Oficina de Introdução ao Software R

UFABC - Campus São Bernardo do Campo

http://bit.ly/rUFABC

Beatriz Milz
29/11/2018

Sobre a oficina de introdução ao R

- Oficina introdutória
- Oferecida pelo LAPLAN e MACROAMB.
- 2a Edição (a primeira aconteceu em outubro no IEE)
- A oficina tem como objetivo introduzir os pesquisadores sobre as possibilidades de utilização do software R como ferramenta para manipulação e análise de dados em pesquisas e projetos científicos.
- Cooperar para a difusão de **ferramentas gratuitas** e colaborativas na elaboração de pesquisas e produção de conhecimento científico.

Instrutora

- **Beatriz Milz**
  - Bacharel em Gestão Ambiental (EACH/USP).
  - Mestre em Ciências no PPG-Análise Ambiental Integrada (UNIFESP/Diadema).
  - Atualmente no processo seletivo de Doutorado em Ciência Ambiental no PROCAM/IEE/USP.
  - Co-organizadora do Meetup R-Ladies São Paulo.
  - Email: beatriz.milz@hotmail.com
  - Github: @beatrizmilz (repositório de código)

Organizadora e Monitora

- **Rosana Laura da Silva**
  - Mestranda em Planejamento e Gestão do Território - UFABC
  - Bacharel em Ciência e Tecnologia - UFABC
  - Bacharel em Engenharia Ambiental e Urbana - UFABC
  - LAPLAN - Laboratório de Planejamento e Gestão do Território
  - MACROAMB - Projeto Temático FAPESP: Governança Ambiental da Macrometrópole Paulista faca à variabilidade climática.
  - Email: laura.ufabc@gmail.com

Organizadora e Monitora

- **Bruna de Souza Fernandes**
  - Cursando Bacharel em Ciência e Humanidades - UFABC
  - Cursando Bacharel em Planejamento Territorial - UFABC
  - LAPLAN - Laboratório de Planejamento e Gestão do Território
  - MACROAMB - Projeto Temático FAPESP: Governança Ambiental da Macrometrópole Paulista faca à variabilidade climática.
  - Email: bsfernandes17@gmail.com

Monitora

- **Alissa Munerato**
  - Co-organizadora do Meetup R-Ladies São Paulo.
  - Cursando Bacharelado em Ciência e Tecnologia - UFABC
  - Cursando Bacharelado em Neurociência - UFABC
  - Email: alissamunerato@gmail.com

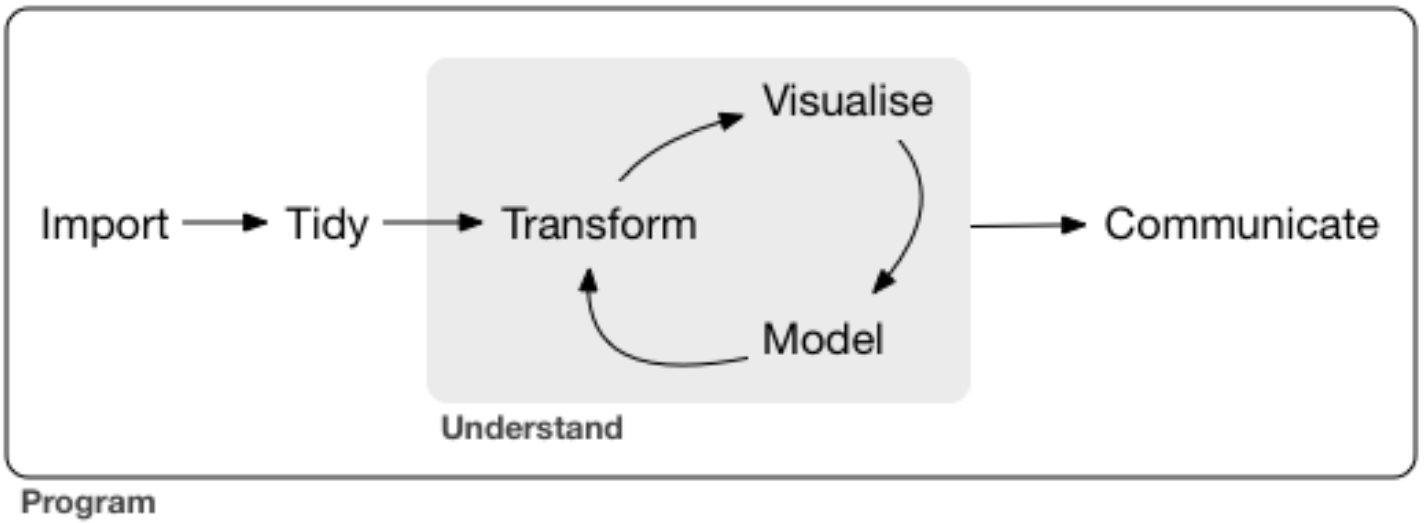
Introdução

O que é o R?

> R is a free software environment for statistical computing and graphics. (https://www.r-project.org/)
>
> R é um ambiente de software livre para computação estatística e gráficos.

- O R é open-source;
- Muito usado por cientistas de dados, estatísticos e pesquisadores.

Ciclo da ciência de dados



Fonte: Livro R for Data Science

O R é uma linguagem de programação.

Qual é a vantagem?

- É um texto
- É reprodutível
- Dá para compartilhar!

Reproducible Research / Ciência Reprodutível

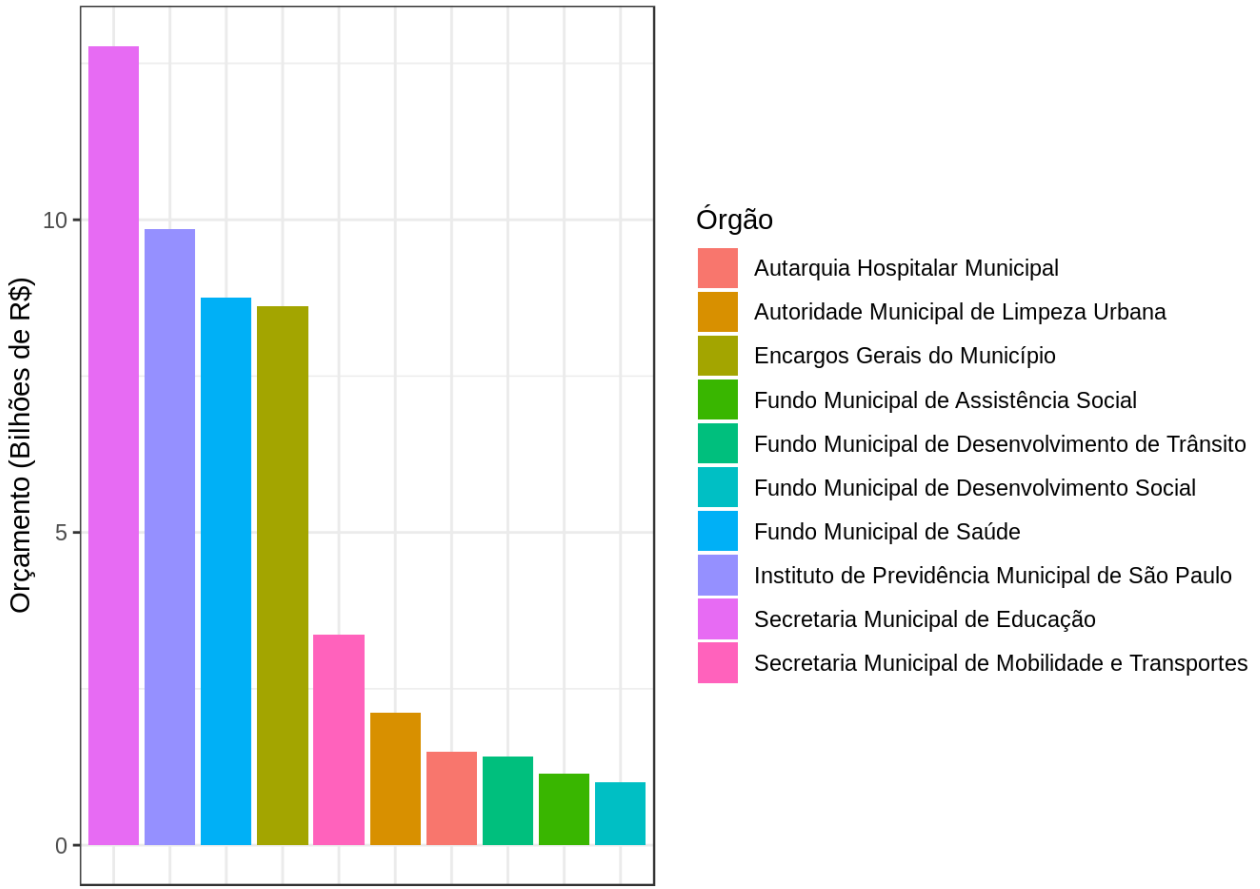
> *"The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified."*
>
> *"O objetivo da pesquisa reprodutível é vincular instruções específicas à análise de dados e dados experimentais para que os estudos possam ser recriados, melhor compreendidos e verificados."*

