

G. T. N. ARTS COLLEGE, (AUTONOMOUS)

A Telugu Linguistic Minority Institution | Affiliated to Madurai Kamaraj University

Re-accredited with 'A++' Grade by NAAC in the 2nd Cycle with CGPA 3.56

Recognized under DBT STAR COLLEGE SCHEME

Old Karur Road, Dindigul – 624 005.



DEPARTMENT OF MATHEMATICS

DBT STAR COLLEGE SCHEME

R PROGRAMMING - 20UMAS5P

LABORATORY MANUAL (2023 - 2024)

Supported by



**Department of
BioTechnology,
Government
of India**

सत्यमेव जयते

INDEX

Sl. No.	Program	Page No.
	Introduction	1
	Syntax	6
1.	Write a Program in R to create Vectors	8
2.	Write a Program in R to create a Matrices	10
3.	Write a Program in R to Calculate the Mean, Median and Mode of a set of observation	13
4.	Write a Program in R to Calculate standard deviation of a set of observation	14
5.	Write a Program in R test of significance using Chi-square test	15
6.	Write a program in R test of significance using Students t-test	17
7.	Write a program in R to calculate Spearman's Rank Correlation Coefficients	18
8..	Write a program in R to find regression coefficients and draw regression lines	19
9.	Write a program in R test of significance using F- test	21
10.	Write a program in R to Create Different charts for visualization of given set of given set of data	22

1. OBJECTIVES OF R PROGRAMMING LAB

The main objectives of the R Programming Lab are:

- To introduce students to the fundamentals of R programming language.
- To learn how to write and execute R scripts.
- To understand basic data structures such as vectors, matrices, lists, factors, and data frames.
- To perform statistical analysis including measures of central tendency, dispersion, hypothesis testing, correlation, and regression.
- To create and interpret data visualizations using bar charts, pie charts, histograms, and scatter plots.
- To use built-in mathematical functions for computation.
- To develop problem-solving skills in statistics using R.
- To cultivate analytical thinking for real-world data analysis.

2. INTRODUCTION TO R PROGRAMMING

R is a powerful, open-source programming language designed for statistical analysis, data visualization, and scientific computation. It provides an extensive environment that includes a wide range of statistical techniques—from simple statistics to advanced modeling.

R is widely used in:

- Data Science
- Machine Learning
- Artificial Intelligence
- Predictive Analytics
- Bioinformatics
- Finance
- Academic Research

3. HISTORY OF R

- R was developed in 1993 by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand.

- It was based on the **S language**, created at Bell Laboratories by John Chambers.
- R became an official GNU project and follows an open-source model.
- The first stable version, R 1.0.0, was released in 2000.
- The Comprehensive R Archive Network (CRAN) hosts more than 18,000 packages.
- Today, R is maintained by the R Foundation and supported by a global community of developers, analysts, and researchers.

4. WHY DO USE R?

R is preferred because:

- Free and Open Source – No license required.
- Statistical Strength – Designed for statistical modelling and analysis.
- Data Visualization – Produces high-quality graphics.
- Package Support – Thousands of libraries for every field.
- Cross-platform – Works on Windows, Linux, macOS.
- Community Support – Strong global community.
- Integration – Works with Python, C, C++, SQL, Hadoop, TensorFlow.
- Used Worldwide by statisticians, data scientists, and researchers.

5. STRUCTURE OF R PROGRAM

An R program typically contains:

- Comments

Code: `# This is a comment`

- Variables and Data Structures

Code: `x <- 10`

`y <- c(1,2,3)`

- Functions

Code: `mean(x)`

- Input / Output Statements

Code: `print("Hello")`

- Control Statements

Code: `if(x > 5) { print("Greater") }`

- Loops

Code: `for(i in 1:5) print(i)`

- Plots

Code: `plot(x, y)`

6. BASIC STEPS IN R

- Install R and R Studio.
- Open R console or R Studio IDE.
- Write an R script (.R file).
- Use <- to assign values.
- Create vectors, matrices, lists, or data frames.
- Use built-in functions (mean, sd, plot, print).
- Execute commands line-by-line or run entire script.
- View results in the Output or Plots window.

7. HOW TO INSTALL R

Step 1: Download R

- Visit: <https://cran.r-project.org>
- Choose your operating system (Windows / Linux / macOS)
- Download the latest version of R.

