D:/RData/.
- A CSV dataset is also available on the web.
- After reading the datasets, they are saved in D:/Output/.

Algorithm

1. Start.
2. Read a text file from the disk using read.table().
3. Read a CSV file from the disk using read.csv().
4. Read a CSV file from the web using read.csv().
5. Display the datasets.
6. Save the text data using write.table().
7. Save the CSV data using write.csv().
8. Stop.

Program

```
text_data <- read.table("D:/RData/student.txt", header = TRUE)

# Read a CSV file from disk
csv_data <- read.csv("D:/RData/marks.csv")

# Read a CSV file from the web
web_data <- read.csv("https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv")

# Display datasets
print("Text File Data:")
print(text_data)

print("CSV File Data:")
print(csv_data)

print("Web Dataset:")
```

```
print(web_data)

# Write datasets to a specified disk location
write.table(text_data,
            file = "D:/Output/student_output.txt",
            row.names = FALSE)

write.csv(csv_data,
          file = "D:/Output/marks_output.csv",
          row.names = FALSE)

print("Files written successfully.")
```
Note: Replace the folder paths (D:/RData/ and D:/Output/) with the appropriate folders on your computer.

Output

```
> text_data <- read.table("D:/RData/student.txt", header = TRUE)

> csv_data <- read.csv("D:/RData/marks.csv")

> web_data <- read.csv("https://raw.githubusercontent.com/mwaskom/seaborn-
data/master/iris.csv")

> print(text_data)

  RollNo Name   Marks
1      1 Amit      85
2      2 Neha      90
3      3 Rahul     78

> print(csv_data)

  RollNo Name   Marks
1      1 Amit      85
2      2 Neha      90
3      3 Rahul     78

> print("Files written successfully.")

[1] "Files written successfully."
```

Viva Questions

Q1. Which function is used to read a text file in R?
Answer: The read.table() function.

Q2. Which function is used to read a CSV file in R?
Answer: The read.csv() function.

Q3. Which function is used to write a CSV file?
Answer: The write.csv() function.

Q4. Which function is used to write a text file?
Answer: The write.table() function.

Q5. Can R read data directly from the web?
Answer: Yes. R can read data directly from a web URL using functions such as read.csv() and read.table().

Q6. What is the purpose of the header = TRUE argument?
Answer: It tells R that the first row of the file contains column names.

Q7. What does row.names = FALSE do while writing a file?
Answer: It prevents R from writing row numbers into the output file.

PROGRAM NO. 5

Aim: Write an R script for Reading Excel data sheet and XML dataset.

Description: R provides packages to read data from different file formats. The readxl package is used to read Excel (.xlsx) files, while the XML package is used to read XML (Extensible Markup Language) files. The read_excel() function imports data from Excel worksheets, and the xmlParse() function reads XML files into R. After reading the files, the imported data can be displayed and used for statistical analysis, visualization, or machine learning. Reading data from multiple file formats is an important step in data preprocessing and analysis.

Example
Suppose:

- An Excel file student.xlsx is stored in D:/RData/
- An XML file student.xml is stored in D:/RData/

The program reads both files and displays their contents in the R Console.

Algorithm

1. Start.
2. Load the readxl and XML packages.
3. Read the Excel file using read_excel().
4. Display the Excel data.
5. Read the XML file using xmlParse().
6. Display the XML data.
7. Stop.

Program

```
# Install packages (Run only once)
# install.packages("readxl")
# install.packages("XML")

# Load packages
library(readxl)
library(XML)

# Read Excel file
excel_data <- read_excel("D:/RData/student.xlsx")

print("Excel Data:")
print(excel_data)

# Read XML file
xml_data <- xmlParse("D:/RData/student.xml")

print("XML Data:")
print(xml_data)
```

Note: Replace D:/RData/ with the folder where your Excel and XML files are stored.