Fonte: CRAN Task View: Reproducible Research

O que podemos fazer com o R?

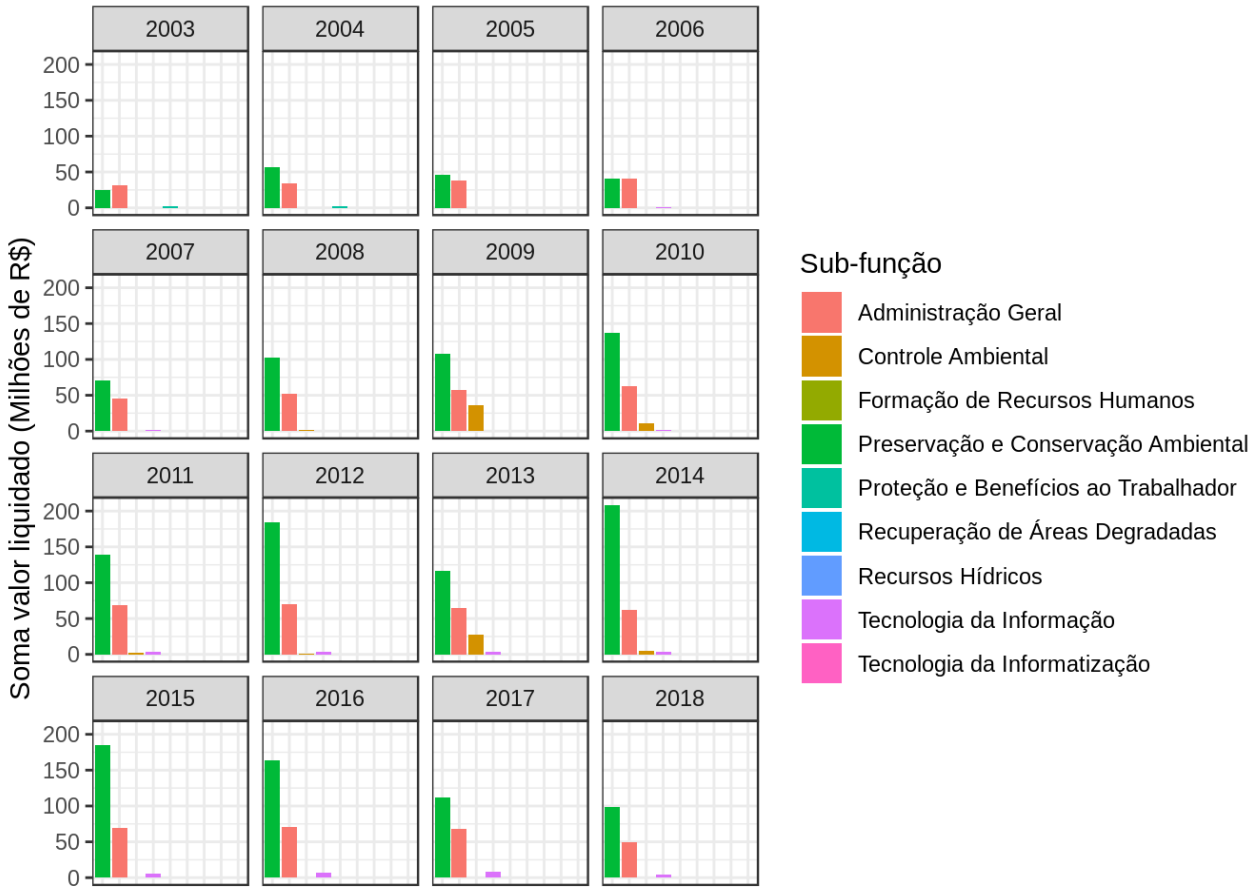
- Análise de dados - Estatística, modelagem, etc.
- Visualização de dados
- Apresentações - Pacote Xaringan - Material: Comunicando seus resultados e criando apresentações com R
- Relatórios dinâmicos
- Escrever livros - Pacote Bookdown
- Mineração de dados
- Muito mais ...

