

Matrizes:

O que é uma matriz? Matriz é uma estrutura de dado, em que você consegue armazenar um conjunto ordenado de dados de **MESMO TIPO** e se refere a cada dado pela sua **posição** (Linha e Coluna).

O vetor é um tipo de matriz, porém é de apenas uma dimensão.

Na matriz abaixo, temos 3 linhas e 4 colunas.

Para definir esta matriz no R, devemos fazer o seguinte:

Posição	1	2	3	4
1	10	-5	8	7
2	0	5	6	2
3	1	-7	11	4

```
RGui (64-bit)
Arquivo Editar Pacotes Janelas Ajuda

R Console
> matA <- matrix(c(10, 0, 1, -5, 5, -7, 8, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
> matA
     [,1] [,2] [,3] [,4]
[1,]  10   -5   8   7
[2,]   0   5   6   2
[3,]   1  -7  11   4
> |

Sem nome - Editor R
matA <- matrix(c(10, 0, 1, -5, 5, -7, 8, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
matA
```

Adicionando outra matriz, matB.

```
RGui (64-bit)
Arquivo Editar Pacotes Janelas Ajuda

R Console
> matA <- matrix(c(10, 0, 1, -5, 5, -7, 8, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
> matA
     [,1] [,2] [,3] [,4]
[1,]  10   -5   8   7
[2,]   0   5   6   2
[3,]   1  -7  11   4
> matB <- matrix(c(1:12), nrow = 3, ncol = 4)
> matB
     [,1] [,2] [,3] [,4]
[1,]   1   4   7  10
[2,]   2   5   8  11
[3,]   3   6   9  12
> |

Sem nome - Editor R
matA <- matrix(c(10, 0, 1, -5, 5, -7, 8, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
matA
matB <- matrix(c(1:12), nrow = 3, ncol = 4)
matB
```

Podemos utilizar os operadores para realizar as operações com matrizes.

Adição	<code>matA + matB</code>
Subtração	<code>matA - matB</code>
Multiplicação	<code>matA %*% matB</code>
Transposta	<code>t(matA)</code> ou <code>t(matB)</code>
Inversa	<code>solve(matA)</code> ou <code>solve(matB)</code>

Aplicação dos operadores de adição e subtração de matrizes.

```
R Console
> matA <- matrix(c(10, 0, 1, -5, 5, -7, 9, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
> matA
      [,1] [,2] [,3] [,4]
[1,]  10   0   1  -5
[2,]   5   5  -7   9
[3,]   6  11   7   2
> matB <- matrix(c(1:12), nrow = 3, ncol = 4)
> matB
      [,1] [,2] [,3] [,4]
[1,]   1   4   7  10
[2,]   2   5   8  11
[3,]   3   6   9  12
> matA + matB
      [,1] [,2] [,3] [,4]
[1,]  11  -1  15  17
[2,]   7  10  14  13
[3,]   9   7  20  16
> matA - matB
      [,1] [,2] [,3] [,4]
[1,]   9  -9   1  -3
[2,]  -2   0  -2  -9
[3,]  -2 -13   2  -8
~|

Sem nome - Editor R
matA <- matrix(c(10, 0, 1, -5, 5, -7, 9, 6, 11, 7, 2, 4), nrow = 3, ncol = 4)
matA
matB <- matrix(c(1:12), nrow = 3, ncol = 4)
matB
matA + matB
matA - matB
```

Aplicação do operador multiplicação e transposta.

```
R Console
> matC <- matrix(c(1:9), nrow = 3, ncol = 3)
> matD <- matrix(c(9:1), nrow = 3, ncol = 3)
> matC
      [,1] [,2] [,3]
[1,]   1   4   7
[2,]   2   5   8
[3,]   3   6   9
> matD
      [,1] [,2] [,3]
[1,]   9   6   3
[2,]   8   5   2
[3,]   7   4   1
> matC %*% matD
      [,1] [,2] [,3]
[1,]  90  54  18
[2,] 114  69  24
[3,] 138  84  30
> t(matC)
      [,1] [,2] [,3]
[1,]   1   2   3
[2,]   4   5   6
[3,]   7   8   9
>

Sem nome - Editor R
matC <- matrix(c(1:9), nrow = 3, ncol = 3)
matD <- matrix(c(9:1), nrow = 3, ncol = 3)
matC
matD
matC %*% matD
t(matC)
```

