Date: 20/09/24

Program-01

Develop a program to create two 3x3 matrices A and B perform the following operations

a) transpose of the matrix

b) Addition

c) Subtraction

d) Multiplication

```
A <- matrix(nrow=3, ncol=3, data= c(1,2,3,4,
                    5,6,7,8,9))
B <- matrix(nrow=3, ncol=3, data= c(1,2,3,4,
                    5,6,7,8,9))
add <- A+B
sub <- A-B
pro <- A%*%B
transpose1 <- t(A)
transpose2 <- t(B)
print (add)
print (sub)
print (pro)
print (transpose1)
print (transpose2)
```

output:

```
> print (add)
2    8    14
4    10   16
6    12   18

> print (sub)
0    0    0
0    0    0
0    0    0

> print (pro)
30   66   102
36   81   126
42   96   150

> print (transpose1)
1    2    3
4    5    6
7    8    9

> print (transpose2)
1    2    3
4    5    6
7    8    9
```

Date: 20/09/24

Program - 02

Develop a R program which generate prime number upto specify number

```
prime2<-function(n){
if(n==1){
    is.prime <- FALSE
}else if(n==2){
    is.prime <- TRUE
}else{
    is.prime <- TRUE
    for (m in 2:(n/2)){
        if (n %% m == 0) is.prime <- FALSE
    }
}
return (is.prime)
}
n<- 100
primes <- c()
for (x in 1:n){
    if(prime2 (x)== TRUE)
    {
        primes <- c (primes, x)
    }
}
print (primes)
```

output :

```
print(primes)
```

2 3 5 7 11 13 17 19 23 29 31 37
41 43 47 53 59 61 67 71 73 79 83 89
97

Date: 4/10/24

Program - 03

Develop R program to create a data frame with following details & do the following operation

ItemCode	ItemCategory	ItemPrice
1001	Electronics	700
1002	Desktop Supplies	300
1003	office Supplies	350
1004	USB	400
1005	CD Drive	800

a) Subset the data frame & display the datas of only those items who's price is greater than or equal to 350

b) Subset the data frame & display only the items where the category is either office supplies or desktop supplies

c) Create another data frame called item details with three different fields item Code, item Qty on Hand, ItemRecorderLvl and merge the two frames.

output :

ItemCode	ItemCategory	Item Price
1001	Electronics	700
1002	Desktop Supplies	300
1003	Office Supplies	350
1004	USB	400
1005	CD Drive	800

a)

ItemCode	ItemCategory	ItemPrice
1001	Electronics	700
1003	Office Supplies	350
1004	USB	400
1005	CD Drive	800

b)

ItemCode	ItemCategory	item Price
1002	Desktop Supplies	300
1003	Office Supplies	350

c)

ItemCode	ItemQtyHand	ItemRecorderLvl
1001	10	100
1002	2	500
1003	3	800
1004	4	200
1005	5	600
1		

```
item <- data.frame (
      itemCode = c (1001, 1002, 1003, 1004, 1005),
      itemCategory = c (" Electronics", " Desktop
                 supplies", " Office Supplies", "USB",
             " CD Drive"),
      itemPrice = c (700, 300, 350, 400, 800)
)
print (items)
a) subset-price <- subset (items, itemPrice >=350)
print (subset-price)
b) subset- category <- subset (items, itemCategory
     %in% c (" office supplies", "Desktop Supplies"))
print ( subset-category)
c) item-details <- data. frame (
        itemCode = c (1001, 1002, 1003, 1004, 1005),
        itemQtyOnHand = c (10 , 2, 3, 4, 5),
        itemRecordLvl = c (100, 500, 800, 200, 600)
)
print (item-details)
merge-data = merge (items , item-details,
              by = " itemCode")
print (merge-data)
```

After merging

itemCode	itemCategory	itemPrice	itemQtyHand	item RecordLvl
1001	Electronics	700	10	100
1002	Desktop Supplies	300	2	500
1003	Office Supplies	350	3	800
1004	USB	400	4	200
1005	CD Drive	800	5	600

Date: 18/10/24

program-04

Develop a program to find the factorial of given number using recursive function.

```
factorial-recursive <- function(n){
        if (n==0){
            return(1)
        }else{
            return (n*factorial-recursive(n-1))
        }
}
num=5
if( num<0){
        cat (" Factorial is not defined for negative
              numbers\n")
}else{
        cat (" The factorial of", num, "is",
              factorial-recursive(num)"\n")
}.
```

Output 1:
num=5
The factorial of 5 is 120.

output 2:
num=0

The factorial of 0 is 1.

output 3:
num=-1

Factorial is not defined for negative numbers.