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



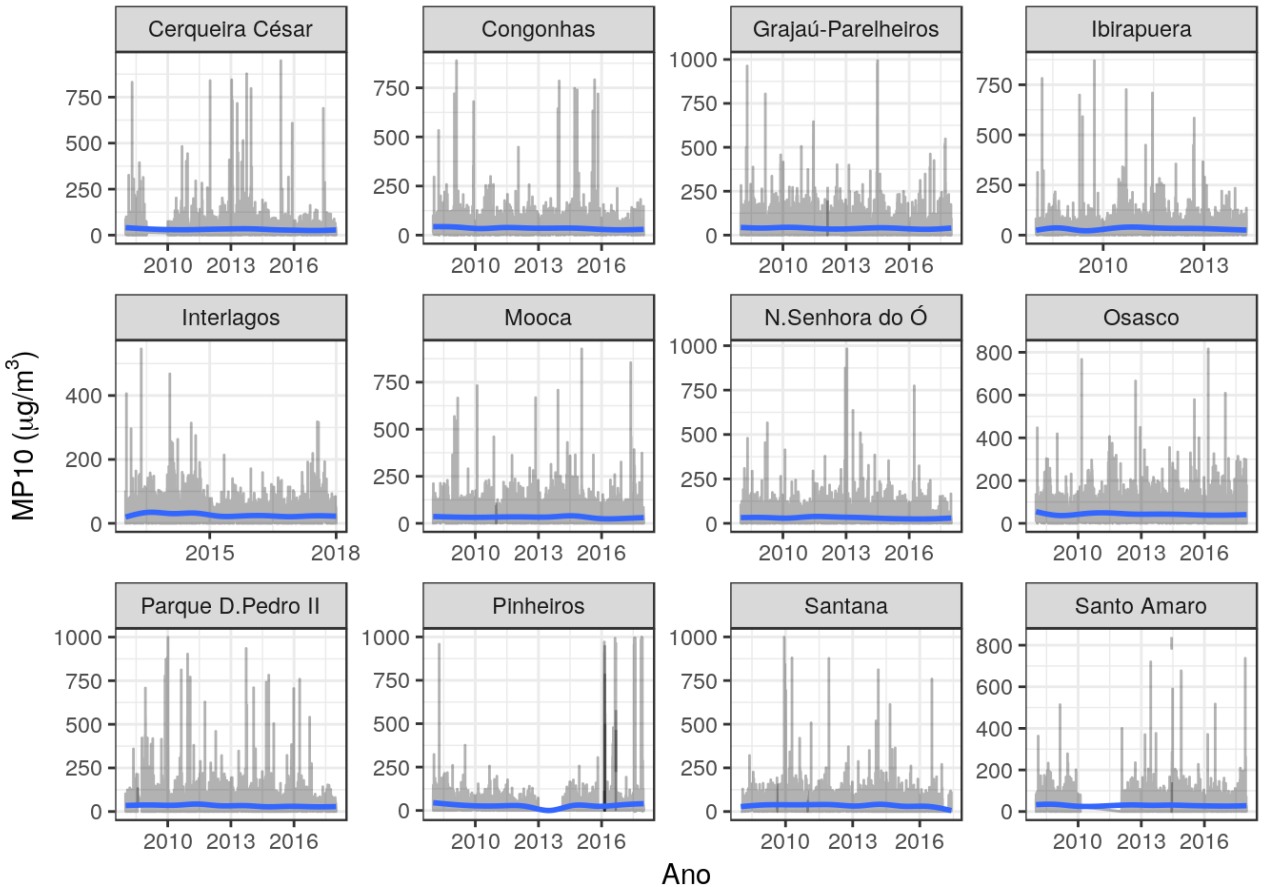
Fonte: Explorando o orçamento da Prefeitura Municipal de São Paulo

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



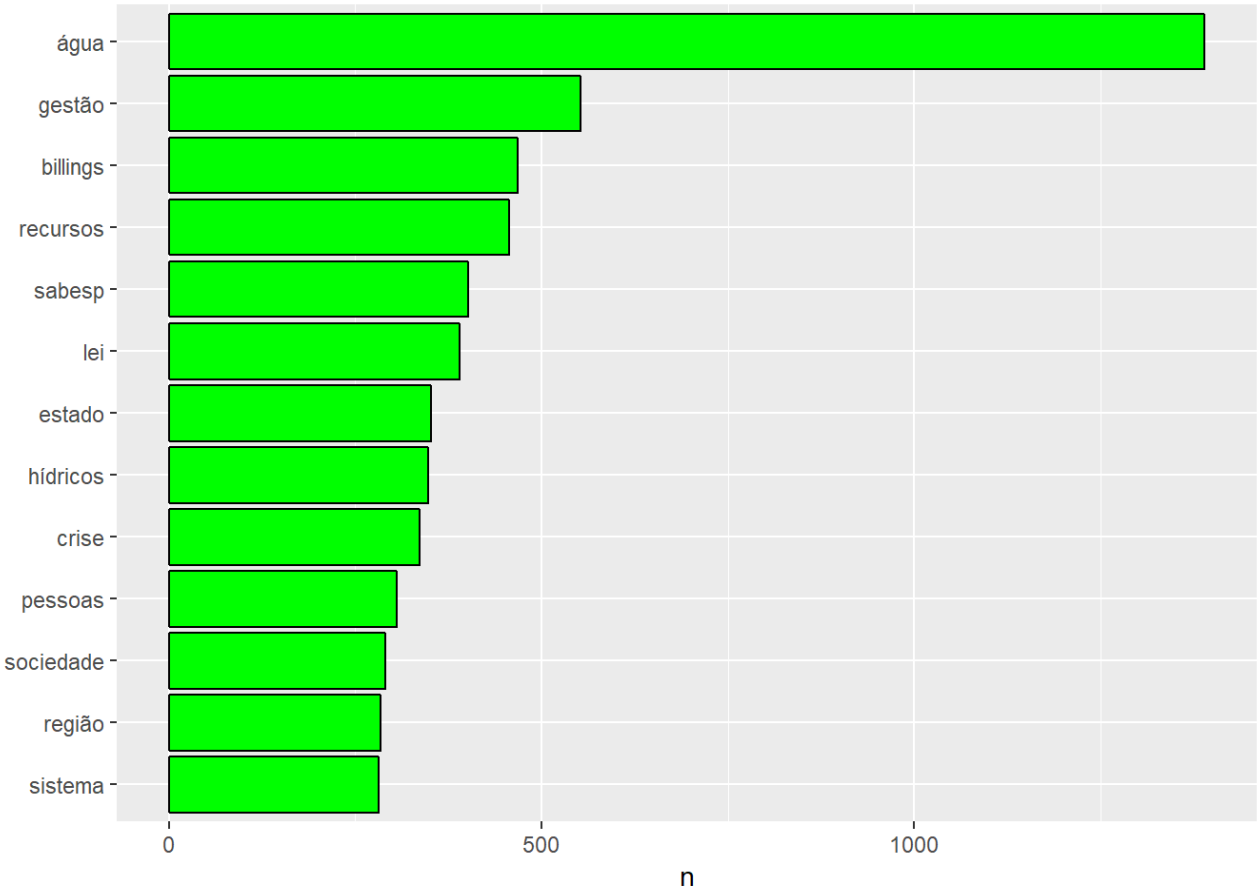
Fonte: Explorando o orçamento da Prefeitura Municipal de São Paulo

