Exp. 3: Demonstrate Matrix operations in R

Aim: To experiment with matrices, a two-dimensional data type/objects of R.

- Matrices are the R objects in which the elements are arranged in a two-dimensional rectangular layout.

- They contain elements of the same atomic types.

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Exp. 3: Demonstrate Matrix operations in R

- A Matrix is created using the **matrix()** function.

Syntax: The basic syntax for creating a matrix in R is −

matrix(data, nrow, ncol, byrow, dimnames)

Following is the description of the parameters used −

- **data** is the input vector which becomes the data elements of the matrix.
- **nrow** is the number of rows to be created.
- **ncol** is the number of columns to be created.
- **byrow** is a logical clue. If TRUE then the input vector elements are arranged by row.
- **dimname** is the names assigned to the rows and columns.

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

# Elements are arranged sequentially by row:

```
        M <- matrix(c(3:14), nrow = 4, ncol=3, byrow =
TRUE)
        print(M)
```

# it produces the following result :

```
          [,1]   [,2]   [,3]
  [1,]     3      4      5
  [2,]     6      7      8
  [3,]     9     10     11
  [4,]    12     13     14
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

# Elements are arranged sequentially by column:

```
M       N <- matrix(c(3:14), nrow = 4, ncol=3,byrow =
FALSE)
        print(N)
```

# it produces the following result :

```
        [,1]    [,2]    [,3]
[1,]      3       7      11
[2,]      4       8      12
[3,]      5       9      13
[4,]     6       10      14
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

```
# Define the column and row names.
        rownames = c("row1", "row2", "row3", "row4")
        colnames = c("col1", "col2", "col3")

        P <- matrix(c(3:14), nrow =4, byrow = TRUE, dimnames = list(rownames, colnames))
        print(P)
```

```
# it produces the following result :
        col1    col2    col3
row1     3       4       5
row2     6       7       8
row3     9      10      11
row4    12      13      14
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

Accessing Elements of a Matrix:

- Elements of a matrix can be accessed by using the column and row index of the element.
- Use the **indexes (starting with 1) to access a row or a column** or an element.

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

```
# Define the column and row names.
        rownames = c("row1", "row2", "row3", "row4")
        colnames = c("col1", "col2", "col3")

# Create the matrix.
        P <- matrix(c(3:14), nrow =4, byrow = TRUE, dimnames = list(rownames, colnames))

# Access the element at 1st row and 3rd column.
    print(P[1,3])

# it produces the following result :
        5
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

# Access the element at 4th row and 2nd column.

```
print(P[4,2])
```

# it produces the following result :

```
13
```

# Access the only the 2nd row .

```
print(P[2,])
```

# it produces the following result :

```
col1   col2   col3
 6      7      8
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

# Access the only the 3rd column.

```
print(P[,3])
```

# it produces the following result :

```
row1   row2   row3   row4
 5      8      11     14
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

Creating a Matrix by joining multiple vectors:

- **cbind()** **and** **rbind()** both create matrices by combining several vectors of the same length.

- **cbind()** combines vectors as columns, while **rbind()** combines them as rows.

Exp. 3: Demonstrate Matrix operations in R

# Example:

```
x <- 1:5
y <- 6:10
z <- 11:15
```

# Create a matrix where x, y and z are columns

```
cbind(x, y, z)
```

# it produces the following result :

```
      x    y    z
[1,]  1    6    11
[2,]  2    7    12
[3,]  3    8    13
[4,]  4    9    14
[5,]  5    10   15
```

# Create a matrix where x, y and z are rows

```
rbind(x, y, z)
```

# it produces the following result :

```
   [,1] [,2] [,3] [,4] [,5]
x   1    2    3    4    5
Y   6    7    8    9    10
z   11   12   13   14   15
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

Matrix Computations:

- Various mathematical operations are performed on the matrices using the R operators. The result of the operation is also a matrix.

- The **dimensions (number of rows and columns) should be same** for the matrices involved in the operation.

- **Operations on Matrix:**
  - ➢ Addition
  - ➢ Subtraction
  - ➢ Division
  - ➢ Multiplication
  - ➢ Transpose of a Matrix
  - ➢ Creating an Identity Matrix
  - ➢ Returns the diagonal elements of Matrix
  - ➢ Dimensions of a Matrix
  - ➢ Determinant of a Matrix
  - ➢ Inverse of a Matrix

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

#Matrix Addition:

```
m=matrix(rep(2,9),nrow=3)
n=matrix(rep(3,9),nrow=3)
p=m+n
cat("Result of addition","\n")
print(p)
```

# it produces the following result :

```
        [,1]    [,2]    [,3]
[1,]    5        5       5
[2,]    5        5       5
[3,]    5       5        5
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

#Matrix Subtraction:

```
m=matrix(rep(2,9),nrow=3)
n=matrix(rep(3,9),nrow=3)
p=m-n
cat("Result of subtraction is","\n")
print(p)
```

# it produces the following result :

```
        [,1]    [,2]    [,3]
[1,]     -1      -1      -1
[2,]     -1      -1      -1
[3,]     -1      -1      -1
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

```
#Matrix Multiplication:
        m=matrix(rep(2,9),nrow=3)
        n=matrix(rep(3,9),nrow=3)
        p=m*n
        cat("Result of Multiplication is","\n")
        print(p)
```

```
# it produces the following result :

             [,1]    [,2]    [,3]
        [1,]  6       6       6
        [2,]  6       6       6
        [3,]  6       6       6
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

#Matrix Division:

```
m=matrix(rep(2,9),nrow=3)
n=matrix(rep(3,9),nrow=3)
p=m/n
cat("Result of Division is","\n")
print(p)
```

# it produces the following result :

```
          [,1]        [,2]        [,3]
[1,]   0.6666667   0.6666667   0.6666667
[2,]   0.6666667   0.6666667   0.6666667
[3,]   0.6666667   0.6666667   0.6666667
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

#Check the Matrix Equality:

```
m=matrix(c(1:9),nrow=3)
n=matrix(rep(3,9),nrow=3)
p=(m==n)
cat("Result of Division is","\n")
print(p)
```

# it produces the following result :

```
       [,1]    [,2]    [,3]
[1,]  FALSE   FALSE   FALSE
[2,]  FALSE   FALSE   FALSE
[3,]  TRUE    FALSE   FALSE
```

Exp. 3: Demonstrate Matrix operations in R

Example: Create a matrix taking a vector of numbers as input.

```
#To create an identity matrix
        p=diag(x,nrows, ncols)
        p=diag(3)
        print(p)
```

# it produces the following result :

```
        [,1]     [,2]    [,3]
[1,]     1        0       0
[2,]     0        1       0
[3,]     0        0       1
```

Exp. 3: Demonstrate Matrix operations in R

#To create an identity matrix

```
p=diag(3)
print(p)
```

# it produces the following result :

```
        [,1]    [,2]    [,3]
[1,]    1       0       0
[2,]    0       1       0
[3,]    0       0       1
```

#To find the determinant of a square matrix

```
p=diag(3)
print(p)
print(det(p))
```

# it produces the following result :

```
1
```

#Dimension of a matrix

```
m=matrix(c(1:9),nrow=3)
y=dim(m)
print(y)
```

# it produces the following result :

```
[1] 3 3
```