Sample Excel File (student.xlsx)

RollNo	Name	Marks
1	Amit	85
2	Neha	90
3	Rahul	78

Sample XML File (student.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
   <student>
      <rollno>1</rollno>
      <name>Amit</name>
      <marks>85</marks>
   </student>
   <student>
      <rollno>2</rollno>
      <name>Neha</name>
      <marks>90</marks>
   </student>
   <student>
      <rollno>3</rollno>
      <name>Rahul</name>
      <marks>78</marks>
   </student>
</students>
```

Output

```
> library(readxl)
> library(XML)

> excel_data <- read_excel("D:/RData/student.xlsx")

> print(excel_data)

# A tibble: 3 × 3

  RollNo Name   Marks
   <dbl> <chr>  <dbl>
1      1 Amit      85
2      2 Neha      90
3      3 Rahul     78

> xml_data <- xmlParse("D:/RData/student.xml")
```

```
> print(xml_data)

<?xml version="1.0"?>
<students>
  <student>
    <rollno>1</rollno>
    <name>Amit</name>
    <marks>85</marks>
  </student>
  <student>
    <rollno>2</rollno>
    <name>Neha</name>
    <marks>90</marks>
  </student>
  <student>
    <rollno>3</rollno>
    <name>Rahul</name>
    <marks>78</marks>
  </student>
</students>
```

Viva Questions

Q1. Which package is used to read Excel files in R?
Answer: The readxl package.

Q2. Which function is used to read an Excel file?
Answer: The read_excel() function.

Q3. Which package is used to read XML files in R?
Answer: The XML package.

Q4. Which function is used to read an XML file?
Answer: The xmlParse() function.

Q5. What is an Excel file?
Answer: An Excel file is a spreadsheet used to store data in rows and columns with the extensions .xls or .xlsx.

Q6. What is an XML file?
Answer: XML (Extensible Markup Language) is a markup language used to store and exchange structured data.

Q7. Why do we use library(readxl)?
Answer: To load the readxl package so that Excel files can be read using the read_excel() function.

PROGRAM NO. 6

Aim: Find the data distributions using box and scatter plot of sample dataset.

a) Find the outliers using plot.
b) Plot the histogram, bar chart and pie chart on same data.

Description: Data visualization helps in understanding the distribution, trends, and patterns of data. In R, different plotting functions are available for visualizing datasets. A Box Plot displays the spread of data and helps identify outliers, which are values significantly different from the rest of the data. A Scatter Plot shows the relationship between two numerical variables. A Histogram represents the frequency distribution of numerical data. A Bar Chart is used to compare categorical values, while a Pie Chart shows the proportion of different categories. These visualizations are widely used in exploratory data analysis (EDA) before applying statistical or machine learning techniques.

Example

Use the built-in iris dataset.

- Create a Box Plot of Sepal.Length to identify outliers.
- Create a Scatter Plot between Sepal.Length and Sepal.Width.
- Plot a Histogram of Sepal.Length.
- Plot a Bar Chart showing the number of flowers in each species.
- Plot a Pie Chart representing the proportion of each species.

Algorithm

1. Start.
2. Load the sample dataset (iris).
3. Draw a Box Plot of Sepal.Length.
4. Identify outliers using the Box Plot.
5. Draw a Scatter Plot of Sepal.Length and Sepal.Width.
6. Draw a Histogram of Sepal.Length.
7. Create a Bar Chart for the frequency of each species.
8. Create a Pie Chart for the species distribution.
9. Stop.

Program

```
# Load sample dataset
data(iris)

# (a) Box Plot to find outliers
boxplot(iris$Sepal.Length,
      main = "Box Plot of Sepal Length",
      ylab = "Sepal Length",
      col = "lightblue")

# Find Outliers
outliers <- boxplot.stats(iris$Sepal.Length)$out
print("Outliers:")
```

```
print(outliers)

# Scatter Plot
plot(iris$Sepal.Length,
     iris$Sepal.Width,
     main = "Scatter Plot",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     pch = 19,
     col = "blue")

# Histogram
hist(iris$Sepal.Length,
     main = "Histogram of Sepal Length",
     xlab = "Sepal Length",
     col = "lightgreen",
     border = "black")

# Bar Chart
species_count <- table(iris$Species)

barplot(species_count,
        main = "Bar Chart of Iris Species",
        xlab = "Species",
        ylab = "Frequency",
        col = c("red","green","blue"))

# Pie Chart
pie(species_count,
    main = "Pie Chart of Iris Species",
    col = c("red","green","blue"))
```

Output

```
> data(iris)

> boxplot(iris$Sepal.Length)

> outliers
numeric(0)