Date: 18/10/24

program-05

Develop R program using built in data set named mammals to find pearson correlation and compare it with spearman correlation

a) Pearson and Spearman Correlation

```
data(mammals)
mammals
head (mammals)

# Calculate Pearson correlation
pearson-corr <- cor(mammals$body,
    mammals$brain, method = "pearson")
print(paste (" Pearson Correlation: ",
    pearson-corr))

# Calculate Spearman Correlation
spearman-corr <- cor(mammals$body,
    mammals$brain, method = " spearman")
print (paste(" Spearman Correlation : ",
    spearman-corr))
```

b) Plot the data

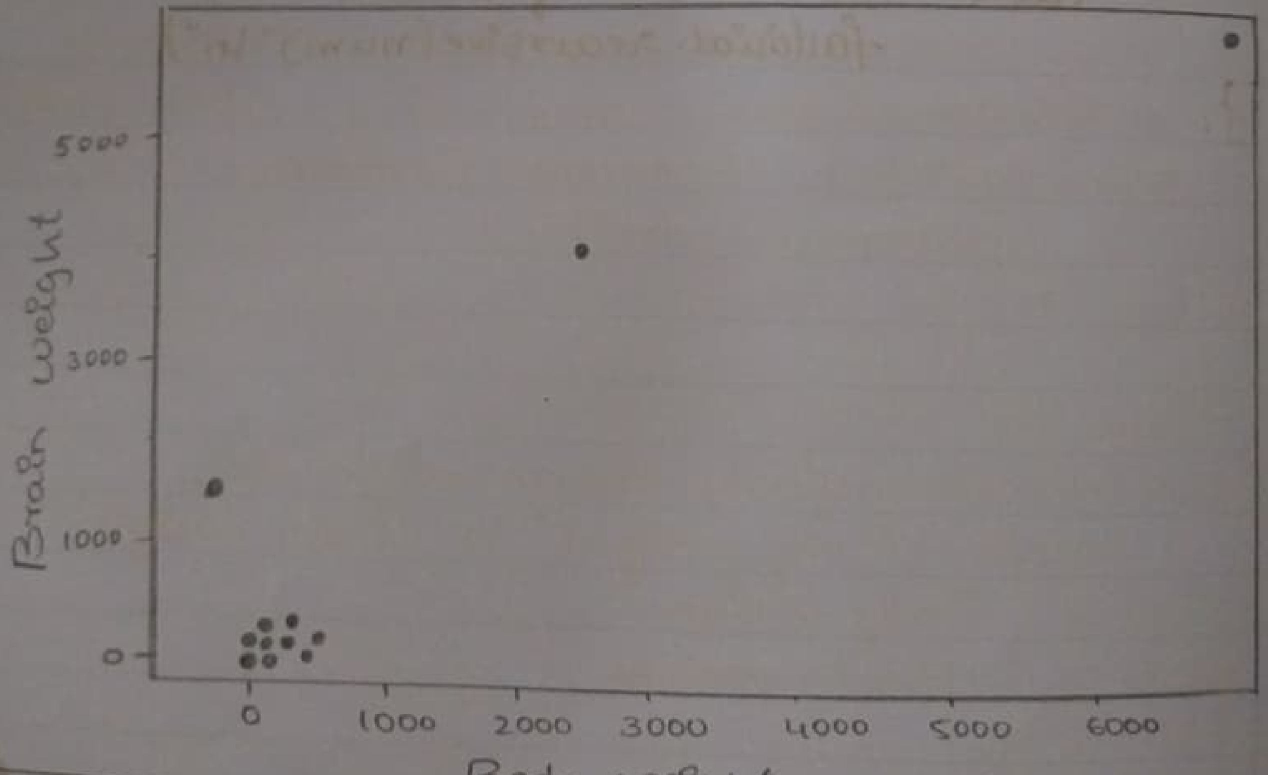
```
plot(mammals$body, mammals$brain,
    main = "Body weight" vs Brain weight"
xlab = " Body weight",
ylab = " Brain weight",
pch = 19,
col = "blue")
```

Output :
a) head (mammals)

	body	brain
Arctic fox	3.385	44.5
Owl monkey	0.480	15.5
Mountain beaver	1.350	8.1
Cow	465.000	423.0
Grey wolf	36.330	119.5
Goat	27.660	115.0

" Pearson Correlation : 0.9341638423 23355"
" Spearman Correlation : 0.9534986212 77599"