Aplicação da inversa.

```
R Console
> matE <- matrix(c(1,3,-2,5,7,2,0,1,1), nrow = 3, ncol = 3)
> matF <- matrix(c(1.5,2,-3,4,10,-2,0,1.6,10), nrow = 3, ncol = 3)
> matE
      [,1] [,2] [,3]
[1,]   1   5   0
[2,]   3   7   1
[3,]  -2   2   1
> matF
      [,1] [,2] [,3]
[1,]  1.5   4  0.0
[2,]  2.0  10  1.6
[3,] -3.0 -2 10.0
> solve(matE)
      [,1] [,2] [,3]
[1,] -0.25  0.25 -0.25
[2,]  0.25 -0.05  0.05
[3,] -1.00  0.60  0.40
> solve(matF)
      [,1] [,2] [,3]
[1,]  1.8561151 -0.7194245  0.11510791
[2,] -0.4460432  0.2697842 -0.04316547
[3,]  0.4676259 -0.1618705  0.12589928
>

Sem nome - Editor R
matE <- matrix(c(1,3,-2,5,7,2,0,1,1), nrow = 3, ncol = 3)
matF <- matrix(c(1.5,2,-3,4,10,-2,0,1.6,10), nrow = 3, ncol = 3)
matE
matF
solve(matE)
solve(matF)
```

Para saber a dimensão de uma matriz, basta digitar o comando “dim(objeto)”.

```

R Console
> novamatriz <- matrix(c(1,2,3,-1,-5,6,2,7,10), nrow = 3, ncol = 3)
> novamatriz
      [,1] [,2] [,3]
[1,]    1  -1    2
[2,]    2  -5    7
[3,]    3    6   10
> novamatriz[2,]
[1] 2 -5 7
> novamatriz[,2]
[1] -1 -5 6
> novamatriz[2,2]
[1] -5
> dim(novamatriz)
[1] 3 3

```

```

Sem nome - Editor R
novamatriz <- matrix(c(1,2,3,-1,-5,6,2,7,10), nrow = 3, ncol = 3)
novamatriz      Impressão de toda a matriz
novamatriz[2,]  Impressão da linha 2.
novamatriz[,2]  Impressão da coluna 2.
novamatriz[2,2] Impressão do elemento na posição: linha 2 e coluna 2.
dim(novamatriz) Dimensão da matriz.

```

A vantagem de trabalhar com matrizes é que podemos armazenar muitas informações em apenas um objeto, porém essas informações DEVEM ser do MESMO TIPO de dados. Para armazenar VÁRIOS TIPOS de dados não podemos mais utilizar o conceito de matriz e sim o conceito de data.frame.

Para armazenar valores em um data.frame:

```

R Console
> m1 <- matrix(c(1,2.5,3.2,-3.5), nrow = 2, ncol = 2)
> m2 <- matrix(c("1","2","3","4"), nrow = 2, ncol = 2)
> m3 <- matrix(c(1:20), nrow = 4, ncol = 5)
>
> k <- data.frame(m1,m2,m3)
> k
  X1  X2 X1.1 X2.1 X1.2 X2.2 X3 X4 X5
1 1.0 3.2 1 3 1 5 9 13 17
2 2.5 -3.5 2 4 2 6 10 14 18
3 1.0 3.2 1 3 3 7 11 15 19
4 2.5 -3.5 2 4 4 8 12 16 20
>

```

```

Sem nome - Editor R
m1 <- matrix(c(1,2.5,3.2,-3.5), nrow = 2, ncol = 2)
m2 <- matrix(c("1","2","3","4"), nrow = 2, ncol = 2)
m3 <- matrix(c(1:20), nrow = 4, ncol = 5)

k <- data.frame(m1,m2,m3)
k

```

Em resumo: A principal diferença é que em um “data.frame”, podemos armazenar dados de diferentes tipos e em matrizes devem ser do mesmo tipo!

