Introdução ao R

Ma. Pétala Tuy
Cientista de Dados – ATOS Brasil
Pesquisadora associada do Centro de Excelência em Pesquisa Aplicada
em Inteligência Artificial para a Industria do SENAI CIMATEC/ATOS

Conteúdo

- O que é R? Por que R?
- Como Instalar
- Primeiro Contato com o R
- Help no R
- R como Calculadora
- If, else e else if, For, Ifelse, While
- Vetorização / Reciclagem
- Funções
- Estruturas de dados
- Dimensões e comprimento
- Subsetting
- Explorando o conjunto de dados!

O que é R?

- R é uma linguagem com **foco explícito em estatística**. Se a matemática é o fator de maior peso na construção do seu projeto, não hesite em escolhê-la.

- R foi desenvolvida por **estatísticos para estatística**. Possui excelente documentação e um riquíssimo ecossistema de pacotes para diferentes áreas de aplicação.

Por que R?

- **De graça!**
- Menor curva de aprendizado
- Escrito por estatísticos, para estatísticos
- Atualizado e com grande gama de funções analíticas
- Comunidade ativa e crescente
- Permite **analisar**, **manusear** e **limpar** dados com mais **facilidade** e **agilidade**
- Possui sistema e **pacotes** de **visualização** de dados que é **referência**
- Tem, potencialmente, **capacidade** de **interagir** com qualquer **software** e **linguagem** de programação, inclusive com o SAS Excel
- Facilita a análise de dados colaborativa

Como Instalar

- Baixar o R:

https://cran.r-project.org/bin/windows/base/

- Baixar o RStudio:

https://rstudio.com/products/rstudio/download/

RStudio

File Edit Code View Plots Session Project Build Tools Help

Go to file/function

Project: (None)

Untitled1*

Source on Save

Run

Source

```
1  library(ggplot2)
2  df<-data.frame(var=rnorm(1000),group=rep(LETTERS[1:4],250))
3  qplot(group,var,geom="boxplot",data=df,fill=group)
```

2:1 (Top Level) R Script

Workspace History

Import Dataset

Data

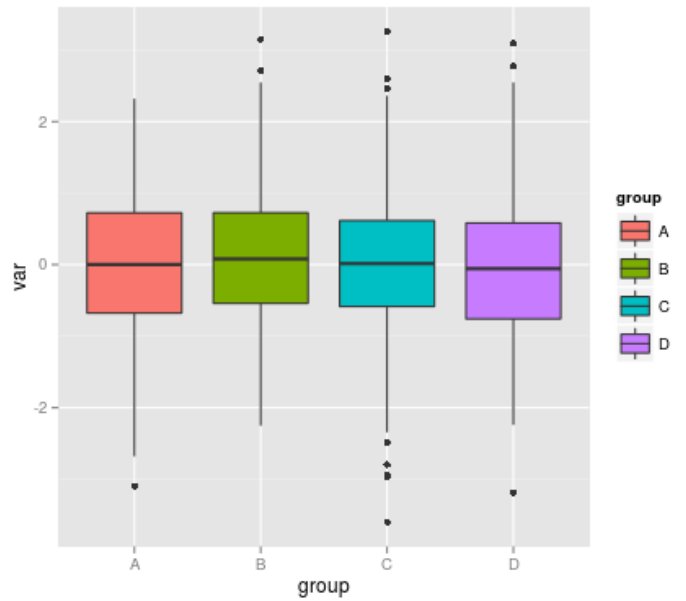
df	1000 obs. of 2 variables

Console ~/

```
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i686-pc-linux-gnu (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(ggplot2)
> df<-data.frame(var=rnorm(1000),group=rep(LETTERS[1:4],250))
> qplot(group,var,geom="boxplot",data=df,fill=group)
>
```

Files Plots Packages Help

Zoom Export Clear All



Primeiro Contato com o R

- getwd()
- setwd()
- install.packages("<nome_do_pacote>")
- library(<nome_do_pacote>)
- Instale o pacote “ggplot2”

Help no R

Google

R Como Calculadora

Operadores Aritméticos

Operador	Descrição
x + y	Adição de x com y
x - y	Subtração de y em x
x * y	Multiplicação de x e y
x / y	Divisão de x por y
x^y ou x**y	x elevado a y-ésima potência
x%%y	Resto da divisão de x por y (módulo)
x%/%y	Parte inteira da divisão de x por y

R Como Calculadora

Operadores Lógicos

operador	descricao
x < y	x menor que y?
x <= y	x menor ou igual a y?
x > y	x maior que y?
x >= y	x maior ou igual a y?
x == y	x igual a y?
x != y	x diferente de y?
!x	Negativa de x
x | y	
x & y	x e y são verdadeiros?
xor(x, y)	x ou y são verdadeiros (apenas um deles)?