Output :
b) Body weight vs Brain weight



Date: 15/11/24

program-06

Use datalist air quality along with the arguments to find the following
a) assign names to the X-axis & Y-axis.
b) change the dimension of the axis
c) remove the axis & plot the hystogram
d) plot the density curve for the hystogram

a)
```
# load the airquality dataset
data ("airquality")
head ("airquality")
hist (airquality$Ozone, main = " levels in
   New York (may-sep 1973)",
xlab = " Ozone",
ylab = " Frequency",
 col = 'lightblue")
```

b)
```
hist(airquality$Ozone, main=" Levels in New York
 (may-sep) 1973)",
    xlab = " Ozone",
    ylab = "Frequency",
    col  = " lightblue",
    border = " white")
```

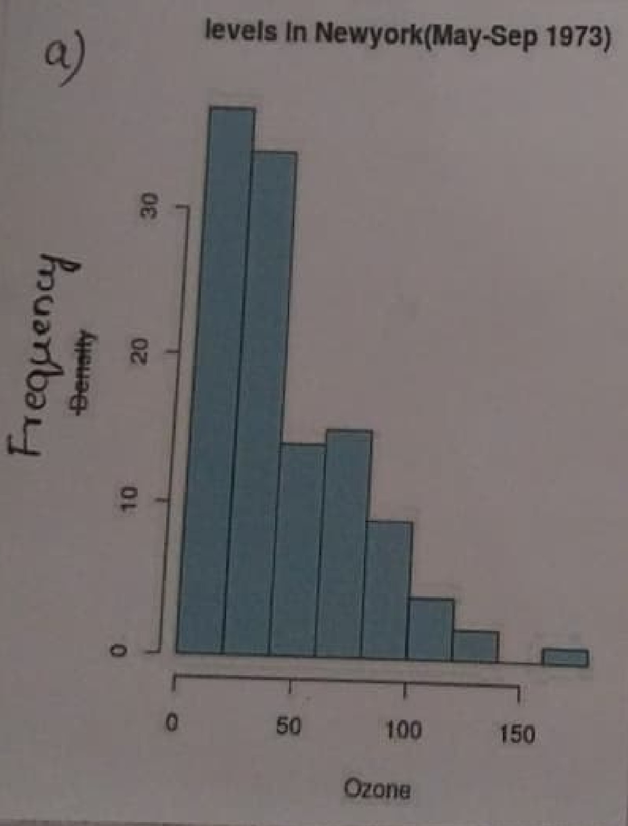
Output :

> head (airquality)

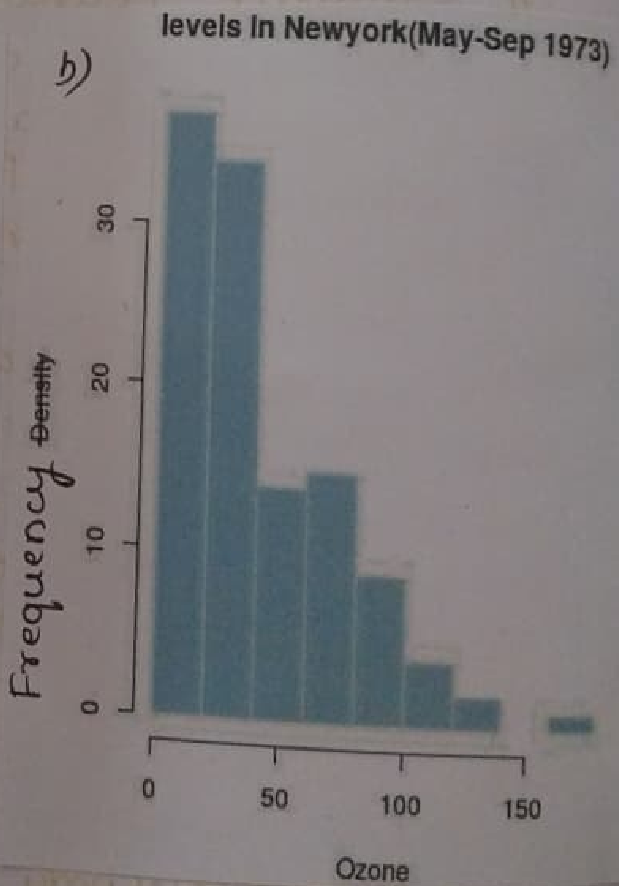
	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6

a)

a)



b)