Exercícios:

Este é o momento de fixação, portanto **PRATIQUE!**

- 1) Qual a principal diferença entre uma matriz e um vetor?
- 2) Escreva a sintaxe para ler uma matriz de 3 linhas e 4 colunas (iniciando em 1 e terminando em 12).
- 3) Escreva a mesma matriz do número 2, em uma forma reduzida.
- 4) Insira duas matrizes genéricas, sendo as dimensões (linha x coluna). matriz1(2x3) e matriz2(2x3) e efetue as seguintes operações:
 - a) Some as duas matrizes.
 - b) Subtraia a matriz1 da matriz2.
 - c) É possível fazer a operação de multiplicação das matrizes?
 - d) Então, atualize a matriz2, transformando ela em sua **transposta**.
 - e) Qual o operador para efetuar a multiplicação de matrizes?
 - f) Agora efetue a multiplicação da matriz1 com a matriz2.
 - g) Agora efetue a multiplicação da matriz2 com a matriz1.
 - h) É possível calcular a inversa da matriz1 e da matriz2?

- i) Então, crie uma matriz de ordem quadrada ($n \times n$) e calcule a sua inversa.
 - j) Escreva os operadores para se trabalhar com matrizes: adição, subtração, multiplicação, inversa e transposta.
- 5) Como você pode verificar a dimensão de uma matriz?
 - 6) Qual a principal diferença entre trabalhar com **matriz** e **data.frame**?
 - 7) Como você faria uma busca de um valor dentro de uma matriz(4 x 5):
 - a) Toda a linha 2
 - b) Toda a coluna 3
 - c) O elemento matriz(4,4).

Respostas:

- 1) Um vetor é uma matriz de apenas uma dimensão. As matrizes de mais de uma dimensão, possuem linhas e colunas, em que é possível armazenar valores em cada posição (linha e coluna), sendo sempre valores de mesmo tipo.
- 2) `lendoMatriz <- matrix(c(1,2,3,4,5,6,7,8,9,10,11,12), nrow = 3, ncol = 4)`
`print(lendoMatriz)`
- 3) `lendoMatriz <- matrix(c(1:12), nrow = 3, ncol = 4)`
`print(lendoMatriz)`
- 4) a) e b)

The screenshot shows two windows from the R environment. The 'R Console' window displays the following commands and their outputs:

```
> matriz1 <- matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3)
> matriz2 <- matrix(c(-1,4,5,7,0,2), nrow = 2, ncol = 3)
> matriz1 + matriz2
      [,1] [,2] [,3]
[1,]  0    8    5
[2,]  6   11    8
> matriz1 - matriz2
      [,1] [,2] [,3]
[1,]  2   -2    5
[2,] -2   -3    4
> |
```

The 'Sem nome - Editor R' window shows the same code being typed:

```
matriz1 <- matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3)
matriz2 <- matrix(c(-1,4,5,7,0,2), nrow = 2, ncol = 3)
matriz1 + matriz2
matriz1 - matriz2
```

- c) Não é possível, pois as matrizes precisam ter a seguinte propriedade. matriz1($m \times n$) e matriz2 ($n \times p$), ou seja, o número de coluna da primeira matriz, deve ser igual ao número de linhas da segunda matriz a ser multiplicada. (Cuidado que o operador de matriz é o "%*%" e não o "*").
- d) `matriz2 <- t(matriz2)`
- e) `%*%`
- f) `matriz1 %*% matriz2`
- g) `matriz2 %*% matriz1`
- h) Não, pois para uma matriz admitir inversa, ela precisa ser de ordem quadrada (mesmo número de linhas e colunas).
- i) `matriz3 <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)`
`solve(matriz3)`
- #Uma matriz só admite inversa, se o determinante for diferente de zero.
- j) `+`, `-`, `%*%`, `solve(objeto)`, `t(objeto)`.

- 5) `dim(objetomatriz)`

- 6) Em matrizes, podemos armazenar apenas os dados com o mesmo tipo em posições (linhas x colunas) e em data.frame podemos armazenar em cada posição diferentes tipos de dados, inclusive podemos armazenar um vetor/matriz dentro de cada posição do data.frame.
- 7) a) matriz[2,]
b) matriz[,3]
c) matriz[4,4]