

# **Oficina de R - Básico**

**R-Ladies São Paulo**

**02/02/2019**

# What is R-Ladies?

gender diversity

## How?

## For whom?

# Mission

# How did R-Ladies begin?



**Gabriela de Queiroz**

**October 1, 2012**

**San Francisco, California (United States)**

138

R-Ladies groups on meetup.com



44

R-Ladies Countries

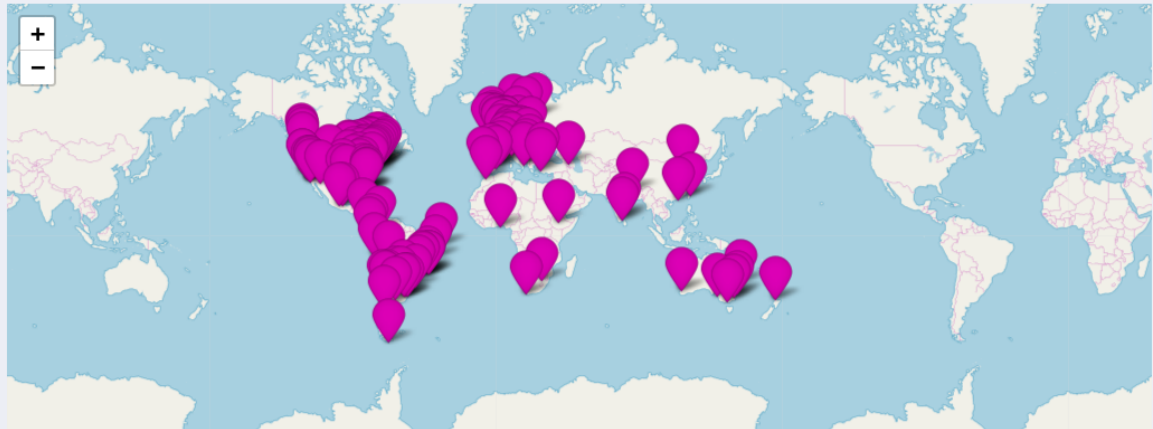


138

R-Ladies Cities



39133

R-Ladies members on  
meetup.com

# **Code of conduct**

# **How to get involved**

# R-Ladies São Paulo



**First R-Ladies São Paulo Meetup**

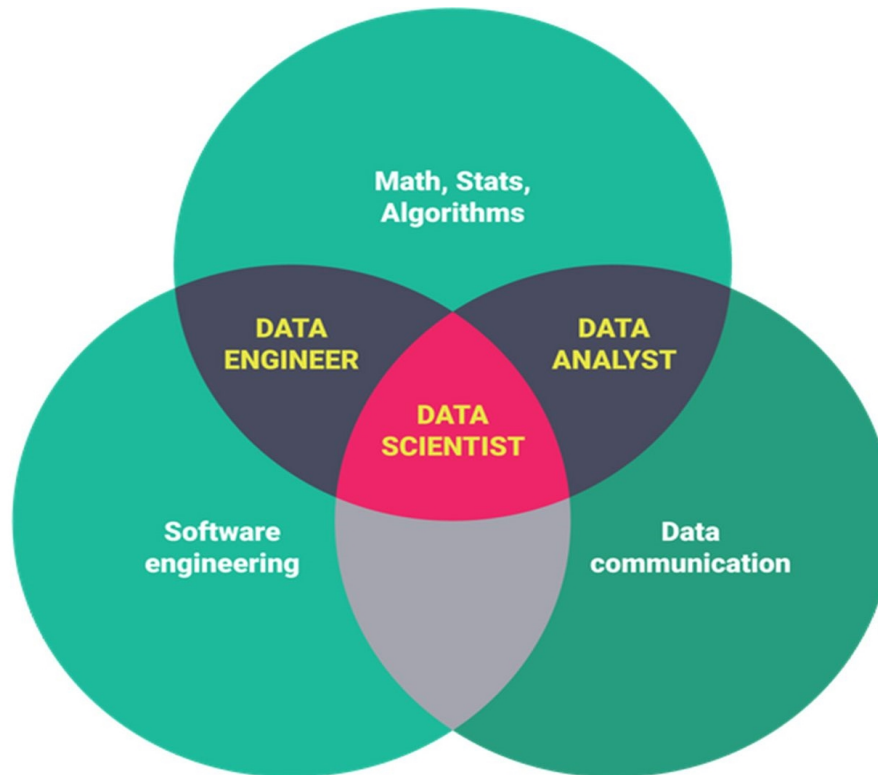


# Cronograma de hoje

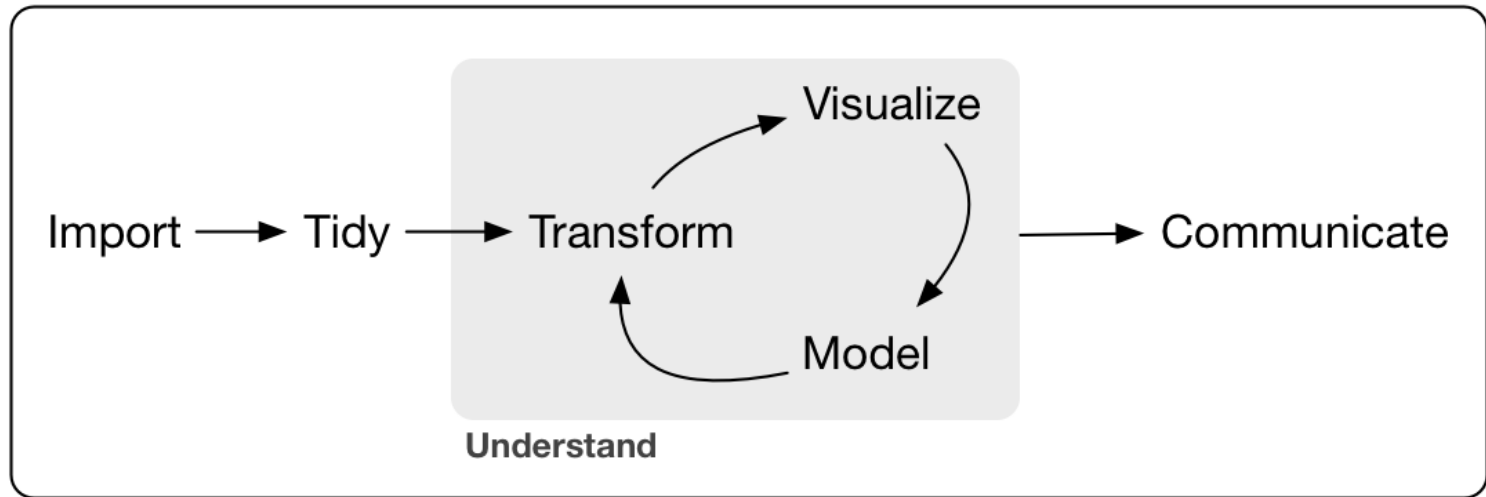
- 
- 
- 
- 
-

# O que é Data Science?

Data Science    Ciência de Dados



# Ciclo da Ciência de Dados



Program

Fonte:

# Ciência de Dados



TADA

DATA



SORTED



ARRANGED



PRESENTED  
VISUALLY



DATASCIENCE **GO**

Fonte:

# O que é o R?

- 
- 



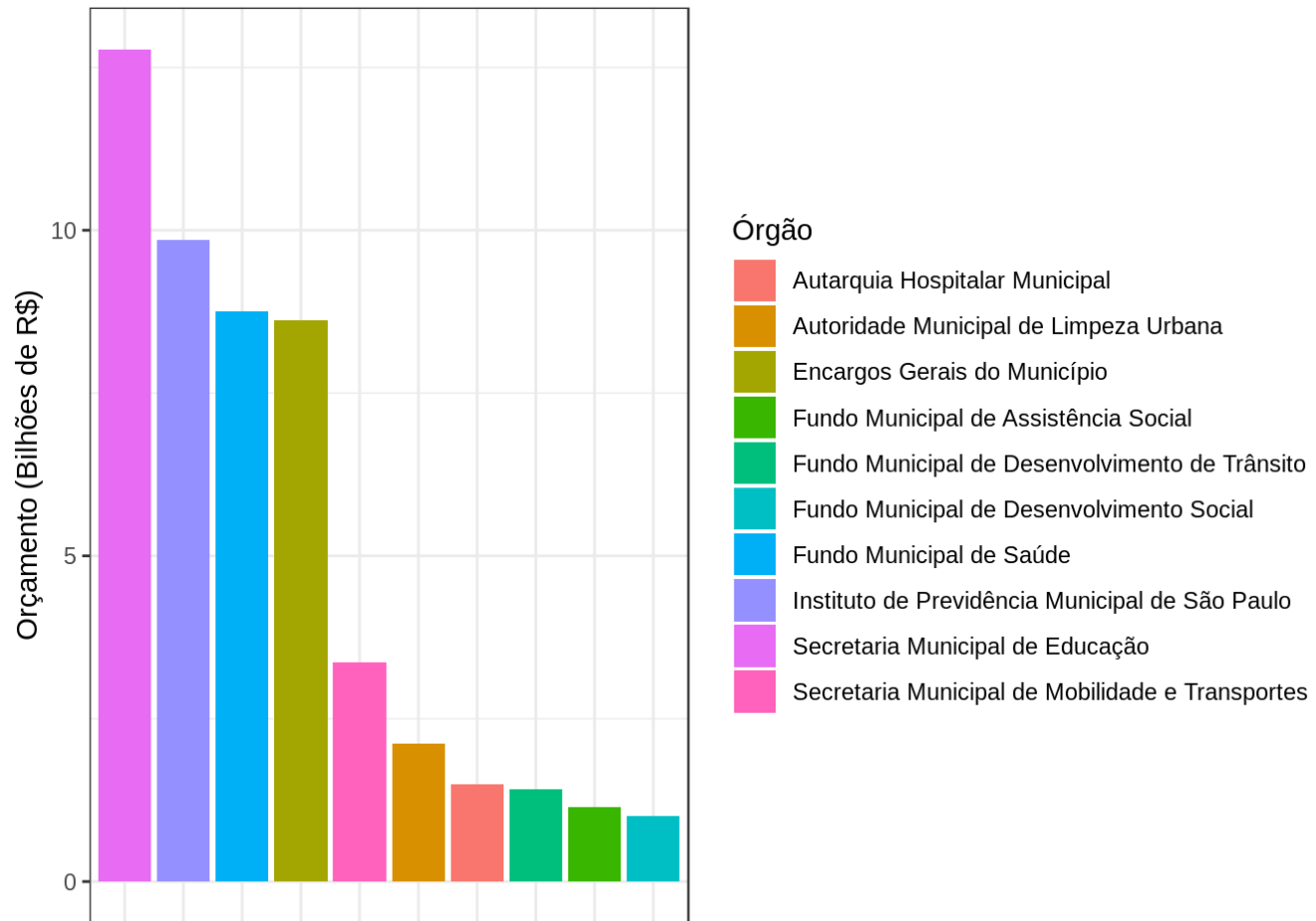
# Por que usar o R?

