R Matrix Operations

Matrix

Matrix can be created by specifying number of rows and columns. “byrow = TRUE” means elements will be filled in row wise manner.

```
M = matrix( c('a','a','b','c','b','a'), nrow=2,ncol=3,byrow = TRUE)
print(M)
```

```
##      [,1] [,2] [,3]
## [1,] "a"  "a"  "b"
## [2,] "c"  "b"  "a"
```

If “byrow = TRUE” is not gievn then elements will be filled in column wise manner. In this example elements 3 to 14 are arranged in 4 rows.

```
M <- matrix(c(1:12), nrow = 4)
print(M)
```

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

```
M <- matrix(c(1:14), nrow = 4)
```

```
## Warning in matrix(c(1:14), nrow = 4): data length [14] is not a sub-multiple or
## multiple of the number of rows [4]
```

```
print(M)
```

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

In case rows or columns are not specified then, matrix with single column and no of rows equal to number of elements is created.

```
M <- matrix(c(3:14))
print(M)
```

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

Accessing elements of a Matrix

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

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

```
cat("\n Print element row 1 col 3 M[1,3]",M[1,3])
```

```
##
##  Print element row 1 col 3 M[1,3] 11
```

```
cat("\n Print row 1 M[1,]",M[1,])
```

```
##
##  Print row 1 M[1,] 3 7 11
```

```
cat("\n Print col 3 M[,3]",M[,3])
```

```
##
##  Print col 3 M[,3] 11 12 13 14
```

```
M[2,3] = 20  #Assigning value 20 to the element at 2nd row and 3rd column
cat("\n After assigning 20 to M[2,3] \n")
```

```
##
##  After assigning 20 to M[2,3]
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   20
## [3,]    5    9   13
## [4,]    6   10   14
```

```
M[3,] = 300 # This is vectorized ... all elements in row 3 will be 300
cat("\n After assigning 300 to row 3 \n")
```

```
##
##  After assigning 300 to row 3
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   20
## [3,]  300  300  300
## [4,]    6   10   14
```

```
M[,2] = 23.4
cat("\n After assigning 23.4 to column 2 \n")
```

```
##
##  After assigning 23.4 to column 2
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3 23.4   11
## [2,]    4 23.4   20
## [3,]  300 23.4  300
## [4,]    6 23.4   14
```

```
M <- matrix(c(3:14), nrow = 4)
print(M - 30) # create a copy of matrix
```

```
##      [,1] [,2] [,3]
## [1,]  -27  -23  -19
## [2,]  -26  -22  -18
## [3,]  -25  -21  -17
## [4,]  -24  -20  -16
```

```
print(M) # original matrix is still the same
```

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

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

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

```
M[2,2] = M[2,2] + 10
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4   18   12
## [3,]    5    9   13
## [4,]    6   10   14
```

```
M[3,] = M[3,] - 30
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4   18   12
## [3,]  -25  -21  -17
## [4,]    6   10   14
```

```
M[,1] = M[,1] * 0.1
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]  0.3    7   11
## [2,]  0.4   18   12
## [3,] -2.5  -21  -17
## [4,]  0.6   10   14
```

```
M = M + 10101
print(M)
```

```
##         [,1]  [,2]  [,3]
## [1,] 10101.3 10108 10112
## [2,] 10101.4 10119 10113
## [3,] 10098.5 10080 10084
## [4,] 10101.6 10111 10115
```

Change Matrix Elements based on Condition

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

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

```
print(M[M < 6])
```

```
## [1] 3 4 5
```

```
M[M < 6] = 100
print("Modified Array is ")
```

```
## [1] "Modified Array is "
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]  100    7   11
## [2,]  100    8   12
## [3,]  100    9   13
## [4,]    6   10   14
```

```
M[M < 100] = M[M < 100] * 20
print("Modified Array is ")
```

```
## [1] "Modified Array is "
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]  100  140  220
## [2,]  100  160  240
## [3,]  100  180  260
## [4,]  120  200  280
```

```
# modify elements in 2nd row to 200 if element is less than 10
M <- matrix(c(3:14), nrow = 4)
M[2,][M[2,] < 10] = 200
M
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]  200  200   12
## [3,]    5    9   13
## [4,]    6   10   14
```

Add new column to matrix

here we use 'cbind()' function

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

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

```
M = cbind(M, c(0,0,0,0))

print("Modified matrix")
```

```
## [1] "Modified matrix"
```

```
print(M)
```

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

```
M = cbind(M, c(50,51,52))
```

```
## Warning in cbind(M, c(50, 51, 52)): number of rows of result is not a multiple
## of vector length (arg 2)
```

```
print(M)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    3    7   11    0   50
## [2,]    4    8   12    0   51
## [3,]    5    9   13    0   52
## [4,]    6   10   14    0   50
```

```
M = cbind(M, -1)
print(M)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    3    7   11    0   50   -1
## [2,]    4    8   12    0   51   -1
## [3,]    5    9   13    0   52   -1
## [4,]    6   10   14    0   50   -1
```

Add new row to matrix

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

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

```
M = rbind(M , c(0))
print(M)
```

