A LABORATORY REPORT FILE ON

R - PROGRAMMING LAB

BACHELOR OF TECHNOLOGY

SUBJECT CODE: CS111791; 7TH SEMESTER

GUIDED BY

Mr. Rajeshwar Kumar Dewangan

(Assistant Professor)

Computer Science & Engineering

SUBMITTED BY

Name:- .....

University Roll No:- .....

Enrollment No:- .....

Specialization:- .....

Section:- .....

Computer Science & Engineering

SESSION: 2024-25

ज्ञानादेव तु कैवल्यम्

(Shri Gangajali Education Society)

Established 1999

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Shri Shankaracharya Technical Campus, Bhilai

An Autonomous Institute

Approved by AICTE, New Delhi, Affiliated to

Chhattisgarh Swami Vivekanand Technical University, Bhilai

Khapri, Junwani Road, Bhilai, District-Durg, Chhattisgarh Pin Code- 490020, INDIA.

Ph. No.:0788-4088888, Fax No.: 0788-2298606, E-mail : info@sstc.ac.in, Web : www.sstc.ac.in

CERTIFICATE

Shri Shankaracharya Technical Campus, Bhilai

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

This is certify that Mr./Ms. ..... whose University Roll No. ....., Enrollment No. ..... is a student of B.TECH. (COMPUTER SCIENCE & ENGINEERING) – 7TH SEMESTER. Under Specialization .....and Section ..... at SHRI SHANKARACHARYA TECHNICAL CAMPUS, BHILAI CG INDIA. has completed his/ her Practical work in the SUBJECT :- R- PROGRAMMING LAB & SUBJECT CODE :- (CS111791) is accepted & approved after proper evaluation as a creditable work required as per the norms of CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY, BHILAI CG INDIA. in the laboratory of this college in the SESSION 2024-25 .

(Signature of the Guide)
Mr. Rajeshwar Kumar Dewangan
Assistant Professor (CSE)
SSTC, Bhilai

(Signature of the Student)
Place:-_____
Date:-_____

(Signature of the HOD)
Dr. (Mrs.) Samta Gajbhiye
Professor & Head of Department (CSE)
SSTC, Bhilai

ज्ञानादेव तु कैवल्यम्

Student Name :- ---------- **University Roll No. :-** ----------

Branch/ Semester :- B.Tech. (CSE) - 7th SEM **Enrollment No. :-** ----------

Subject/ Code :- R- Programming Lab (CS111791) **Specialization/ Section :-** ----------

INDEX

Sr. No.	List of Programs/ Experiment Description	Page No.	Date of Performing	Date of Submission	Signature/ Remarks
1.	Write an R program to check whether a year(integer) entered by the user is a leap year or not?	1			
2.	Write an R program to find the sum of natural without formula using the if-else statement and while loop	1			
3.	Write an R program that prints the grades of the students according to the marks obtained. The grading of the marks should be as follows. Marks Grades 800-1000 A+, 700 – 800 A, 500 – 700 B+, 400-500 B, 150 – 400 C, less than 150 D	2			
4.	Write an R program to make a simple calculator that can add, subtract, multiply and divide using switch cases and functions.	3			
5.	Write an R program to perform searching within a list (1 to 50). If the number is found in the list, print that the search is successful otherwise print that the number is not in the list.	4			
6.	Write an R program to create a list and data frame that stores the marks of any three subjects for 10 students. Find out the total marks, average, maximum marks and minimum marks of every subject.	5			
7.	Write an R program to import data from Excel to CSV files and apply data viewer functions like rm(), dim(), head(), tail(), sorting, filtering, searching to view few set of rows.	6-8			
8.	Write an R program to create two 3 X 3 matrices A and B and perform the following operations: a) Transpose of the matrix A. b) Addition of matrices A and B. c) Subtraction of matrices A and B.	9			
9.	Write an R program to create a list containing strings, numbers, vectors and logical values and do the following manipulations over the list. a. Access the first element in the list b. Give the names to the elements in the list c. Add element at some position in the list d. Remove the element e. Print the fourth element f. Update the third element	10-11			
10.	Let us use the built-in dataset air quality which has Daily air quality measurements in New York, May to September 1973. Create a histogram by suing appropriate arguments for the following statements: a. Assigning names, using the air quality data set b. Change colours of the Histogram c. Remove Axis and Add Labels to Histogram d. Change Axis limits of a Histogram e. Create a Histogram with density and Add Density curve to the Histogram	12-13			

Student Name :- ---------- **University Roll No. :-** ----------

Branch/ Semester :- B.Tech. (CSE) - 7th SEM **Enrollment No. :-** ----------

Subject/ Code :- R- Programming Lab (CS111791) **Specialization/ Section :-** ----------

INDEX

Program No-01

Write an R program to check whether a year(integer) entered by the user is a leap year or not?