- 
- 
- 
- 
-

# O que podemos fazer com o R?

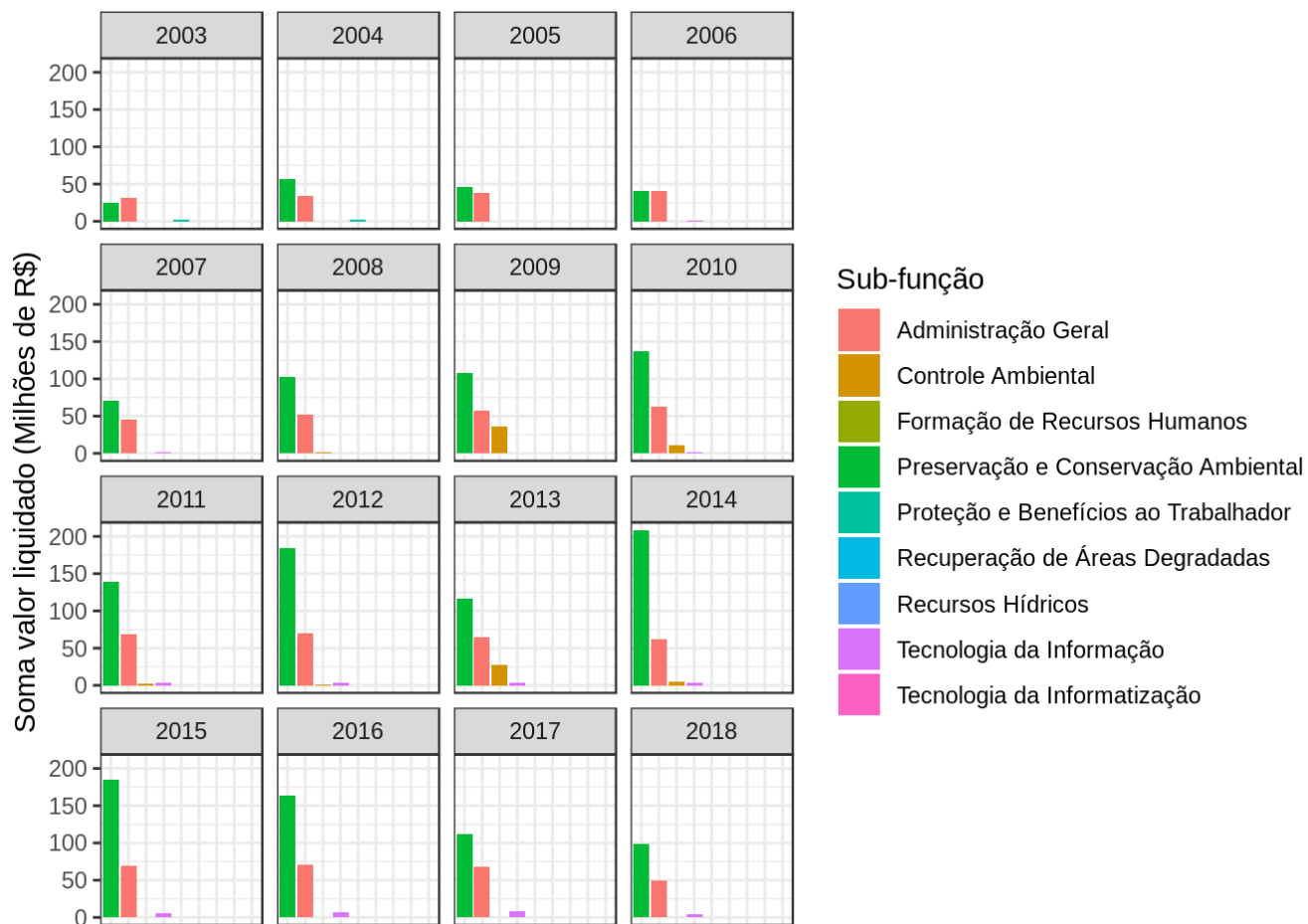
- 
- 
- 
- 
- 
- 
-

## Exemplo: Gráfico elaborado com R - Proposta Orçamentária PMSP 2019 - 10 maiores orçamentos

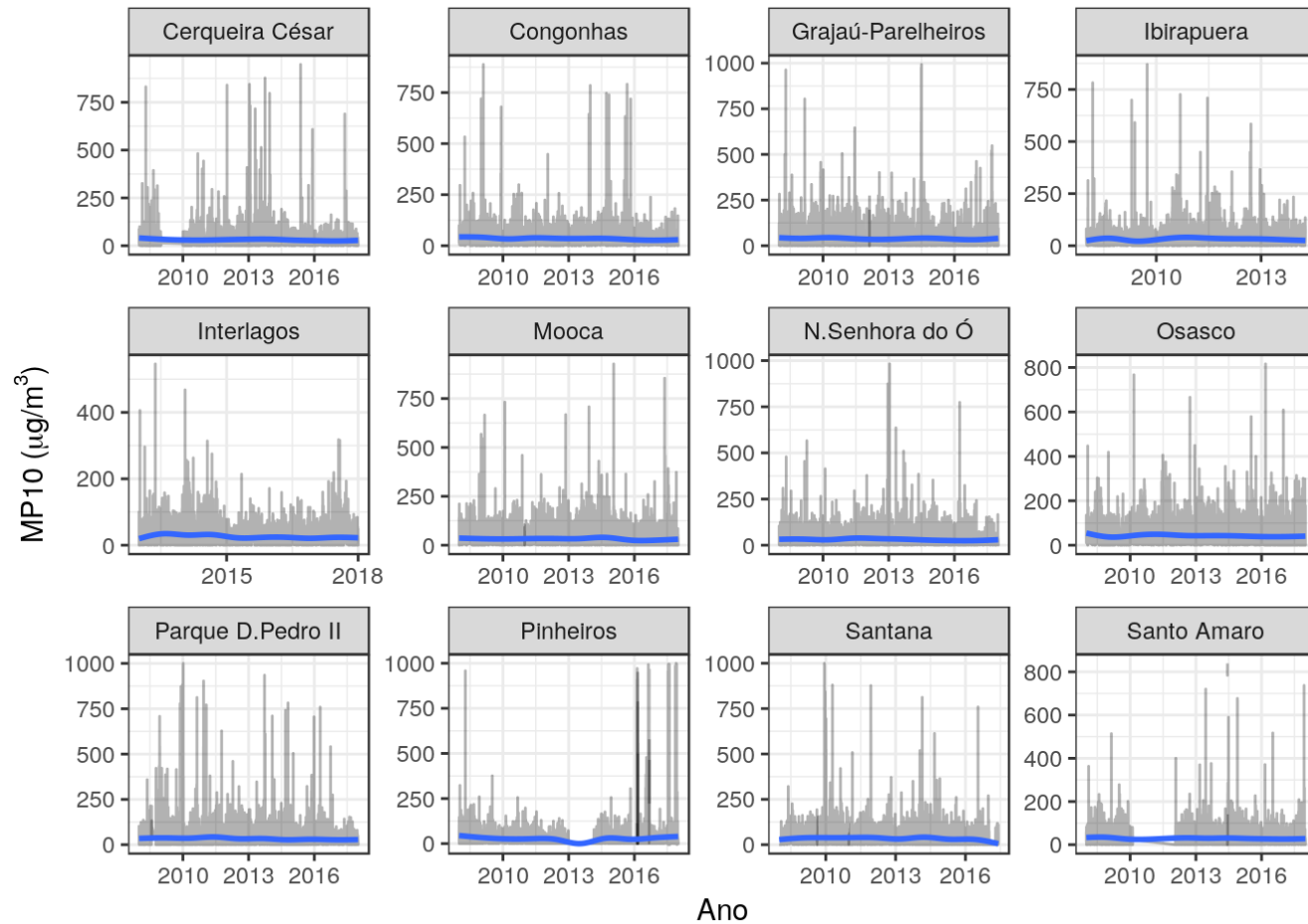




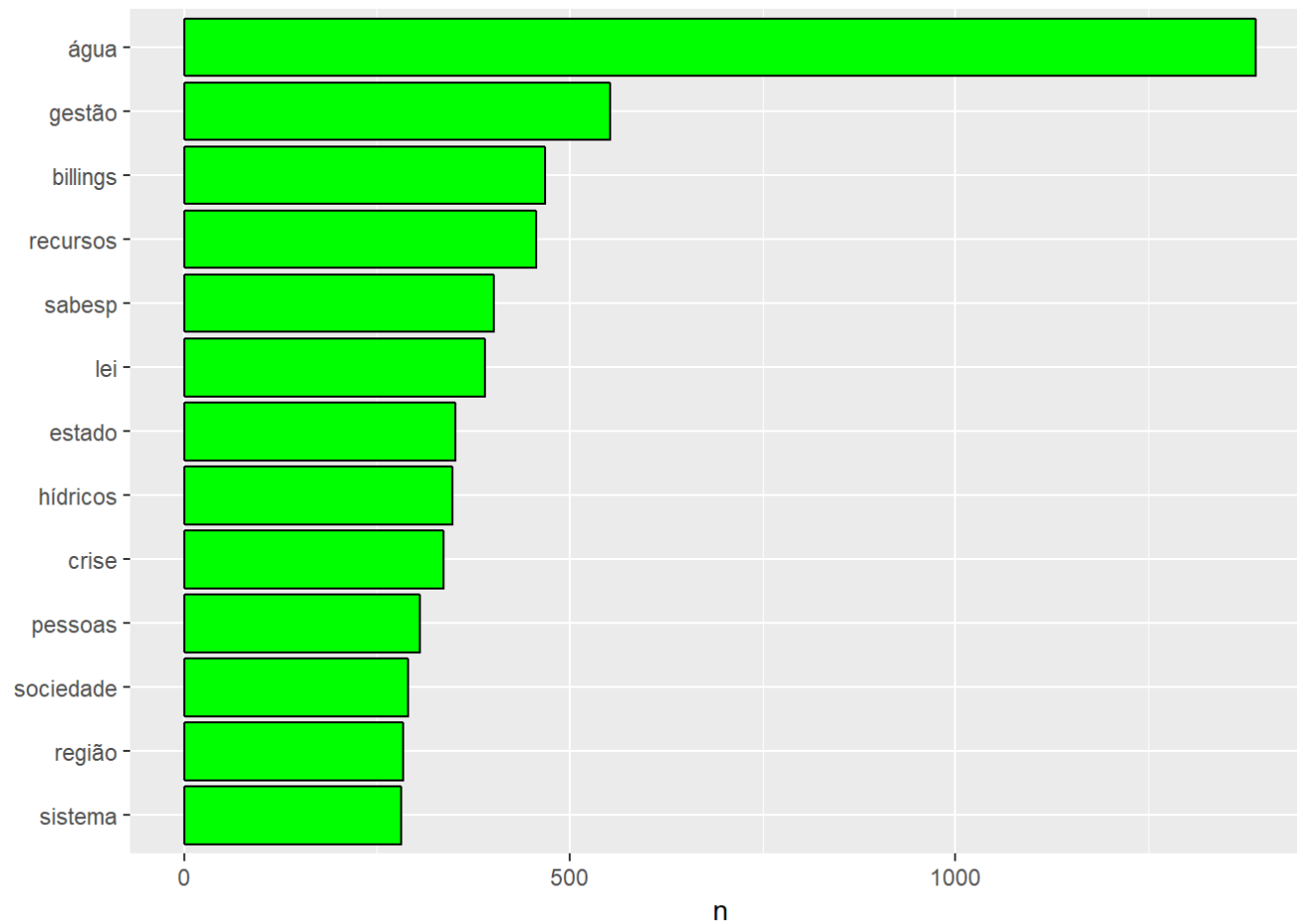
# Exemplo: Gráfico elaborado com R - Execução Orçamentária PMSP na função Gestão Ambiental



