

Curso de R para Geociências

Marcus Suassuna Santos

26/10/2020

Estrutura do Curso

- O que é o R?
- Porque usar o R em geociências?
- Instalação e interface do R
- Comandos básicos
- Leitura de dados em R
- Manuseio de dados (usando R básico e `dplyr`)
- Estruturas de controle
- Gráficos
- Funções
- Datas e horários
- Tópicos especiais aplicáveis à previsão

Instrutor

2008: Engenheiro Civil, UnB

2011: Msc. Saneamento, Meio Ambiente e Recursos Hídricos - UFMG

2013: CPRM

2015: **Abandonei Excel**

2016: Data Science Specialization - Johns Hopkins University

2019: Mastering Software Development in R Specialization - Johns Hopkins University

2019: Python for Everybody Specialization - Michigan University

2019: Dsc. Tecnologia Ambiental e Recursos Hídricos - UnB

...

Principais recursos

- [R Programming for Data Science](#), Roger D. Peng, 2019. Johns Hopkins Bloomberg School of Public Health.
- Stat 133 Class Notes - Spring, 2011, Phil Spector.
- Livro de Hidrologia Estatística, Naghettini e Pinto, 2007
- Material da Data Science Specialization.
- [Stack Overflow](#)
- [R for Data Science](#)
- www.r-bloggers.com R-Bloggers.

O que é a linguagem R

R é um dialeto da linguagem S (?!)

S é uma linguagem desenvolvida por John Chambers e outros em 1962 nos laboratórios da companhia telBell, naquele momento, parte da AT&T. Foi desenvolvida como ambiente de análises estatísticas internas, implementada como biblioteca da linguagem Fortran.

Em 1988 foi reescrita em C. A versão 4 da linguagem S, que é a versão utilizada hoje.

Filosofia linguagem S:

Enraizada na análise de dados, e não na programação tradicional. Como tornar a análise de dados mais fácil, primeiro para eles próprios e, finalmente, para os outros?

O que é a linguagem R

Linguagem S só é acessível como um pacote comercial S-PLUS.

1991: Ross Ihaka e Robert Gentleman criam a linguagem no Departamento de Estatística da University of Auckland:

Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. Journal of Computational and Graphical Statistics, 5(3):299–314, 1996

1995: GNU - General Public License - R software livre. Código fonte acessível.

1996: Lista de e-mails, tutoriais e ajuda

1997: Formação do Core Group

2000: Versão 1.0.0 lançada

O que é a linguagem R

- [R Programming for Data Science](#), Roger D. Peng, 2019. Johns Hopkins Bloomberg School of Public Health.

Porque usar o R em geociências?

```
url <- "https://cran.r-project.org/web/packages/available_packages_by_date.html"
```

```
page <- read_html(url)
```

```
pkgs <- page %>%
```

```
  html_node("table") %>%
```

```
  html_table() %>%
```

```
  mutate(count = rev(1:nrow(.))) %>%
```

```
  mutate(Date = as.Date(Date)) %>%
```

```
  mutate(Month = format(Date, format="%Y-%m")) %>%
```

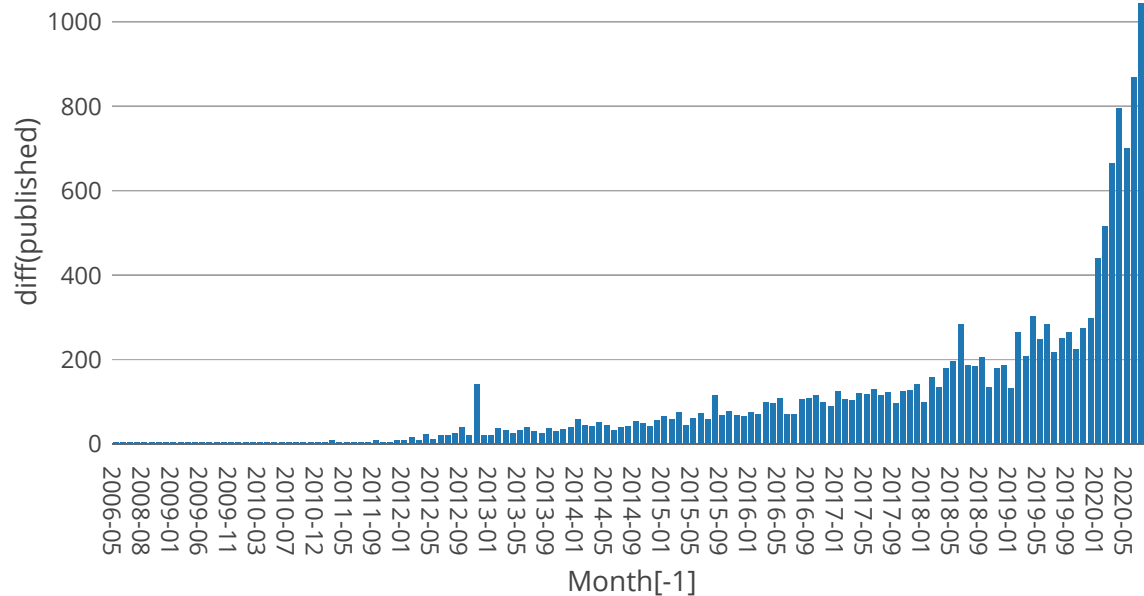
```
  group_by(Month) %>%
```

```
  summarise(published = min(count)) %>%
```

```
  mutate(Date = as.Date(as.yearmon(Month))) -> pkgs
```