Program:

```
year<- as.integer(readline(prompt="Enter a year: "))
if((year %% 4) == 0) {
  if((year %% 100) == 0) {
    if((year %% 400) == 0) {
      cat(paste(year,"is a leap year"))
    } else {
      cat(paste(year,"is not a leap year"))
    }
  } else {
    cat(paste(year,"is a leap year"))
  }
} else {
  cat(paste(year,"is not a leap year"))
}
```

Program No-01 Output:

```
Enter a year: 2023
2023 is not a leap year

Enter a year: 2020
2020 is a leap year
```

Program No-02

Write an R program to find the sum of natural without formula using the if-else statement and while loop

Program:

```
# Take input from the user
num<- as.integer(readline(prompt = "Enter a number: "))
if(num < 0) {
  cat("Enter a positive number")
} else {
  sum<- 0
  # use while loop to iterate until zero
  while(num > 0) {
    sum<- sum + num
    num<- num - 1
  }
  cat(paste("The sum is", sum))
}
```

Program No-02 Output:

```
Enter a number: 10
The sum is 55
Enter a number: -18
Enter a positive number
```

Program No-03

Write an R program that prints the grades of the students according to the marks obtained. The grading of the marks should be as follows. Marks Grades 800-1000 A+, 700 – 800 A, 500 – 700 B+, 400-500 B, 150 – 400 C, less than 150 D

Program:

```
marks<- as.integer(readline(prompt="Enter your total marks: "))
if(marks>=800|marks>=1000) {
  # This block executes when the boolean expression 1 is true.
  cat("Your grade is A+")
} else if(marks>=700|marks>800) {
  # This block executes when the boolean expression 2 is true.
  cat("Your grade is A")
} else if(marks>=500|marks>700) {
  # This block executes when the boolean expression 3 is true.
  cat("Your grade is B+")
} else if(marks>=400|marks>500) {
  # This block executes when the boolean expression 4 is true.
  cat("Your grade is B")
} else if(marks>=150|marks>400) {
  # This block executes when the boolean expression 5 is true.
  cat("Your grade is C")
} else {
    # This block executes when none of the above condition is true.
  cat("Your grade is D")
}
```

Program No-03 Output:

```
Enter your total marks: 889
Your grade is A+

Enter your total marks: 445
Your grade is B

Enter your total marks: 125
Your grade is D
```

Program No-04

Write an R program to make a simple calculator that can add, subtract, multiply and divide using switch cases and functions.

Program:

```
cat("1) For Addition\n")
cat("2) For Subtraction\n")
cat("3) For Multiplication\n")
cat("4) For Division\n")
n1<-as.integer(readline(prompt="Enter first number:"))
n2<-as.integer(readline(prompt="Enter second number:"))
choice<-as.character(readline(prompt="Enter your choice:"))

add<- function(x,y){
  return(x+y)
}
subtract<- function(x,y){
  return(x-y)
}
multiply<- function(x,y){
  return(x*y)
}
divide<- function(x,y){
  return(x/y)
}

# Syntax of switch(expression, case1, case2, case3....)
# Using this cased on matching value as a string(character)
switch(choice,
       "1"=cat("Addition of two number =",add(n1,n2)),
       "2"=cat("Subtraction of two number =",subtract(n1,n2)),
       "3"=cat("Multiplication of two number =",multiply(n1,n2)),
       "4"=cat("Division of two number =",divide(n1,n2))
)
```

Program No-04 Output:

```
1) For Addition
2) For Subtraction
3) For Multiplication
4) For Division
Enter first number: 5
Enter second number: 6
Enter your choice: 1
Addition of two number = 11

Enter your choice: 3
Multiplication of two number = 30
```

Program No-05

Write an R program to perform searching within a list (1 to 50). If the number is found in the list, print that the search is successful otherwise print that the number is not in the list.