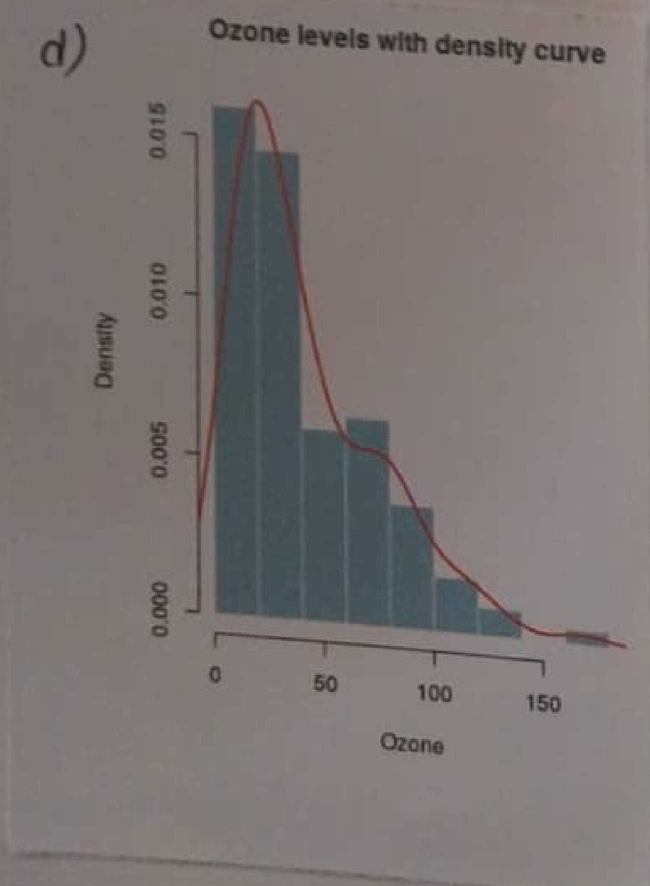
```
c) data ("airquality")
  head ("airquality")
hist (airquality$Ozone, main= levels in NewYork(
        may-sep 1973"),
        xlab = " Ozone",
        ylab = " Frequency",
        col  = " lightblue",
        border = " white",
        axes  = FALSE)


d) hist(airquality$Ozone , main =" Ozone levels with
        density curve",
        xlab = "Ozone",
        ylab = "Density",
        col  = "lightblue",
        border = " white",
        freq = FALSE)
lines(density(na.omit(airquality$ozone)),
        col ="red",
        lwd = 2)
```

()

levels in Newyork(May-Sep 1973)
Frequency
Ozone

d)



Ozone levels with density curve
Density
0.000
0.005
0.010
0.015
0
50
100
150
Ozone

Date: 6/12/24

program - 07

Assess the Financial Statement of an Organization being supplied with 2 vectors of data: Monthly Revenue and Monthly expense for the finacial Year. You can create your own sample data vector for this experiment. Calculate the following finacial metrics.

a. Profit for each month
b. Profit after tax for each month (Tax Rate is 30%.)
c. Profit margin for each month equals to profit after tax divide by revenue
d. Good Months - where the profit after tax was greater than the mean for the year
e Bad Months - where the profit after tax was less than the mean for the year
f The best month - where the profit after tax was max for the year
g. The worst month - where the profit after tax was min for the year

```
# Create sample data for Monthly Revenue and
  Expenses.

monthly_revenue <- c(20000,22000,25000,18000.
30000,31000,27000,29000,26000,24000,21000,
23000)

monthly_expenses <- c(15000,18000,20000,17000,
25000,24000,23000,26000,22000,20000,19000,21000)
```

```
# Calculate profit for each month
profit <- monthly_revenue - monthy_expenses

# Calculate Profit after tax for each month
(Tax Rate = 30%)
tax_rate <- 0.30
profit_after_tax <- profit * (1 - tax_rate)

# Calculate Profit Margin for each month
profit_margin <- (profit_after_tax / monthly_revenue)
    * 100

# Identify Good and Bad months

mean_profit_after_tax <- mean(profit_after_tax)
good_months <- profit_after_tax > mean_profit_after_
    tax
bad_months <- profit_after_tax < mean_profit_after_
    tax

#Identity the Best and Worst months
best_month <- which.max(profit_after_tax)
worst_month <- which.min (profit_after_tax)

# Format values as required

formatted_profit <- round (profit / 1000, 2) # in units of $1000
formatted_profit_after_tax <- round (profit_after_tax /
    1000, 2) # units of $1000
formatted_profit_margin <- round (profit_margin, 0)
```

```
# Combine results into a data frame

results <- data.frame(
    Month = 1:12,
    Revenue_in_1000s = monthly_revenue/1000,
    Expenses_in_1000s = monthly_expenses/1000,
    Profit_in_1000s = formatted_profit,
    Profit_After_tax_in_1000s = formatted_profit_after
     tax,
    Profit_Margin_Percentage = formatted_profit_m
      argin,
    Good_Month = good_months,
    Bad_Month = bad_months,
    Best_Month = Month == best_month,
    Worst_Month = Month == worst_month.
)

# Save results to a CSV file
write.csv(results, "finacial_statement.csv",
 row.names = FALSE)

# Print the results
print(results)
```