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



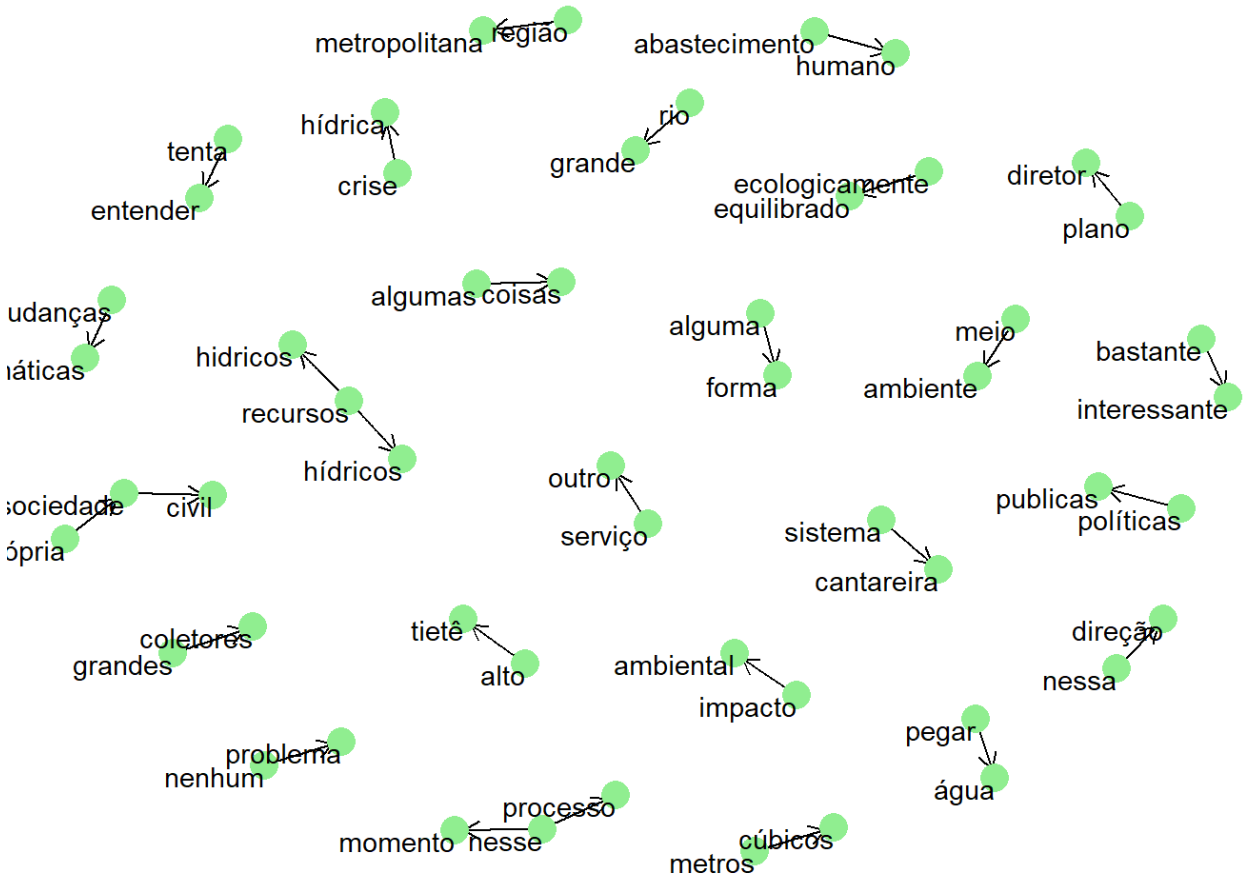
Fonte: Rpollution

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



Fonte: Ana Lu Spinola

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



Fonte: Ana Lu Spinola

Exemplo: Nuvem de palavras

Exemplo de dashboard

- Polling Data - http://pollingdata.com.br/ - pesquisas eleitorais
- Dashboard em Shiny - R

RStudio

RStudio é o IDE da Linguagem R, ou seja, o ambiente que utilizamos para editar e executar os códigos em R.

- Facilita a utilização do R.

Instalação R e R Studio

- Instalação do R
- Instalação do R Studio

Fonte: Maria Marinho

Projetos

> *"A good project layout will ultimately make your life easier: It will help ensure the integrity of your data; It makes it simpler to share your code with someone else (a lab-mate, collaborator, or supervisor); It allows you to easily upload your code with your manuscript submission; It makes it easier to pick the project back up after a break."*
>
> *"Um bom layout de projeto facilitará sua vida: ajudará a garantir a integridade de seus dados; facilita o compartilhamento de seu código com outra pessoa (colega de laboratório, colaborador ou orientador); ele permite que você facilmente faça o upload do seu código com a submissão do seu manuscrito; torna-se mais fácil recuperar o projeto depois de um intervalo. "*

Fonte: SW Carpentry

Boas práticas para organizar seu projeto

*1. **Tratar dados como somente leitura:** esse é provavelmente o objetivo mais importante da configuração de um projeto. Os dados geralmente consomem tempo e/ou são caros para coletar. Trabalhar com eles interativamente (por exemplo, no Excel), onde eles podem ser modificados, significa que você nunca tem certeza de onde os dados vieram, ou como eles foram modificados desde a coleta. Portanto, é uma boa ideia tratar seus dados como "somente leitura".*

*2. **Qualquer coisa gerada pelos seus scripts deve ser tratada como descartável:** todos devem poder ser criados novamente a partir dos seus scripts. Existem várias maneiras diferentes de gerenciar essa saída. Acho útil ter uma pasta de saída com subdiretórios diferentes para cada análise separada. Isso fica mais fácil depois, já que muitas das análises são exploratórias e não acabam sendo usadas no projeto final, e algumas análises são compartilhadas entre os projetos.*

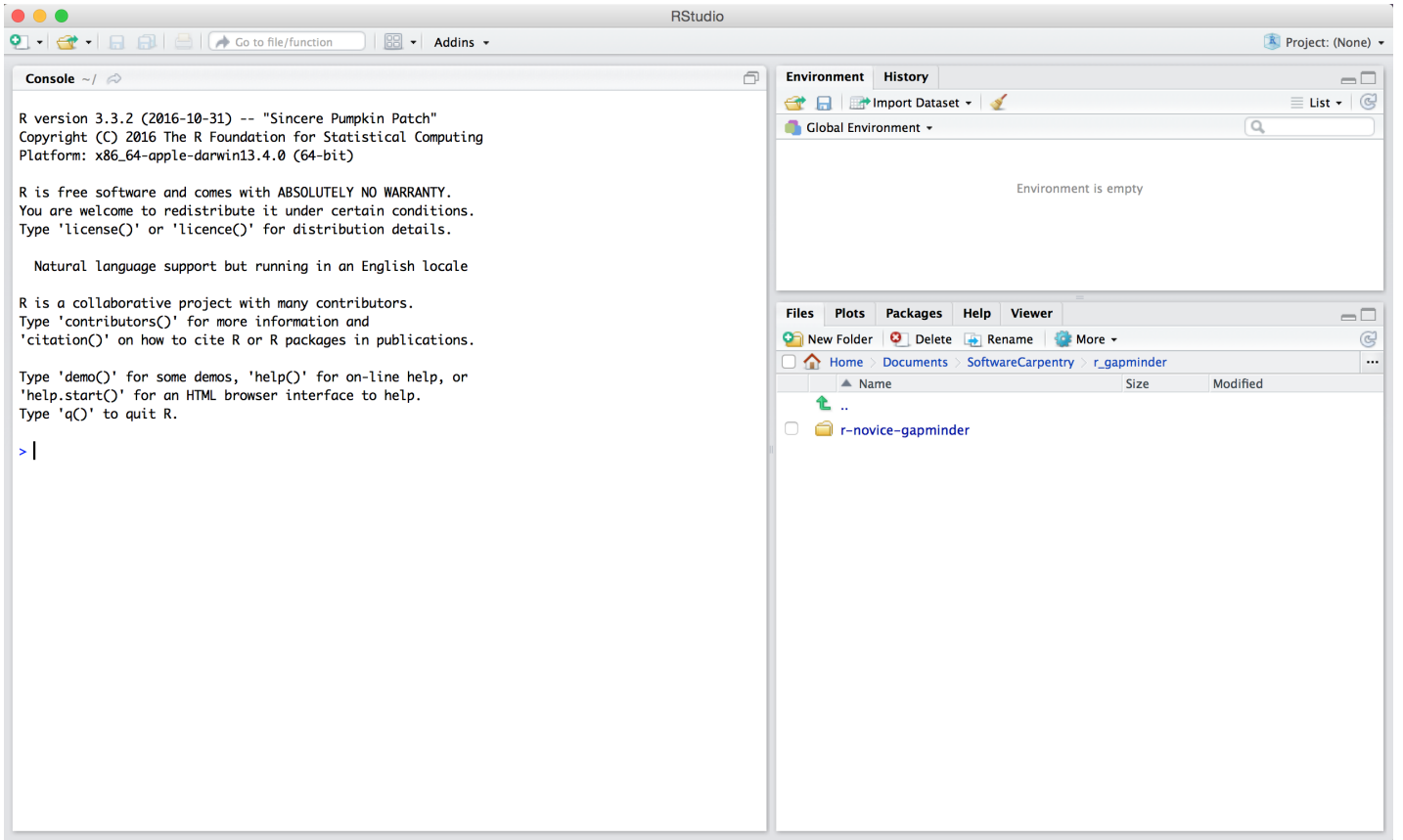
Fonte: SW Carpentry

Criando um projeto

1. Clique na opção **“File”** do menu, e então em **“New Project”**.
2. Clique em **“New Directory”**.
3. Clique em **“New Project”**.
4. Escreva o nome do diretório (pasta) onde deseja manter seu projeto, ex “my_project”.
5. Clique no botão **“Create Project”**.

