

# Comandos básicos do R

*Bárbara Costa*

*08 de Maio de 2017*

## Criar um diretório do curso

A primeira coisa a se fazer é criar uma pasta para você armazenar todos os arquivos deste curso. Depois de criar a pasta para o curso, faça o seguinte: Verifique o diretório de trabalho atual. Obs: Para rodar o comando `getwd()` clique `crtl + enter` no windows ou `command + enter` no mac. Siga da mesma forma para qualquer outro comando que quiser rodar abaixo.

```
getwd()
```

Se necessário, mude o diretório de trabalho (para trabalhar dentro da sua pasta do curso). Para isso, use o comando `setwd()` e o caminho do seu computador até o diretório criado. Exemplo: `setwd("~/Desktop/disciplina")`.

Dica: use “” e a tecla `tab` para ajudar a visualizar os caminhos do computador.

Se usou o comando `setwd()` verifique em seguida o diretório atual:

```
getwd()
```

Você também tem a opção de fazer isso utilizando a janela no canto inferior direito do RStudio da seguinte forma: selecione a pasta que deseja e na opção **More** selecione **Set As Working Directory**.

## Carregue alguns pacotes

observação: se ainda não instalou algum desses pacotes, faça isso primeiro. Exemplo: `install.packages("ggplot2")`

```
#carregando
```

```
require (ggplot2)
```

```
require (evolqg)
```

Pergunta: quando voce roda os dois comandos abaixo, o que vc acontece?

```
# familiarize-se com o pacote
```

```
help(package="ggplot2")
```

```
help(package="evolqg")
```

## Algumas operacoes aritimeticas basicas no R!

```
4 + 9
```

```
## [1] 13
```

```
4 - 5
```

```
## [1] -1
```

```
4 * 5
```

```
## [1] 20
```

```
4 / 5
```

```
## [1] 0.8
```

```
4^5
```

```
## [1] 1024
```

```
2^2
```

```
## [1] 4
```

```
(4 + 5 ) * 7 - (36/18)^3
```

```
## [1] 55
```

```
(4 + 5 * 7 - 36/18)^3
```

```
## [1] 50653
```

```
# raiz quadrada
```

```
sqrt(9)
```

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

```
# logaritmo natural ou neperiano
```

```
log(10)
```

```
## [1] 2.302585
```

```
# logaritmo base 10
```

```
log(10, base = 10)
```

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

```
# também logaritmo de base 10
```

```
log10(10)
```

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

```
# exponencial
```

```
exp(1)
```

```
## [1] 2.718282
```

Exercício: crie dois objetos, a e b, para as duas primeiras linhas acima (4+9 e 4-5). Faça a soma da diferença desses dois objetos, elevado ao quadrado. Resposta: 196

## Criação de Vetores

Para criar um vetor, podemos usar a função c (c = colar, concatenar). Essa função simplesmente junta todos os argumentos dados a ela, formando um vetor:

```
vector.a <- c(1, 10, 3.4, 16, 23, 91, 46 )
```

```
vector.b <- c(1, 10, 3.4, pi, pi/4, exp(-1), log( 2.23 ), sin(pi/7) )
```

```
vector.c <- c(6.1, 7.5, 7.0, 8.8, 9.1, 6.3)
```

Exercício: Você criou alguns objetos. Como vcs podem visualizá-lo no RStudio? Qual a classe desses objetos? Agora, crie um vetor com 6 elementos da classe character. Dica use aspas. Como vc apagaria esse vetor que acabou de criar? Dica: veja o help da função rm.

