

Programação no R - Aula 5

Disciplina: Lógica da programação (de computadores) e
análise de dados no R

Prof. Maurício Garcia de Camargo. IO-FURG.

2025-10-03



Revisão de vetores

Os cinco fundamentos da programação são:

1. Function (abstração)

```
1 f_soma = function(n1,n2) {  
2   print(paste('Soma = ',n1+n2))  
3 }  
4 f_soma(7,5)
```

2. IF

3. ELSE

4. FOR

5. WHILE



Revisão de vetores

Criando vetores

```
1 v1 = c(2, 4, 5, 7, 11) # Criando um vetor numérico
2 v2 = c('A', 'B', 'C')  # Criando um vetor de strings
3 v3 = 1:5                # Sequência simples
4 v4 = rep(12, 5)         # Função rep() para criar um vetor
```

Funções vetoriais

```
1 length(v1) # Número de elementos do vetor
2 sum(v1)    # Soma dos elementos do vetor
```

Índices de vetores

```
1 x = c(3, 8, 5, 12, 4)
2 x[1]                # Primeiro o elemento
3 x[2:5]              # Do segundo ao quinto elemento
4 x[c(2, 4)]          # Segundo E quarto elemento
5 ##### Atribuir valores a um elemento do vetor #####
6 x[3] = 99           # Atribuir um elemento individual
7 x = c(x, 33)        # Concatenar um elemento a um vetor
```



Novidades sobre vetores

```
1 x = c(0, 3, 8, 5, 12, 4)
```

```
2 x[-1] # Todos, exceto o primeiro
```

```
[1] 3 8 5 12 4
```

```
1 x[c(-2, -3)] # Todos, exceto o segundo e o terceiro
```

```
[1] 0 5 12 4
```

```
1 x[x >= 7] # Filtragem lógica
```

```
[1] 8 12
```

```
1 x[(x < 5) & (x != 0)] # Filtragem lógica com operadores booleanos
```

```
[1] 3 4
```



Matrizes

Um vetor numérico pode ser compreendido como uma linha (ou coluna) de números.

```
1 x = c(1, 2, 3, 4, 5)
2 x
```

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

Uma matriz é uma junção de dois ou mais vetores, que podem ser combinados usando a função **rbind** para unir linhas ou **cbind** para unir colunas.

```
1 m = rbind(c(1, 2, 3), c(4, 5, 6))
2 m
```

```
      [,1] [,2] [,3]
[1,]     1     2     3
[2,]     4     5     6
```

Existem muitas outras maneiras de contruir matrizes.



Transformando um vetor em matriz

Seja o vetor v:

```
1 v = c(1, 2, 3, 4, 5, 6)
2 v
```

```
[1] 1 2 3 4 5 6
```

Para transformá-lo em matriz:

```
1 m = matrix(v, nrow = 2)
2 m
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Repare nos índices das linhas e colunas.



Índices de matrizes

Exemplo de matriz com 4 linhas e 3 colunas:

```
1 v = 1:12
```

```
2 v
```

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

```
1 m = matrix(v, nrow = 4)
```

```
2 m
```

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

O índice da matriz é dados por $m[\text{linha}, \text{coluna}]$

Vetor tem 1 dimensão (*linhas*).

Matriz tem 2 dimensões (*linhas e colunas*).



Mais sobre índices de matrizes

```
1 m
```

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

```
1 ### Para selecionar a terceira linha inteira:
```

```
2 m[3,]
```

```
[1] 3 7 11
```

```
1 ### Para selecionar a segunda coluna inteira:
```

```
2 m[,2]
```

```
[1] 5 6 7 8
```

```
1 ### Para selecionar as 3 primeiras linhas da segunda coluna:
```

```
2 m[1:3,2]
```

```
[1] 5 6 7
```



Exemplos de índices

A lógica de utilização de índices é a mesma de vetores, porém com duas dimensões.

```
1 m
```

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

```
1 m[2,3] #Extrai a segunda linha e a terceira coluna
```

```
[1] 10
```

```
1 m[c(1,3),c(2,3)] #Extrai a primeira e terceira linha e segunda e quarta col
```

	[,1]	[,2]
[1,]	5	9
[2,]	7	11



Exercício de índices em matrizes

Exercício 1. Seja a matriz **m**, criada da seguinte maneira:

```
1 m = matrix(1:8,nrow = 4)
2 m
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

- A. Extraia o valor da primeira linha e primeira coluna.
- B. Extraia o valor da última linha e última coluna.
- C. Extraia todos os valores da primeira linha.
- D. Extraia todos os valores da última coluna.
- E. Extraia os segundo e terceiro valores das linha da primeira coluna.



Back to programming - lembrando o laço FOR

O laço **FOR** deve ser usada quando o número exato de repetições é conhecido. Utiliza uma *variável de controle* que deve ser do tipo número inteiro.

Esqueleto da estrutura de repetição **FOR** é:

```
1 #for (i in itens) {  
2 #   #Instruções que dependem de "i" (ou qualquer #outro nome de variável)  
3 }
```

Onde *i* é uma variável de controle que serve para representar qualquer um dos valores de *itens*, que pode ser um vetor, por exemplo.



Usando FOR em matrizes

Para percorrer todos os elementos da matriz, é preciso usar dois laços FOR, um para o número de linhas (**nrow**) e outro para o número de colunas (**ncol**):

```
1 m = matrix(1:8,nrow = 4)
2 m
```

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

```
1 for (i in 1:nrow(m))
2   for (j in 1:ncol(m))
3     print(m[i,j])
```

```
[1] 1
[1] 5
[1] 2
[1] 6
[1] 3
[1] 7
[1] 4
[1] 8
```



Exercícios com matrizes

Exercício 2.

Crie uma matriz **m** com elementos sequenciais de 1 até 16 distribuídos em 4 linhas.

Exercício 3.

Substitua na matriz **m** o elemento da linha 2 e coluna 3 pelo valor de zero.



Exercícios com matrizes

Exercício 4.

Com a mesma matriz m , faça um laço FOR para acumular a soma de todos os elementos da matriz.

Exercício 5.

Com a mesma matriz m , faça um laço que irá acumular (em um vetor vazio) a soma de cada linha.



Exercícios com matrizes

Exercício 6.

Com a matriz **m**, conte o número de elementos que são maiores do que 12.

Exercício 7.

Crie uma função que receba uma matriz e devolva um vetor contendo os valores da diagonal principal (quando $i=j$).



Exercícios com matrizes

Exercício 8 (DESAFIO).

Crie uma função que receberá dois números (a e b) e imprima a tabuada de a até b.

Por exemplo: imprimir a tabuada de 1 até 10.

“Tabuada do 1”

“1 x 1 = 1”

“1 x 2 = 2”

... “Tabuada do 2”

“2 x 1 = 2”

“2 x 2 = 4”

... etc até a tabuada do 10.