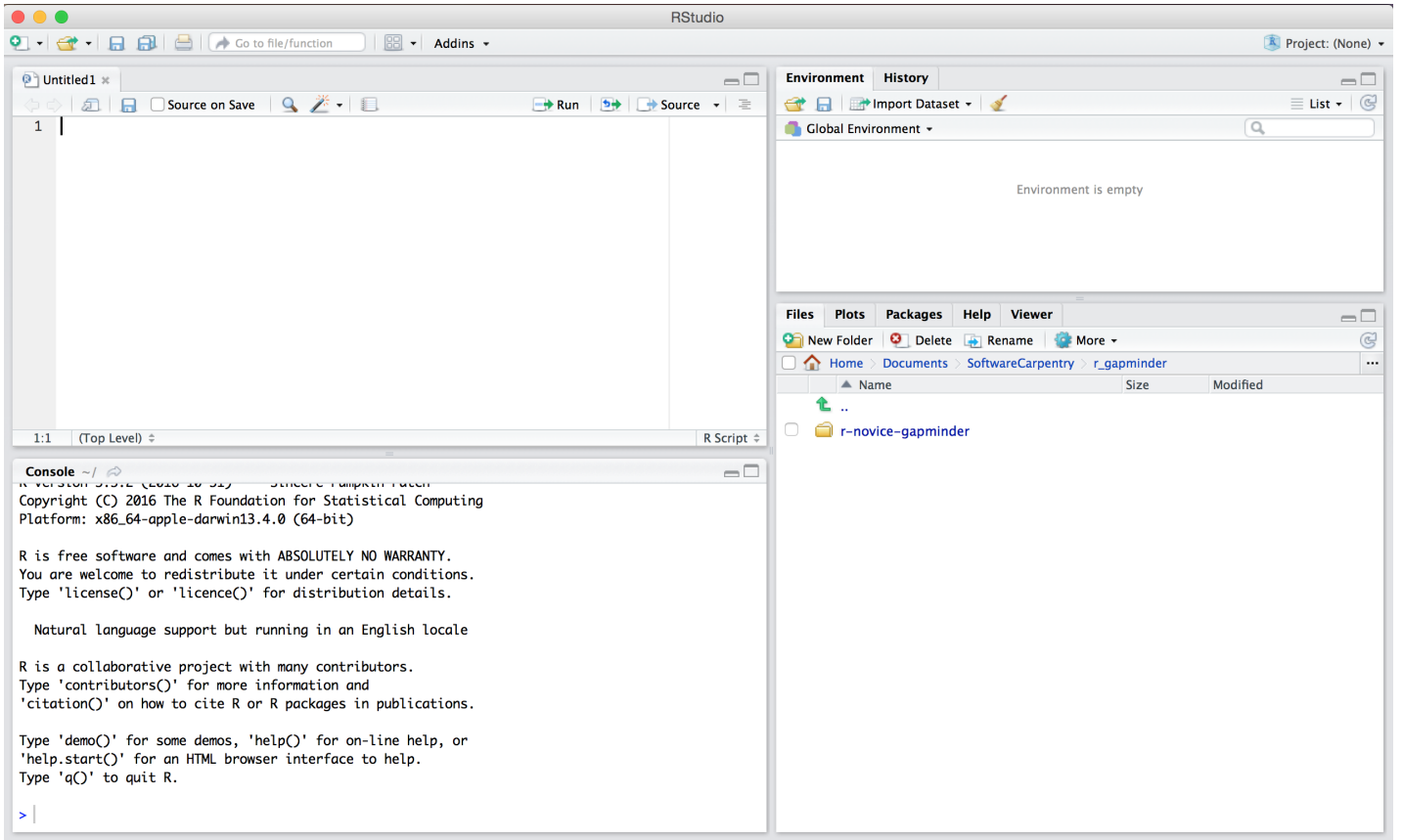
OBS: Crie um novo script para escrever seus códigos! **File -> New File -> RScript**

