Matrices in R

Initialize a matrix

We can make a 3x4 matrix of zeroes with the following code:

```
matrix(0, 3, 4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
```

We can initialize a matrix using a vector of values. Note: if the size of the vector is greater or smaller than the number of cells in the matrix, a warning will be received, but the matrix will be anyway initialized:

```
m_init = 1:15
m = matrix(m_init, 3, 4)
```

```
## Warning in matrix(m_init, 3, 4): la longitud de los datos [15] no es un
## submúltiplo o múltiplo del número de columnas [4] en la matriz
```

```
print(m)
```

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

Change the dimension of a matrix / vector

We can transform a vector into a matrix with the `dim()` function. We simply assign a vector that will act as its new dimension

```
dim(m) = c(4,3)
print(m)
```

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

Accessing matrix values

Similar to vectors, they can be easily accessed with th "[]" operator

```
print(m[1,3])
```

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

Note: take care when using indexes out of rnge, R doesn't provide a warning an works a bit unconventional (both the follwoing examples should fail).

```
print(m[-1,3])
```

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

```r
print(m[0,2])
```

```
## integer(0)
```

We can get an entire row of the matrix by omitting the column index (but keeping the comma).

```r
print(m[1,])
```

```
## [1] 1 5 9
```

We can also retrieve multiple rows or columns by providing a vector or sequence with their indices.

```r
print(m[,2:3])
```

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