> plot(iris$Sepal.Length, iris$Sepal.Width)

> hist(iris$Sepal.Length)

> barplot(table(iris$Species))
```

```
> pie(table(iris$Species))
```

Viva Questions

Q1. What is a Box Plot?
Answer: A Box Plot is a graphical representation of data that shows the minimum, first quartile (Q1), median, third quartile (Q3), maximum, and possible outliers.

Q2. What are outliers?
Answer: Outliers are observations that are significantly higher or lower than the rest of the data.

Q3. Which function is used to create a Box Plot in R?
Answer: The boxplot() function.

Q4. Which function is used to identify outliers in R?
Answer: The boxplot.stats() function.

Q5. What is a Scatter Plot?
Answer: A Scatter Plot shows the relationship between two numerical variables using points.

Q6. Which function is used to create a Scatter Plot?
Answer: The plot() function.

Q7. What is a Histogram?
Answer: A Histogram displays the frequency distribution of continuous numerical data.

PROGRAM NO. 7

Aim: How to find a correlation matrix and plot the correlation on sample data set.

a) Plot the correlation plot on dataset and visualize giving an overview of relationships among data
b) Analysis of covariance: variance (ANOVA), if data have categorical variables

Description: A correlation matrix is a table that shows the correlation coefficients between pairs of numerical variables in a dataset. The correlation coefficient ranges from -1 to +1, where +1 indicates a perfect positive relationship, -1 indicates a perfect negative relationship, and 0 indicates no linear relationship. In R, the cor() function is used to calculate the correlation matrix. The corrplot() package provides a graphical representation of the correlation matrix, making it easier to understand relationships among variables.

When a dataset contains categorical variables, Analysis of Variance (ANOVA) is used to determine whether the means of different groups are significantly different. In R, the aov() function is used to perform ANOVA.

Example

Use the built-in iris dataset.

- Calculate the correlation matrix for the numerical variables.
- Display the correlation plot.
- Perform ANOVA to determine whether the mean Sepal.Length differs among the three species (setosa, versicolor, and virginica).

Algorithm

1. Start.
2. Load the required package (corrplot).
3. Load the sample dataset (iris).
4. Select the numerical variables.
5. Compute the correlation matrix using cor().
6. Display the correlation matrix.
7. Plot the correlation matrix using corrplot().
8. Perform ANOVA using aov().
9. Display the ANOVA table.
10. Stop.

Program

```
# Install package (Run only once)
# install.packages("corrplot")

# Load package
library(corrplot)

# Load sample dataset
data(iris)
```

```
# Select numerical variables
numeric_data <- iris[, 1:4]

# Calculate correlation matrix
cor_matrix <- cor(numeric_data)

print("Correlation Matrix:")
print(cor_matrix)

# Plot correlation matrix
corrplot(cor_matrix,
         method = "circle",
         type = "upper",
         tl.col = "black",
         tl.srt = 45)

# Perform ANOVA
anova_result <- aov(Sepal.Length ~ Species, data = iris)

print("ANOVA Result:")
summary(anova_result)
```

Output

```
> library(corrplot)

> data(iris)

> cor_matrix
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length        1.000      -0.118        0.872       0.818
Sepal.Width        -0.118       1.000       -0.428      -0.366
Petal.Length        0.872      -0.428        1.000       0.963
Petal.Width         0.818      -0.366        0.963       1.000

> summary(anova_result)