Step 2: Install R

- Double-click the downloaded installer.
- Follow standard installation steps.
- Select default settings.

Step 3: Install RStudio (Optional but Recommended)

- Visit <https://rstudio.com>
- Download R Studio Desktop (Free).
- Install and open R Studio.
- R Studio provides a user-friendly interface.

8. INPUT AND OUTPUT FUNCTIONS IN R

Input Methods

- `readline()` – For simple text input
Code: `name <- readline("Enter your name: ")`
- `scan()` – For numeric input
Code: `numbers <- scan()`
- `read.csv()` – For reading CSV files
Code: `data <- read.csv("file.csv")`
- Output Methods
 - `print()`
Code: `print("Hello R")`
 - `cat()`
 - Code: `cat("The value is", 10)`
 - `paste()`
 - Code: `paste("Sum =", 10 + 20)`

9. INSTALL RSTUDIO:

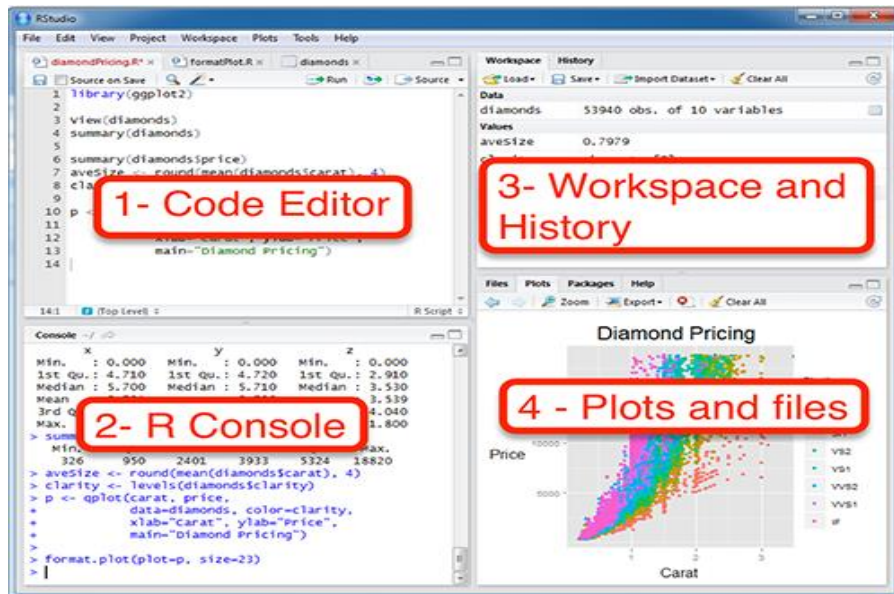
Go to: <https://posit.co/download/rstudio-desktop/>

Install the free desktop version.

RStudio provides:

- Script Editor
- Console
- Environment viewer

- Plot window
- Package manager



10. ALGORITHM TO RUN AN R PROGRAM

- Step 1: Start R or RStudio
- Step 2: Create a new script file
- Step 3: Type the R code
- Step 4: Save the program
- Step 5: Run the cript using:
 - Run Button
 - Or press Ctrl+Enter
- Step 6: Check output in the console.

11. MATHEMATICAL FUNCTIONS IN R

- Basic Arithmetic
 - $a + b$
 - $a - b$
 - $a * b$
 - a / b
 - $a \% \% b$ # Modulus

- `a %/% b` # Integer division
- `a ^ b` # Power

Built-in Mathematical Functions

Function	Meaning	Example
<code>abs(x)</code>	Absolute value	<code>abs(-10)</code>
<code>sqrt(x)</code>	Square root	<code>sqrt(25)</code>
<code>log(x)</code>	Natural log	<code>log(10)</code>
<code>exp(x)</code>	Exponential	<code>exp(2)</code>
<code>sin(x)</code>	Sine	<code>sin(pi/2)</code>
<code>cos(x)</code>	Cosine	<code>cos(0)</code>
<code>tan(x)</code>	Tangent	<code>tan(pi/4)</code>
<code>mean(x)</code>	Average	<code>mean(c(1,2,3))</code>
<code>median(x)</code>	Middle value	<code>median(x)</code>
<code>sd(x)</code>	Standard deviation	<code>sd(x)</code>
<code>sum(x)</code>	Sum	<code>sum(x)</code>
<code>prod(x)</code>	Product	<code>prod(x)</code>

