

Aula 02 - Trabalhando com matrizes

Laís Salgueiro, Murilo Dias e Yan Soares

2022-09-27 16:34:02

Contents

1	Introdução ao R	2
1.1	Retomando aula anterior	2
1.2	Matrizes	2
1.2.1	Exemplo 1 - Criando uma matriz simples	3
1.2.2	Exemplo 2 - Contextualizando (matrizes mais complexas)	4

1 Introdução ao R

1.1 Retomando aula anterior

O R é um programa estatístico muito utilizado em pesquisas científicas. Uma das vantagens deste programa é a *flexibilidade* tanto na manipulação de planilhas de dados, quanto a análise estatísticas decorrentes e na produção de gráficos.

- Vocês aprenderam
 - A utilizar o R, como calculadora
 - Várias operações matemáticas
 - Objetos, vetores, matriz, data frame e listas
 - Estrutura dos objetos: numérico, inteiro, caracter...
 - Alguma dúvida e/ou consideração?
- Alguns dos pacotes implementados no R são muito uteis, sendo cobrados em vagas de empregos em algumas instituições, como **UNESCO**, **ONU**, **MAPBiomias** e **WWF**
 - **plyr**
 - **dplyr**
 - **tidyverse**
 - E por aí vai



Figure 1: Alguns dos pacotes que são pedidos para vagas de emprego

1.2 Matrizes

- O que é uma matriz?
 - Um conjunto de dados organizado em linhas e colunas (bidimensional)
 - Como mostrando anteriormente, nós podemos criar matrizes de diferentes formas
 - Uma maneira simples de criá-las é utilizando a função **matrix**, na qual permite criar uma matriz com valores atribuídos previamente
 - Dê uma olhada nessa função no console digite “**?matrix**” e veja os atributos dessa função
 - Sobre o *help* no R
 - * Apenas uma **interrogação** irá procurar nos pacotes carregados previamente
 - * Duas **interrogações** irá procurar sobre aquela função em toda a documentação do R

1.2.1 Exemplo 1 - Criando uma matriz simples

Aqui nós criamos uma matriz simples com um padrão de preenchimento pré-definido

- Como está definida essa matriz?
 - Vocês sabem o que os argumentos querem dizer?
 - Utilizem o *help* mais uma vez para investigar
 - Que tal testarmos de outro jeito?

```
matrix(1:15, byrow=T, nrow=3)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    6    7    8    9   10
## [3,]   11   12   13   14   15
```

Aqui nós repetimos a mesma matriz, no entanto, com outras especificações nos argumentos

```
matrix(1:15, byrow=F, ncol=3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

Caso a gente queira criar uma matriz de dados diferente, isto é, número maior de linhas e colunas. Como faríamos?

- Caso a gente continue trabalhando com sequência de números precisamos pegar uma sequência que seja múltipla do número de colunas que desejamos
- Por exemplo, **4 linhas** ou **colunas 32**
- Vamos ver isso na prática

```
matrix(1:32, byrow=T, ncol=4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
## [4,]   13   14   15   16
## [5,]   17   18   19   20
## [6,]   21   22   23   24
## [7,]   25   26   27   28
## [8,]   29   30   31   32
```

1.2.2 Exemplo 2 - Contextualizando (matrizes mais complexas)

Agora vamos pensar em um caso hipotético (não tanto assim), em que nós iremos construir uma matriz de dados a partir dos valores de arrecadação das bilheterias do **Jurassic Park (JP)**.

- Que tal olharmos a bilheteria da trilogia dos anos 90 e início dos anos 2000?
 - Jurassic Park – Parque dos Dinossauros - arrecadação nos EUA 357 milhões; arrecadação no exterior 621 milhões
 - O mundo perdido: Jurassic Park - arrecadação nos EUA 229 milhões; arrecadação no exterior 389 milhões
 - Jurassic Park III - arrecadação nos EUA 181 milhões; arrecadação no exterior 187 milhões

```
jp1993 <- c(357, 621)
jp1997 <- c(229, 389)
jp2001 <- c(181, 187)
```

Alguém tem ideia em como construir uma matriz com os vetores acima?

```
jp <- matrix(c(jp1993,jp1997,jp2001), byrow = T, nrow = 3, ncol = 2)
jp
```

```
##      [,1] [,2]
## [1,]  357  621
## [2,]  229  389
## [3,]  181  187
```

```
rownames(jp) <- c("jp1993","jp1997","jp2001")
colnames(jp) <- c("USA","EXTERIOR")
jp
```

```
##      USA EXTERIOR
## jp1993 357      621
## jp1997 229      389
## jp2001 181      187
```

Agora vamos navegar dentro da matriz para navegar entre as colunas é muito simples, pois basta utilizar o **\$**. Também podemos mostrar o nosso conjunto de dados com algumas outras funções sendo elas **head**, **tail** e **length**, por exemplo.

```
head(jp,2)
```

```
##      USA EXTERIOR
## jp1993 357      621
## jp1997 229      389
```

```
tail(jp,2)
```

```
##          USA EXTERIOR
## jp1997 229      389
## jp2001 181      187
```

```
length(jp)
```

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

Caso a gente queira saber qual foi a arrecadação total de cada um dos filmes, como faríamos isso? Alguém tem ideia? Para isso é simples, a função **rowSums** faz a soma dos valores das linhas e a partir daí podemos lidar de diferentes maneiras.

```
(lucro <- rowSums(jp))
```

```
## jp1993 jp1997 jp2001
##    978    618    368
```

```
#ou
lucro
```

```
## jp1993 jp1997 jp2001
##    978    618    368
```

```
#qual o lucro total dos três primeiros filmes?
sum(lucro) # a partir do vetor que nos criamos anteriormente
```

```
## [1] 1964
```

```
sum(jp[,c(1,2)]) # a partir da matriz de dados
```

```
## [1] 1964
```

```
jp[3,2] # mostrando valor específico (3 linha e 2 coluna)
```

```
## [1] 187
```

Poderíamos criar essa matriz de maneira mais eficiente? Alguém sabe como poderíamos fazer?

```
bilheteria <- c(357, 621, 229, 389, 181, 187)
filmes <- c("jp1993", "jp1997", "jp2001")
colnomes <- c("USA", "EXTERIOR")
jp <- matrix(bilheteria,
             nrow = 3,
             byrow = TRUE,
             dimnames = list(filmes, colnomes))
jp
```

```
##          USA EXTERIOR
## jp1993 357      621
## jp1997 229      389
## jp2001 181      187
```

```
View(jp)
```

O que acharam? Será que conseguimos fazer a mesma operação anterior (somar valores das linhas) para obter o lucro total a partir da nova matriz? Que tal juntarmos o vetor **lucro** com o nosso conjunto de dados **jp**. Para isso nós utilizaremos o a função **cbind** (combina colunas) e a **rbind** combina linhas.

```
lucro <- rowSums(jp)
(jpt<- cbind(jp,lucro))
```

```
##          USA EXTERIOR lucro
## jp1993 357      621   978
## jp1997 229      389   618
## jp2001 181      187   368
```

```
colnames(jpt)
```

```
## [1] "USA"      "EXTERIOR" "lucro"
```

```
rownames(jpt)
```

```
## [1] "jp1993" "jp1997" "jp2001"
```