Porque usar o R em geociências?

CRAN packages published ever since.



Atualmente, existem 15788 pacotes do R disponíveis no repositório oficial CRAN. 10 pacotes que identifiquei com aplicações em geologia e 41 em hidrologia.

Porque usar o R em geociências?

Hidrologia

- [R in Hydrology](#) , Slater, et al., Hydrol. Earth Syst. Sci., 23, 2939–2963, 2019.

Geociências

- [Potential use of R-statistical programming in the field of geoscience](#), , R. M. Bishwal, 2017 2nd International Conference for Convergence in Technology (I2CT), Mumbai, 2017, pp. 979-982, doi: 10.1109/I2CT.2017.8226275.

GIS

- [Whitebox](#) , Whitebox R package for geospatial analysis, which is an R frontend of a stand-alone executable command-line program called WhiteboxTools.

Porque usar o R em geociências?

Pacotes USGS

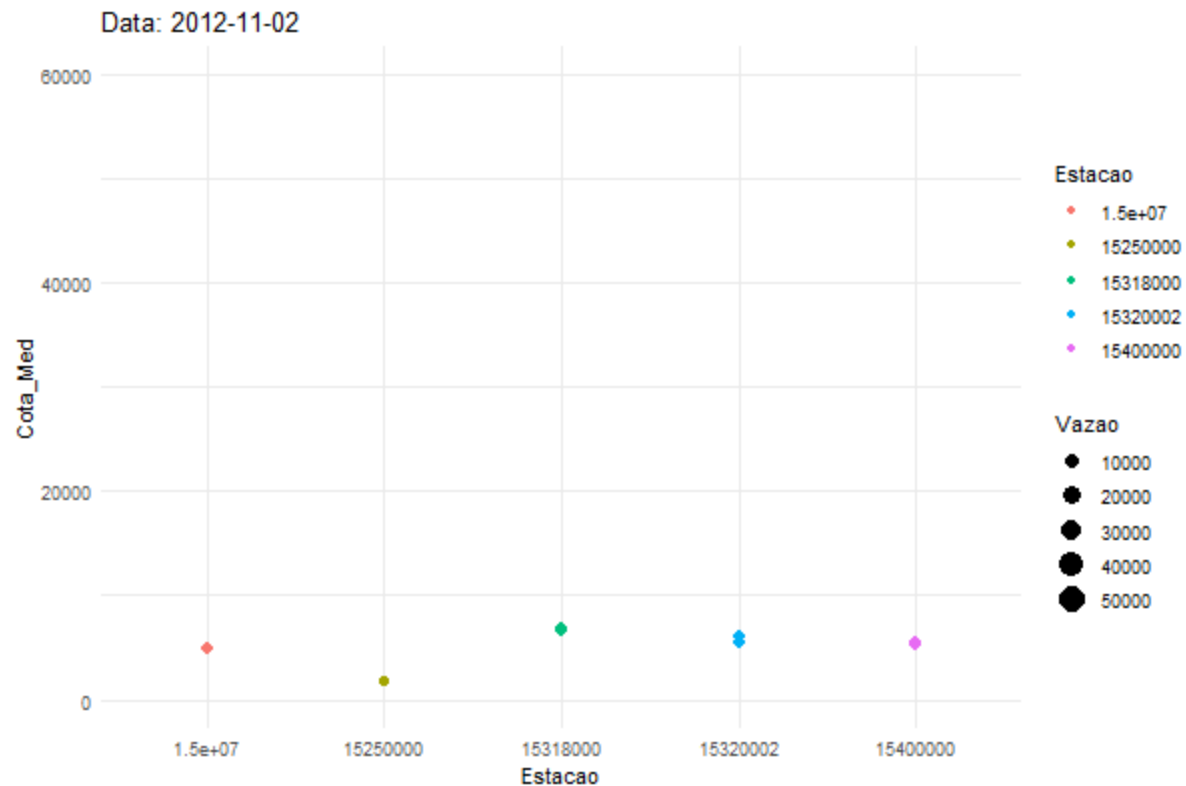
- [USGS R Packages](#)