Program:

```
# Create a list of numbers from 1 to 50
my_list <- 1:50

# Define a function to search for a number in the list
search_list <- function(num) {
  if (num %in% my_list) {
    cat("The search is successful.")
  } else {
    cat("The number is not in the list.")
  }
}

# Call the function with a number to for search
num<-as.numeric(readline(prompt="Please enter the number to search in the list: "))
search_list(num)
```

Program No-05 Output:

```
Please enter the number to search in the list: 25
The search is successful.

Please enter the number to search in the list: 55
The number is not in the list.
```

Program No-06

Write an R program to create a list and data frame that stores the marks of any three subjects for 10 students. Find out the total marks, average, maximum marks and minimum marks of every subject.

Program:

```
# Create a list of 10 students
# students <- paste0("Student ", 1:10)
students <- c("Aarav", "Rakesh", "Naresh", "Kiran", "Vikas", "Shyam", "Aarush", "Yug", "Namita", "Rahul")

# Create a list of 3 subjects
subjects <- c("Maths", "Science", "English")

# Create a data frame to store the marks of each student
marks <- data.frame(Maths=c(71, 63, 73, 84, 95, 45, 57, 68, 39, 27),
                    Science=c(87, 62, 55, 39, 27, 69, 82, 98, 97, 19),
                    English=c(38, 36, 66, 76, 78, 68, 97, 79, 47, 34))

# Convert the data frame and list to a data frame with help of column bind
marks_df <- as.data.frame(do.call(cbind, marks))

# Add row names to the data frame
rownames(marks_df) <- students
is.data.frame(marks_df)

# Calculate the total marks, average, maximum marks, and minimum marks of every subject
total_marks <- colSums(marks_df)
average_marks <- colMeans(marks_df)
max_marks <- apply(marks_df, 2, max)
min_marks <- apply(marks_df, 2, min)

# Print the results
cat("Total marks:\n")
print(total_marks)
cat("\nAverage marks:\n")
print(average_marks)
cat("\nMaximum marks:\n")
print(max_marks)
cat("\nMinimum marks:\n")
print(min_marks)
```

 # data frame of 10 students with subject's marks

	Maths	Science	English
Aarav	71	87	38
Rakesh	63	62	36
Naresh	73	55	66
Kiran	84	39	76
Vikas	95	27	78
Shyam	45	69	68
Aarush	57	82	97
Yug	68	98	79
Namita	39	97	47
Rahul	27	19	34

Program No-06 Output:

Total marks:

```
 Maths Science English
   622     635     619
```

Average marks:

```
 Maths Science English
  62.2    63.5    61.9
```

Maximum marks:

```
 Maths Science English
    95      98      97
```

Minimum marks:

```
 Maths Science English
    27      19      34
```

Program No-07

Write an R program to import data from Excel to CSV files and apply data viewer functions like rm(), dim(), head(), tail(), sorting, filtering, searching to view few set of rows.

Program:

```
#Installing xlsx package using R Console
install.packages("xlsx")

# Loading the library readxl and xlsx package into R workspace.
library(readxl)
library("xlsx")

# Verifying the package is installed.
any(grepl("xlsx",installed.packages()))

# Getting and printing current working directory.
print(getwd())

# Setting the current working directory.
setwd("D:/R_Program")

# Import data from a path to Excel file.xlsx
excel_data <- read.xlsx("D:/R_Program/car-speed.xlsx", sheetIndex = 1)

# Write the data to a CSV file
write.csv(excel_data, "D:/R_Program/car-speed.csv")

# Read the data from a CSV file to Dataframes
csv_data <- read.csv("D:/R_Program/car-speed.csv")
View(csv_data)
is.data.frame(csv_data)

# All the variable names
names(csv_data)
# Show the name of each column in the data frames
colnames(csv_data)

# The summary function is useful to quickly summarize the values in data frame
summary(csv_data)

# Remove the data frames from the workspace
rm(csv_data)

# Get the dimensions (rows and columns) of the data
dim(csv_data)
ncol(csv_data)
nrow(csv_data)
```

```
# View the first 6 rows of the data frame
head(csv_data)

# View the last 6 rows of the data frame
tail(csv_data)

# Find position of a matched pattern in a data frame
grep("Color", colnames(csv_data))

# Sort the data by a variable
sorted_data <- csv_data[order(csv_data$State),]

# Filter the data by a condition
filtered_data1 <- subset(csv_data, State == "Australia")
filtered_data2 <- subset(csv_data, Speed > 35)

# Search for a value in the data
search_result <- csv_data[csv_data$State == "NewMexico",]

#-----------------------------------------------------#
# Loading the dplyr package
install.packages("dplyr")
library("dplyr")

# Filter the data by a condition using dplyr package
filter(csv_data, State == "India")
filter(csv_data, Speed > 35)

#select/Search rows where 45 appears in any column using pipe(%>%) operator
csv_data %>% filter_all(any_vars(. %in% c(45)))
```