## Vetores: Operações Estatísticas

Faça a média do vetor.a, variância do vetor.b, busque pelo maior valor do vetor.c e, por fim, faça o desvio padrão do vetor.a. Dica! Use a função help para pesquisar sobre essas operações: `?mean` `help(mean)` `?var` `help(sd)`

Respostas: media = 27.2 variância = 10.6227 máximo = 9.1 desvio padrão = 31.92616

Execute o comando abaixo e veja o help dela. O que ela fez? `summary(vector.a)`

## Vamos trabalhar com os Tipos de Objetos no R

### Vetor:

```
meu.vetor <- c(10.5,11.3,12.4,5.7)
meu.vetor
```

```
## [1] 10.5 11.3 12.4 5.7
```

Coloque nomes para cada elemento do objeto meu.vetor. Dica: use a função **names**.

```
names(meu.vetor) <- c("a", "b", "c", "d")
```

Usando a indexação busque apenas o segundo elemento do objeto “meu.vetor”. Dica: use colchetes.

### Matriz:

```
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
```

```
args(matrix)
```

```
## function (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
## NULL
```

Olhe o help da função `matrix` e olhe também os argumentos dela. O que significa o argumento “byrow”? Crie um objeto `minha.outra.matriz` agora preenchendo as matrizes pelas linhas.

### Array:

```
my.array <- array(1:36, dim=c(3,4,3))
my.array
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   13   16   19   22
## [2,]   14   17   20   23
## [3,]   15   18   21   24
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]   25   28   31   34
## [2,]   26   29   32   35
## [3,]   27   30   33   36
```

### Data frame:

```
nomes <- 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   Ded     1936     V
## 3  Mussum    1941     F
## 4 Zacarias   1934     F
```

Execute as fun  es **head()** e **tail()** para o objeto trabalhos. Usando **head()**, solicite as 10 primeiras linhas de **trapalhoes**. Dica: olhe o help da fun   o. Com qual fun   o voc   olharia a estrutura do objeto? Dica: olhe o help da fun   o **str**.

## Lista:

```
minha.lista <- list(um.vetor=1:5, uma.matriz=matrix(1:6,2,3), um.dframe=data.frame(seculo=c("XIX", "XX",
```

Abra o objeto **minha.lista**. Com qual fun   o voc   observaria a estrutura desta lista? Use tamb  m a fun   o **names** e veja o resultado.

## Algumas fun   es b  sicas no R

Execute a fun   o que indica a classe dos objetos **minha.lista**, **trapalhoes**, **my.array** e **minha.matriz**. Depois, execute os comandos abaixo e veja o que cada um faz. N  o deixe de ver tamb  m o help de cada uma das fun   es.

```
# atributo de um objeto
attributes(minha.lista)
```

```
## $names
## [1] "um.vetor"  "uma.matriz" "um.dframe"
```

```
attributes(trapalhoes)
```

```
## $names
## [1] "nomes"    "ano.nasc" "vive"
```

```
##  
## $class  
## [1] "data.frame"  
##  
## $row.names  
## [1] 1 2 3 4
```

```
attributes(my.array)
```

```
## $dim  
## [1] 3 4 3
```

```
attributes(minha.matriz)
```

```
## $dim  
## [1] 3 4
```

```
# dimensões do objeto
```

```
dim(my.array)
```

```
## [1] 3 4 3
```

```
# numero de linhas
```

```
nrow(minha.matriz)
```

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

```
# número de colunas
```

```
ncol(minha.matriz)
```

```
## [1] 4
```

## Vamos trabalhar com um objeto do pacote datasets

Observação: se ainda não instalou esse pacote, faça isso primeiro. Você já sabe como fazer para instalar um pacote :). Lembrete: use aspas. Pronto! Com a função data carregue os dados do objeto iris:

```
data(iris)
```

Agora, abra o objeto iris e olhe o conteúdo dele. Explore o objeto iris com o comando **str** e indique qual a classe do objeto e nomes das partes.

## Indexação:

Vamos selecionar colunas específicas de iris.

```
iris$Sepal.Length
```

```
##      [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##     [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##     [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##     [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##     [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##     [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
##    [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
##    [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
##    [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris$Sepal.Width
```

```
##      [1] 3.5 3.0 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 3.7 3.4 3.0 3.0 4.0 4.4 3.9
##     [18] 3.5 3.8 3.8 3.4 3.7 3.6 3.3 3.4 3.0 3.4 3.5 3.4 3.2 3.1 3.4 4.1 4.2
##     [35] 3.1 3.2 3.5 3.6 3.0 3.4 3.5 2.3 3.2 3.5 3.8 3.0 3.8 3.2 3.7 3.3 3.2
##     [52] 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7 2.0 3.0 2.2 2.9 2.9 3.1 3.0 2.7
##     [69] 2.2 2.5 3.2 2.8 2.5 2.8 2.9 3.0 2.8 3.0 2.9 2.6 2.4 2.4 2.7 2.7 3.0
##     [86] 3.4 3.1 2.3 3.0 2.5 2.6 3.0 2.6 2.3 2.7 3.0 2.9 2.9 2.5 2.8 3.3 2.7
##    [103] 3.0 2.9 3.0 3.0 2.5 2.9 2.5 3.6 3.2 2.7 3.0 2.5 2.8 3.2 3.0 3.8 2.6
##    [120] 2.2 3.2 2.8 2.8 2.7 3.3 3.2 2.8 3.0 2.8 3.0 2.8 3.8 2.8 2.8 2.6 3.0
##    [137] 3.4 3.1 3.0 3.1 3.1 3.1 2.7 3.2 3.3 3.0 2.5 3.0 3.4 3.0
```

```
iris$Species
```

```
##      [1] setosa      setosa      setosa      setosa      setosa      setosa
```



```
## [7] setosa      setosa      setosa      setosa      setosa      setosa
## [13] setosa      setosa      setosa      setosa      setosa      setosa
## [19] setosa      setosa      setosa      setosa      setosa      setosa
## [25] setosa      setosa      setosa      setosa      setosa      setosa
## [31] setosa      setosa      setosa      setosa      setosa      setosa
## [37] setosa      setosa      setosa      setosa      setosa      setosa
## [43] setosa      setosa      setosa      setosa      setosa      setosa
## [49] setosa      setosa      versicolor versicolor versicolor versicolor
## [55] versicolor versicolor versicolor versicolor versicolor versicolor
## [61] versicolor versicolor versicolor versicolor versicolor versicolor
## [67] versicolor versicolor versicolor versicolor versicolor versicolor
## [73] versicolor versicolor versicolor versicolor versicolor versicolor
## [79] versicolor versicolor versicolor versicolor versicolor versicolor
## [85] versicolor versicolor versicolor versicolor versicolor versicolor
## [91] versicolor versicolor versicolor versicolor versicolor versicolor
## [97] versicolor versicolor versicolor versicolor virginica  virginica
## [103] virginica  virginica  virginica  virginica  virginica  virginica
## [109] virginica  virginica  virginica  virginica  virginica  virginica
## [115] virginica  virginica  virginica  virginica  virginica  virginica
## [121] virginica  virginica  virginica  virginica  virginica  virginica
## [127] virginica  virginica  virginica  virginica  virginica  virginica
## [133] virginica  virginica  virginica  virginica  virginica  virginica
## [139] virginica  virginica  virginica  virginica  virginica  virginica
## [145] virginica  virginica  virginica  virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

Você fazer o mesmo usando colchetes. Faça o teste. Lembre-se de usar as aspas dentro dos colchetes se for designar os nomes das colunas ou coloque simplesmente o número dela.

Agora, busque todas as linhas da coluna 3 do objeto `minha.matriz`. Você já sabe fazer :)

Use `colnames(minha.matriz) <- c("a", "b", "c", "d")` para colocar nomes nas colunas da matriz. Abra o objeto `minha.matriz` de novo para conferir. Em seguida, busque as linhas 1 e 3, da coluna "d".

Busque no objeto `trapalhoes` a sua classe. Em seguida, busque todas as linhas da coluna "nomes".

Agora, busque a linha 4 de todas as colunas.

Fim!