

Linguagem R de programação Estatística

Professor Eduardo Monteiro de Castro Gomes

Matrizes

As matrizes são objetos com comportamento semelhante aos vetores mas apresentam duas dimensões organizando seus elementos em linhas e colunas

```
mat1 <- matrix(data = 1:12, nrow = 3)
mat1
```

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

na definição da matriz acima foram definidos os valores que compõe a matriz e foi definida uma das dimensões da matriz. A outra dimensão não precisou ser definida pois as matrizes em R sempre devem ser completas em sua forma retangular e assim se existem 12 elementos dispostos em 3 linhas consequentemente serão utilizadas 4 colunas. Nos casos em que as dimensões e a quantidade valores não for equivalente o princípio de reciclagem será utilizado para que a matriz tenha a forma retangular e seja completa com elementos. Nos casos em que a reciclagem não é completa uma mensagem de aviso é passada.

```
mat2 <- matrix(1:5, nrow = 2)
```

```
## Warning in matrix(1:5, nrow = 2): data length [5] is not a sub-multiple or
## multiple of the number of rows [2]
```

```
mat2
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    1
```

Note que os números das matrizes foram preenchidos por colunas, mas pode-se optar por fazer o preenchimento por linhas utilizando o argumento *byrow = TRUE* na definição da matriz

```
mat3 <- matrix(data = 1:12, nrow = 3, byrow = TRUE)
mat3
```

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

Os princípios de reciclagem se aplicam também para as operações com matrizes conforme ilustrado a seguir

```
mat3 * 10
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   10   20   30   40
## [2,]   50   60   70   80
## [3,]   90  100  110  120
```

```
mat3 * c(-10,10,1000)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  -10  -20  -30  -40
## [2,]   50   60   70   80
```

```
## [3,] 9000 10000 11000 12000
```

A indexação dos elementos na matrix segue o mesmo princípio de endereçamento considerado para os vetores, mas agora com um endereço de linha e outro de coluna na forma [linha , coluna]. Por exemplo, o elemento que está na segunda linha e terceira coluna da matriz *mat3*, que é o número 7, pode ser acessado por:

```
mat3[ 2 , 3 ]
```

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

Podendo-se omitir o endereço de uma das dimensões para indicar o acesso a toda uma linha ou coluna. No exemplo a seguir é selecionada toda a quarta coluna da matriz *mat3* omitindo-se o endereço de linha e selecionando a quarta coluna.

```
mat3[,4]
```

```
## [1] 4 8 12
```

É possível utilizar vetores para indicar endereços de linhas e colunas e assim para selecionar a primeira e terceira linha da matriz *mat3* pode-se utilizar um vetor conforme o exemplo

```
mat3[ c(1,3) , ]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    9   10   11   12
```

As matrizes também são objetos homogêneos, de forma que todos os elementos contidos em uma matriz devem ser do mesmo tipo. Assim como ocorre para os vetores a linguagem faz a coerção dos elementos inseridos para que todos sejam conforme no sentido de serem do mesmo tipo.

É muito comum que as informações armazenadas contenham tipos diferentes de informações envolvendo números, nomes, categorias. Para armazenar informações de diferentes tipos em um mesmo objeto pode se utilizar um *data.frame*.

Dataframe

O dataframe é um objeto semelhante a matriz por ter duas dimensões sendo elas linhas e colunas mas o dataframe tem a propriedade que cada coluna pode ser de um tipo diferente. Pode-se considerar que um dataframe é um agrupamento de vetores em colunas, onde cada coluna deve ser homogênea e possuir elementos de um único tipo.

Nesse contexto, de uma forma geral, as linhas representam as observações ou indivíduos e as colunas representam as diferentes variáveis coletadas para cada observação ou indivíduo.

O exemplo a seguir ilustra um dataframe que armazena informações médicas de personagens fictícios.

```
tabela_medica <- data.frame(nome = c("Ana", "Bia", "Carol", "Daniela", "Fernanda"),
                             idade = c(30, 40, 50, 60, 70),
                             numeroconsultas = c(3, 7, 1, 6, 2),
                             tem_plano = c(TRUE, FALSE, FALSE, TRUE, TRUE))
```

```
tabela_medica
```

```
##      nome idade numeroconsultas tem_plano
## 1    Ana   30              3      TRUE
## 2    Bia   40              7      FALSE
## 3  Carol   50              1      FALSE
## 4 Daniela   60              6      TRUE
## 5 Fernanda  70              2      TRUE
```

O acesso aos elementos do dataframe podem ser feitos de forma semelhante ao acesso na matriz, pelo endereço de linha e coluna [linha,coluna] e as diferentes colunas podem também ser acessadas por seus nomes de duas formas conforme os seguintes exemplos:

```
tabela_medica$idade
```

```
## [1] 30 40 50 60 70
```

```
tabela_medica[,c("nome", "tem_plano")]
```

```
##      nome tem_plano
## 1    Ana      TRUE
## 2    Bia     FALSE
## 3   Carol     FALSE
## 4 Daniela     TRUE
## 5 Fernanda    TRUE
```

- Ordenação do data.frame

É comum o interesse em ordenar os dados por uma variável de interesse. Deve-se observar que é fundamental preservar as informações de cada um dos indivíduos. Deve-se determinar a ordem em que as linhas do data.frame será apresentado e a função *order* será utilizada. No exemplo seguinte a tabela médica é ordenada pelo número de consultas, de forma que as pessoas com mais consultas são apresentadas no topo da tabela.

```
tabela_medica
```

```
##      nome idade numeroconsultas tem_plano
## 1    Ana    30                3      TRUE
## 2    Bia    40                7     FALSE
## 3   Carol    50                1     FALSE
## 4 Daniela    60                6      TRUE
## 5 Fernanda    70                2      TRUE
```

```
tabela_medica[order(tabela_medica$numeroconsultas, decreasing = TRUE),]
```

```
##      nome idade numeroconsultas tem_plano
## 2    Bia    40                7     FALSE
## 4 Daniela    60                6      TRUE
## 1    Ana    30                3      TRUE
## 5 Fernanda    70                2      TRUE
## 3   Carol    50                1     FALSE
```

Todas as colunas de um dataframe devem ter a mesma dimensão ou número de elementos.