

- [Tutorial](#)
- [Exercícios](#)
- [Apostila](#)

## 3. Leitura e Manipulação de Dados

### Leitura de Dados

#### Entrada de Dados Diretamente no R

##### Função "c()" (concatenate ou combine)

As funções de criação de vetores já foram detalhadas na [seção anterior](#). Basta lembrar aqui que todas elas são usadas para entrar diretamente dados em vetores no R:

```
> meu.vetor <- c(10.5, 11.3, 12.4, 5.7)
> meu.vetor
[1] 10.5 11.3 12.4 5.7
>
> vetor.vazio <- c()
> vetor.vazio
NULL
```

##### Função "matrix()"

A função `matrix` cria uma matriz com os valores do argumento `data`. O números de linhas e colunas são definidos pelos argumentos `nrow` e `ncol`:

```
> minha.matriz <- matrix(data=1:12, nrow=3, ncol=4)
> minha.matriz
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Como o *default* do argumento `data` é `NA`, se ele é omitido o resultado é uma matriz vazia:

```
> matriz.vazia <- matrix(nrow=3, ncol=4)
> matriz.vazia
      [,1] [,2] [,3] [,4]
[1,]  NA  NA  NA  NA
[2,]  NA  NA  NA  NA
[3,]  NA  NA  NA  NA
```

Também por *default*, os valores são preenchidos por coluna. Para preencher por linha basta o alterar o argumento `byrow` para `TRUE`:

```
> minha.matriz <- matrix(data=1:12,nrow=3,ncol=4,byrow=T)
> minha.matriz
  [,1] [,2] [,3] [,4]
[1,]  1   2   3   4
[2,]  5   6   7   8
[3,]  9  10  11  12
```

Se o argumento `data` tem menos elementos do que a matriz, eles são repetidos até preenchê-la:

```
> elementos <- matrix(c("ar","água","terra","fogo","Leeloo"),ncol=4,nrow=4)
Warning message:
comprimento dos dados [5] não é um submúltiplo ou múltiplo do número de
linhas [4] na matrix
> elementos
  [,1]  [,2]  [,3]  [,4]
[1,] "ar"   "Leeloo" "fogo"  "terra"
[2,] "água" "ar"     "Leeloo" "fogo"
[3,] "terra" "água"  "ar"     "Leeloo"
[4,] "fogo"  "terra" "água"  "ar"
```

## Função "data.frame()"

Com a função `data.frame` reunimos vetores de mesmo comprimento em um só objeto:

```
> nome <- c("Didi","Dedé","Mussum","Zacarias")
> ano.nasc <- c(1936,1936,1941,1934)
> vive <- c("V","V","F","F")
> trapalhoes <- data.frame(nomes,ano.nasc,vive)
> trapalhoes
  nomes ano.nasc vive
1  Didi   1936   V
2  Dedé   1936   V
3 Mussum  1941   F
4 Zacarias 1934   F
>
# O mesmo, em um só comando:
> trapalhoes <- data.frame(nomes=c("Didi","Dedé","Mussum","Zacarias"),
ano.nasc=c(1936,1936,1941,1934), vive=c("V","V","F","F"))
```

## Função "edit()"

Esta função abre uma interface simples de edição de dados em formato planilha, e é útil para pequenas modificações. Mas para salvar as modificações atribua o resultado da função `edit` a um objeto:

```
trapalhoes.2<-edit(trapalhoes)
```



|   | nomes    | ano.nasc | vive |
|---|----------|----------|------|
| 1 | Didi     | 1936     | V    |
| 2 | Dedé     | 1936     | V    |
| 3 | Mussum   | 1941     | F    |
| 4 | Zacarias | 1934     | F    |

## Dados que já Estão em Arquivos

### Leitura e Exportação de Arquivos-Texto: "read.table()" e "write.table()"

Para conjuntos de dados grandes, é mais prático gerar um arquivo de texto (ASCII) a partir de uma planilha ou banco de dados, e usar a função `read.table` para ler os dados para um objeto no R.

Para criar um objeto com os dados do arquivo [gbmam93.csv \(apagar extensão .pdf\)](#), por exemplo, digitamos:

```
> gbmam93 <- read.table(file="gbmam93.txt", header=T, row.names=1, sep=",")
> gbmam93
  a b c d e f g h i j k l m n o p q r s
1  1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1
2  1 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 1
3  1 0 0 1 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1
4  1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
5  1 0 0 1 1 0 1 0 1 0 0 1 1 0 0 0 1 0 0
6  1 1 1 0 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1
7  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
8  1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 1
9  1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0
10 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0
11 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
12 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1
13 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1
14 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
```

O argumento `header=T` indica que a primeira linha são os nomes das variáveis, assim como `row.names=1` indica que a primeira coluna deve ser usada para os nomes das linhas. O argumento `sep` indica qual é o sinal de separação de cada registro, no caso vírgulas.

Esses e os outros argumentos da função a tornam extremamente flexível para ler dados em arquivos texto. Consulte a ajuda para mais informações, e também para conhecer as variantes `read.csv` e `read.delim`.

Para exportar um objeto para um arquivo texto, use a função `write.table`, que tem a mesma

lógica.

## Conjuntos de Dados Distribuídos com os Pacotes do R

Muitos pacotes do R incluem conjuntos de dados para exemplos, treinamento e verificação de análises. Se o pacote já está carregado (funções `library` ou `require`) todos os seus objetos estão disponíveis, inclusive os objetos de dados. Incluindo as séries temporais de número de peles de linces caçados no Canadá, analisadas pelo ecólogo Charles Elton obtém-se:

```
> lynx
Time Series:
Start = 1821
End = 1934
Frequency = 1
 [1] 269 321 585 871 1475 2821 3928 5943 4950 2577 523 98 184 279
409
 [16] 2285 2685 3409 1824 409 151 45 68 213 546 1033 2129 2536 957
361
 [31] 377 225 360 731 1638 2725 2871 2119 684 299 236 245 552 1623
3311
 [46] 6721 4254 687 255 473 358 784 1594 1676 2251 1426 756 299 201
229
 [61] 469 736 2042 2811 4431 2511 389 73 39 49 59 188 377 1292
4031
 [76] 3495 587 105 153 387 758 1307 3465 6991 6313 3794 1836 345 382
808
 [91] 1388 2713 3800 3091 2985 3790 674 81 80 108 229 399 1132 2432
3574
[106] 2935 1537 529 485 662 1000 1590 2657 3396
```

Como qualquer objeto de um pacote, `lynx` tem um arquivo de ajuda, que é exibido com o comando `help(lynx)` ou `?lynx`:

```
lynx                package:datasets                R Documentation

Annual Canadian Lynx trappings 1821-1934

Description:

  Annual numbers of lynx trappings for 1821-1934 in Canada. Taken
  from Brockwell & Davis (1991), this appears to be the series
  considered by Campbell & Walker (1977).

Usage:

  lynx

Source:
```

Brockwell, P. J. and Davis, R. A. (1991) \_Time Series and Forecasting Methods.\_ Second edition. Springer. Series G (page 557).

#### References:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) \_The New S Language\_. Wadsworth & Brooks/Cole.

Campbell, M. J. and A. M. Walker (1977). A Survey of statistical work on the Mackenzie River series of annual Canadian lynx trappings for the years 1821-1934 and a