

Introdução ao R
2. Noções básicas (a)
1/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

Introdução ao uso do



em Ciências da Saúde

2. Noções básicas para o trabalho em R
- a. Aprendendo uma linguagem

Fúlvio Borges Nedel

Departamento de Saúde Pública – SPB
Centro de Ciências da Saúde – CCS
Universidade Federal de Santa Catarina – UFSC

Grups de Recerca d'Amèrica i Àfrica Llatines – GRAAL
<http://graal.uab.cat>

4 de dezembro de 2017

Roteiro



Introdução ao R

2. Noções básicas (a)

2/8

Fúlvio Nedel

SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

1 Workspace

2 Vetor

3 Função

4 Criar um banco de dados

Workspace: o espaço de trabalho

- ★ O R trabalha com **objetos** em **espaços de trabalho**
- ★ Os objetos podem ser de diferentes **classes**
- ★ O *workspace* contém ainda os **pacotes**, chamados automaticamente na inicialização do programa ou por demanda do usuário
- ★ Um pacote contém as **funções**, comandos para a execução das ações desejadas
- ★ O R trabalha com os objetos e funções **em memória – não sobre o arquivo de dados mas uma cópia**, que permanece ativa durante a sessão de trabalho

Introdução ao R

2. Noções básicas (a)

4/8

Fábio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

Workspace

- ★ Permite, entre outras coisas, trabalhar com diferentes bancos de dados ao mesmo tempo
- ★ As ações – modificação e criação de objetos – realizadas numa *sessão de trabalho* são descartadas ao encerrar a sessão, a menos que se grave um arquivo com os elementos desejados do espaço de trabalho.
- ★ É recomendável que o diretório de trabalho seja o mesmo com os arquivos de dados e outros relacionados à análise

Introdução ao R

2. Noções básicas (a)

5/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

Noções básicas

■ O sinal do console:

- “>” indica a primeira linha do comando
- “+” indica sua continuação, quando houver

■ Pacote / biblioteca (library)

base, foreign, Hmisc, epicalc, Epi, ...

■ Objeto

- É a forma com que são armazenados em memória os elementos (criados ou carregados) na sessão de trabalho
- Banco de dados, vetor, matriz, tabela, função, resultado da função, fórmula, ...
- São criados pelo sinal de destinação (“<- ”)

```
> x <- 3
> x
[1] 3
```
- Vários objetos podem ser salvos em um único arquivo



Introdução ao R

2. Noções básicas (a)

6/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

Noções básicas

■ Vetor

- Sequência de valores de um mesmo tipo de dados, em uma ordem específica
- É o objeto mais simples do R
- Podem ser **numéricos** (e sofrer operações matemáticas) ou **alfanuméricos** (e ser combinados)

```
> x <- 3:7      > z <- x + y
> x             > z
[1] 3 4 5 6 7    [1] 10 10 10 10 10
> y <- 7:3      > w <- z + sqrt(4)
> y             > w
[1] 7 6 5 4 3    [1] 12 12 12 12 12
```

```
> x <- 'banana'
> y <- 'laranja'
> z <- x + y
Erro em x + y :
  argumento não-numérico
  para operador binário
> z <- c(x,y)
>
[1] "banana"   "laranja"
```

Introdução ao R

2. Noções básicas (a)

7/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

Noções básicas

■ Vetor

Seleção de valores específicos de um objeto

```
> animais <- c('cachorro', 'galinha', 'cachoro', 'gato',  
  'esquilo', 'tagtaguga', 'gralha', 'sangue-de-boi',  
  'cachorro')  
  
> animais  
[1] "cachorro"      "galinha"       "cachoro"  
[4] "gato"          "esquilo"        "tagtaguga"  
[7] "gralha"         "sangue-de-boi"   "cachorro"  
  
> animais[c(3,6)] <- c('cachorro', 'tartaruga') # arrumo o 3º  
  e o 6º registro  
  
> animais[3:6] # observo os valores do 3º ao 6º registro  
[1] "cachorro" "gato"    "esquilo"  "tartaruga"
```



Noções básicas

■ Função `função(x, ...)`

- `x = objeto`
- `... demais argumentos`

- É um comando direcionado a um *objeto* qualquer do espaço de trabalho (*workspace*)
- Tem *argumentos* com as opções de execução
 - Podem ser obrigatórios ou não
 - O nome do objeto geralmente é obrigatório
- Eventualmente a função pode não exigir a definição de nenhum argumento, como em
 - `ls()` # lista os objetos do espaço de trabalho
 - `help()` # abre a ajuda sobre a função
`'help()'`
 - `par()` # apresenta os parâmetros ativos de construção de gráficos

Noções básicas

■ Função `função(x, ...)`

- `x = objeto`
 - `... demais argumentos`
- É um comando direcionado a um *objeto* qualquer do espaço de trabalho (*workspace*)
 - Tem *argumentos* com as opções de execução
 - Podem ser obrigatórios ou não
 - O nome do objeto geralmente é obrigatório
 - Eventualmente a função pode não exigir a definição de nenhum argumento, como em

- `ls()` # lista os objetos do espaço de trabalho
- `help()` # abre a ajuda sobre a função
`'help()'`
- `par()` # apresenta os parâmetros ativos de construção de gráficos

“Comentário” (#):
o que vem após a cerquilha não é interpretado como comando.