12. SYNTAX IN R

- Assignment Operator
 - Code: `x <- 10`
 - `y = 20`
- Vector
 - Code: `v <- c(1,2,3)`
- Matrix
 - Code: `m <- matrix(1:9, nrow=3)`
- Function Definition
 - Code: `fun <- function(a, b){`
`return(a + b)`
`}`

- Condition (if)
 - Code: `if(x > 10) { print("Large") }`
- Loop
 - `for(i in 1:5) print(i)`

13. OUTCOME OF R LAB

Upon completion of the R programming laboratory, students will:

- ✓ Understand fundamentals of R and statistical programming.
- ✓ Be capable of writing and debugging R programs.
- ✓ Perform statistical analyses and hypothesis testing.
- ✓ Create visually appealing charts and graphs.
- ✓ Work with datasets using vectors, matrices, and data frames.
- ✓ Apply R in research areas such as data science, analytics, and machine learning.
- ✓ Gain practical experience in data manipulation and interpretation.

Problem 1:

Write a Program in R to create Vectors

Aim: To create a vectors in R programming.

Problem:

```
#To create vectors
```

```
vector1=1:9
```

```
print("Vector1")
```

```
print(vector1)
```

```
vector2=9:1
```

```
print("Vector2")
```

```
print(vector2)
```

```
a=vector1+vector2
```

```
print("Addition of two vectors")
```

```
print(a)
```

```
b=vector1-vector2
```

```
print("Subtraction of two vectors")
```

```
print(b)
```

```
c=vector1*vector2
```

```
print("Multiplication of two vectors")
```

```
print(c)
```

```
d=vector1/vector2
```

```
print("Division of two vectors")
```

```
print(d)
```

Output:

[1] "Vector1"

[1] 1 2 3 4 5 6 7 8 9

[1] "Vector2"

[1] 9 8 7 6 5 4 3 2 1

[1] "Addition of two vectors"

[1] 10 10 10 10 10 10 10 10 10

[1] "Subtraction of two vectors"

[1] -8 -6 -4 -2 0 2 4 6 8

[1] "Multiplication of two vectors"

[1] 9 16 21 24 25 24 21 16 9

[1] "Division of two vectors"

[1] 0.1111111 0.2500000 0.4285714 0.6666667 1.0000000 1.5000000 2.3333333

[8] 4.0000000 9.0000000

Problem 2:

Write a Program in R to create Matrices

Aim: To create a matrix in R programming.

Problem:

```
#create 4x4 matrices
m1=matrix(c(1:16),nrow=4)
print("Matrix-1")
print(m1)
m2=matrix(c(17:32),nrow=4)
print("Matrix-2")
print(m2)
print("Addition of two matrices")
result=m1+m2
print(result)
print("Subtraction of two matrices")
result=m1-m2
print(result)
print("Multiplication of two matrices")
result=m1*m2
print(result)
print("Division of two matrices")
result=(m1/m2)
print(result)
print("Transpose of matrix-1")
result=(t(m1))
print(result)
```

Output:

[1] "Matrix-1"

[1] [2] [3] [4]

[1,] 1 5 9 13

[2,] 2 6 10 14

[3,] 3 7 11 15

[4,] 4 8 12 16

[1] "Matrix-2"

[1] [2] [3] [4]

[1,] 17 21 25 29

[2,] 18 22 26 30

[3,] 19 23 27 31

[4,] 20 24 28 32

[1] "Addition of two matrices"

[1] [2] [3] [4]

[1,] 18 26 34 42

[2,] 20 28 36 44

[3,] 22 30 38 46

[4,] 24 32 40 48

[1] "Subtraction of two matrices"

[1] [2] [3] [4]

[1,] -16 -16 -16 -16

[2,] -16 -16 -16 -16

[3,] -16 -16 -16 -16

[4,] -16 -16 -16 -16

[1] "Multiplication of two matrices"

[,1] [,2] [,3] [,4]

[1,] 17 105 225 377

[2,] 36 132 260 420

[3,] 57 161 297 465

[4,] 80 192 336 512

[1] "Division of two matrices"

[,1] [,2] [,3] [,4]