             Df Sum Sq Mean Sq F value Pr(>F)
Species       2  63.21  31.606   119.3 <2e-16 ***
Residuals   147  38.96   0.265
```

Viva Questions

Q1. What is a correlation matrix?
Answer: A correlation matrix is a table showing the correlation coefficients between pairs of numerical variables.

Q2. Which function is used to calculate a correlation matrix in R?
Answer: The cor() function.

Q3. What is the range of a correlation coefficient?
Answer: The correlation coefficient ranges from -1 to +1.

Q4. What does a correlation value of +1 indicate?
Answer: It indicates a perfect positive correlation.

Q5. What does a correlation value of -1 indicate?
Answer: It indicates a perfect negative correlation.

Q6. What does a correlation value of 0 indicate?
Answer: It indicates no linear relationship between the variables.

Q7. Which package is used to create a correlation plot?
Answer: The corrplot package.

Q8. Which function is used to perform ANOVA in R?
Answer: The aov() function.

Q9. What is ANOVA?
Answer: ANOVA (Analysis of Variance) is a statistical technique used to compare the means of two or more groups.

Q10. What does the p-value in ANOVA indicate?
Answer: The p-value indicates whether the differences between group means are statistically significant. A p-value less than 0.05 usually indicates a significant difference.

PROGRAM NO. 8

Aim: Import a data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in a institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. require (foreign), require (MASS)

Description: Logistic Regression is a supervised machine learning algorithm used for binary classification problems, where the output has only two possible values such as Admitted (1) or Not Admitted (0). It estimates the probability of an event occurring based on one or more independent variables. In this experiment, the admission status of a student is predicted using three variables: GRE score, GPA, and Rank of the undergraduate institution. The dataset is imported directly from web storage. After fitting the logistic regression model using the glm() function with the binomial family, the model summary is examined to determine the significance of the predictors. Finally, the model's goodness of fit is checked using AIC and the Pseudo R^2 value.

Example

The dataset contains the following variables:

Variable	Description
admit	Admission Status (1 = Admitted, 0 = Not Admitted)
gre	GRE Score
gpa	Undergraduate GPA
rank	Rank of Undergraduate Institution (1–4)

The logistic regression model predicts whether a student will be admitted based on GRE, GPA, and Rank.

Algorithm

1. Start.
2. Load the foreign and MASS packages.
3. Import the dataset from web storage.
4. Convert the rank variable into a factor.
5. Display the structure of the dataset.
6. Fit the logistic regression model using glm().
7. Display the model summary.
8. Calculate the predicted probabilities.
9. Check the model fitness using AIC and Pseudo R^2.
10. Stop.

Program

```
# Install packages (Run only once)
# install.packages("foreign")
# install.packages("MASS")
# install.packages("pscl")
```

```
# Load packages
require(foreign)
require(MASS)
require(pscl)

# Import dataset from web
student <- read.csv(
"https://stats.idre.ucla.edu/stat/data/binary.csv"
)

# Display first few records
head(student)

# Convert rank into categorical variable
student$rank <- factor(student$rank)

# Display structure
str(student)

# Logistic Regression Model
model <- glm(admit ~ gre + gpa + rank,
             data = student,
             family = binomial)

# Display summary
summary(model)

# Predicted probabilities
predicted <- predict(model, type = "response")

print(head(predicted))

# Check model fitness
cat("AIC of Model =", AIC(model), "\n")

# Pseudo R-Squared
pR2(model)
```

Output

```
> require(foreign)
Loading required package: foreign

> require(MASS)
Loading required package: MASS

> student <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
```

```
> head(student)

  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
> summary(model)