Program No-07 Output:

 # import data from Excel (car-speed.xlxs) to CSV files and apply data View() function

	Color	Speed	State
1	Blue	32	NewMexico
2	Red	45	India
3	Blue	35	SouthAfrica
4	White	34	Australia
5	Red	25	Australia
6	Blue	41	Australia
7	Yellow	80	India
8	Green	76	SouthAfrica
9	Blue	88	India
10	Yellow	22	NewMexico

 # Read the data from a CSV file to Dataframes after that remove the Dataframes from the workspace using rm() function

```
csv_data <- read.csv("D:/R_Program/car-speed.csv")
rm(csv_data)
```

 # Get the dimensions (rows and columns) of the data

```
dim(csv_data)      [1] 10  4
ncol(csv_data)     [1] 4
nrow(csv_data)     [1] 10
```

 # View the first 6 rows of the data frame

```
head(csv_data)
```

	Color	Speed	State
1	Blue	32	NewMexico
2	Red	45	India
3	Blue	35	SouthAfrica
4	White	34	Australia
5	Red	25	Australia
6	Blue	41	Australia

 # View the last 6 rows of the data frame

```
tail(csv_data)
```

	Color	Speed	State
5	Red	25	Australia
6	Blue	41	Australia
7	Yellow	80	India
8	Green	76	SouthAfrica
9	Blue	88	India
10	Yellow	22	NewMexico

 # Sort the data by a variable

```
sorted_data <- csv_data[order(csv_data$State),]
```

	Color	Speed	State
4	White	34	Australia
5	Red	25	Australia
6	Blue	41	Australia
2	Red	45	India
7	Yellow	80	India
9	Blue	88	India
1	Blue	32	NewMexico
10	Yellow	22	NewMexico
3	Blue	35	SouthAfrica
8	Green	76	SouthAfrica

 # Filter the data by a condition

```
filtered_data1 <- subset(csv_data, State == "Australia")
```

	Color	Speed	State
4	White	34	Australia
5	Red	25	Australia
6	Blue	41	Australia

```
filtered_data2 <- subset(csv_data, Speed > 35)
```

	Color	Speed	State
2	Red	45	India
6	Blue	41	Australia
7	Yellow	80	India
8	Green	76	SouthAfrica
9	Blue	88	India

 # Search for a value in the data

```
search_result <- csv_data[csv_data$State == "NewMexico",]
```

	Color	Speed	State
1	Blue	32	NewMexico
10	Yellow	22	NewMexico

Program No-08

Write an R program to create two 3 X 3 matrices A and B and perform the following operations:
a) Transpose of the matrix A.
b) Addition of matrices A and B.
c) Subtraction of matrices A and B.