RStudio



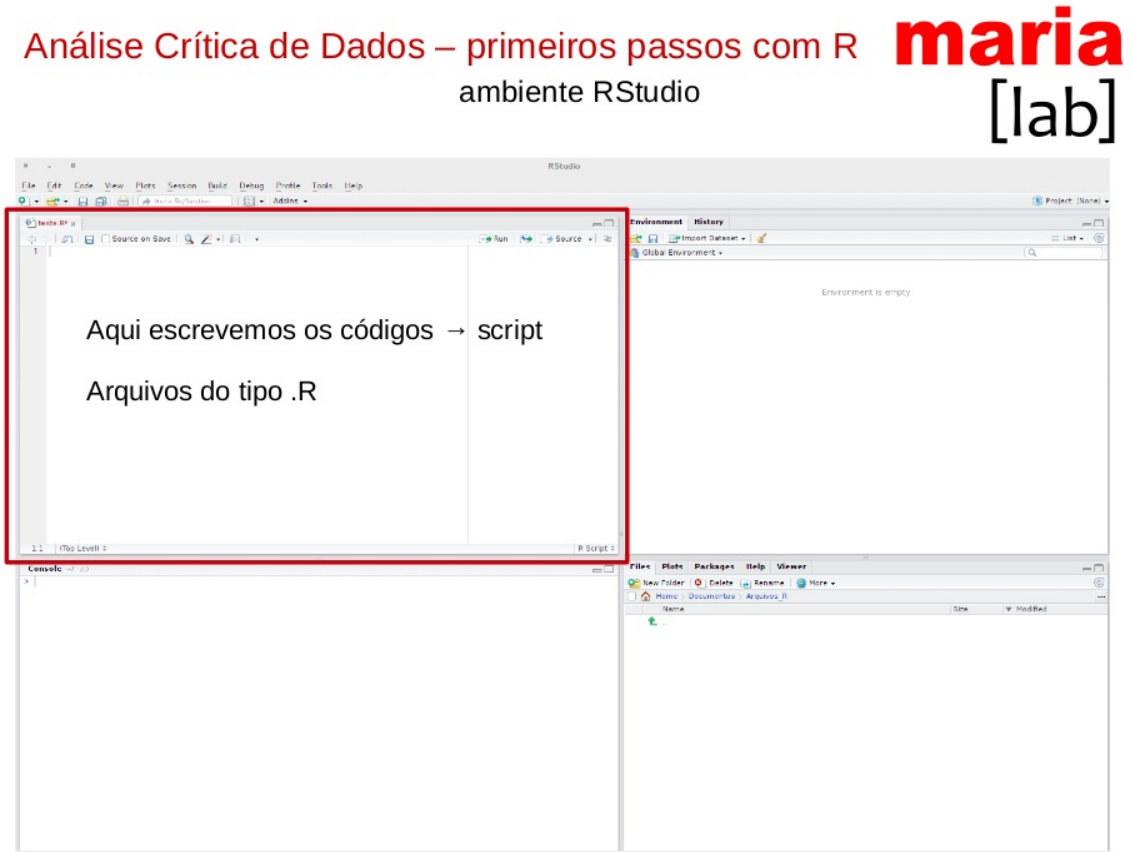
Fonte: SW Carpentry

RStudio



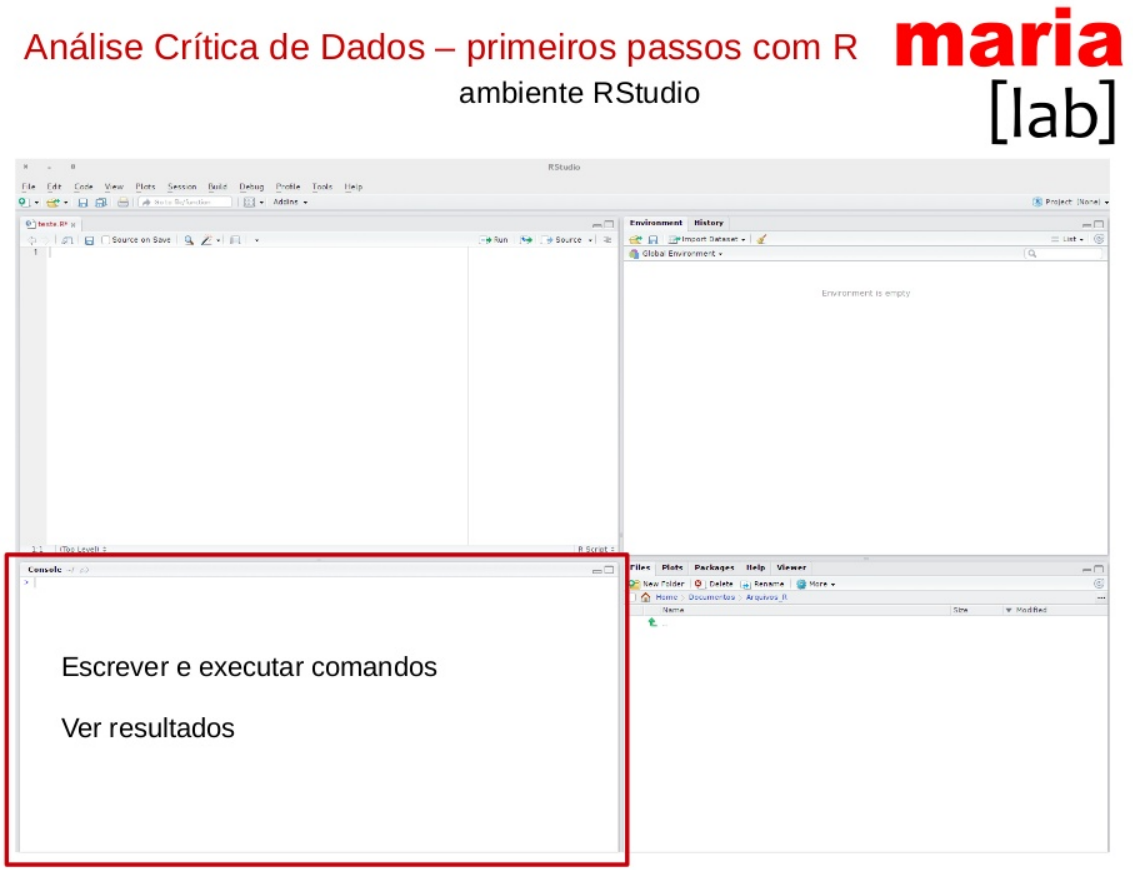
Fonte: SW Carpentry

RStudio



Fonte: Haydee Svab

RStudio



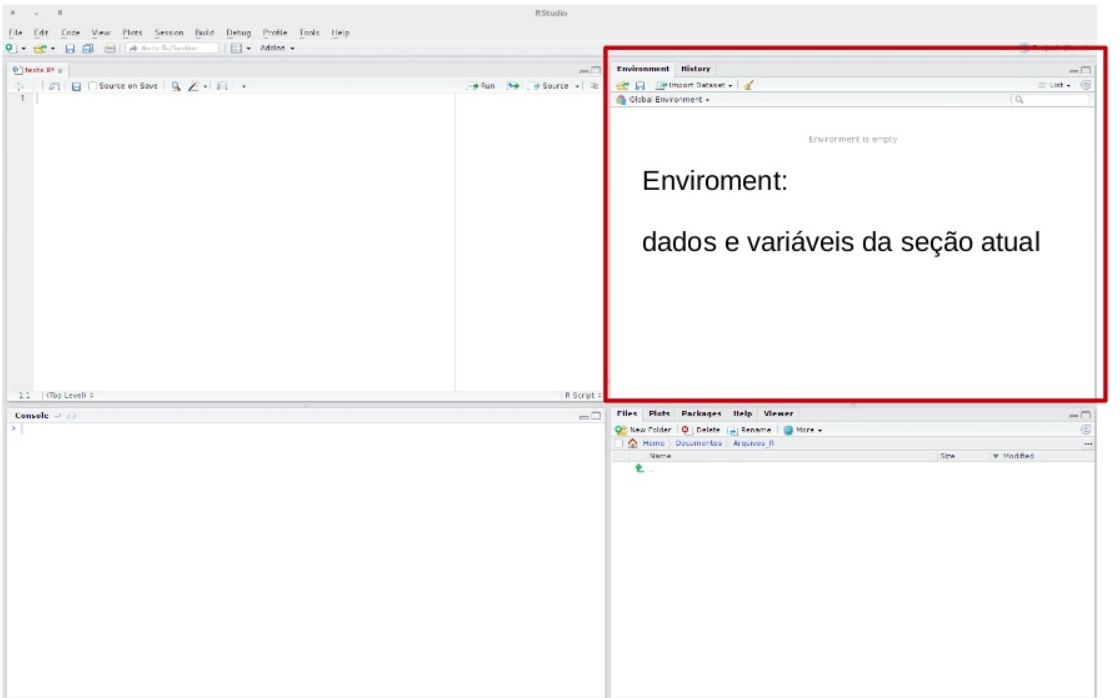
Fonte: Haydee Svab

RStudio

Análise Crítica de Dados – primeiros passos com R

ambiente RStudio

maria [lab]



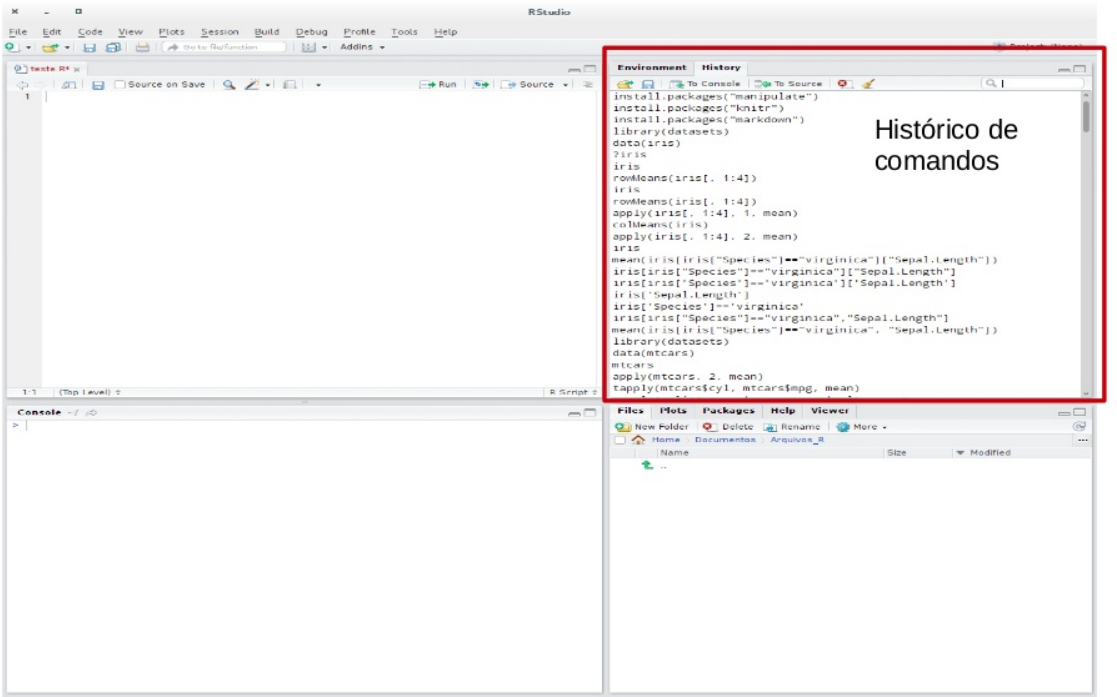
Fonte: Haydee Svab

RStudio

Análise Crítica de Dados – primeiros passos com R

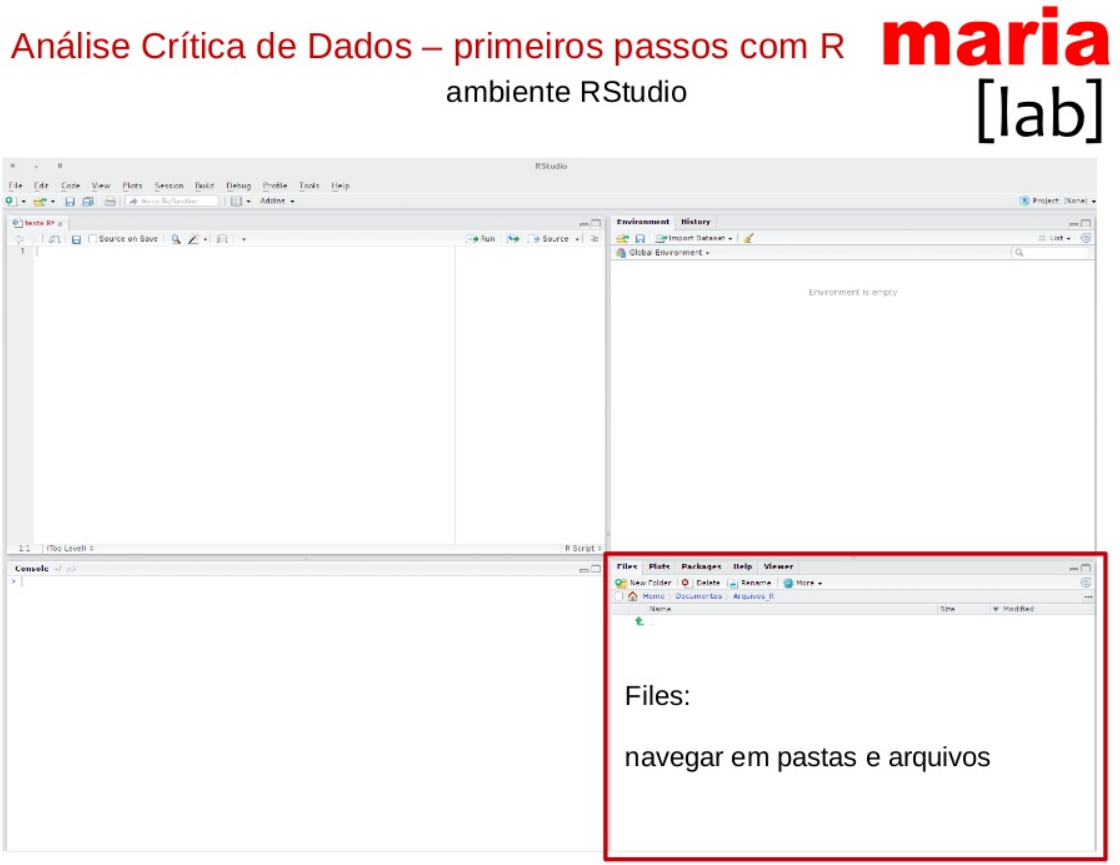
ambiente RStudio

maria [lab]



Fonte: Haydee Svab

RStudio



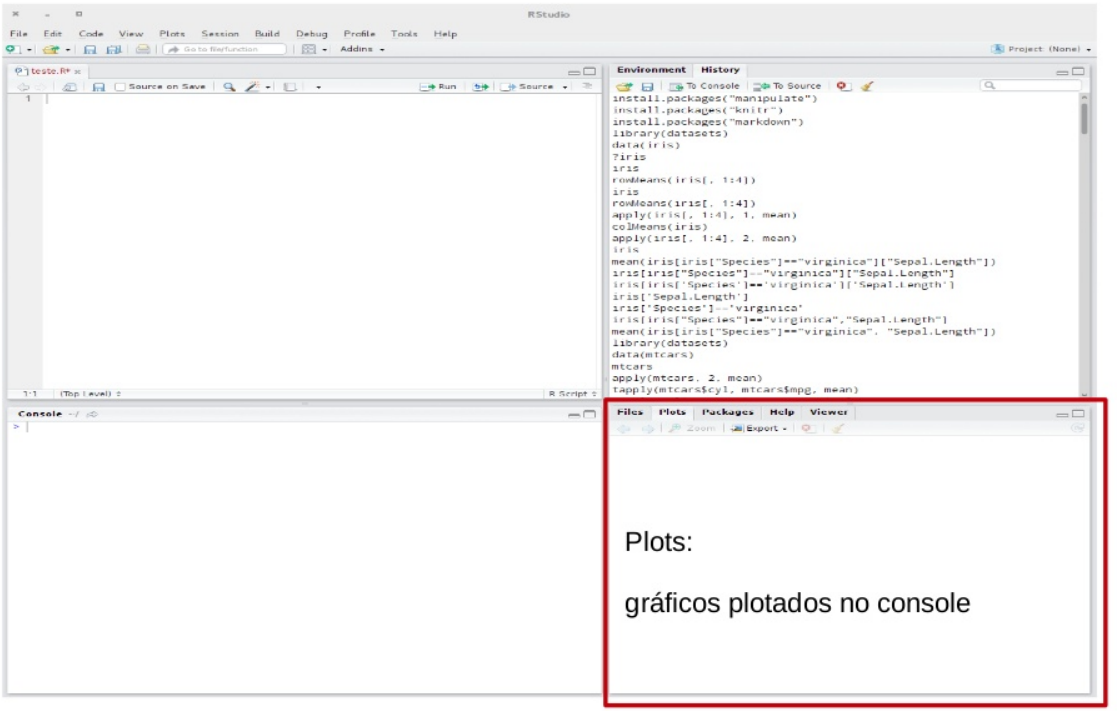
Fonte: Haydee Svab

RStudio

Análise Crítica de Dados – primeiros passos com R

ambiente RStudio

maria [lab]