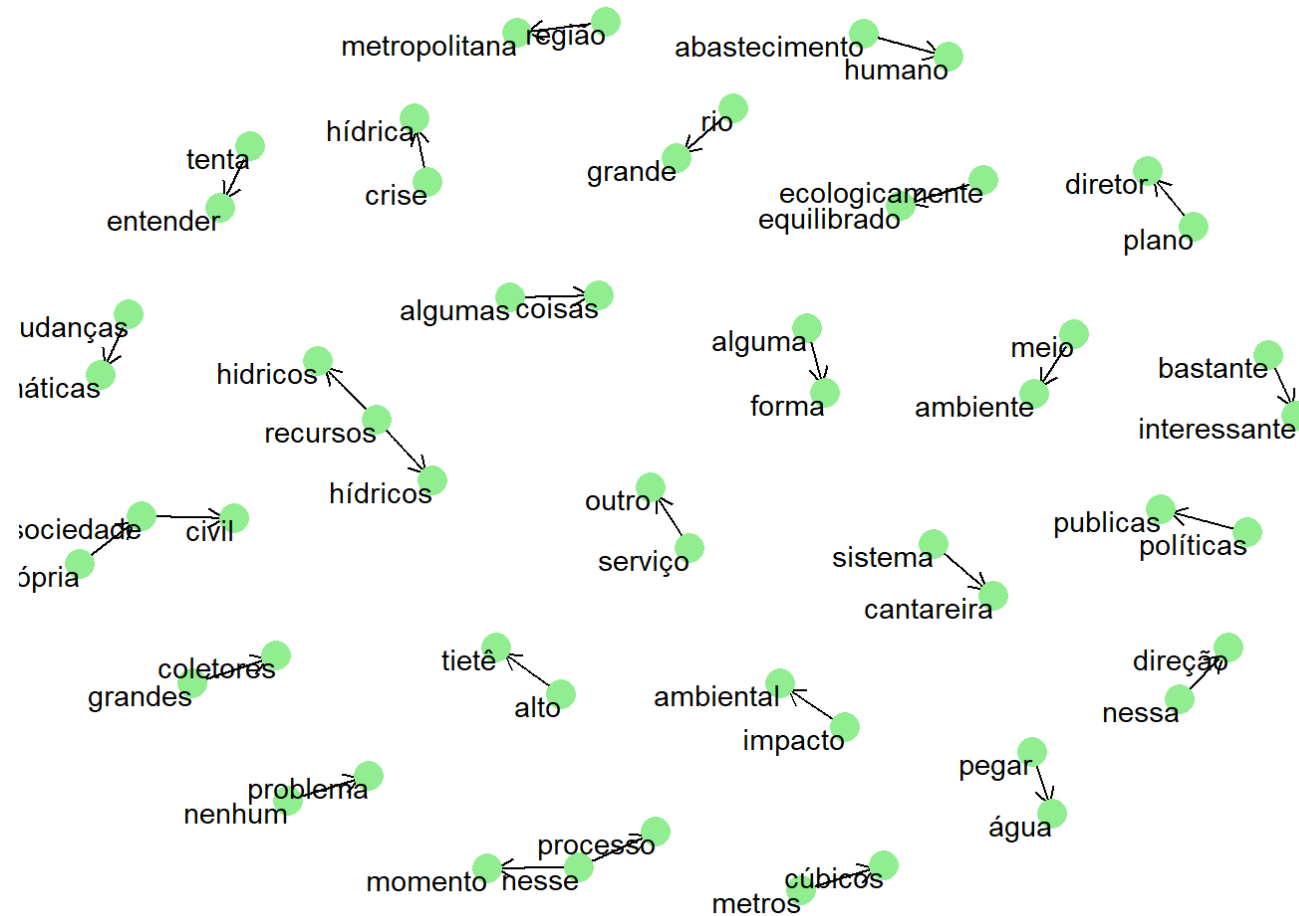
## Exemplo: Gráfico elaborado com R - Material Particulado 10 - Dados CETESB - RPollution;