Program:

```
# Creating two vector as a matrixA & matrixB
MatrixA <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
MatrixB <- c(9, 8, 7, 6, 5, 4, 3, 2, 1)

# Creating First matrix A
myMatrixA <- matrix(MatrixA, nrow = 3, ncol = 3)
cat("Matrix A:")
print(myMatrixA)

# Creating Second matrix B
myMatrixB <- matrix(MatrixB, nrow = 3, ncol = 3)
cat("Matrix B:")
print(myMatrixB)

# Transpose of matrix A
transposeA <- t(myMatrixA)
cat("Transpose of Matrix A:")
print(transposeA)

# Addition of matrices A and B
matrixC_Addition <- myMatrixA + myMatrixB
cat("Result of Addition:")
print(matrixC_Addition)

# Subtraction of matrices A and B
matrixC_Subtraction <- myMatrixA - myMatrixB
cat("Result of Subtraction:")
print(matrixC_Subtraction)
```

Program No-08 Output:

```
Matrix A:> print(myMatrixA)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

Matrix B:> print(myMatrixB)
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1

Transpose of Matrix A:>
print(transposeA)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

Result of Addition:>
print(matrixC_Addition)
     [,1] [,2] [,3]
[1,]   10   10   10
[2,]   10   10   10
[3,]   10   10   10

Result of Subtraction:>
print(matrixC_Subtraction)
     [,1] [,2] [,3]
[1,]   -8   -2    4
[2,]   -6    0    6
[3,]   -4    2    8
```

Program No-09

Write an R program to create a list containing strings, numbers, vectors and logical values and do the following manipulations over the list.

a. Access the first element in the list
b. Give the names to the elements in the list
c. Add element at some position in the list
d. Remove the element
e. Print the fourth element
f. Update the third element

Program:

```
# Create a list containing strings, numbers, vectors, and logical values
list_data <- list("Python", "PHP", c(5, 7, 9, 11), TRUE, 125.17, 75.83)
print(list_data)

# a. Access the first element in the list
first_element <- list_data[[1]]
cat(first_element)

# b. Give names to the elements in the list
names(list_data) <- c("Language1", "Language2", "Numbers", "Flag", "Value1", "Value2")
print(list_data)

# c. Add an element at a specific position in the list
list_data[["NewElement"]] <- "New Value"
print(list_data)

# d. Remove an element from the list
list_data[["Language2"]] <- NULL
print(list_data)

# e. Print the fourth element
fourth_element <- list_data[["Flag"]]
cat(fourth_element)

# f. Update the third element and print the modified list
list_data[["Numbers"]][3] <- 99
print("Data of the list:")
print(list_data)
```

Program No-09 Output: # Create a list containing strings, numbers, vectors and logical values

[[1]]
[1] "Python"

[[2]]
[1] "PHP"

[[3]]
[1] 5 7 9 11

[[4]]
[1] TRUE

[[5]]
[1] 125.17

[[6]]
[1] 75.83

 # a. Access the first element in the list Python

 # b. Give names to the elements in the list

$Language1
[1] "Python"

$Language2
[1] "PHP"

$Numbers
[1] 5 7 9 11

$Flag
[1] TRUE

$Value1
[1] 125.17

$Value2
[1] 75.83

 # c. Add an element at a specific position in the list

$Language1
[1] "Python"

$Language2
[1] "PHP"

$Numbers
[1] 5 7 9 11

$Flag
[1] TRUE

$Value1
[1] 125.17

$Value2
[1] 75.83

$NewElement
[1] "New Value"

 # d. Remove an element from the list

$Language1
[1] "Python"

$Numbers
[1] 5 7 9 11

$Flag
[1] TRUE

$Value1
[1] 125.17

$Value2
[1] 75.83

$NewElement
[1] "New Value

 # e. Print the fourth element TRUE

 # f. Update the third element and print the modified list

$Language1
[1] "Python"

$Numbers
[1] 5 7 99 11

$Flag
[1] TRUE

$Value1
[1] 125.17

$Value2
[1] 75.83

$NewElement
[1] "New Value"

Program No-10

Let us use the built-in dataset air quality which has Daily air quality measurements in New York, May to September 1973. Create a histogram by suing appropriate arguments for the following statements:

a. Assigning names, using the air quality data set

b. Change colours of the Histogram

c. Remove Axis and Add Labels to Histogram

d. Change Axis limits of a Histogram

e. Create a Histogram with density and Add Density curve to the Histogram

Program:

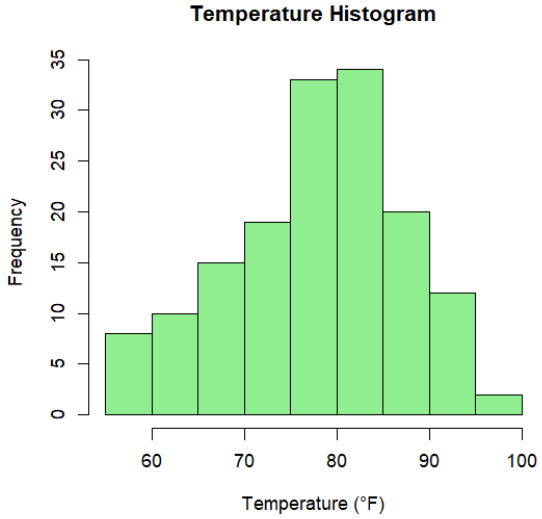
```
# Load the airquality dataset
data("airquality")
# Assign meaningful names to the columns using air quality data with help of airquality dataset
colnames(airquality) <- c("Ozone", "Solar.R", "Wind", "Temp", "Month", "Day")

# Change colours of the histogram with temperature data
hist(airquality$Temp, col = "lightgreen", main = "Temperature Histogram", xlab = "Temperature (°F)")

# Create a histogram without axis and add custom axis labels
hist(airquality$Temp, col = "skyblue", main = "Temperature Histogram", xlab = "Temperature (°F)", axes = FALSE)

# Add custom axis labels
axis(1, at = seq(50, 100, by = 10), labels = paste(seq(50, 100, by = 10), "°F"))
axis(2, las = 1)

# Create a histogram with custom x-axis limits
hist(airquality$Temp, col = "orange", main = "Temperature Histogram", xlab = "Temperature (°F)", xlim = c(50, 100))

# Create a density histogram with add a density curve
hist(airquality$Temp, col = "pink", main = "Temperature Histogram (Density)", xlab = "Temperature (°F)", freq = FALSE)

# Add a density curve
lines(density(airquality$Temp), col = "red", lwd = 2)
```

Program No-10 Output:

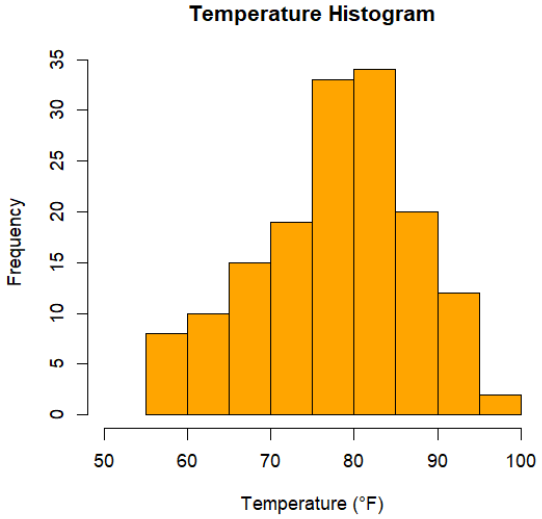
 # a. Assign meaningful names to the columns using air quality data with help of airquality dataset

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8	72	5	2
3	12	149	12.6	74	5	3
…	…	…	…	…	…	…
…	…	…	…	…	…	…
151	14	191	14.3	75	9	28
152	18	131	8	76	9	29
153	20	223	11.5	68	9	30

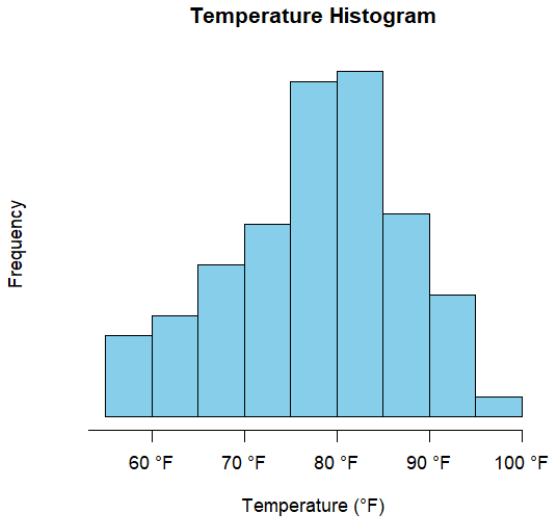
 # b. Change colours of the histogram with temperature data



