Additional Material for Creating Matrices using Data Frames

Sudhakar Kumar

1 March 2019

Creating vectors in R

A **vector** is a sequence of data elements of the same type. In **R** language, vectors are created using the `c` function. For example, suppose we want to create a vector with its elements as `(1,2,3,4)`. Then, we use the following command

```
c(1,2,3,4)
```

```
## [1] 1 2 3 4
```

We can also use the command as follows

```
c(1:4)
```

```
## [1] 1 2 3 4
```

If we want to create a vector of consecutive numbers, the `:` operator comes very handy. Hence, for creating a vector with elements from 1 to 4, we can type the command as follows

```
vec <- 1:4
vec
```

```
## [1] 1 2 3 4
```

Since, a vector must have data elements of the same type, this function `c` will try and coerce elements to the same type, if they are different.

Coercion is from lower to higher types from logical to integer to double to character (DataMentor 2018). For example, if we create a vector with data elements of different type as given below

```
vec <- c(1, 5.4, TRUE, "hello")
vec
```

```
## [1] "1"     "5.4"   "TRUE"  "hello"
```

As we see that we have inserted different data types like numeric, logical and character in this vector `vec`. However, **R** language will coerce all the elements to character type. Let us find out the data type of our vector `vec` (using the `class` or `typeof` function).

```
class(vec)
```

```
## [1] "character"
```

More complex sequences can be created using the `seq` function, like defining the number of points in an interval, or the step size. For example, we want to create a vector with elements from 0 to 5 with a step size of 0.5. For this, we type the command as given below

```
seq(from = 0, to = 5, by=0.5)
```

```
##  [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Creating matrices using vectors

Matrix can be created using the `matrix` function. Dimension of the matrix can be defined by passing appropriate value for arguments `nrow` and `ncol`. Providing values for both dimension is not necessary. If one of the dimension is provided, the other is inferred from length of the data. Suppose we want to create a 3×3 matrix with numbers from 1 to 9, we type the command as follows

```
matrix(data = 1:9, nrow = 3, ncol = 3)
```

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

Here, we can see that the numbers from 1 to 9 has been arranged column wise. If we want the numbers (1 to 9) to be arranged in row wise format, we type the command as given below

```
matrix(data = 1:9, nrow = 3, ncol = 3, byrow = TRUE)
```

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

Consider two different vectors as given below

```
vec1 <- c(9:12)
vec2 <- c(13:16)
vec1; vec2
```

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

```
## [1] 13 14 15 16
```

Now, we want to create a 4×2 matrix using these two vector `vec1` and `vec2`. For this, we will first create a single vector by joining these two vectors, as given below

```
vec_final <- c(vec1, vec2)
vec_final
```

```
## [1]  9 10 11 12 13 14 15 16
```

Then, we use the `matrix` function and apply the arguments as given below:

- **data** = `vec_final`
- **nrow** = 4
- **ncol** = 2

So, we type the command as given below

```
mat <- matrix(data = vec_final, nrow = 4, ncol = 2)
mat
```

```
##      [,1] [,2]
## [1,]    9   13
## [2,]   10   14
## [3,]   11   15
## [4,]   12   16
```

References

DataMentor. 2018. “R Vector: Create, Modify and Access Vector Elements.” https://www.datamentor.io/r-programming/vector/.