Date: 6/12/24

program-08

Design a data frame in R 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"

```
# Step 1: Create a data frame with 20 employee
details
employee_data <- data.frame(
id = 1:20,
name = c("Alice", "Bob", "Charlie", "David" "Eva",
"Frank", "Grace", "Helen", "Ian", "Jack", "Kate", "Liam",
"Mona", "Nina", "Oscar", "Paul", "Quinn", "Rita",
"Steve", "Tina"),
salary = c(15000, 22000, 18000, 25000, 27000 19000,
30000, 23000, 21000, 24000, 20000, 17000, 28000,
26000, 31000, 16000, 22000, 29000, 18000, 27000),
```

```
start-date = as.Date(c("2010-01-15", "2012-03-18",
"2015-05-10", "2018-07-20", "2020-09-25", "2016-11-30",
"2017-02-14", "2019-04-25", "2013-06-18", "2021-08-05",
"2014-10-15", "2011-12-20", "2018-02-28", "2017-09-30",
"2015-01-02", "2013-03-22", "2020-05-16", "2021-07-19",
"2014-08-23", "2016-11-11")),
dept = c("IT", "HR", "Finance", "IT", "IT", "HR",
"IT", "Finance", "HR", "IT", "Finance", "IT", "HR",
"IT", "Finance", "HR", "IT", "IT", "Finance", "IT")
)

# Step2: Save the data frame to a csv file
named "input.csv"

write.csv(employee_data, "input.csv", row.
  names = FALSE)

# step 3: Import the data back into R

imported_data <- read.csv("input.csv")

# step 4: Perform the analysis
  # a) Find the total number of rows and
     columns
num_rows <- nrow(imported_data)
num_cols <- ncol(imported_data)

# b) Find the maximum salary
  max_salary <- max(imported_data$salary)
```

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

employee-max-salary ← imported-data[imported-data$salary == max-salary,]

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

it-employees ← subset(imported-data, dept== "IT")

e) Retrieve employees in the IT Department with salary > 20000 and save to "output.csv"

it-employees-high-salary ← subset(it-employees, salary > 20000)
write.csv(it-employees-high-salary, "output.csv", row.names = FALSE)

Print results

cat("Total rows:", num-rows, "\n")
cat("Total columns:", num-cols, "\n")
cat("Maximum salary:", max-salary, "\n")
cat("Employee ~~max~~ with max salary :\n")
print(employee-max-salary)
cat("Employees in IT.Department:\n")
print(it-employees)
cat("IT employees with salary > 20000 saved to 'output.csv'\n")

Date: 20/09/24

Program-09

Demostrate the steps for installing of R and R-studio perform the following

a) Assign different types of values to variably and display the type of variable, Assign different types such as Double, Integer, logical, complex and character and understand the difference between each data type.

b) Demonstrate Arithmetic and logical operation with simple example

c) Demonstrate generation of sequence and creation of vectors

d) Demonstrate the creation of matrix from vectors using Binding function

e) Demonstrate the creation of matries.

f) Demonstrate element extraction from vectors matrices and array

Numeric

```
V <- 23.5
print(class(V))
```

logical

```
V <- TRUE
print(class(V))
```

Integer

```
V <- 22
print(class(V))
```

character

```
v <- a
print (class(v))
apple <- c ("red", "green", "Yellow");
print (class (apple))
```

Complex

```
v <- 3+4i
print (class(v))
```

b) Arithmetic operation

```
v <- c (2, 5, 6)
t <- c (8, 3, 4)
print (v+t)
```

Logical operation

```
v <- c(0,1,0)
t <- c(1,1,0)
print (v&t)
```

c) # Generate a sequence from 1 to 10

```
seq1 <- seq (1, 10)
print ( seq1)
```

```
# Create a numeric vector
numeric-vector <- c(1, 2, 3, 4, 5)
print( numeric-vector)
```

d) # define two matrices
mat 1 <- matrix (c(1,2,3,4) , nrow = 2 , ncol = 2)
mat 2 <- matrix (c(5,6,7,8), nrow = 2 , ncol = 2)

# Addition of two matrices
mat_add <- mat 1 + mat 2
print (mat_add)

e) row 1 <- c (1, 2, 3)
row 1 <- c (4, 5, 6)
row 3 <- c (7, 8, 9)
a <- rbind (row 1, row 2, row 3)
print (a)