## Exemplo: Frequencia de Palavras - TESE Doutorado PROCAM/USP Ana Lucia Spinola;



**Exemplo: BIGRAM - TESE Doutorado PROCAM/USP Ana Lucia Spinola;**



# **R Básico**

# Instalação R e R Studio

- 
-

# RStudio



# Projetos





# Boas práticas para organizar seu projeto



**Tratar dados como somente leitura:**

**Qualquer coisa gerada pelos seus scripts deve ser tratada como descartável:**

# Criando um projeto

**“File”**

**“New Project”**

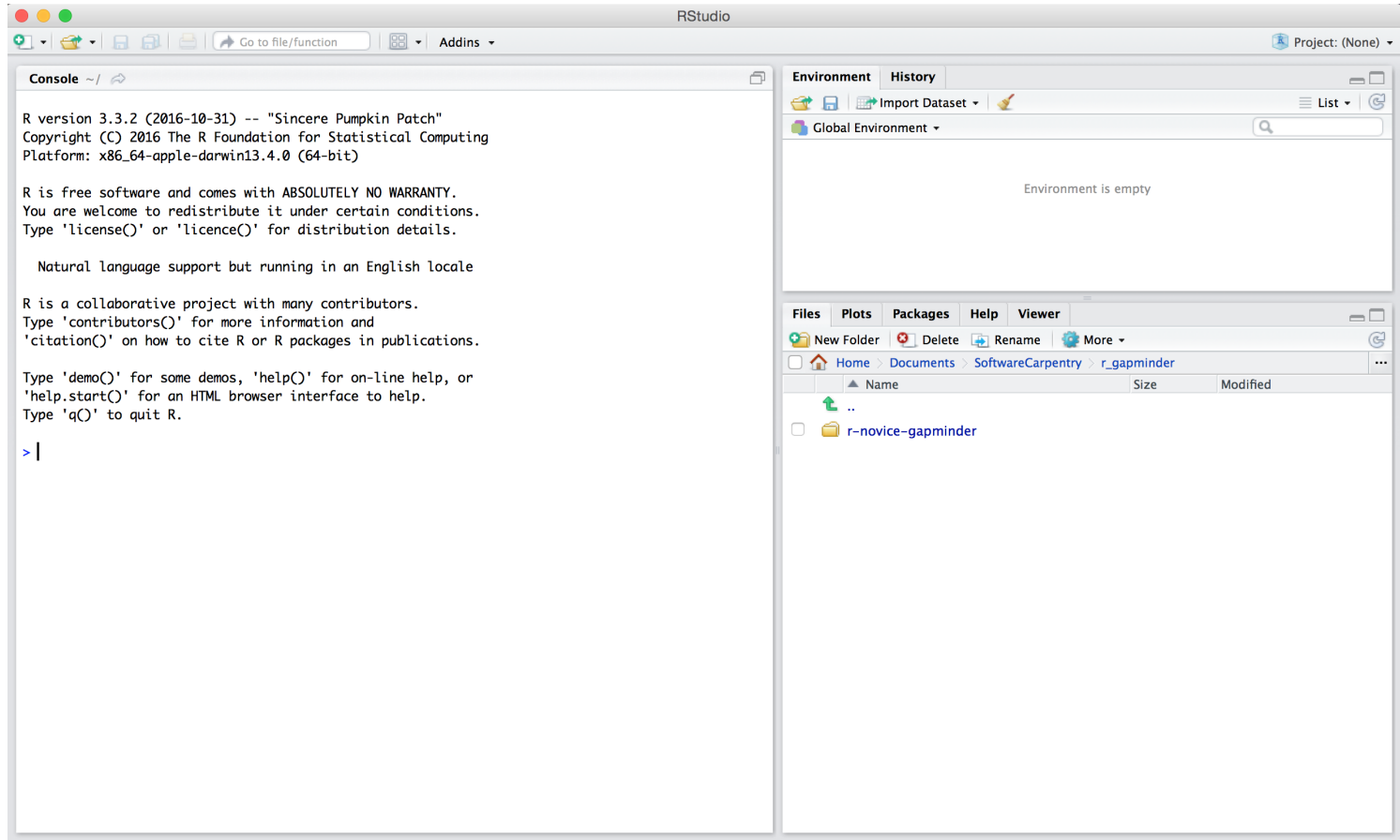
**“New Directory”**

**“New Project”**

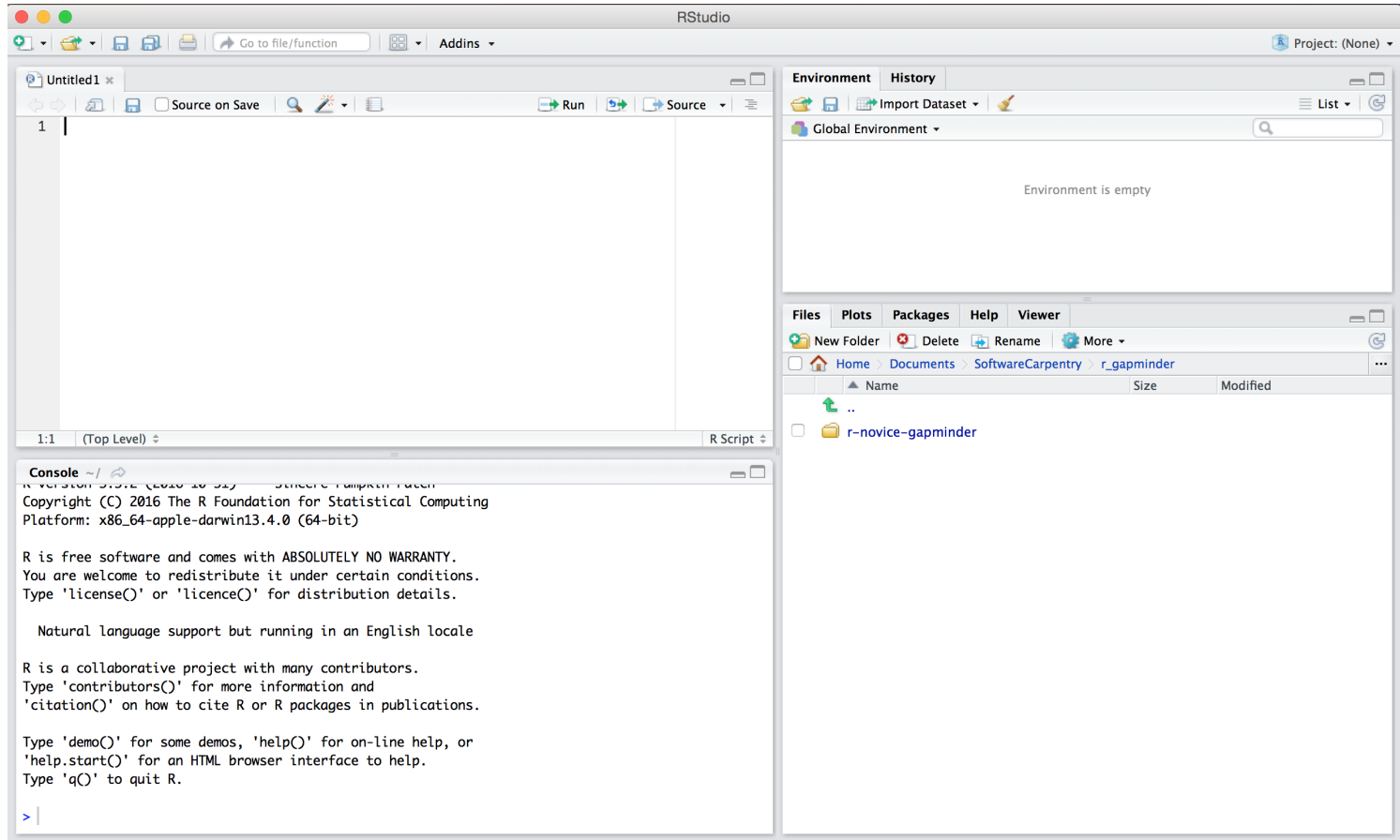
**“Create Project”**

**File -> New File -> RScript**

# RStudio



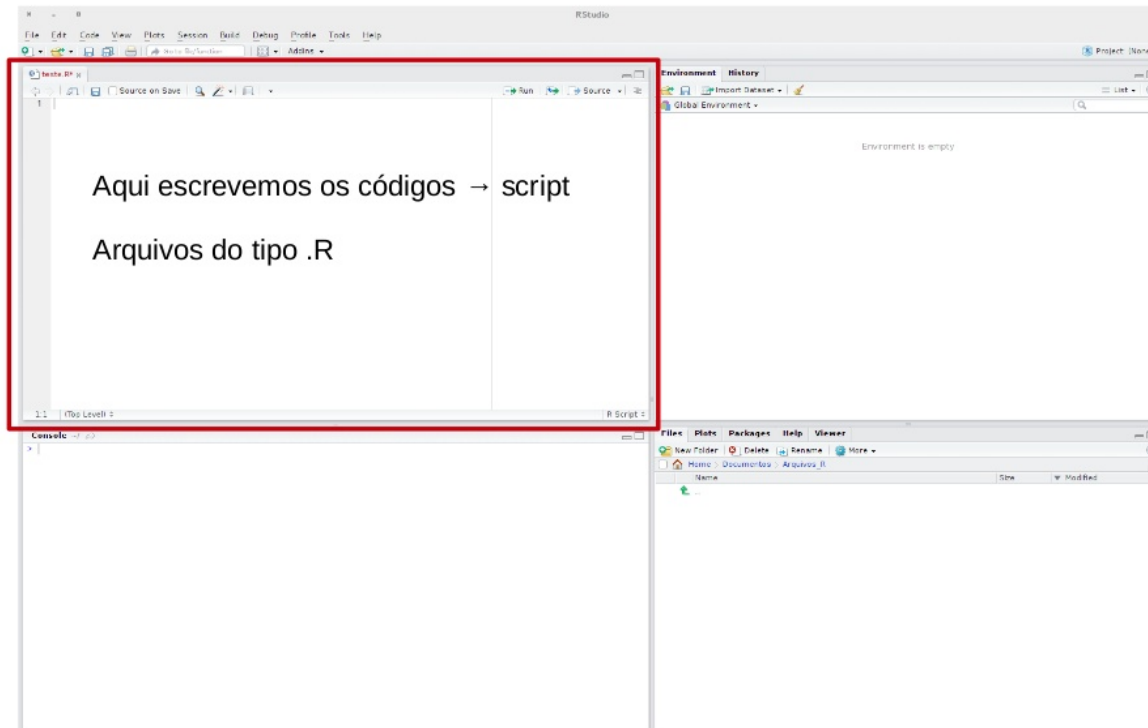
# RStudio



# RStudio

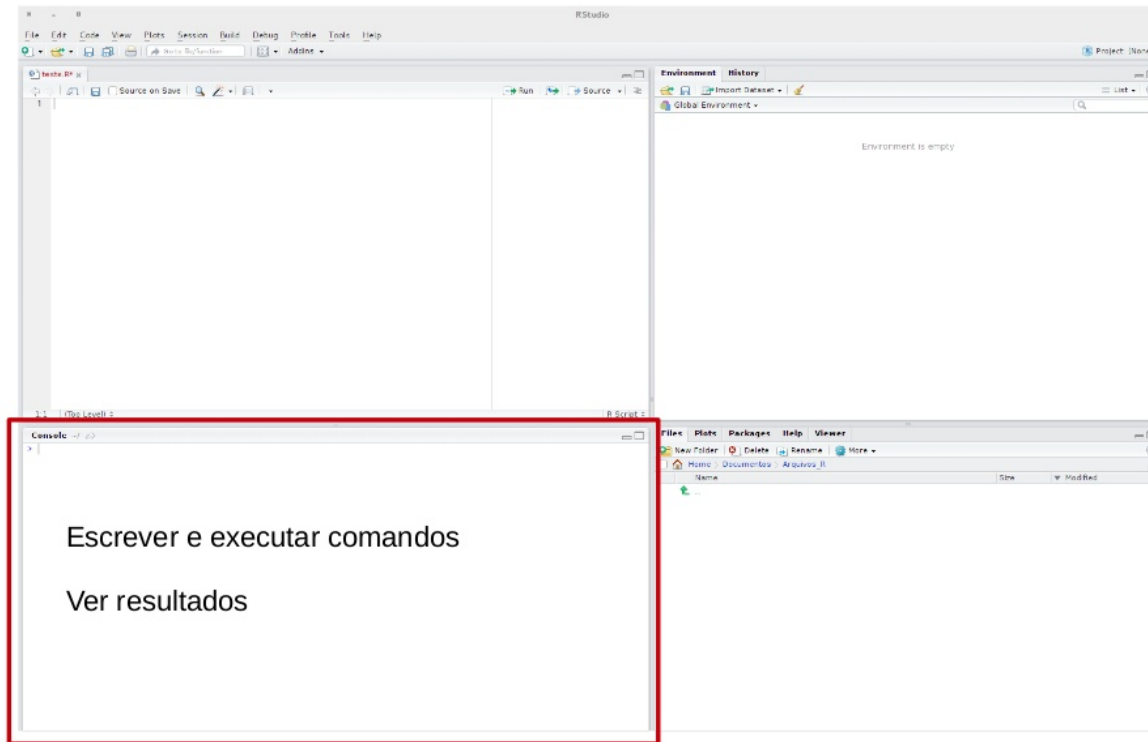
Análise Crítica de Dados – primeiros passos com R  
ambiente RStudio

**maria**  
[lab]



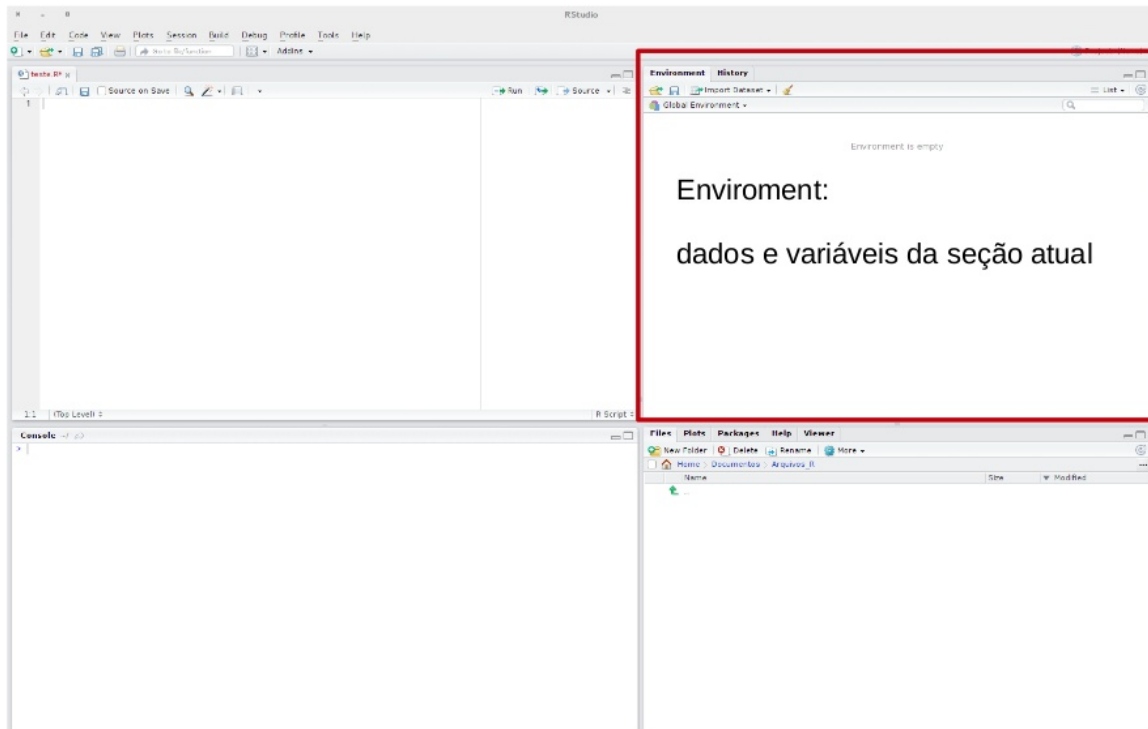
# RStudio

Análise Crítica de Dados – primeiros passos com R **maria**  
ambiente RStudio [lab]