If, else e else if

```
# If, else e else if

condicao_1 = T
condicao_2 = F

if(condicao_1){

  print('Condição 1 é verdadeira')

} else if (condicao_2){

  print('Condição 2 é verdadeira')

} else {

  print('Nenhuma condição é verdadeira')

}
```

If, else e else if

- Além de TRUE e FALSE, o R aceita 1 e 0, respectivamente
- Objetos character, **NA**, **NaN** e list não são interpretáveis como lógicos
- Else e else if são opcionais.

For()

```
# For

for (i in 1:10){
  print(paste0('i = ',i))
}
```

```
[1] "i = 1"
[1] "i = 2"
[1] "i = 3"
[1] "i = 4"
[1] "i = 5"
[1] "i = 6"
[1] "i = 7"
[1] "i = 8"
[1] "i = 9"
[1] "i = 10"
```

Ifelse()

```
x <- c(6:-4)

sqrt(x)  #- gives warning

sqrt(ifelse(x >= 0, x, NA))  # no warning
```

```
> x <- c(6:-4)
> x
 [1]  6  5  4  3  2  1  0 -1 -2 -3 -4
> sqrt(x)  #- gives warning
 [1] 2.449490 2.236068 2.000000 1.732051 1.414214 1.000000 0.000000      NaN
 [9]      NaN      NaN      NaN
Warning message:
In sqrt(x) : NaNs produced
>
> sqrt(ifelse(x >= 0, x, NA))  # no warning
 [1] 2.449490 2.236068 2.000000 1.732051 1.414214 1.000000 0.000000       NA
 [9]       NA       NA       NA
```

While()

```
# While (NÃO EXECUTAR)

while(T){
  print('Loop infinito')
}
```

Vetorização

- Vetores são objetos que guardam dados.

```
# Vetorização
numeros = 1:5
numeros

log10(numeros)

2^numeros
```

```
> numeros = 1:5
> numeros
[1] 1 2 3 4 5
> log10(numeros)
[1] 0.0000000 0.3010300 0.4771213 0.6020600 0.6989700
> 2^numeros
[1]  2  4  8 16 32
```

Reciclagem

- Exemplo: Soma de vetores de tamanhos diferentes

```
# Reciclagem
x <- c(1,5)
x
y <- c(1,10,100,1000)
y
x + y
```

```
> x <- c(1,5)
> x
[1] 1 5
> y <- c(1,10,100,1000)
> y
[1]    1   10  100 1000
> x + y
[1]    2   15  101 1005
```

Funções

- Funções também são objetos!
- Funções podem ser passadas como **argumentos** de outras **funções**
- Use suas funções como se tivessem vindas com o R:

  nome_da_funcao(...)

- Crie uma função sempre que for **repetir** o **código** e for mudar poucas coisas entre essas repetições

Funções

```r
# função que ecoa uma palavra
ecoar <- function(palavra, n_ecos = 3) {
  paste(c(rep(palavra, n_ecos), "!"), collapse = " ")
}

ecoar("eco")
ecoar("eco",5)
```

```
> ecoar("eco")
[1] "eco eco eco !"
> ecoar("eco",5)
[1] "eco eco eco eco eco !"
```

Funções

Funções básicas	Significado
require(nome do pacote)	Ativa o pacote entre parênteses
install.packages("pacotes")	Instala o pacote selecionado, quando está conectada a internet
Help(nome da função)	Fornece informações da função entre parênteses
read.table(nome do arquivo)	Importação de arquivos no formato txt, para o R

Funções estatísticas	Significado
max(x)	Valor máximo
min(x)	Valor mínimo
sum(x)	Soma dos valores
mean(x)	Média
median(x)	Mediana
sd(x)	Desvio padrão
var(x)	Variância

Funções matemáticas	Significado
length(x)	Tamanho do vetor
log(x)	Logaritmo na base e
exp(x)	Exponencial
sqrt(x)	Raiz quadrada

Estruturas de dados no R

- Estruturas de dados básicas no R:

Vetores - **Fatores** - **Matrizes** - **Listas** – **DataFrames**

- **Vetor**: homogêneo e unidimensional
- **Fator:** heterogêneo e unidimensional
- **Matriz**: homogêneo e bidimensional
- **Lista**: heterogêneo
- **Data frame**: heterogêneo

Estruturas de dados no R

- **Vetores**: Sequências de valores numéricos ou não

```
x <- c(1, 2, 5, 7,10) ;x ;typeof(x)
x <- c(1, 2, 5, 7, 10.5) ;x ;typeof(x)
x <- c(T, F, TRUE, FALSE) ;x ;typeof(x)
x <- c("string 1", "string 2") ;x ;typeof(x)
```

Estruturas de dados no R

- Quando dois tipos de objetos são inseridos uma estrutura homogênea (vetores ou matrizes), o R converte converterá o objeto para o tipo mais flexível, na ordem:

1. **Logical**
2. **Integer**
3. **Double**
4. **Character**

- Na lista acima, character é o tipo mais flexível.

