

Introdução à Computação (MAC113 - FEA – 2020)

Ana C V de Melo

Tipos Compostos: matrizes

[Aulas Anteriores](#)

[Matrizes](#)

[Data-frames](#)

Aulas Anteriores

O que já vimos

- ▶ Como ler e gravar dados em um arquivo
- ▶ Tipos simples: inteiros, reais, caracteres
- ▶ Tipos Compostos. Vetores e Listas: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos.

O que iremos ver nas próximas aulas

Tipos Compostos:

- ▶ Matrizes: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos.
- ▶ Data-frames: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos.

Matrizes

Lembrando Vetores...

- ▶ Criar vetores

```
v1 <- c(1,6,3) # sequência de valores
print(v1)
```

```
## [1] 1 6 3
```

```
v2 <- 1:5 # valores consecutivos
print(v2)
```

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

```
vc <- c("azul", "vermelho", "amarelo")
print(vc)
```

```
## [1] "azul" "vermelho" "amarelo"
```

- ▶ linear (uma sequência de valores - unidimensional)

Para que queremos Matrizes - bidimensional

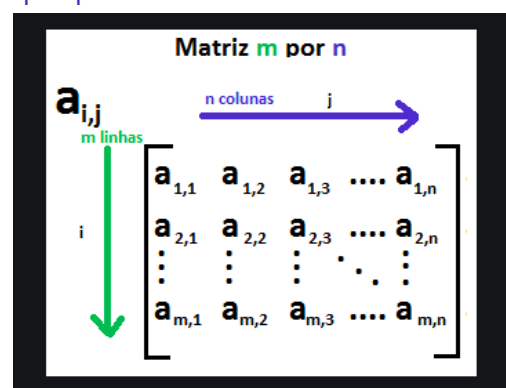


Figure 1:

Matrizes - Criar

- conjunto de elementos formado por linhas X colunas

```
m1 <- matrix(c(1,30,10,5)) # uma única coluna
print(m1)

##      [,1]
## [1,]    1
## [2,]   30
## [3,]   10
## [4,]    5

m2<- matrix(10:13,nrow = 2) # duas linhas - o número
print(m2)                  # de colunas é calculado

##      [,1] [,2]
## [1,]   10  12
## [2,]   11  13
```

Criar Matrizes - faltam valores

- e quando faltam valores?

```
m4 <- matrix(c(1:4,8:10), nrow = 3)

## Warning in matrix(c(1:4, 8:10), nrow = 3): data length [7] is not
## multiple or multiple of the number of rows [3]

print(m4)

##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2

► começa a repetir os valores iniciais ...
```

Acesso a 1 elemento da matriz - não existe

```
m4[4,1] # linha 4, coluna 1 (elemento não existe)

Error in m4[4, 1] : subscript out of bounds
```

Seleciona alguns elementos da matriz

```
print(m4)

##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2

m4[2:3, 1:3] # linhas 2 a 3, cols 1 a 3

##      [,1] [,2] [,3]
## [1,]    2    8    1
## [2,]    3    9    2
```

Criar Matrizes - a partir de vetores existentes

- Matrizes

```
print(v1)

## [1] 1 6 3

m3 <- matrix(c(v1,4:8),ncol = 4) # quatro colunas - número
print(m3)                       # de linhas calculado

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

► Observe que podemos usar vetores predefinidos
  ► a ordem dos elementos
  ► todos os valores são do mesmo tipo
```

Acesso a 1 elemento da matriz - existe

```
print(m4)

##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2

m4[2,2] # linha 2, coluna 2

## [1] 8
```

Seleciona alguns elementos da matriz

```
print(m4)

##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2

m4[c(1,2), c(2,3)] # linhas 1 e 2, colunas 2 e 3

##      [,1] [,2]
## [1,]    4   10
## [2,]    8    1
```

Seleciona alguns elementos da matriz - 1 linha

```
print(m4)

##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2

m4[2,] # todos os elems da linha 2

## [1] 2 8 1
```

Seleciona alguns elementos da matriz - 1 coluna

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2
```

```
m4[,3]      # todos os elems da coluna 3
```

```
## [1] 10  1  2
```

Modifica 1 elemento da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8    1
## [3,]    3    9    2
```

```
m4[2,3] <- 20
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8   20
## [3,]    3    9    2
```

Modifica 1 linha da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]    2    8   20
## [3,]    3    9    2
```

```
m4[2,] <- c(20,6,7)
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]   20    6    7
## [3,]    3    9    2
```

Modifica 1 linha da matriz - faltando elementos

```
print(m4)
m4[2,] <- c(30,60)      # falta o 3o elemento - não modifica
print(m4)
```

```
Error in m4[2, ] <- c(20, 6) :
  number of items to replace is not a multiple of replacement
  [,1] [,2] [,3]
[1,]    1    4   10
[2,]   20    6    7
[3,]    3    9    2
```

Modifica 1 coluna da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   10
## [2,]   20    6    7
## [3,]    3    9    2
```

```
m4[,3] <- c(20,30,40)
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   20
## [2,]   20    6   30
## [3,]    3    9   40
```

Adiciona 1 linha à matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   20
## [2,]   20    6   30
## [3,]    3    9   40
```

```
m4 <- rbind(m4,c(20,30,40)) # rbind() row bind
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   20
## [2,]   20    6   30
## [3,]    3    9   40
## [4,]   20   30   40
```

Adiciona 1 coluna à matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4   20
## [2,]   20    6   30
## [3,]    3    9   40
## [4,]   20   30   40
```

```
m4 <- cbind(m4,c(20,30,40,50)) # cbind() column bind
print(m4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4   20   20
## [2,]   20    6   30   30
## [3,]    3    9   40   40
## [4,]   20   30   40   50
```

Remove 1 linha da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4   20   20
## [2,]   20    6   30   30
## [3,]    3    9   40   40
## [4,]   20   30   40   50
```

```
m4 <- m4[-c(2),] # remove a linha 2
print(m4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4   20   20
## [2,]    3    9   40   40
## [3,]   20   30   40   50
```

Remove 1 coluna da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1    4   20   20  
## [2,]    3    9   40   40  
## [3,]   20   30   40   50
```

```
m4 <- m4[, -c(3)] # remove a coluna 3  
print(m4)
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4   20  
## [2,]    3    9   40  
## [3,]   20   30   50
```

Remove todos os elementos da matriz

```
print(m4)
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4   20  
## [2,]    3    9   40  
## [3,]   20   30   50
```

```
m4 <- NULL  
print(m4)
```

```
## NULL
```

Importância

- ▶ as planilhas são armazenadas como data-frames
- ▶ precisamos aprender a manipular data-frames
- ▶ eles são formados por listas
- ▶ estrutura bidimensional formada por listas

Data-frames

As estruturas que estamos aprendendo...

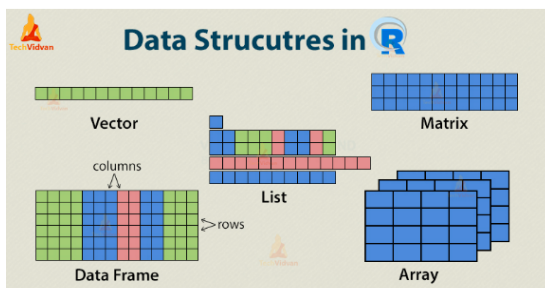


Figure 2:

Próxima Aula