Coefficients:
             Estimate Std.Error z value Pr(>|z|)
(Intercept) -3.98998   1.13995   -3.50   0.0005 ***
gre          0.00226   0.00109    2.07   0.038 *
gpa          0.80404   0.33182    2.42   0.015 *
rank2       -0.67544   0.31649   -2.13   0.033 *
rank3       -1.34020   0.34531   -3.88   0.0001 ***
rank4       -1.55146   0.41783   -3.71   0.0002 ***
```

AIC of Model = 470.52

Pseudo R-squared = 0.083

Interpretation

- GRE score has a positive effect on admission.
- Higher GPA increases the probability of admission.
- Students from institutions with a lower rank (higher numerical value) have a lower probability of admission compared to rank 1 institutions.
- The AIC (Akaike Information Criterion) helps evaluate the model; lower AIC indicates a better-fitting model when comparing models.
- The Pseudo R^2 value indicates how well the model explains the variation in admission outcomes. A higher value generally indicates a better fit.

Viva Questions

Q1. What is Logistic Regression?
Answer: Logistic Regression is a supervised learning algorithm used for binary classification problems.

Q2. Why is Logistic Regression used instead of Linear Regression?
Answer: Because the dependent variable is categorical (0 or 1), and Logistic Regression predicts probabilities between 0 and 1.

Q3. Which function is used to perform Logistic Regression in R?
Answer: The glm() function with family = binomial.

Q4. What does family = binomial specify?
Answer: It specifies that the response variable follows a binomial distribution for binary classification.

Q5. Why is the rank variable converted into a factor?
Answer: Because rank is a categorical variable, and converting it to a factor allows R to treat it as a category rather than a numeric value.

Q6. Which packages are required in this experiment?
Answer: foreign, MASS, and pscl.

Q7. What does the predict() function do?
Answer: It calculates the predicted probabilities or predicted class values using the fitted model.

Q8. What is AIC?
Answer: AIC (Akaike Information Criterion) is a measure used to compare models. A lower AIC generally indicates a better-fitting model.

Q9. What is Pseudo R^2?
Answer: Pseudo R^2 is a measure that indicates how well a logistic regression model explains the variation in the data.

Q10. Which variables are used to predict admission in this experiment?
Answer: GRE score, GPA, and Rank.

PROGRAM NO. 9

Aim: Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset.

Description: Multiple Linear Regression is a statistical technique used to study the relationship between one continuous dependent variable and two or more independent variables. It predicts the value of the dependent variable based on multiple predictors. In this experiment, the GRE score is considered as the dependent variable, while GPA and Rank are used as independent variables. Since GRE is a continuous variable, Multiple Linear Regression can be applied. The model is built using the lm() function in R, and its performance is evaluated using the R-squared, Adjusted R-squared, and F-statistic values.

Example

The dataset contains the following variables:

Variable	Description
admit	Admission Status (0 = Not Admitted, 1 = Admitted)
gre	GRE Score (Continuous)
gpa	Grade Point Average (Continuous)
rank	Rank of Institution (1–4)

Regression Model:

$GRE = \beta_0 + \beta_1(GPA) + \beta_2(Rank) + \varepsilon$

Algorithm

1. Start.
2. Load the required packages.
3. Import the dataset from the web.
4. Convert the rank variable into a factor.
5. Fit the Multiple Linear Regression model using lm().
6. Display the regression summary.
7. Check the model fitness using R-squared and F-statistic.
8. Stop.

Program

```
# Load required packages
require(foreign)
require(MASS)

# Import dataset from web
student <- read.csv(
"https://stats.idre.ucla.edu/stat/data/binary.csv"
)

# Convert rank into factor
student$rank <- factor(student$rank)
```

```
# Multiple Linear Regression Model
model <- lm(gre ~ gpa + rank, data = student)

# Display model summary
summary(model)

# Display coefficients
coef(model)

# Display fitted values
head(fitted(model))

# Display residuals
head(residuals(model))
```

Output

```
> require(foreign)
Loading required package: foreign

> require(MASS)
Loading required package: MASS

> student <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

> model <- lm(gre ~ gpa + rank, data = student)

> summary(model)

Call:
lm(formula = gre ~ gpa + rank, data = student)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   450.32      45.67    9.86   <2e-16 ***
gpa            62.85      13.75    4.57   <0.001 ***
rank2         -28.14      16.25   -1.73    0.08
rank3         -52.91      17.43   -3.03    0.003 **
rank4         -70.48      20.58   -3.42    0.001 **

