

Matrices

In this lesson, we will discuss matrices.

We'll cover the following

- Creating Matrices
 - Why Do We Need Matrices If We Have Arrays?
 - Syntax
- Accessing and Manipulating Matrices

Matrices are R objects where elements are arranged in a **two-dimensional** rectangular layout.

Like arrays, they contain elements of the same data type.

Creating Matrices

A Matrix is created using the `matrix()` function. The `matrix()` function takes an atomic vector as input. We can also define how many rows should be in the matrix by setting the `nrow` argument to a number. Furthermore, we can also set the `ncol` argument, which tells R how many columns to include in the matrix.

Why Do We Need Matrices If We Have Arrays?

Matrices in R language are only 2 dimensional. A matrix is just a more **convenient constructor**. There are many functions and methods especially mathematical and statistical methods that only accept `2D` arrays. So, to ensure that the programmer does not make the mistake of making the `2D` array an `nD` array, a convenient object specifically for this task **matrix** is used.

Syntax

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

Here, data is the input **vector**, `nrow` is the number of rows to be created, `ncol` is the number of columns to be created, `byrow` is a **logical clue**: which if set to `TRUE`

arranges elements **row-wise** and `dimnames` is the parameter that lets us assign names to the rows and columns.

Let's have a look at the code to create an empty matrix:

```
myMatrix <- matrix(nrow = 2, ncol = 2)
print(myMatrix)
```



Creating an empty matrix

Let's use a populated vector to create a matrix:

```
myVector <- c(1, 2, 3, 4)
myMatrix <- matrix(myVector, nrow = 2, ncol = 2)
print(myMatrix)
```



Creating a simple matrix

Let's play around with the `matrix()` function.

 `byrow=TRUE`

 `byrow=FALSE`

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



Manipulating the byrow parameter

We can also set the names of **rows** and **columns**. In the following code snippet, we name rows from `r1...r4` and name columns from `c1...c3`.

```
# Define the column and row names.
rownames = c("r1", "r2", "r3", "r4")
colnames = c("c1", "c2", "c3")
myMatrix <- matrix(c(1:12), nrow = 4, ncol = 3, byrow = TRUE, dimnames = list(rownames, colnames))
print(myMatrix)
```





Setting names for rows and columns

Accessing and Manipulating Matrices

Elements of a matrix can be accessed and updated by using the row and column index of the element inside square brackets `[]`.

```
myMatrix <- matrix(c(1:12), nrow = 4, byrow = TRUE)
print(myMatrix[2, 3])
```



Accessing element at row 2 and column 3

Changing the value of an element in a matrix can be done by first accessing the element using square brackets and then assigning it a different value or by using the column-wise single index.

 Using_Row_Column_Index

 Using_Single_Index

```
myMatrix <- matrix(c(1:12), nrow = 4, byrow = TRUE)
myMatrix[2, 3] = 0
print(myMatrix)
```



Changing element at row 2 and column 3

Let's test your concepts on **arrays and matrices** with a small exercise.