Estruturas de dados no R

- **Fatores**: Vetores de inteiros utilizados para ordenar strings

- Contém apenas valores pré-definidos, chamados *levels*;

```
c5 <- c("M", "F", "F", "F", "M", "M"); c5 ; typeof(c5)
f5 <- as.factor(c5); f5 ; typeof(f5)
f5 <- factor(c5, levels = c("M","F")); f5 ; typeof(f5)
```

Estruturas de dados no R

- Sempre tome cuidado ao converter factors em objetos numéricos

```
f <- factor(c("2", "3", "1", "10"))
as.numeric(f)
as.numeric(as.character(f))
```

```
> as.numeric(f)
[1] 3 4 1 2
> as.numeric(as.character(f))
[1]  2  3  1 10
```

Estruturas de dados no R

- **Matrizes**: Combinações de vetores

```
> matrix(1:12, ncol=3)
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

Estruturas de dados no R

- **Listas**: Utilizados para combinar diferentes objetos em um objeto único.

```
pessoa <- list(idade=21, nome='Fred', score=c(65,78,55))
```

```
> pessoa$idade
[1] 21
> pessoa$nome
[1] "Fred"
> pessoa$score
[1] 65 78 55
```

Estruturas de dados no R

- **Data Frames**: Tabela de uma base de dados, em que cada linha corresponde a um registo (linha) da tabela

```
df <- data.frame(x = 1:4,
                 y = c("um", "dois", "tres", "quatro"),
                 z = T)
```

```
> df
  x       y    z
1 1      um TRUE
2 2    dois TRUE
3 3    tres TRUE
4 4 quatro TRUE
```

Estruturas de dados no R

- É possível combinar data frames usando as funções **rbind()** e **cbind()**

```
df1 <- data.frame(x = 3:4, y = c("s", "s"), z = T)
df2 <- data.frame(x = 1:2, y = c("n", "n"), z = F)
rbind(df1, df2)
cbind(df1, df2)
```

```
> rbind(df1, df2)
  x y     z
1 3 s  TRUE
2 4 s  TRUE
3 1 n FALSE
4 2 n FALSE
> cbind(df1, df2)
  x y    z x y     z
1 3 s TRUE 1 n FALSE
2 4 s TRUE 2 n FALSE
```

Subsetting

- Chamamos de *subsetting* a seleção de um subconjunto de um objeto. No R, existem três tipos principais de operação de subsetting:

```
x <- c(13, 8, 5, 3, 2, 1, 1)
x[c(1,2,3)]
order(x)
x[order(x)]
x[-c(2, 5,6)]
```

```
> x <- c(13, 8, 5, 3, 2, 1, 1)
> x[c(1,2,3)]
[1] 13  8  5
> order(x)
[1] 6 7 5 4 3 2 1
> x[order(x)]
[1]  1  1  2  3  5  8 13
> x[-c(2, 5,6)]
[1] 13  5  3  1
```

Estruturas de dados no R

- Lendo um data frame no formato excel.

```
df <- read_excel("banco_estadual_COVID19_15_07_2020_reduzido.xlsx")
head(df)
```

```
> df
     DATA DA COLETA DO TESTE IDADE EM ANOS            TIPO DE TESTE RESULTADO DO TESTE
  1:              07/06/2020            31     RT-PCR EM TEMPO REAL            NEGATIVO
  2:                    <NA>            40                     <NA>                <NA>
  3:                    <NA>            36     RT-PCR EM TEMPO REAL            NEGATIVO
  4:              30/06/2020            34     RT-PCR EM TEMPO REAL            POSITIVO
  5:              25/05/2020            50 TESTE RAPIDO - ANTICORPO            POSITIVO
 ---
```

Estruturas de dados no R

- Escrevendo um data frame no formato csv

- Lendo um data frame no formato csv

```
# Escrevendo arquivo csv
write.csv(head(df), file = 'Covid.csv')

# Lendo arquivo csv
read.csv('Covid.csv')
```

Dimensões e comprimentos

```
dim(c(1,2,3))
nrow(c(1,2,3))
ncol(c(1,2,3))
length(c(1,2,3))

dim(df)
nrow(df)
ncol(df)
length(df)
```

```
> dim(c(1,2,3))
NULL
> nrow(c(1,2,3))
NULL
> ncol(c(1,2,3))
NULL
> length(c(1,2,3))
[1] 3
> dim(df)
[1] 5837   29
> nrow(df)
[1] 5837
> ncol(df)
[1] 29
> length(df)
[1] 29
```

Explorando o conjunto de dados

- Renomear colunas
- Configurar datas
- Removendo colunas desnecessárias
- Transformando variáveis em numéricas
- Subsetting data frame