Manual de Métodos Estatísticos para Recursos Hídricos com exemplos em R

- [Statistical Methods in Water Resources - USGS](#)

Porque usar o R em geociências?

- Interface gráfica



GIF Madeira

Porque usar o R em geociências?

Aplicações web em Shiny

- Modelo Hidrológico [Smap](#)
- Dados do [hidroweb](#)
- [Análise de frequência](#)

Porque usar o R em geociências?

Em resumo

- Simplicidade da linguagem
- Excelente desempenho na manipulação de grandes bancos de dados
- Acesso a métodos estatísticos consolidados
- Acesso a diversas ferramentas de visualização de dados
- Acesso a métodos analíticos de ponta e ferramentas compartilhadas internacionalmente - não só pelo uso de pacotes, mas sobretudo, usando fóruns e tutoriais.

Instalação

Download de 2 programas: R e RStudio

- [R](#)

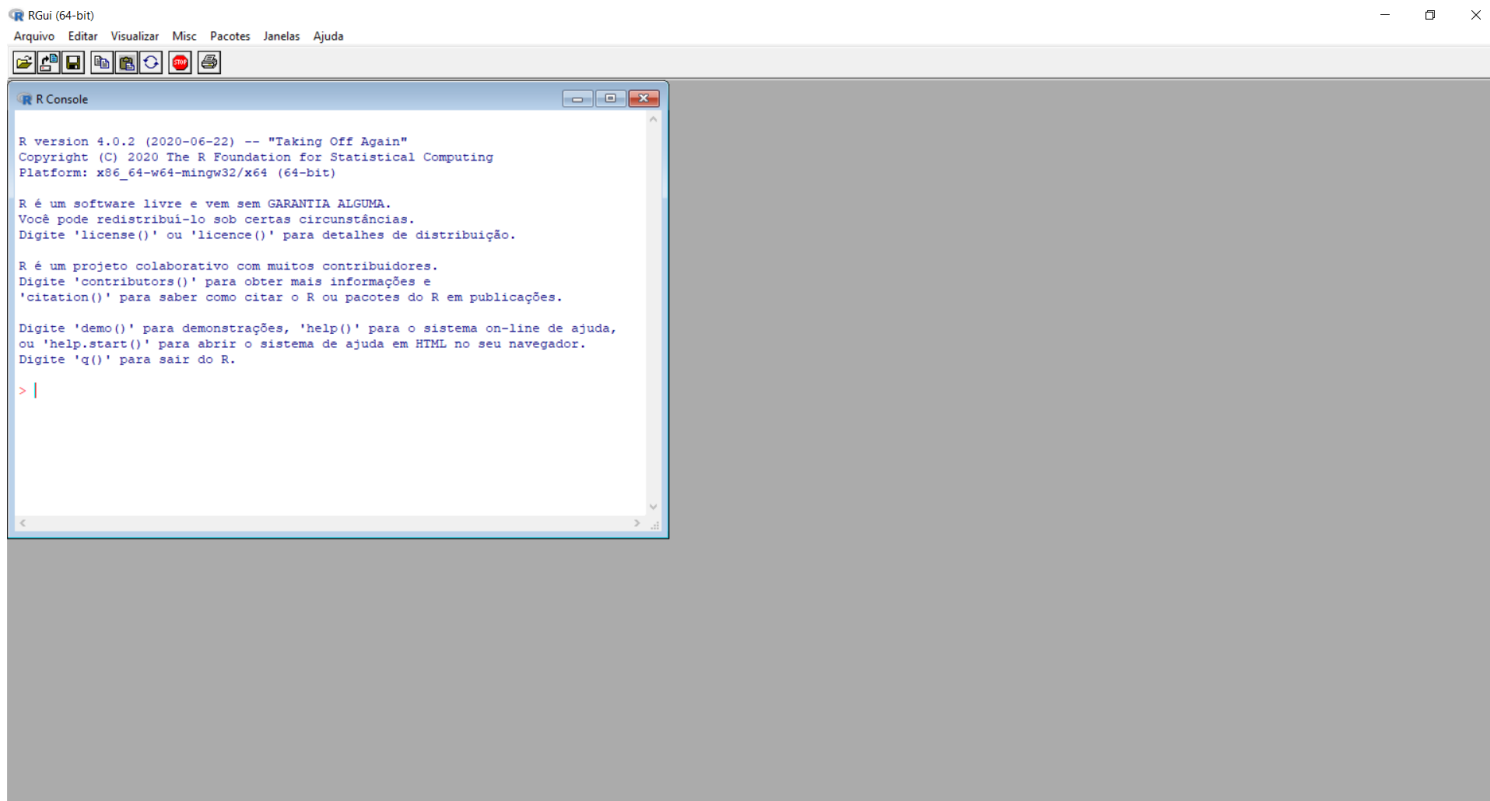


- [R Studio](#)