[1,] 0.05882353 0.2380952 0.3600000 0.4482759

[2,] 0.11111111 0.2727273 0.3846154 0.4666667

[3,] 0.15789474 0.3043478 0.4074074 0.4838710

[4,] 0.20000000 0.3333333 0.4285714 0.5000000

[1] "Transpose of matrix-1"

[,1] [,2] [,3] [,4]

[1,] 1 2 3 4

[2,] 5 6 7 8

[3,] 9 10 11 12

[4,] 13 14 15 16

Problem 3:

Write a Program in R to Calculate the Mean, Median and Mode of a set of observation

Aim: To calculate the Mean, Median and Mode in R programming.

Problem:

```
#create a vector
```

```
x<-c(12,7,3,4.2,18,2,54,-21,8,-5)
```

```
#Find mean
```

```
result.mean<-mean(x,trim=0.3)
```

```
print(result.mean)
```

```
#find the median
```

```
median.result<-median(x)
```

```
print(median.result)
```

```
getmode<-function(v)
```

```
{
```

```
  uniqv<-unique(v) uniqv[which.max(tabulate(match(v,uniqv)))]
```

```
}
```

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

```
#calculate the mode using the user function
```

```
result<-getmode(v)
```

```
print(result)
```

Output:

```
[1] 5.55
```

```
[1] 5.6
```

```
[1] 2
```

Problem 4:

Write a Program in R to Calculate standard deviation of a set of observation

Aim: To calculate the standard deviation in R programming.

Problem:

```
v<-c(1.8,3.7,9.2,4.7,6.1,2.8,6.1,2.2,1.4,7.9)
```

```
s2<-sd(v)
```

```
print(s2)
```

Output:

```
[1] 2.676004
```

Problem 5:

Write a Program in R test of significance using Chi-square test.

Aim: To calculate the test of significance using Chi-square test.

Example: 1

```
observed<-c(20,18,13,23,26,11,15)
```

```
expected<-c(1/7,1/7,1/7,1/7,1/7,1/7,1/7)
```

```
chisq.test(x=observed,p=expected)
```

Example: 2

From the following data find whether there is any significant linking in the habit of taking

soft drinks among the categories of employees.

Employees Soft drinks	Clerks	Teacher	officer	Total
Pepsi	10	25	65	100
Thumpsup	15	30	65	110
Bovonto	50	60	30	140
Total	75	115	160	350

Problem:

```
mytable<-c(10,25,65,15,30,65,50,60,30)
```

```
colnames(mytable)=c("clerks","teachers","officers")
```

```
rownames(mytable)=c("pepsi","thumps up","bovonto")
```

```
mytable
```

```
chisq.test(mytable,correct=false)
```

```
chisq.test(mytable)
```

Output:

	clerks	teachers	officers
pepsi	10	25	65
thumbs up	15	30	65
bovonto	50	60	30

Pearson's Chi-squared test

data: mytable

X-squared = 60.234, df = 4, p-value = 2.591e-12

Pearson's Chi-squared test

data: mytable

X-squared = 60.234, df = 4, p-value = 2.591e-12

Problem 6:

Write a program in R test of significance using Students T-test

Aim: To test the significance using students T-test in R programming.

A random sample of 10 girls has the following I.Q 70,120,110,88,95,98,107,100 and population mean of I.Q is 100.

Problem:

```
#read the data
```

```
x<-c(70,120,110,101,88,83,95,98,107,100)
```

```
#calculating p-values in R this looks like
```

```
n<-length(x)
```

```
t_start<-(mean(x)-100)/(sd(x)/sqrt(n))
```

```
t_start
```

Output:

```
[1] -0.620336
```

Problem 7:

Write a program in R to calculate Spearman's Rank Correlation Coefficients

Aim:

To calculate Spearman's Rank Correlation Coefficients in R programming.

Find the rank correlation coefficient between the height in cm. and weight in kg of 6 soldiers in Indian army.