# RStudio

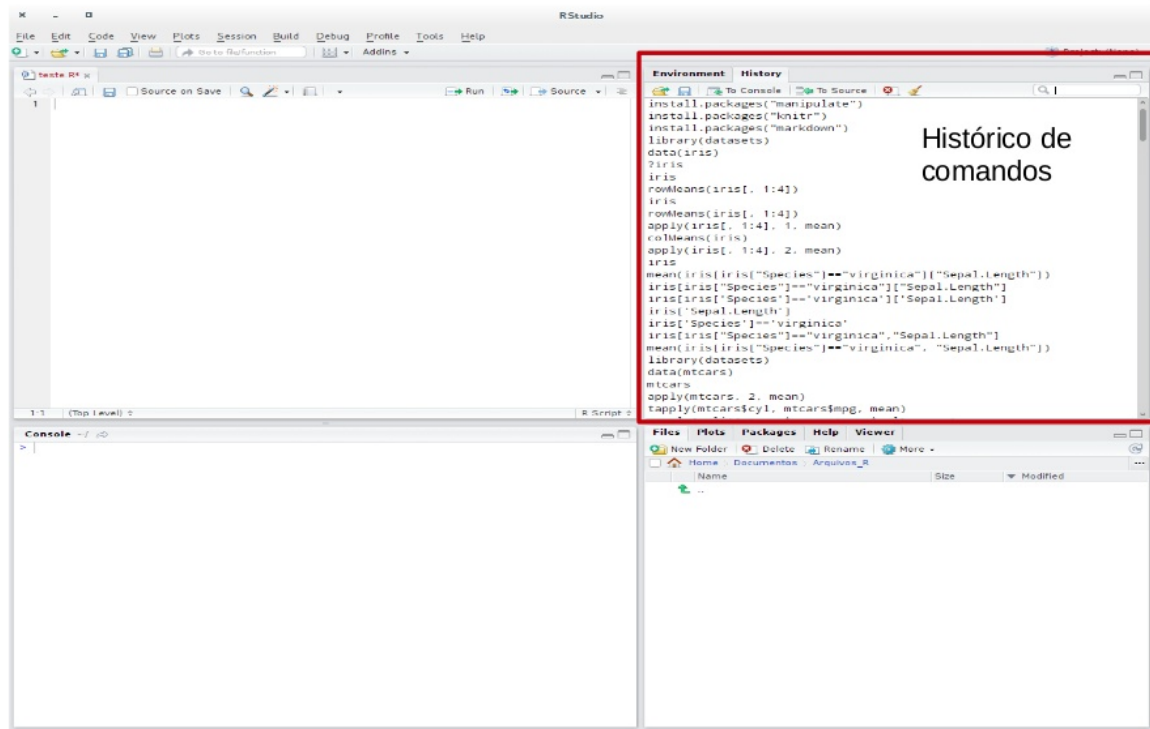
Análise Crítica de Dados – primeiros passos com R **maria**  
ambiente RStudio [lab]



# RStudio

Análise Crítica de Dados – primeiros passos com R  
ambiente RStudio

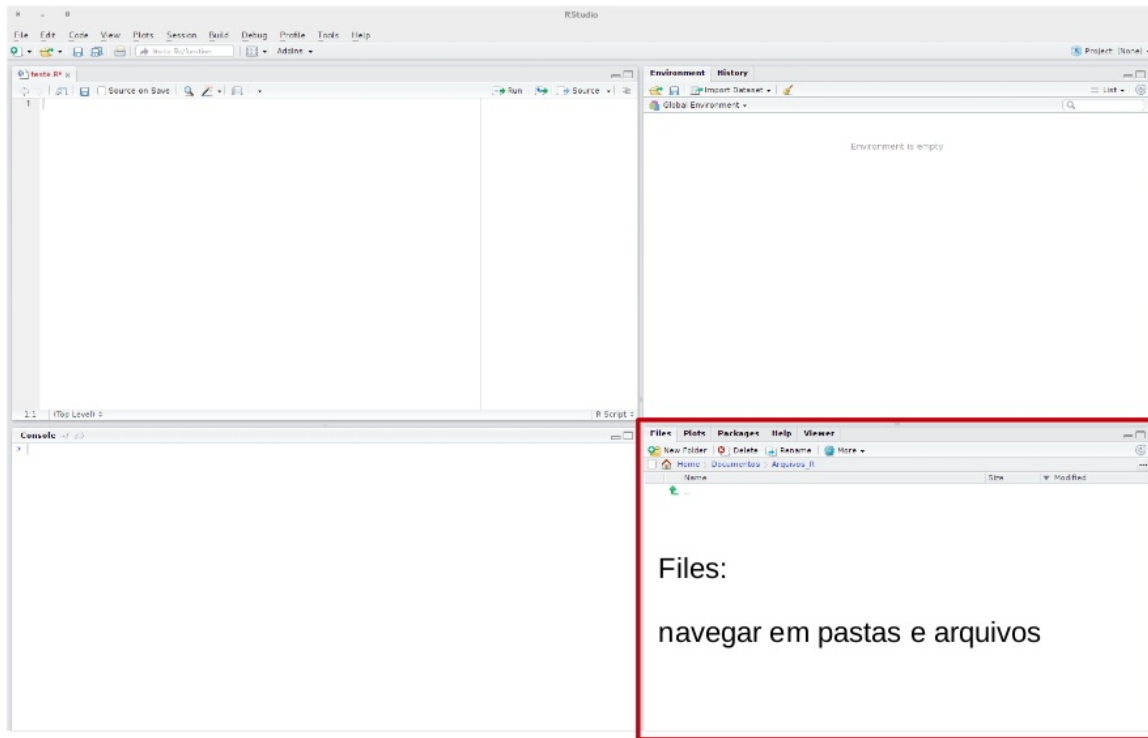
**maria**  
[lab]





# RStudio

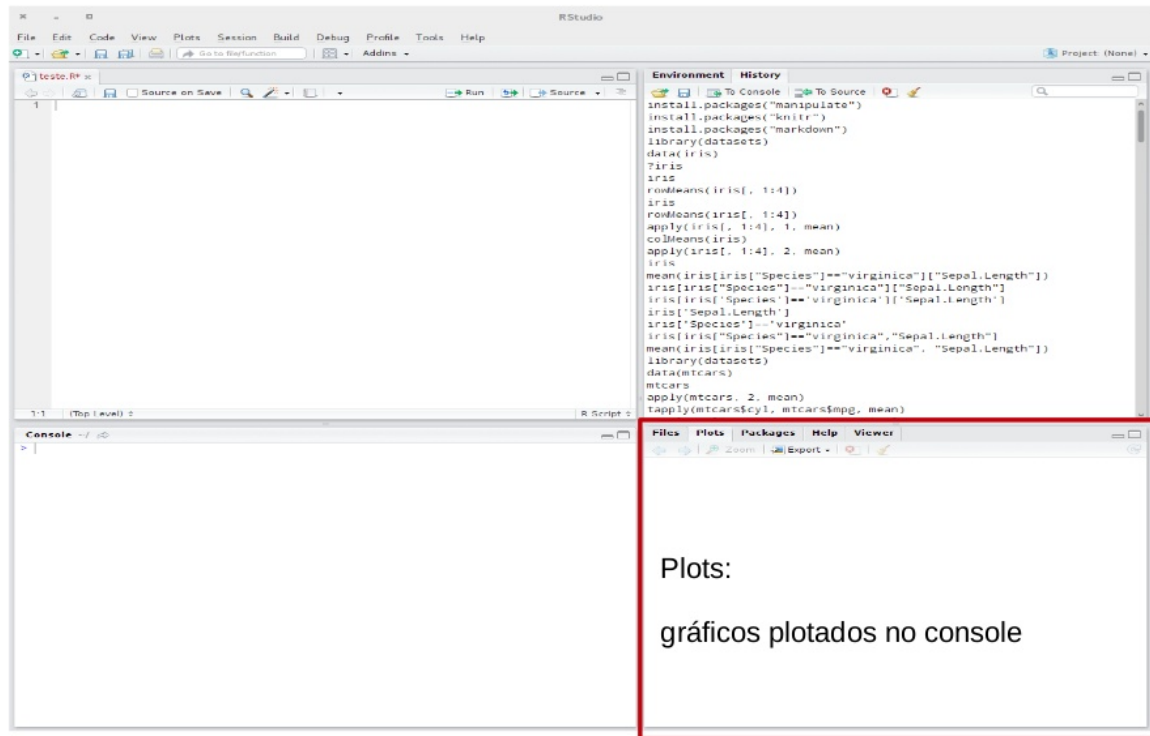
Análise Crítica de Dados – primeiros passos com R **maria**  
ambiente RStudio [lab]