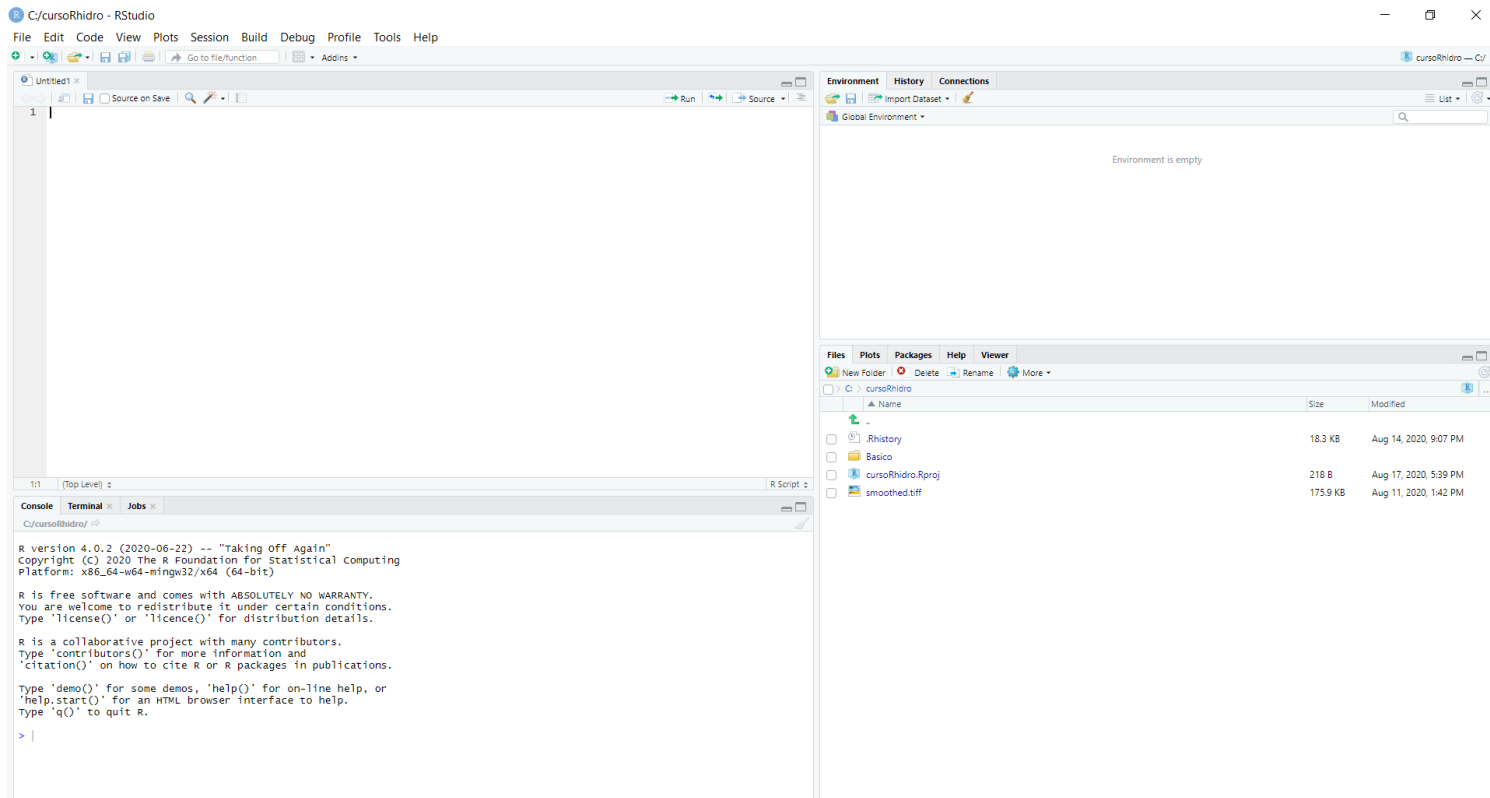


RStudio é uma plataforma com inúmeras ferramentas úteis e que roda a linguagem R. É possível usar só o R por sua interface, mas só é possível usar o RStudio com o R instalado na máquina.

Aparência do R e RStudio

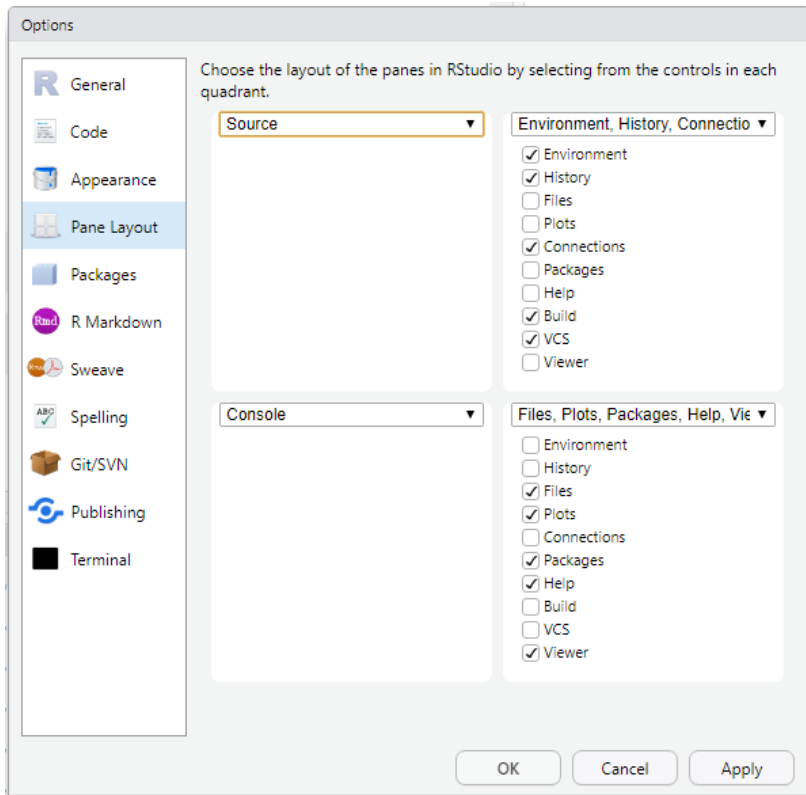


Aparência do R e RStudio



Aparência do R e RStudio

O layout é ajustável em `view -> Panes -> Pane Layout`



Comandos no R

Antes de tudo!!! Saber onde se está trabalhando!

```
getwd();setwd("C:/cursoRhidro")
```

```
## [1] "C:/cursoRhidro/Basico"
```

```
dir()
```

```
## [1] "Aulas PDF"          "Basico"              "Controle"  
## [4] "cursoRhidro.Rproj" "Funcoes"             "Graficos"  
## [7] "Leitura"            "Manuseio"           "manuseio.txt"  
## [10] "soma2.R"
```

```
source("soma2.R")
```

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

Comandos no R

R é utilizável como uma calculadora

```
setwd("C:/cursoRhidro/Basico")
```

```
2 + 2
```

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

```
12 * 3546541322
```

```
## [1] 42558495864
```

Comandos no R

R é utilizável como uma calculadora

```
2 / 2
```

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

```
12 ** 4
```

```
## [1] 20736
```

```
12 ^ 4
```

```
## [1] 20736
```

Comandos no R

R trabalha igualmente com escalares e vetores

```
c(1,3,5,7) + c(2,4,6,8)
```

```
## [1] 3 7 11 15
```

```
c(1,3,5,7) * c(2,4,6,8)
```

```
## [1] 2 12 30 56
```

A função `c()` é usada para concatenar valores, criando vetores de valores da mesma classe.