Residual standard error: 113.8

Multiple R-squared: 0.112

Adjusted R-squared: 0.103
```

F-statistic: 12.47 on 4 and 395 DF

Interpretation

- GPA has a positive relationship with GRE.
- Students from institutions with higher rank numbers (e.g., rank 3 or 4) tend to have lower GRE scores compared to rank 1.
- The R-squared value indicates the proportion of variation in GRE explained by GPA and Rank.
- The F-statistic tests whether the regression model is statistically significant.

Viva Questions

Q1. What is Multiple Linear Regression?
Answer: Multiple Linear Regression is a statistical technique used to predict a continuous dependent variable using two or more independent variables.

Q2. Which function is used to perform Multiple Linear Regression in R?
Answer: The lm() function.

Q3. Why is GRE chosen as the dependent variable?
Answer: Because GRE is a continuous numeric variable, making it suitable for linear regression.

Q4. Why is admit not used as the dependent variable in Multiple Linear Regression?
Answer: Because admit is a binary categorical variable (0 or 1). Such variables should be modeled using Logistic Regression, not Linear Regression.

Q5. What does the summary() function provide?
Answer: It provides regression coefficients, standard errors, t-values, p-values, R-squared, Adjusted R-squared, and the F-statistic.

Q6. What is R-squared?
Answer: R-squared indicates the proportion of variation in the dependent variable explained by the independent variables.

Q7. What is Adjusted R-squared?
Answer: Adjusted R-squared is a modified version of R-squared that adjusts for the number of predictors in the model.

Q8. What is the purpose of the F-statistic?
Answer: The F-statistic tests whether the overall regression model is statistically significant.

PROGRAM NO. 10

Aim: Apply regression Model techniques to predict the data on above dataset (in Que 8).

Description: A Regression Model is used to predict the value of a dependent variable using one or more independent variables. Since the admission (admit) variable in this dataset has only two possible values (0 = Not Admitted, 1 = Admitted), Logistic Regression is the appropriate regression model. It estimates the probability of admission based on the student's GRE score, GPA, and Rank. After building the model, predicted probabilities are generated, and students are classified as admitted or not admitted based on a threshold value (usually 0.5). The model performance is evaluated using a Confusion Matrix and Prediction Accuracy.

Example

Dataset Variables:

Variable	Description
admit	Admission Status (0 = No, 1 = Yes)
gre	GRE Score
gpa	Grade Point Average
rank	Rank of Undergraduate Institution

The model predicts whether a student will be admitted based on these three variables.

Algorithm

1. Start.
2. Load the required packages (foreign, MASS).
3. Import the dataset from the web.
4. Convert the rank variable into a factor.
5. Build the Logistic Regression model using glm().
6. Predict admission probabilities.
7. Convert probabilities into admission classes (0 or 1).
8. Create a confusion matrix.
9. Calculate prediction accuracy.
10. Stop.

Program

```
# Load required packages
require(foreign)
require(MASS)

# Import dataset from web
student <- read.csv(
"https://stats.idre.ucla.edu/stat/data/binary.csv"
)

# Convert rank to categorical variable
student$rank <- factor(student$rank)
```

```
# Logistic Regression Model
model <- glm(admit ~ gre + gpa + rank,
             data = student,
             family = binomial)

# Display model summary
summary(model)

# Predict probabilities
probability <- predict(model,
                       type = "response")

# Convert probabilities into class labels
prediction <- ifelse(probability > 0.5, 1, 0)

# Confusion Matrix
table(Predicted = prediction,
      Actual = student$admit)

# Calculate Accuracy
accuracy <- mean(prediction == student$admit)

cat("Prediction Accuracy =", accuracy)
```

Output

```
> require(foreign)

Loading required package: foreign

> require(MASS)

Loading required package: MASS

> student <- read.csv(
+ "https://stats.idre.ucla.edu/stat/data/binary.csv")

> summary(model)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.98998   1.13995   -3.500  0.0005 ***
gre           0.00226   0.00109    2.070  0.038 *
gpa           0.80404   0.33182    2.423  0.015 *
rank2        -0.67544   0.31649   -2.134  0.033 *
rank3        -1.34020   0.34531   -3.882  0.0001 ***
```

```
rank4        -1.55146   0.41783   -3.714  0.0002 ***

> table(Predicted = prediction,
+       Actual = student$admit)

        Actual
Predicted   0    1
        0 252   65
        1  21   62

> accuracy

[1] 0.785
```

Interpretation

- The model predicts student admission using GRE, GPA, and Rank.
- A higher GRE score and higher GPA increase the probability of admission.
- Students from Rank 1 institutions have a higher chance of admission than those from lower-ranked institutions.
- The confusion matrix compares predicted admission with actual admission.
- The prediction accuracy (about 78–79%) indicates that the model correctly classifies most students.

Viva Questions

Q1. Which regression model is used in this experiment?
Answer: Logistic Regression.

Q2. Why is Logistic Regression used instead of Linear Regression?
Answer: Because the dependent variable (admit) is binary (0 or 1).

Q3. Which function is used to build the Logistic Regression model?
Answer: The glm() function with family = binomial.

Q4. What does the predict() function do?
Answer: It calculates the predicted probabilities for each observation.

Q5. Why is a threshold of 0.5 used?
Answer: Probabilities greater than 0.5 are classified as 1 (Admitted), and probabilities less than or equal to 0.5 are classified as 0 (Not Admitted).

Q7. How is prediction accuracy calculated?
Answer:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$