# RStudio

Análise Crítica de Dados – primeiros passos com R  
ambiente RStudio

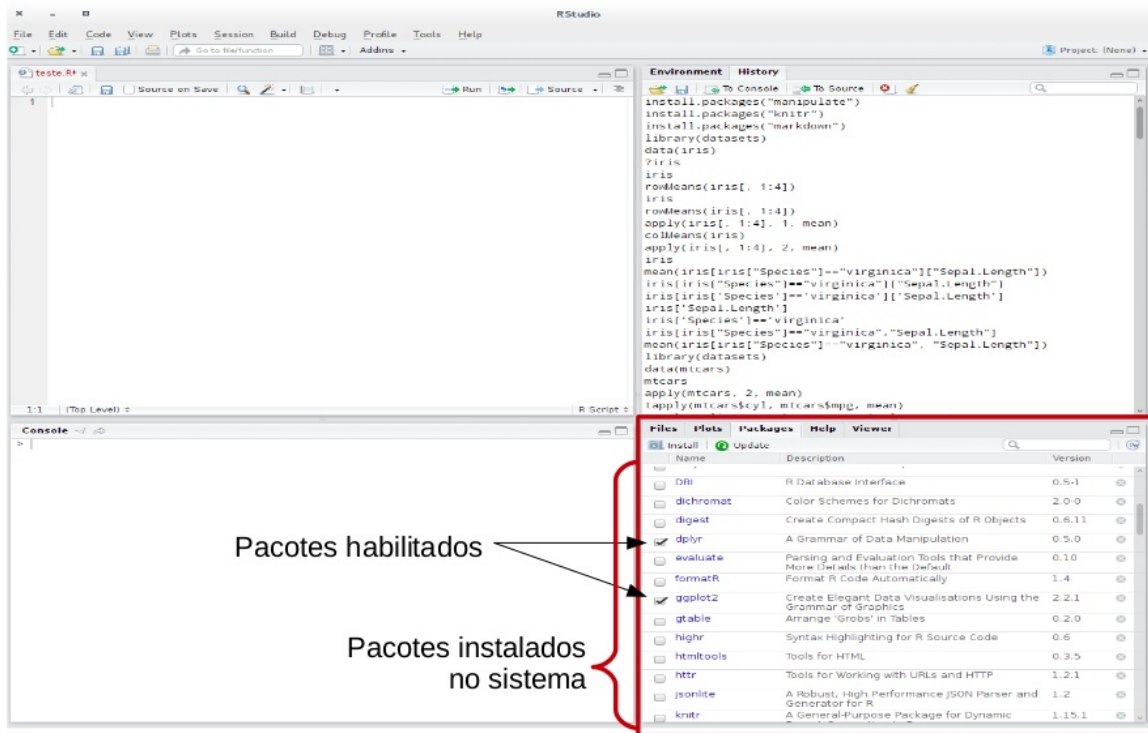
**maria**  
[lab]



# RStudio

Análise Crítica de Dados – primeiros passos com R  
ambiente RStudio

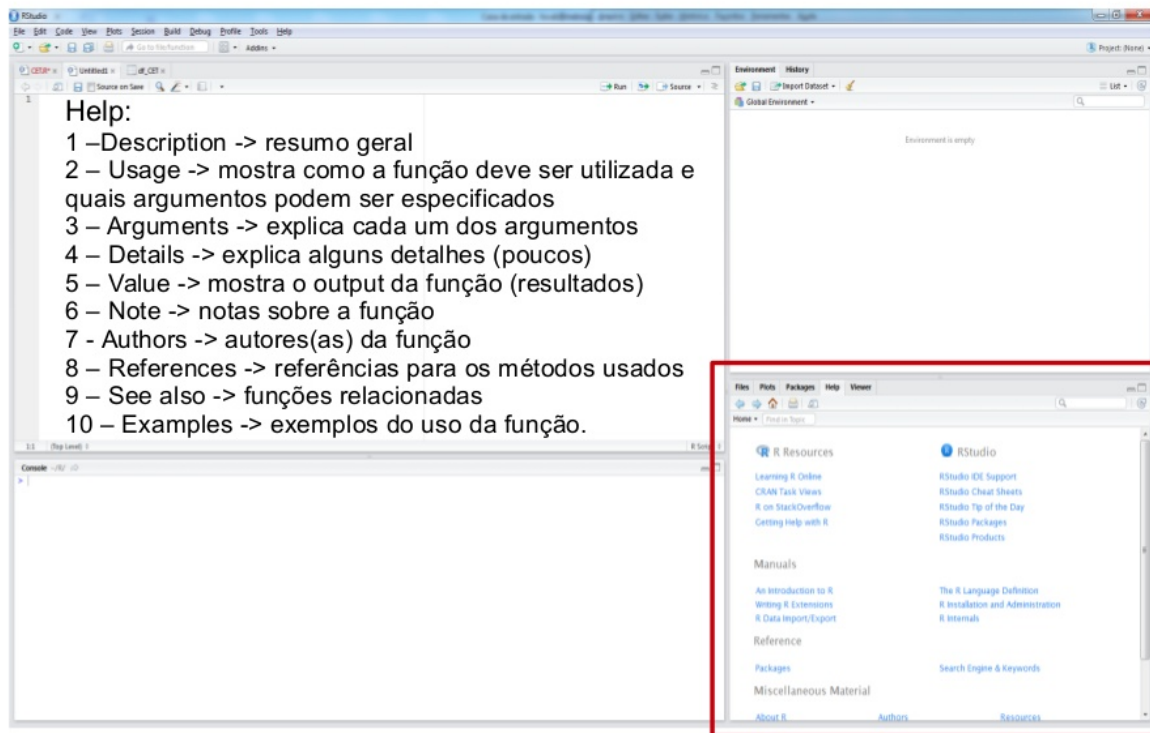
**maria**  
[lab]



# RStudio

Análise Crítica de Dados – primeiros passos com R  
ambiente RStudio

**maria**  
[lab]



# Help!

- **help** **?**

```
help(sum)  
?sum
```

- -

# Pacotes no R

## Pacotes



# Instalação de Pacotes:

- 

```
install.packages("tidyverse")
```

- 

```
devtools::install_github("tidyverse/dplyr")
```

# Carregar pacotes:

- 

```
library(tidyverse)
```

# Dicas sobre Pacotes

**não**



# Pacotes - CRAN Task View

- 
- 
-

# Atalhos importantes

- 
- -
-

# R como calculadora

```
2 + 5    # adição
```

```
## [1] 7
```

```
9 - 4    # subtração
```

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

```
5 * 2    # multiplicação
```

```
## [1] 10
```

```
7 / 5    # divisão
```

```
## [1] 1.4
```

# R como calculadora

```
9 %% 4    # resto da divisão de 9 por 4
```

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

```
7 %/% 4    # parte inteira da divisão de 4 por 3
```

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

```
8 ^ 2      # potenciação
```

```
## [1] 64
```

```
sqrt(1024) # radiciação
```

```
## [1] 32
```

# Funções matemáticas

```
sin(1) # trigonometry functions
```

```
## [1] 0.841471
```

```
log(1) # natural logarithm
```

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

```
log10(10) # base-10 logarithm
```

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

```
exp(0.5) #  $e^{(1/2)}$ 
```

```
## [1] 1.648721
```

# Nomes de objetos e variáveis

- 

- 

**usar\_snake\_case**

-

# Criando objetos no R

- `nome_objeto <- valor`
- `nome_objeto <-`
- `nome_objeto <-`

`nome_do_objeto <- valor`

# Exemplos de objetos e atribuição de valores

```
nome_empregado <- "Tom Cruise de Souza e Silva"  
nome_empregado
```

```
## [1] "Tom Cruise de Souza e Silva"
```

```
horas_trabalhadas <- 160  
horas_trabalhadas
```

```
## [1] 160
```

```
salario <- 3984.23  
salario
```

```
## [1] 3984.23
```

```
ativo <- TRUE  
ativo
```

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



# Classes Básicas ou Atômicas do R

- **Character**
- **Integer**
- **Numeric**
- **Complex**
- **Logical**
- **Factor**

# Exemplos: character

```
"escola"
```

```
## [1] "escola"
```

```
"2019"
```

```
## [1] "2019"
```

```
"I love pinschers."
```

```
## [1] "I love pinschers."
```

# Exemplos: integer

```
10L #Um número inteiro pode ser representado acompanhado de um L
```

```
## [1] 10
```

```
2019L
```

```
## [1] 2019
```

# Exemplos: numeric

```
10
```

```
## [1] 10
```

```
2019
```

```
## [1] 2019
```

```
5.44
```

```
## [1] 5.44
```

# Exemplos: complex

```
4 + 9i
```

```
## [1] 4+9i
```

# Exemplos: logical

TRUE

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

FALSE

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

# Exemplos: factor

```
escolaridade <- c("Médio", "Superior", "Fundamental", "Fundamental", "Médio")  
fator <- as.factor(escolaridade)  
fator
```

```
## [1] Médio      Superior    Fundamental Fundamental Médio  
## Levels: Fundamental Médio Superior
```

**forcats**

# Função class:

## class

```
nome_filme <- "Bohemian Rhapsody"  
class(nome_filme)
```

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

```
ano_inteiro <- 2018L  
class(ano_inteiro)
```

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

```
ano <- 2018  
class(ano)
```

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



# Função class

```
motor <- 1.5  
class(motor)
```

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

```
passou_enem <- TRUE  
class(passou_enem)
```

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

# Tipos de objetos:

- **Vector**
- **Matrix**
- **List**
- **Data.frame**

# Exemplo: Vector

```
semestre1 <- c("janeiro", "fevereiro", "março", "abril", "maio", "junho")  
notas_alunos <- c(5, 6.5, 10, 0.5, 2.75, 3, 9)
```

```
vetor1 <- 1:5  
vetor1 / 5    #Objeto vetor1 dividido por 5
```

```
## [1] 0.2 0.4 0.6 0.8 1.0
```

```
vetor2 <- 6:10  
vetor1 * vetor2
```

```
## [1] 6 14 24 36 50
```

# Exemplo: Matrix

```
primeira_matriz <- matrix(1:8, nrow = 2, ncol = 4)  
primeira_matriz
```

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

```
dim(primeira_matriz)
```

```
## [1] 2 4
```

# Exemplo: List

```
wizards <- list("Harry Potter", 18, TRUE, c("Hermione Granger", "Rony Weasley"))  
class(wizards)
```

```
## [1] "list"
```

```
harry_friends <- c("Hermione Granger", "Rony Weasley")  
class(harry_friends)
```

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

```
is.list(harry_friends)
```

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

# Exemplo: Data.frame

```
df <- starwars
```

```
head(df)
```

```
## # A tibble: 6 x 13
##   name    height    mass hair_color skin_color eye_color birth_year gender
##   <chr>    <int> <dbl> <chr>      <chr>      <chr>      <dbl> <chr>
## 1 Luke...    172     77 blond      fair        blue         19  male
## 2 C-3PO     167     75 <NA>       gold        yellow       112  <NA>
## 3 R2-D2      96     32 <NA>       white, bl... red          33  <NA>
## 4 Dart...   202    136 none       white       yellow       41.9 male
## 5 Leia...   150     49 brown      light       brown        19  female
## 6 Owen...   178    120 brown, gr... light       blue         52  male
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

## Tibble

# Dataframes - Tidy data

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174604898
China	1999	214258	127291272
China	2000	216766	128042583

variables

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174604898
China	1999	214258	127291272
China	2000	216766	128042583

observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174604898
China	1999	214258	127291272
China	2000	216766	128042583

values

# Funções úteis para trabalhar com dataframes

- 
- 
-



# Conversão de classes:

- 
- 
- 
-

# Exemplo de conversão de classes

```
vetor <- 0:9  
vetor
```

```
## [1] 0 1 2 3 4 5 6 7 8 9
```

```
class(vetor)
```

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

```
vetor_numeric <- as.numeric(vetor)  
vetor_numeric
```

```
## [1] 0 1 2 3 4 5 6 7 8 9
```

```
class(vetor_numeric)
```

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

# Exemplo de conversão de classes

```
vetor_character <- as.character(vetor)
vetor_character
```

```
## [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

```
class(vetor_character)
```

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

```
vetor_logical <- as.logical(vetor)
vetor_logical
```

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

```
class(vetor_logical)
```

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

# Operadores Lógicos

- ==
- !=
- >
- >=
- <
- <=
- !
- &
- |

**NA**

**NAs**

**NA**

# NA

```
NA > 10
```

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

```
10 == NA
```

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

```
NA + 10
```

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

```
NA / 2
```

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

# NA

```
NA == NA
```

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

**is.na()**

# Estrutura Condicional: IF

```
mes <- "Novembro"

if(mes == "Novembro") {
  print("O Natal é mês que vem!") # mostra essa mensagem.
}
```

```
## [1] "O Natal é mês que vem!"
```



# Estrutura Condicional: IF e ELSE

```
mes <- "Dezembro"

if(mes == "Novembro") {
  print("O Natal é mês que vem!") # mostra essa mensagem.
} else if(mes == "Dezembro") {
  print("O Natal chegou!") # mostra essa mensagem.
}
```

```
## [1] "O Natal chegou!"
```

# Estrutura de Repetição: FOR

```
df <- starwars  
  
lista_personagens <- head(df$name, 10)  
  
for(i in seq_along(lista_personagens)) {  
  print(lista_personagens[i])  
}
```

```
## [1] "Luke Skywalker"  
## [1] "C-3PO"  
## [1] "R2-D2"  
## [1] "Darth Vader"  
## [1] "Leia Organa"  
## [1] "Owen Lars"  
## [1] "Beru Whitesun lars"  
## [1] "R5-D4"  
## [1] "Biggs Darklighter"  
## [1] "Obi-Wan Kenobi"
```

# Tidyverse

# 0 Tidyverse



# Pacotes do Tidyverse

- **ggplot2**
- **dplyr**
- **tidyr**
- **readr**
- **purrr**
- **tibble**
- **magrittr**
- **stringr**
- **forcats**
- **lubridate**

# dplyr

dplyr

%>%

pipe

%>%

# Os 6 verbos do dplyr

- **filter**
- **arrange**
- **select**
- **mutate**
- **summarise**
- **group\_by**

# O operador %>%, o Pipe

```
forno(misture(junte(ingredientes)))
```

```
ingredientes %>% junte %>% misture %>% forno
```



# O operador `%>%`, o Pipe

**Atalho**

# Importação de arquivos

**readr**

- 
- 
- 
- 
- 

**readxl**

- 

**haven**

- 
- 
-

# E lá vamos nós!



# Instruções

- 
- 
- 
- 
-

# Importação de arquivos

```
library(tidyverse)
```

```
# Uma outra opção é carregar somente o(s) pacote(s) que irá utilizar.  
#library(dplyr)
```

```
# Importa o arquivo csv para o objeto df_titanic  
df_titanic <- read_csv("data/titanic.csv")
```

# View e glimpse

- **View**
- **glimpse()**

```
df_titanic %>% glimpse()
```

```
## Observations: 891
## Variables: 12
## $ id_passageiro    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...
## $ sobreviveu       <chr> "nao", "sim", "sim", "sim", "nao", "nao", "nao"...
## $ classe          <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2,...
## $ nome             <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John ...
## $ sexo             <chr> "masculino", "feminino", "feminino", "feminino"...
## $ idade            <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 2...
## $ irmaos_conjuge   <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0,...
## $ pais_crianças    <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0,...
## $ passagem        <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1...
## $ tarifa           <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.458...
## $ cabine           <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, N...
## $ embarque        <chr> "Southampton", "Cherbourg", "Southampton", "Sou..."
```

# filter

```
# Selecciona os sobreviventes.  
df_titanic %>% filter(sobreviveu == "sim")
```

```
## # A tibble: 342 x 12  
##   id_passageiro sobreviveu classe nome  sexo  idade irmaos_conjuge  
##           <int> <chr>      <int> <chr> <chr> <dbl>      <int>  
## 1             2  sim          1 Cumi... femi...   38          1  
## 2             3  sim          3 Heik... femi...   26          0  
## 3             4  sim          1 Futr... femi...   35          1  
## 4             9  sim          3 John... femi...   27          0  
## 5            10  sim          2 Nass... femi...   14          1  
## 6            11  sim          3 Sand... femi...    4          1  
## 7            12  sim          1 Bonn... femi...   58          0  
## 8            16  sim          2 Hewl... femi...   55          0  
## 9            18  sim          2 Will... masc...   NA          0  
## 10           20  sim          3 Mass... femi...   NA          0  
## # ... with 332 more rows, and 5 more variables: pais_crianças <int>,  
## #   passagem <chr>, tarifa <dbl>, cabine <chr>, embarque <chr>
```

## Tibble

# filter

```
# Cria um objeto e atribui a ele as linhas com os sobreviventes.  
sobreviventes <- df_titanic %>% filter(sobreviveu == "sim")
```

```
# Crianças com menos de 12 anos que sobreviveram.  
criancas_sobreviventes <- df_titanic %>% filter(sobreviveu == "sim" & idade < 12)
```

```
# Embarque realizado nos locais: Southampton ou Queenstow.  
embarque <- df_titanic %>% filter(embarque == "Southampton" |  
                                embarque == "Queenstow")
```

```
# A instrução acima pode ser reescrita com o operador %in%:  
embarque <- df_titanic %>% filter(embarque %in% c("Southampton", "Queenstow"))
```



# filter

```
# Pessoas sem informação de local de embarque.  
# is.na() - função que retorna TRUE se o valor for NA e FALSE se não for.  
sem_embarque <- df_titanic %>% filter(is.na(embarque))
```

```
# Pessoas que tem "Elizabeth" em qualquer posição do campo nome.  
# str_detect - função que retorna TRUE se detectou o valor dado e  
# FALSE, caso não tenha encontrado.  
nome <- df_titanic %>% filter(str_detect(nome, "Elizabeth"))
```

# Desafio 1

# Desafio 1 - Resposta

```
passageiras <- df_titanic %>% filter(sexo == "feminino")
```

```
criancas_Chernbourg <- df_titanic %>% filter(idade < 12 & embarque == "Chernbourg")
```

# arrange

*# Ordena por ordem crescente da coluna nome.*

```
passageiros_ordenados <- df_titanic %>% arrange(nome)
passageiros_ordenados
```

```
## # A tibble: 891 x 12
```

```
##   id_passageiro sobreviveu classe nome  sexo  idade irmaos_conjuge
##         <int> <chr>         <int> <chr> <chr> <dbl>         <int>
## 1           846 nao             3 Abbi... masc...   42             0
## 2           747 nao             3 Abbo... masc...   16             1
## 3           280 sim             3 Abbo... femi...   35             1
## 4           309 nao             2 Abel... masc...   30             1
## 5           875 sim             2 Abel... femi...   28             1
## 6           366 nao             3 Adah... masc...   30             0
## 7           402 nao             3 Adam... masc...   26             0
## 8            41 nao             3 Ahli... femi...   40             1
## 9           856 sim             3 Aks,... femi...   18             0
## 10          208 sim             3 Albi... masc...   26             0
```

```
## # ... with 881 more rows, and 5 more variables: pais_crianças <int>,
```

```
## #   passagem <chr>, tarifa <dbl>, cabine <chr>, embarque <chr>
```

# arrange

```
# Ordena por ordem decrescente de idade e por ordem crescente de nome.
passageiros_ordenados <- df_titanic %>% arrange(desc(idade), nome)
passageiros_ordenados
```

```
## # A tibble: 891 x 12
##   id_passageiro sobreviveu classe nome  sexo  idade irmaos_conjuge
##         <int> <chr>         <int> <chr> <chr> <dbl>         <int>
## 1           631 sim             1 Bark... masc...  80             0
## 2           852 nao             3 Sven... masc...  74             0
## 3           494 nao             1 Arta... masc...  71             0
## 4            97 nao             1 Gold... masc...  71             0
## 5          117 nao             3 Conn... masc... 70.5             0
## 6          746 nao             1 Cros... masc...  70             1
## 7          673 nao             2 Mitc... masc...  70             0
## 8           34 nao             2 Whea... masc...  66             0
## 9          281 nao             3 Duan... masc...  65             0
## 10         457 nao             1 Mill... masc...  65             0
## # ... with 881 more rows, and 5 more variables: pais_crianças <int>,
## #   passagem <chr>, tarifa <dbl>, cabine <chr>, embarque <chr>
```

# filter & arrange

```
# Filtra os sobreviventes homens e ordena por classe.
df_titanic %>%
  filter(sexo == "masculino" & sobreviveu == "sim") %>%
  arrange(classe)
```

```
## # A tibble: 109 x 12
##   id_passageiro sobreviveu classe nome  sexo  idade irmaos_conjuge
##           <int> <chr>      <int> <chr> <chr> <dbl>          <int>
## 1             24 sim          1 Slop... masc... 28              0
## 2             56 sim          1 Wool... masc... NA              0
## 3             98 sim          1 Gree... masc... 23              0
## 4            188 sim          1 "Rom... masc... 45              0
## 5            210 sim          1 Blan... masc... 40              0
## 6            225 sim          1 Hoyt... masc... 38              1
## 7            249 sim          1 Beck... masc... 37              1
## 8            299 sim          1 Saal... masc... NA              0
## 9            306 sim          1 Alli... masc... 0.92            1
## 10           371 sim          1 Hard... masc... 25              1
## # ... with 99 more rows, and 5 more variables: pais_crianças <int>,
## #   passagem <chr>, tarifa <dbl>, cabine <chr>, embarque <chr>
```

# Desafio 2

# Desafio 2 - Resposta

```
passageiros <- df_titanic %>% arrange(desc(classe))
```

```
mulheres <- df_titanic %>%  
  filter(sexo == "feminino") %>%  
  arrange(idade)
```



# select

```
# Selecciona as colunas indicadas.
```

```
df_titanic %>% select(nome, idade, classe, embarque)
```

```
## # A tibble: 891 x 4
```

##	nome	idade	classe	embarque
##	<chr>	<dbl>	<int>	<chr>
## 1	Braund, Mr. Owen Harris	22	3	Southampt...
## 2	Cumings, Mrs. John Bradley (Florence Briggs Th...	38	1	Cherbourg
## 3	Heikkinen, Miss. Laina	26	3	Southampt...
## 4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35	1	Southampt...
## 5	Allen, Mr. William Henry	35	3	Southampt...
## 6	Moran, Mr. James	NA	3	Queenstow
## 7	McCarthy, Mr. Timothy J	54	1	Southampt...
## 8	Palsson, Master. Gosta Leonard	2	3	Southampt...
## 9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Be...	27	3	Southampt...
## 10	Nasser, Mrs. Nicholas (Adele Achem)	14	2	Cherbourg
## #	... with 881 more rows			

# **select**

- **starts\_with**
- **ends\_with**
- **contains**

# select

```
# Selecciona as colunas que começam com "id".  
df_titanic %>% select(starts_with("id"))
```

```
## # A tibble: 891 x 2  
##   id_passageiro idade  
##           <int> <dbl>  
## 1             1    22  
## 2             2    38  
## 3             3    26  
## 4             4    35  
## 5             5    35  
## 6             6    NA  
## 7             7    54  
## 8             8     2  
## 9             9    27  
## 10            10    14  
## # ... with 881 more rows
```

# select

-

```
df_titanic %>% select(-id_passageiro, -nome)
```

```
## # A tibble: 891 x 10
##   sobreviveu classe sexo  idade irmaos_conjuge pais_crianças passagem
##   <chr>         <int> <chr> <dbl>         <int>         <int> <chr>
## 1 nao           3 masc...  22           1           0 A/5 211...
## 2 sim           1 femi...  38           1           0 PC 17599
## 3 sim           3 femi...  26           0           0 STON/02...
## 4 sim           1 femi...  35           1           0 113803
## 5 nao           3 masc...  35           0           0 373450
## 6 nao           3 masc...  NA           0           0 330877
## 7 nao           1 masc...  54           0           0 17463
## 8 nao           3 masc...   2           3           1 349909
## 9 sim           3 femi...  27           0           2 347742
## 10 sim          2 femi...  14           1           0 237736
## # ... with 881 more rows, and 3 more variables: tarifa <dbl>,
## #   cabine <chr>, embarque <chr>
```

# Desafio 3

# Desafio 3 - Resposta

```
tarifa <- df_titanic %>% select(nome, tarifa, classe)
```

```
tarifa_classe <- df_titanic %>%  
  filter(tarifa > 50) %>%  
  select(-irmaos_conjuge, -starts_with("p")) %>%  
  arrange(desc(tarifa), classe)
```

# mutate

```
# Altera a coluna tarifa para o valor da tarifa em reais.  
tarifa_conversao <- df_titanic %>% mutate(tarifa = tarifa * 4.93)
```

```
# Retorna a coluna tarifa para o valor da época.  
tarifa_conversao <- df_titanic %>% mutate(tarifa = tarifa / 4.93)
```

# mutate

```
# Cria no dataset uma nova variável chamada tarifa_reais.
```

```
tarifa_conversao <- df_titanic %>% mutate(tarifa_real = tarifa * 4.93)  
tarifa_conversao
```

```
## # A tibble: 891 x 13
```

```
##   id_passageiro sobreviveu classe nome  sexo  idade irmaos_conjuge  
##           <int> <chr>         <int> <chr> <chr> <dbl>         <int>  
## 1             1 nao             3 Brau... masc... 22             1  
## 2             2 sim             1 Cumi... femi... 38             1  
## 3             3 sim             3 Heik... femi... 26             0  
## 4             4 sim             1 Futr... femi... 35             1  
## 5             5 nao             3 Alle... masc... 35             0  
## 6             6 nao             3 Mora... masc... NA             0  
## 7             7 nao             1 McCa... masc... 54             0  
## 8             8 nao             3 Pals... masc... 2             3  
## 9             9 sim             3 John... femi... 27             0  
## 10           10 sim             2 Nass... femi... 14             1
```

```
## # ... with 881 more rows, and 6 more variables: pais_crianças <int>,  
## #   passagem <chr>, tarifa <dbl>, cabine <chr>, embarque <chr>,  
## #   tarifa_real <dbl>
```



# Desafio 4

# Desafio 4 - Resposta

```
tarifa_conversao <- df_titanic %>%  
  mutate(tarifa_dolar = tarifa * 1.31) %>%  
  arrange(desc(tarifa))
```

# summarize

```
# Calcula a média da variável idade  
# na.rm = TRUE remove os NAs  
df_titanic %>% summarize(mean(idade, na.rm=TRUE))
```

```
## # A tibble: 1 x 1  
##   `mean(idade, na.rm = TRUE)`  
##                               <dbl>  
## 1                               29.7
```

# summarize

```
# Calcula: número de mulheres, mediana geral da tarifa e número de passageiros.  
# No caso abaixo a função sum() retorna o número de mulheres.  
# A função n() mostra o número de linhas (em cada grupo) e  
# costuma ser bastante usada com o summarize.
```

```
df_titanic %>%  
  summarize(  
    mulheres = sum(sexo == "feminino", na.rm = TRUE),  
    mediana_tarifa = median(tarifa, na.rm = TRUE),  
    num_passageiros = n()  
  )
```

```
## # A tibble: 1 x 3  
##   mulheres mediana_tarifa num_passageiros  
##   <int>         <dbl>         <int>  
## 1      314          14.5           891
```

# summarize

```
# Filtra os passageiros homens e calcula a mediana da tarifa.
```

```
df_titanic %>%  
  filter(sexo == "masculino") %>%  
  summarize(  
    mediana_tarifa = median(tarifa, na.rm = TRUE)  
  )
```

```
## # A tibble: 1 x 1  
##   mediana_tarifa  
##           <dbl>  
## 1           10.5
```

# Desafio 5

# Desafio 5 - Resposta

```
media_tarifa <- df_titanic %>% summarize(mean(tarifa, na.rm=TRUE))
```

```
mulheres_tarifa <- df_titanic %>%  
  filter(sexo == "feminino") %>%  
  summarize(  
    mediana_tarifa = median(tarifa, na.rm = TRUE)  
  )
```

# group\_by + summarize

```
# Agrupa pela variável sobreviveu e calcula  
# o número de passageiros por grupo (sim/nao).  
df_titanic %>%  
  group_by(sobreviveu) %>%  
  summarize(num_passageiros = n())
```

```
## # A tibble: 2 x 2  
##   sobreviveu num_passageiros  
##   <chr>         <int>  
## 1 nao             549  
## 2 sim             342
```



# group\_by + summarize

```
# Agrupa pelo local de embarque e calcula a mediana da tarifa de cada grupo.  
df_titanic %>%  
  group_by(embarque) %>%  
  summarize(mediana_tarifa = median(tarifa, na.rm = TRUE))
```

```
## # A tibble: 4 x 2  
##   embarque   mediana_tarifa  
##   <chr>         <dbl>  
## 1 Cherbourg     29.7  
## 2 Queenstown    7.75  
## 3 Southampton    13  
## 4 <NA>          80
```

# Desafios 6

# Desafio 6 - Resposta

```
df_titanic %>%  
  group_by(classe) %>%  
  summarize(qtd_classe = n())
```

```
## # A tibble: 3 x 2  
##   classe qtd_classe  
##   <int>     <int>  
## 1     1         216  
## 2     2         184  
## 3     3         491
```

# Desafio 6 - Resposta

```
df_titanic %>%  
  group_by(sexo) %>%  
  summarize(mediana_tarifa = median(tarifa, na.rm = TRUE))
```

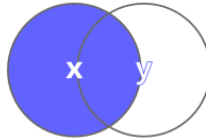
```
## # A tibble: 2 x 2  
##   sexo      mediana_tarifa  
##   <chr>          <dbl>  
## 1 feminino          23  
## 2 masculino        10.5
```

# Join

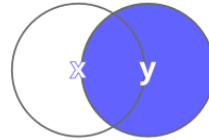
dplyr

## dplyr *joins*

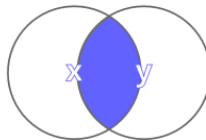
left\_join(x, y)



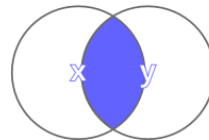
right\_join(x, y)



inner\_join(x, y)

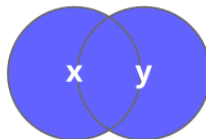


semi\_join(x, y)

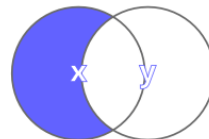


(never duplicate rows of x)

full\_join(x, y)



anti\_join(x, y)



# ggplot2

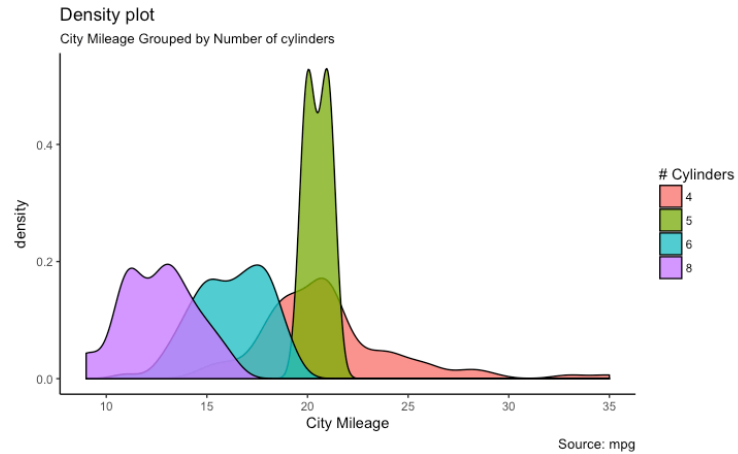
- 

- 

- 

-

# ggplot2 - o pacote dos gráficos



# RMarkdown

RMarkdown

chunk



# Para aprender mais

- 
- 
- 
-

# Referências

- 
- 
- 
- 
- 
- 
- 
-

If you want to go fast,  
*go alone.*

If you want to go far,  
*go together.*

- African Proverb -

**Muito obrigada!**