Comandos no R

Ao usar o operador multiplicativo `*` o R entende que será feita a operação elemento a elemento. Para multiplicação de matrizes, o operador correto é `%*%`. Importante na vetorização!

```
as.vector(c(1,3,5,7)) %*% t(as.vector(c(2,4,6,8)))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    4    6    8
## [2,]    6   12   18   24
## [3,]   10   20   30   40
## [4,]   14   28   42   56
```

Comandos no R

A forma usual de se atribuir um valor a um elemento é usando o operador `<-`.

```
x <- 2
```

```
x
```

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

```
print(x)
```

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

```
class(x)
```

```
## [1] "numeric"
```


Comandos no R

Pode-se atribuir, por exemplo, um valor não numérico a `x`.

```
x <- "Teste"  
print(x)
```

```
## [1] "Teste"
```

```
class(x)
```

```
## [1] "character"
```

Ao atribuir `x` a um novo valor, o valor anterior é apagado, implicitamente. Uma forma de remover valores é usando a função `rm()`,

```
rm(x)
```

Comandos no R

Os operadores `<-`, `=` e `==` todos existem no R. `<-` e `=` são equivalentes, mas `==` funciona para avaliar uma igualdade e não de atribuir um valor a um elemento do R e retorna uma variável lógica **TRUE** ou **FALSE**. Por exemplo:

```
x = 2
```

```
x == 2
```

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

```
x == 3
```

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

Comandos no R

Outros operadores que resultam em lógicos:

```
x != 2
```

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

```
x < 3
```

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

```
x >= 2
```

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

Objetos do R

São cinco os tipos básicos de **Classes** de objetos do R:

- Character
- Número (real)
- Inteiro
- Complexo
- Lógico

A forma mais básica de armazenar objetos dessas classes é na forma de vetor e os vetores podem ser criados, com a função `c()`. A regra geral é que os vetores só armazenam objetos de mesma classe, a única exceção é são objetos do tipo `list`, como veremos mais à frente.

Objetos do R

São cinco os tipos básicos de **Classes** de objetos do R:

- Character
- Número (real)
- Inteiro
- Complexo
- Lógico (TRUE/FALSE)

Via de regra, o R converte números quando se utiliza um valor numérico. Para se criar uma variável do tipo `integer`, deve-se acrescentar a letra `L` ao final:

```
x <- 2L  
class(x)
```

```
## [1] "integer"
```

Objetos do R - coerção de variáveis

Vetores só podem conter variáveis de um tipo:

```
x <- c(1,2,3,4)
class(x)
```

```
## [1] "numeric"
```

```
y <- c(FALSE,TRUE,T,F)
print(y)
```

```
## [1] FALSE TRUE TRUE FALSE
```

```
class(y)
```

```
## [1] "logical"
```

Objetos do R - coerção de variáveis

Vetores só podem conter variáveis de um tipo:

```
x <- c("oi", "bom", "dia", "pessoal")
```

```
x
```

```
## [1] "oi"      "bom"     "dia"     "pessoal"
```

```
class(x)
```

```
## [1] "character"
```

Objetos do R - coerção de variáveis

Vetores só podem conter variáveis de um tipo:

```
x <- c("oi",1,"dia",2)
```

```
x
```

```
## [1] "oi" "1" "dia" "2"
```

```
class(x)
```

```
## [1] "character"
```

COERÇÃO IMPLÍCITA

Objetos do R - coerção de variáveis

Coerção explícita:

```
x <- as.character(c("oi",1,"dia",2))
```

```
x
```

```
## [1] "oi" "1" "dia" "2"
```

```
class(x)
```

```
## [1] "character"
```

Objetos do R - coerção de variáveis

Coerção explícita - às vezes o R não sabe o que fazer e retorna **NA** (Non-Available):

```
x <- as.numeric(c("oi",1,"dia",2))
```

```
## Warning: NAs introduzidos por coerção
```

```
x
```

```
## [1] NA 1 NA 2
```

```
class(x)
```

```
## [1] "numeric"
```

Objetos do R - Vetores

Para criar vetores:

```
x <- c(1,2,3,4,5,6,7,8,9,10)
```

```
x
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
x <- c(1:10)
```

```
x
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
x <- vector("numeric", length = 10)
```

```
x
```

```
## [1] 0 0 0 0 0 0 0 0 0 0
```

Objetos do R - Vetores

Para criar vetores:

```
x <- c(11:20)
```

```
x
```

```
## [1] 11 12 13 14 15 16 17 18 19 20
```

```
x[c(1,3,10)]
```

```
## [1] 11 13 20
```

```
x[c(2,3,11)]
```

```
## [1] 12 13 NA
```

Objetos do R - Matrizes

Matrizes são objetos semelhantes a vetores, porém, com o atributo dimensão `dim`.

```
m <- matrix(nrow = 2, ncol = 3); m
```

```
##      [,1] [,2] [,3]
## [1,]  NA  NA  NA
## [2,]  NA  NA  NA
```

```
attributes(m)
```

```
## $dim
## [1] 2 3
```

```
dim(x)
```

```
## NULL
```

Objetos do R - Matrizes

Matrizes podem ser criadas a partir de vetores, adicionando o atributo dimensão.

```
m <- matrix(x, nrow = 2, ncol = 5); m
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]  11  13  15  17  19  
## [2,]  12  14  16  18  20
```

```
m <- matrix(x, nrow = 2, ncol = 5, byrow = TRUE); m
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]  11  12  13  14  15  
## [2,]  16  17  18  19  20
```

Objetos do R - Matrizes

Para selecionar elementos de uma matriz, é necessário especificar as duas dimensões

```
m <- matrix(1:9, nrow = 3); m
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4    7  
## [2,]    2    5    8  
## [3,]    3    6    9
```

```
m[3,2]
```

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

Objetos do R - Matrizes

Selecionar elementos de matrizes quadradas:

```
diag(m)
```

```
## [1] 1 5 9
```

```
lower.tri(m)
```

```
##      [,1] [,2] [,3]  
## [1,] FALSE FALSE FALSE  
## [2,] TRUE  FALSE FALSE  
## [3,] TRUE   TRUE  FALSE
```


Objetos do R - Matrizes

Selecionar elementos de matrizes quadradas:

```
m[lower.tri(m)] <- NA; m
```

```
##      [,1] [,2] [,3]  
## [1,]   1   4   7  
## [2,]  NA   5   8  
## [3,]  NA  NA   9
```

```
diag(m) <- NA; m
```

```
##      [,1] [,2] [,3]  
## [1,]  NA   4   7  
## [2,]  NA  NA   8  
## [3,]  NA  NA  NA
```

Objetos do R - Matrizes

Matrizes podem ser unidas com simplicidade:

```
x <- c(1:10)
y <- rnorm(10, 20, 5)
uniao <- cbind(x, y); uniao
```

```
##      x      y
## [1,] 1 22.50108
## [2,] 2 21.47637
## [3,] 3 26.04341
## [4,] 4 26.38291
## [5,] 5 23.27002
## [6,] 6 15.88788
## [7,] 7 17.41861
## [8,] 8 25.26503
## [9,] 9 19.81564
## [10,] 10 8.91168
```

Objetos do R - Matrizes

Matrizes podem ser unidas com simplicidade:

```
x <- c(1:10)
y <- rnorm(10, 20, 5)
uniao <- rbind(x, y); uniao

##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
## x  1.00000  2.00000  3.000  4.000000  5.00000  6.00000  7.00000  8.000000
## y 23.65652 24.40404 15.906  8.957705 16.62886 19.38347 23.06969  7.885705
##      [,9]      [,10]
## x  9.00000 10.0000
## y 26.74022 18.2076

dim(uniao)

## [1]  2 10
```

Objetos do R

Operações em/com vetores:

```
x <- c(1:10)
sqrt(9*x/5)
```

```
## [1] 1.341641 1.897367 2.323790 2.683282 3.000000 3.286335 3.549648 3.794733
## [9] 4.024922 4.242641
```

```
x <- c(1:5)
y <- c(1:4)
x+y
```

```
## Warning in x + y: comprimento do objeto maior não é múltiplo do comprimento do
## objeto menor
```

```
## [1] 2 4 6 8 6
```

Objetos do R

Operações em/com vetores:

```
x <- c(1:12); names(x) <- month.abb
```

```
x
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
##  1  2  3  4  5  6  7  8  9 10 11 12
```

```
x[1]
```

```
## Jan  
##  1
```

```
x["Jan"]
```

```
## Jan  
##  1
```

Objetos do R

Operações em/com vetores:

```
x > 6
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
x[x > 6]
```

```
## Jul Aug Sep Oct Nov Dec
## 7 8 9 10 11 12
```

```
sum(x > 6)
```

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

Objetos do R - Listas

Listas são elementos importantes na programação em R. Uma das vantagens de se saber utilizar bem esse tipo de elemento é que nelas é possível armazenar elementos de diferentes tipos.

Objetos do R - Listas

Para criar uma lista:

```
x <- list(1, "Ola", 1+5i, TRUE, 4L)
```

```
x
```

```
## [[1]]  
## [1] 1  
##  
## [[2]]  
## [1] "Ola"  
##  
## [[3]]  
## [1] 1+5i  
##  
## [[4]]  
## [1] TRUE  
##  
## [[5]]  
## [1] 4
```


Objetos do R - Listas

```
lapply(x, class)
```

```
## [[1]]  
## [1] "numeric"  
##  
## [[2]]  
## [1] "character"  
##  
## [[3]]  
## [1] "complex"  
##  
## [[4]]  
## [1] "logical"  
##  
## [[5]]  
## [1] "integer"
```

Objetos do R - Listas

Para selecionar dados de listas, utiliza-se duplo coquete `[[]]`:

```
x <- list(c(1,2,3), c("Ola", "Mundo"), 1+5i, TRUE, 4L)
```

```
x[[1]]
```

```
## [1] 1 2 3
```

```
x[[2]][2]
```

```
## [1] "Mundo"
```

```
x[[5]] + 5
```

```
## [1] 9
```

Objetos do R - Fatores

`factor` é uma importante classe de elemento no R para definir variáveis categóricas. São úteis em modelos de regressão linear com variáveis categóricas de entrada (estações do ano, classes de solo, etc.):

```
x <- factor(c("primavera", "inverno", "primavera", "verão", "outono", "primavera", "outono", "outono", "verão", "verão"))
```

```
## [1] primavera inverno primavera verão outono primavera outono
## [8] outono verão inverno verão primavera primavera outono
## Levels: inverno outono primavera verão
```

```
table(x)
```

```
## x
## inverno outono primavera verão
##      2      4      5      3
```

Objetos do R - Data Frames

A classe de `data.frame` é muito utilizada em R. É o que se utiliza para dados tabulados. São semelhantes às listas, em que todos os elementos do `data.frame` têm o mesmo comprimento.

```
DF <- data.frame(x = c(1:4), y = rnorm(4))
```

```
DF
```

```
##      x      y
## 1 1 -0.2600116
## 2 2 -0.2061026
## 3 3  0.3331557
## 4 4 -1.0635091
```

```
class(DF)
```

```
## [1] "data.frame"
```

Objetos do R - Data Frames

- `matrix`: todos os elementos são da mesma classe;
- `data.frame`: dentro de cada coluna, os objetos pertencem à mesma classe, mas diferentes colunas podem pertencer a diferentes classes;
- `list`: cada elemento pode pertencer a uma classe diferente.

Quando são lidos objetos externos (planilhas excel, arquivos `.txt` ou `.csv`), usando as funções do R `read.table()` ou `read.csv()`, por exemplo, o objeto criado é um `data.frame`.

Objetos do R - Data Frames

Coerção de classe para `data.frame`.

```
m <- matrix(c(1:9), nrow = 3)
as.data.frame(m)
```

```
##   V1 V2 V3
## 1  1  4  7
## 2  2  5  8
## 3  3  6  9
```

Objetos do R - Atributos

Coerção de classe para `data.frame`.

```
m <- matrix(c(1:9), nrow = 3); attributes(m)
```

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

```
attributes(as.data.frame(m))
```

```
## $names  
## [1] "V1" "V2" "V3"  
##  
## $class  
## [1] "data.frame"  
##  
## $row.names  
## [1] 1 2 3
```

Objetos do R - Atributos

Em resumo:

- Operadores matemáticos: + , - , * , / , ** , ^
- Atribuição de valores: <- , -> , =
- Operadores que resultam em valores lógicos: == , != , >= , <= , > , <
- Operações com matrizes, listas, etc: objeto a objeto
- Operações com vetores: %*%
- Tipos básicos: "numeric", "character", "complex", "logical", "integer"
- Tipos de coleções: vetores, matrizes, data frames, listas, **tibble** (tipo especial de data.frame)