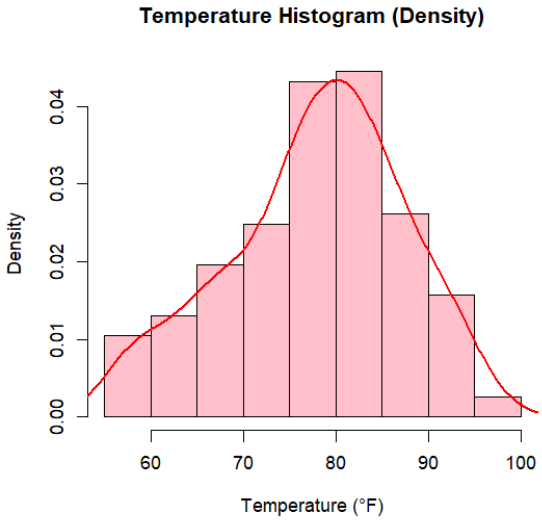
 # d. Create a histogram with custom x-axis limits



 # c. Create a histogram without axis and add custom axis labels



 # e. Create a density histogram with add a density curve



Program No-11

Design a data frame in R for storing about 20 employee details. Create a CSV file named "input.csv" that defines all the required information about the employee such as id, name, salary, start_date, dept. Import into R and do the following analysis.

a. Find the total number rows & columns

b. Find the maximum salary

c. Retrieve the details of the employee with maximum salary

d. Retrieve all the employees working in the IT Department

e. Retrieve the employees in the IT Department whose salary is greater than 20000 and write these details into another file "output.csv".

Program:

```
# Create a data frame with 20 rows and 5 columns
employee_df <- data.frame(
  id = 1:20,
  name = c("John", "Jane", "Bob", "Alice", "Mike", "Sara", "Tom", "Linda", "David", "Mary",
"Jack", "Emily", "Alex", "Olivia", "Daniel", "Sophia", "William", "Ava", "James", "Mia"),
  salary = sample(20000:50000, 20),
  start_date = sample(seq(as.Date('2022/01/01'), as.Date('2022/12/31'), by="day"), 20),
  dept = sample(c("IT", "HR", "Sales"), 20, replace = TRUE)
)

# Write the data frame to a CSV file
write.csv(employee_df, file = "input.csv", row.names = FALSE)

# Import the CSV file into R
employee_df <- read.csv("input.csv")

# Find the total number of rows and columns
n_rows <- nrow(employee_df)
n_cols <- ncol(employee_df)

# Find the maximum salary
max_salary <- max(employee_df$salary)

# Retrieve the details of the employee with maximum salary
max_salary_employee <- employee_df[employee_df$salary == max_salary,]

# Retrieve all the employees working in the IT Department
it_employees <- employee_df[employee_df$dept == "IT",]

# Retrieve the employees in the IT Department whose salary is greater than 20000
it_employees_above_20000 <- it_employees[it_employees$salary > 20000,]

# Write the details of the employees in the IT Department whose salary is greater than 20000
to another CSV file
write.csv(it_employees_above_20000, file = "output.csv", row.names = FALSE)
```

Program No-11 Output:

 # Create 20 employee details such as id, name, salary, start_date, dept. and write the data frame (employee_df) to a CSV (input.csv) file.

id	name	salary	start_date	dept
1	John	27980	17-08-2022	IT
2	Jane	37947	01-03-2022	IT
3	Bob	36803	22-05-2022	HR
4	Alice	31334	25-08-2022	Sales
5	Mike	47297	27-10-2022	Sales
6	Sara	45550	28-09-2022	IT
7	Tom	37487	24-03-2022	IT
8	Linda	40029	21-05-2022	Sales
9	David	23496	19-02-2022	Sales
10	Mary	46308	14-02-2022	HR
11	Jack	44574	11-02-2022	Sales
12	Emily	34584	05-08-2022	Sales
13	Alex	29390	16-04-2022	IT
14	Olivia	39435	20-09-2022	HR
15	Daniel	36983	24-11-2022	HR
16	Sophia	48797	29-12-2022	IT
17	William	39754	24-06-2022	HR
18	Ava	40269	04-07-2022	IT
19	James	39892	19-12-2022	HR
20	Mia	21349	13-11-2022	HR