Height	165	167	166	170	169	172
Weight	61	60	63.5	63	61.5	64

Rank :

Height(x): 6, 4, 5, 2, 3, 1

Weight(y): 5, 6, 2, 3, 4, 1

Problem:

$X=c(6,4,5,2,3,1)$

$Y=c(5,6,2,3,4,1)$

$\text{cor.test}(x,y,\text{method}="spearman")$

Output:

Spearman's rank correlation rho

data: X and Y

$S = 16$, p-value = 0.2972

alternative hypothesis: true rho is not equal to 0

sample estimates:

$\rho = 0.5428571$

Problem 8:

Write a program in R to find regression coefficients and draw regression lines

Aim: To find Regression Coefficients and to draw regression lines.

The following data relate to the class of 10 students ,internal and external exam in the university exam for the maximum of 50 each.

Internal marks	25	28	30	32	35	36	38	39	42	45
External marks	20	26	29	30	25	18	26	35	36	46

(i) Find y when x=30 (ii) Find x when y=25

Problem:

```
x=c(25,28,30,32,35,36,38,39,42,45)
```

```
y=c(20,26,29,30,25,18,26,35,35,46)
```

```
relation=lm(y~x)
```

```
a=data.frame(x=30)
```

```
result=predict(relation,a)
```

```
print(result)
```

```
relation=lm(x~y)
```

```
b=data.frame(y=25)
```

```
result=predict(relation,b)
```

```
print(result)
```

```
plot(x,y,col="blue",main="Regression line y on x",pch=19,abline(lm(y~x)),xlab="Internal  
marks",ylab="External marks")
```

```
plot(y,x,col="blue",main="Regression line x on y",pch=19,abline(lm(x~y)),xlab="Internal  
marks",ylab="External marks")
```

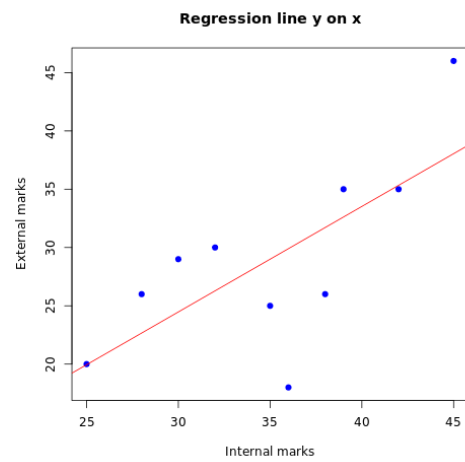
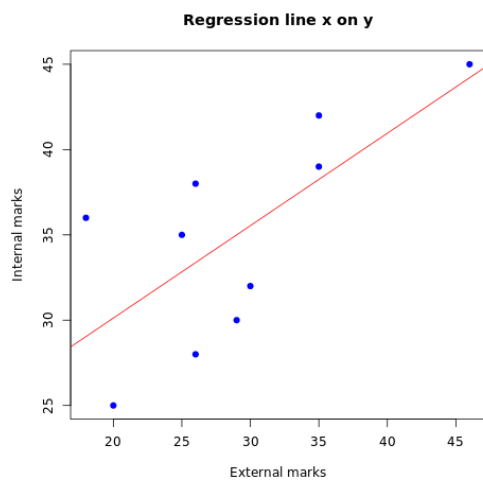
OUTPUT:

[1]1

[1]24.47486

[1]1

[1]32.83278



Problem 9:

Write a program in R test of significance using F- test

Aim: To calculate the test of significance using F test

Two random samples drawn from 2 normal populations are given below.

Test whether the 2 populations have the same variance.

X	17	23	32	25	22	24	28	6	31	33	20	27	N=1
Y	20	16	26	27	23	22	18	24	25	19	-	-	N=1

Problem:

```
x<-c(17,23,32,25,22,24,28,6,31,33,20,27)
```

```
y<-c(20,16,26,27,23,22,18,24,25,19)
```

```
var.test(x,y)
```

OUTPUT:

F test to compare two variances

data: x and y

F = 4.1864, num df = 11, denom df = 9, p-value = 0.04036

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

1.070113 15.020249

sample estimates:

ratio of variances

4.186364

Problem 10:

Write a program in R to Create Different charts for visualization of given set of given set of data.

Aim: To create different charts for visualization of given set of data.

Problem:

```
x<-c(50,20,10,20,30)
```

```
y<-c("RCB","MI","CSK","GT","DC")
```

```
pie(x,y,col=c("red","blue","yellow","orange","skyblue"))
```

```
barplot(x,col=c("red","blue","yellow","orange","skyblue"))
```

Output: