Matrix Operations in R

Mr. Sachin B.

Create matrix using Direct method

```
my_matrix <- matrix(1:20,nrow = 4)

print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

Use of nrow and ncol attribute

```
# Elements are arranged in 4 rows.
I <- matrix(c(1:12), nrow = 4)
print(I)
```

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

```
# Elements are arranged in 4 columns.
J <- matrix(c(1:12), ncol = 4)
print(J)
```

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

Use of byrow attribute

```
# Elements are arranged sequentially by column in 4 rows.
K <- matrix(c(1:12), nrow = 4, byrow = FALSE)
print(K)
```

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

```
# Elements are arranged sequentially by row in 4 rows.
L <- matrix(c(1:12), nrow = 4, byrow = TRUE)
print(L)
```

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

```
# Elements are arranged sequentially by column in 4 columns.
M <- matrix(c(1:12), ncol = 4, byrow = FALSE)
print(M)
```

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

```
# Elements are arranged sequentially by row in 4 columns.
N <- matrix(c(1:12), ncol = 4, byrow = TRUE)
print(N)
```

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

```
# Elements are arranged sequentially by column and in 3 rows and 4 columns.
P <- matrix(c(1:12), nrow = 3, ncol = 4, byrow = FALSE)
print(P)
```

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

If no. of rows and columns are mismatched.

```
# 12 elements and 9 (3*3) places...
Q <- matrix(c(1:12), nrow = 3, ncol = 3, byrow = FALSE)
print(Q)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
# 12 elements and 16 (4*4) places...
R <- matrix(c(1:12), nrow = 4, ncol = 4, byrow = FALSE)
print(R)
```

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

Adding Row and Column name to Matrix using dimnames attribute

```
# Define the column and row names.
rownames = c("row1", "row2", "row3", "row4")
colnames = paste("col",1:3,sep="")

S <- matrix(c(1:12), nrow = 4, byrow = TRUE, dimnames = list(rownames, colnames))
print(S)
```

```
##      col1 col2 col3
## row1    1    2    3
## row2    4    5    6
## row3    7    8    9
## row4   10   11   12
```

To get Structure of Matrix

Use of **dim()**,**nrow()**,**ncol()** and **length()** methods

```
# Create Matrix

my_matrix <- matrix(1:20,nrow = 4)
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```
# dim()
print(dim(my_matrix))
```

```
## [1] 4 5
```

```r
# nrow()
print(nrow(my_matrix))
```

```
## [1] 4
```

```r
# ncol()
print(ncol(my_matrix))
```

```
## [1] 5
```

```r
# length()
print(length(my_matrix))
```

```
## [1] 20
```

Access Elements

```r
# Create Matrix

my_matrix <- matrix(1:20,nrow = 4)
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

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

```
## [1] 9
```

```r
# Access the element at 2nd column and 4th row.
print(my_matrix[4,2])
```

```
## [1] 8
```

```r
# Access only the  2nd row.
print(my_matrix[2,])
```

```
## [1]  2  6 10 14 18
```

```r
# Access only the 3rd column.
print(my_matrix[,3])
```

```
## [1]  9 10 11 12
```

Modifying Matrix

```
# Create Matrix

my_matrix <- matrix(1:20,nrow = 4)
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```
# Re-assignment to change values

my_matrix[3,1] <- 100  # Modifying a Single Element
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]  100    7   11   15   19
## [4,]    4    8   12   16   20
```

```
my_matrix[my_matrix>15] <- 0  # modify elements less than 15
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13    0
## [2,]    2    6   10   14    0
## [3,]    0    7   11   15    0
## [4,]    4    8   12    0    0
```

```
# Re-assignment to change dimension
dim(my_matrix) <- c(5,4)
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11    0
## [2,]    2    7   12    0
## [3,]    0    8   13    0
## [4,]    4    9   14    0
## [5,]    5   10   15    0
```

Transpose, Combine and Deconstruction of Matrix

```
# Create Matrix
my_matrix <- matrix(1:9,nrow = 3)
print(my_matrix)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
# Transpose of Matrix
t_matrix <- t(my_matrix)
print(t_matrix)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
# Combine 2 Matrices
print(cbind(my_matrix,t_matrix))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    4    7    1    2    3
## [2,]    2    5    8    4    5    6
## [3,]    3    6    9    7    8    9
```

```
# Deconstruction 2 Matrices
print(c(my_matrix))
```

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

Matrix can hold only One type (Class) of data

```
# Create Matrix

my_matrix <- matrix(1:20,nrow = 4)
print(my_matrix)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```
# Create Character Vector

new_col <- c('A','B','C','D')
new_row <- c('P','Q','R','S','T')

# Add above Vector into Matrix
cbind(my_matrix,new_col)
```

```
##                              new_col
## [1,] "1" "5" "9"  "13" "17" "A"
## [2,] "2" "6" "10" "14" "18" "B"
## [3,] "3" "7" "11" "15" "19" "C"
## [4,] "4" "8" "12" "16" "20" "D"
```

```
rbind(my_matrix,new_row)
```

```
##         [,1] [,2] [,3] [,4] [,5]
##         "1"  "5"  "9"  "13" "17"
##         "2"  "6"  "10" "14" "18"
##         "3"  "7"  "11" "15" "19"
##         "4"  "8"  "12" "16" "20"
## new_row "P"  "Q"  "R"  "S"  "T"
```

All Numbers will coerced into Character as 'A','B',.. are Characters.

Convert Vector into Matrix

```
# Create Vector

my_vector <- 1:20
print(my_vector)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
# Check Vector Dimension [Vector does not have 'dim' attribute]
print(dim(my_vector))
```

```
## NULL
```

```
# Check Vector Length
print(length(my_vector))
```

```
## [1] 20
```

```
# Adding dimension to Vector so it will no longer be Vector
dim(my_vector)<-c(4,5)
print(my_vector)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

Check: It is Vector or not?

```r
#Check Class
print(class(my_vector))
```

```
## [1] "matrix" "array"
```

```r
# Check Dimension
print(dim(my_vector))
```

```
## [1] 4 5
```

```r
print(attributes(my_vector))
```

```
## $dim
## [1] 4 5
```