 # a. Find the total number rows & columns

n_rows 5L
n_cols 20L

 # b. Find the maximum salary

max_salary 48797L

 # c. Retrieve the details of the employee with maximum salary

16	Sophia	48797	29-12-2022	IT

 # d. Retrieve all the employees working in the IT Department

id	name	salary	start_date	dept
1	John	27980	17-08-2022	IT
2	Jane	37947	01-03-2022	IT
6	Sara	45550	28-09-2022	IT
7	Tom	37487	24-03-2022	IT
13	Alex	29390	16-04-2022	IT
16	Sophia	48797	29-12-2022	IT
18	Ava	40269	04-07-2022	IT

 # e. Retrieve the employees in the IT Department whose salary is greater than 20000 and write these details into another file "output.csv".

16	Sophia	48797	29-12-2022	IT

Program No-12

Create a dataset or table [‘Smart Phone”] in an excel sheet that stores the mobile information [price, company_name, model, sales_percent] of five different companies. Store at least 20 rows. Write an R program to find out the output for the following information.

a. Maximum price of the mobile of each company
b. Minimum price of mobile of each company
c. Average price of mobile of each company
d. Total Price of mobile of each company

Program:

```
# Create a data frame with 20 rows and 4 columns
smartphone_df <- data.frame(
  price = sample(10000:50000, 20),
  company_name = sample(c("Samsung", "Apple", "OnePlus", "Xiaomi", "Realme"), 20, replace = TRUE),
  model = paste0("Model ", 1:20),
  sales_percent = sample(10:50, 20)
)

# Write the data frame to an Excel file
library(openxlsx)
write.xlsx(smartphone_df, file = "Smart Phone.xlsx", sheetName = "Sheet1", rowNames = FALSE)

# Read the data frame from the Excel file
smartphone_df <- read.xlsx("Smart Phone.xlsx", sheetName = "Sheet1")

# Find the maximum price of the mobile of each company
max_price <- aggregate(price ~ company_name, smartphone_df, max)

# Find the minimum price of the mobile of each company
min_price <- aggregate(price ~ company_name, smartphone_df, min)

# Find the average price of the mobile of each company
avg_price <- aggregate(price ~ company_name, smartphone_df, mean)

# Find the total price of the mobile of each company
total_price <- aggregate(price ~ company_name, smartphone_df, sum)
```

Program No-12 Output:

 # Create an excel sheet (Smart_Phone.xlxs) store details of 5 different mobile companies like price, company_name, model, sales_percentage and read the dataframe (smartphone_df) to file

	price	company_name	model	sales_percent
1	26751	Apple	Model 1	43
2	30650	Samsung	Model 2	22
3	42961	OnePlus	Model 3	19
4	37479	OnePlus	Model 4	30
5	28081	OnePlus	Model 5	42
6	14919	Realme	Model 6	35
7	29150	Samsung	Model 7	33
8	30546	OnePlus	Model 8	50
9	22315	Xiaomi	Model 9	47
10	16837	Realme	Model 10	18
11	35977	Samsung	Model 11	24
12	29550	Realme	Model 12	10
13	29480	Apple	Model 13	49
14	40553	Samsung	Model 14	41
15	25397	OnePlus	Model 15	29
16	26115	Realme	Model 16	32
17	30271	Xiaomi	Model 17	13
18	23477	OnePlus	Model 18	12
19	32041	Xiaomi	Model 19	39
20	24594	Realme	Model 20	34

 # a. Maximum price of the mobile of each company

	company_name	**price**
1	Apple	29480
2	OnePlus	42961
3	Realme	29550
4	Samsung	40553
5	Xiaomi	32041

 # b. Minimum price of mobile of each company

	company_name	**price**
1	Apple	26751
2	OnePlus	23477
3	Realme	14919
4	Samsung	29150
5	Xiaomi	22315

 # c. Average price of mobile of each company

	company_name	**price**
1	Apple	28115.5
2	OnePlus	31323.5
3	Realme	22403
4	Samsung	34082.5
5	Xiaomi	28209

 # d. Total Price of mobile of each company

	company_name	**price**
1	Apple	56231
2	OnePlus	187941
3	Realme	112015
4	Samsung	136330
5	Xiaomi	84627