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

```
M = rbind(M , c(100,200,300,400))
```

```
## Warning in rbind(M, c(100, 200, 300, 400)): number of columns of result is not a
## multiple of vector length (arg 2)
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   12
## [3,]    5    9   13
## [4,]    6   10   14
## [5,]    0    0    0
## [6,]  100  200  300
```

```
M = rbind(M , c(-1,-2))
```

```
## Warning in rbind(M, c(-1, -2)): number of columns of result is not a multiple of
## vector length (arg 2)
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   12
## [3,]    5    9   13
## [4,]    6   10   14
## [5,]    0    0    0
## [6,]  100  200  300
## [7,]   -1   -2   -1
```

Reshape Matrix

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

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

```
print("Matrix dimensions are ")
```

```
## [1] "Matrix dimensions are "
```

```
print(dim(M))
```

```
## [1] 4 3
```

```
dim(M) = c(2,6)
print("New dimensions are")
```

```
## [1] "New dimensions are"
```

```
print(dim(M))
```

```
## [1] 2 6
```

```
# NOT Allowed
#dim(M) = c(2,2) # Less values in target
#print("New dimensions are")
#print(dim(M))
#dim(M) = c(1,14) # More values in target
#print("New dimensions are")
#print(dim(M))
M
```

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

Transpose of Matrix

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

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

```
print("transpose is")
```

```
## [1] "transpose is"
```

```
t(M)
```

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

Removing Elements from Matrix

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

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

```
print("Removing  third row")
```

```
## [1] "Removing  third row"
```

```
M[-3,]
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   12
## [3,]    6   10   14
```

```
print("Removing  third & 4 th row")
```

```
## [1] "Removing  third & 4 th row"
```

```
M[c(-3,-4),]
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   12
```

```
# Original matrix is not modified!!
print("Removing  Second column")
```

```
## [1] "Removing  Second column"
```

```
M[,-2]
```

```
##      [,1] [,2]
## [1,]    3   11
## [2,]    4   12
## [3,]    5   13
## [4,]    6   14
```

```
print("Removing  1 &3 column")
```

```
## [1] "Removing  1 &3 column"
```

```
M[,c(-1,-3)]
```

```
## [1]  7  8  9 10
```

```
print("Removing one row and one column ")
```

```
## [1] "Removing one row and one column "
```

```
M[-1,-2]
```

```
##      [,1] [,2]
## [1,]    4   12
## [2,]    5   13
## [3,]    6   14
```

Seach An element in Matrix Find index of an element in matrix

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

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

```
print("Index of 9")
```

```
## [1] "Index of 9"
```

```
which(M == 9)
```

```
## [1] 7
```

```
print("Elements excluding 9")
```

```
## [1] "Elements excluding 9"
```

```
M[-which(M == 9)] # remove single element from matrix
```

```
##  [1]  3  4  5  6  7  8 10 11 12 13 14
```

```
print("Index of even nos")
```

```
## [1] "Index of even nos"
```

```
which(M %% 2 ==0)
```

```
## [1]  2  4  6  8 10 12
```

```
print("Elements excluding even nos")
```

```
## [1] "Elements excluding even nos"
```

```
M[- which(M %%2 ==0)] # remove all even nos from the matrix
```

```
## [1]  3  5  7  9 11 13
```

Matrix Operations

```
M1 <- matrix(c(1:9), nrow = 3)
M2 <- matrix(c(11:19), nrow = 3)
cat("\n After addition \n")
```

```
##
##  After addition
```

```
print(M1+M2)
```

```
##      [,1] [,2] [,3]
## [1,]   12   18   24
## [2,]   14   20   26
## [3,]   16   22   28
```

```
cat("\n After Substraction \n")
```

```
##
##  After Substraction
```

```
print(M1-M2)
```

```
##      [,1] [,2] [,3]
## [1,]  -10  -10  -10
## [2,]  -10  -10  -10
## [3,]  -10  -10  -10
```

```
cat("\n After elementwise Multiplication \n")
```

```
##
##  After elementwise Multiplication
```

```
print(M1*M2)
```

```
##      [,1] [,2] [,3]
## [1,]   11   56  119
## [2,]   24   75  144
## [3,]   39   96  171
```

```
cat("\n After Matrix Multiplication \n")
```

```
##
##  After Matrix Multiplication
```

```
print(M1 %*% M2)
```

```
##      [,1] [,2] [,3]
## [1,]  150  186  222
## [2,]  186  231  276
## [3,]  222  276  330
```

```
cat("\n After Division \n")
```

```
##
##  After Division
```

```
print(M1/M2)
```

```
##            [,1]      [,2]      [,3]
## [1,] 0.09090909 0.2857143 0.4117647
## [2,] 0.16666667 0.3333333 0.4444444
## [3,] 0.23076923 0.3750000 0.4736842
```

```
cat("\n After Matrix Multiplication with scalar (constant)\n")
```

```
##
##  After Matrix Multiplication with scalar (constant)
```

```
print(M1 * 4)
```

```
##      [,1] [,2] [,3]
## [1,]    4   16   28
## [2,]    8   20   32
## [3,]   12   24   36
```