Noções básicas

- **Função** `read.table()`, `help()`, `table()`, ...

```
> library(Hmisc) # carrega o pacote 'Hmisc'  
[...] Attaching package: 'Hmisc' [...]
```

- “Case-sensitive”, diferencia maiúsculas e minúsculas

```
> library(hmisc)  
Erro em library(hmisc) : there is no package called 'hmisc'  
  
> which(animais=='esquilo')  
[1] 5  
  
> animais[5]  
[1] "esquilo"  
  
> which(animais=='Esquilo')  
integer(0)
```

Noções básicas

■ Funções úteis:

- ♦ `c(a, b, ...)`:
- ♦ combina os elementos em uma linha
 - ♦ usada anteriormente para combinar em um novo vetor os objetos `x` ("banana") e `y` ("laranja"):
`z <- c(x,y)`
 - ♦ e para criar um vetor ("animais") com o nome de alguns animais
 - ♦ `animais <- c('cachorro', 'galinha', 'cachorro', 'gato', 'esquilo', 'tagtaguga', 'gralha', 'sangue-de-boi', 'cachorro')`

Noções básicas

■ Funções úteis:

– **cbind(a, b, ...):**

- Combina os vetores em colunas, formando uma matriz.

– **rbind(a, b, ...):**

- Combina os vetores em linhas, formando uma matriz

```
> x <- 3:7  
> y <- 7:3  
> z <- x + y
```

```
> rbind(x, y, z)  
 [,1] [,2] [,3] [,4] [,5]  
 x     3     4     5     6     7  
 y     7     6     5     4     3  
 z    10    10    10    10    10
```

```
> cbind(x, y, z)  
      x y z  
[1,] 3 7 10  
[2,] 4 6 10  
[3,] 5 5 10  
[4,] 6 4 10  
[5,] 7 3 10
```



Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))
```

	x	y	z
1	3	7	10
2	4	6	10
3	5	5	10
4	6	4	10
5	7	3	10

Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))
```

	x	y	z
1	3	7	10
2	4	6	10
3	5	5	10
4	6	4	10
5	7	3	10

Veja uma forma abreviada da
função `print()`: basta colocar
o comando entre parênteses.

Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))
```

	x	y	z
1	3	7	10
2	4	6	10
3	5	5	10
4	6	4	10
5	7	3	10

Veja uma forma abreviada da
função `print()`: basta colocar
o comando entre parênteses.

Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de
dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))  
  
  x y  z      str(banco)  
1 3 7 10  'data.frame': 5 obs. of  3 variables:  
2 4 6 10   $ x: int  3 4 5 6 7  
3 5 5 10   $ y: int  7 6 5 4 3  
4 6 4 10   $ z: int  10 10 10 10 10  
5 7 3 10  
  
Note o modo de chamar a variável:  
banco$y  <nome do data frame>$<nome da variável>  
[1] 7 6 5 4 3
```

Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))
```

	x	y	z
1	3	7	10
2	4	6	10
3	5	5	10
4	6	4	10
5	7	3	10

```
str(banco)
'data.frame': 5 obs. of 3 variables:
 $ x: int 3 4 5 6 7
 $ y: int 7 6 5 4 3
 $ z: int 10 10 10 10 10
```

Note o modo de chamar a variável:

```
banco$y <nome do data frame>$<nome da variável>
[1] 7 6 5 4 3
```

Selecionar o 4º registro da variável 'y':

```
banco[4,2]
[1] 4
```

```
banco[4,"y"]
[1] 4
```

```
banco$y[4]
[1] 4
```

Criar um banco de dados

Introdução ao R
2. Noções básicas (a)
8/8

Fúlvio Nedel
SPB/UFSC

Workspace

Vetor

Função

Criar um banco de dados

- `data.frame()`: cria uma base de dados
- `str()`: apresenta a estrutura do objeto
 - veja também a função `class()` (digite `?class`)

```
(banco <- data.frame(x,y,z))
```

	x	y	z
1	3	7	10
2	4	6	10
3	5	5	10
4	6	4	10
5	7	3	10

```
str(banco)
'data.frame': 5 obs. of 3 variables:
 $ x: int 3 4 5 6 7
 $ y: int 7 6 5 4 3
 $ z: int 10 10 10 10 10
```

Note o modo de chamar a variável:

```
banco$y <nome do data frame>$<nome da variável>
[1] 7 6 5 4 3
```

Selecionar o 4º registro da variável 'y':

```
banco[4,2]
[1] 4
```

```
banco[4,"y"]
[1] 4
```

```
banco$y[4]
[1] 4
```