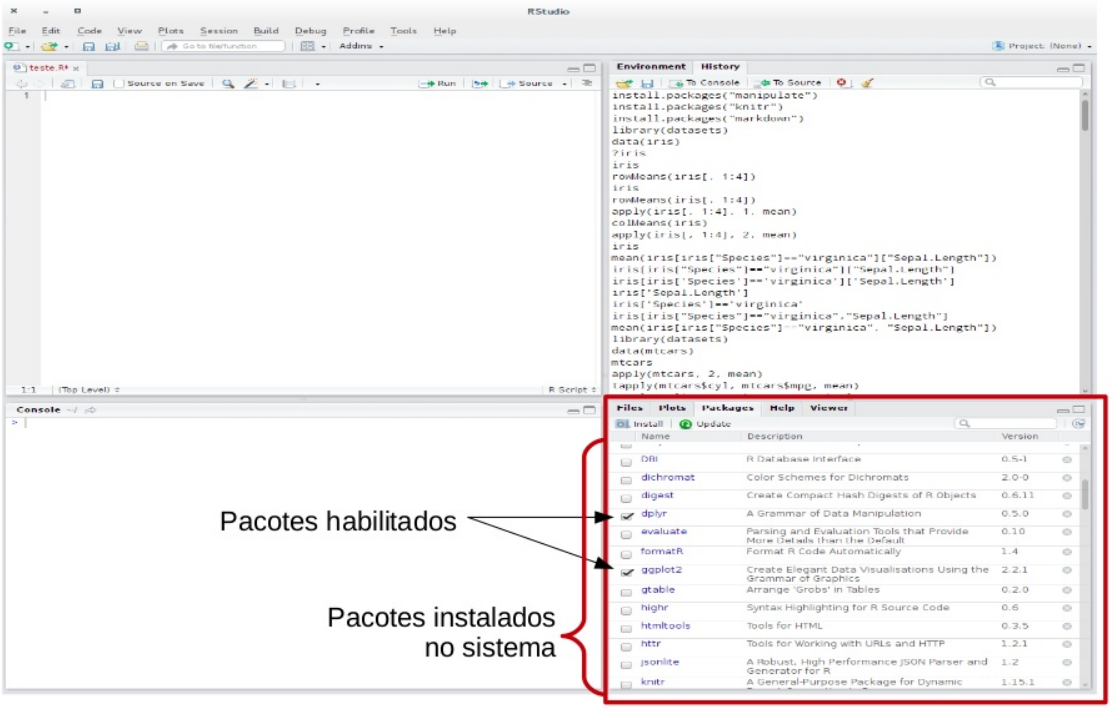
Fonte: Haydee Svab

RStudio

Análise Crítica de Dados – primeiros passos com R

ambiente RStudio

maria [lab]



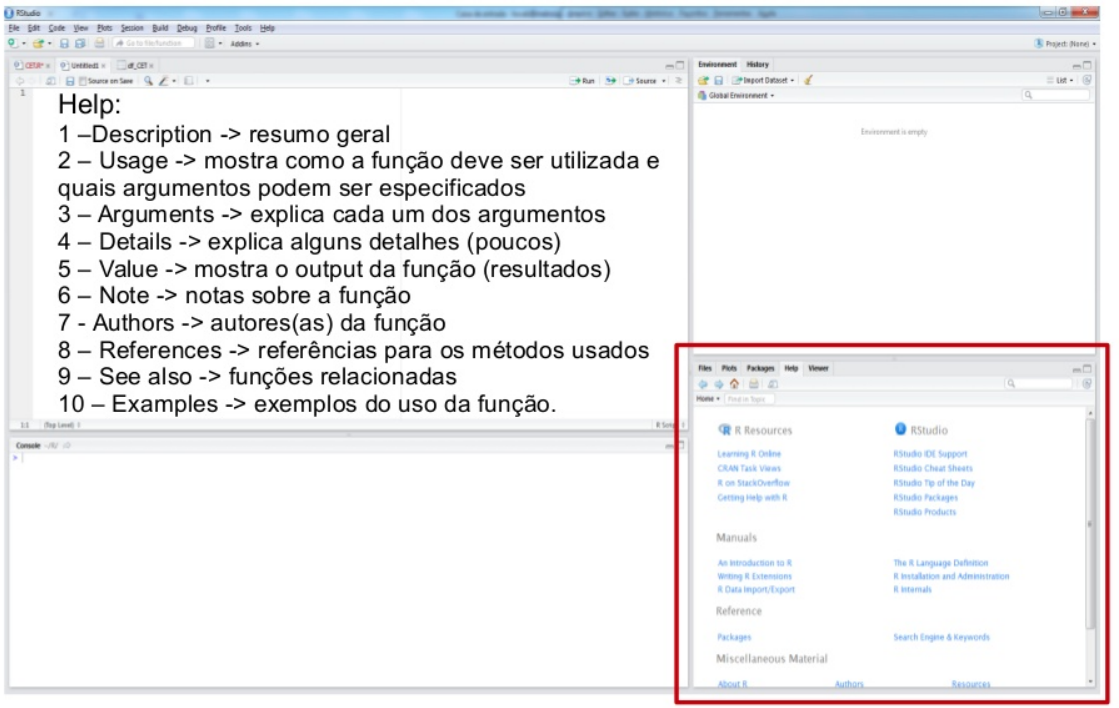
Fonte: Haydee Svab

RStudio

Análise Crítica de Dados – primeiros passos com R

ambiente RStudio

maria [lab]



Fonte: Haydee Svab

Pacotes no R

Pacotes são coleções de funções, dados e documentação que estendem as capacidades do R básico.

Eles precisam ser instalados e carregados.

Fonte: Maria Marinho

Instalação de Pacotes:

- Via CRAN: install.packages("nome-do-pacote").

```
install.packages("dplyr")
```

- Via Github: devtools::install_github("nome-do-repo/nome-do-pacote").

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

Carregar pacotes:

- library(nome-do-pacote)

```
library(dplyr)
```

Fonte: Maria Marinho

Dicas sobre Pacotes

1. Você só precisa instalar o pacote uma vez, mas precisa carregá-lo sempre que começar uma nova sessão;
2. Para instalar o pacote use as aspas;
3. Para carregar o pacote, **não** utilize as aspas.

Fonte: Maria Marinho

Pacotes - CRAN Task View

- CRAN Task View
- CRAN Task View: Analysis of Ecological and Environmental Data - Ex: Hydrology and Oceanography, Climatology, etc.
- CRAN Task View: Analysis of Spatial Data - Ex: Ecological analysis, Geostatistics, etc.

Help!

- Pedir ajuda: **help**(nome_da_funcao) ou **?**nome_da_funcao.

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

- Se a dúvida permanecer, procure no Stack OverFlow, Google.
    - E se ainda tiver dúvidas, pergunte para a comunidade (há grupos no Telegram e outras redes sociais).

Fonte: Maria Marinho

Rmarkdown

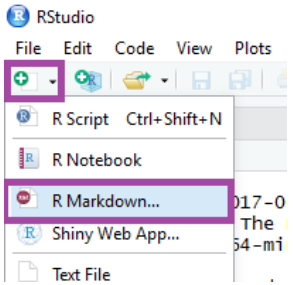
- É um tipo de arquivo que suporta códigos em R, texto, markdown e outros formatos.
- O markdown é uma linguagem de marcação simples.
- Rmarkdown Cheatsheet
- Possibilita exportar diferentes tipos de arquivos.

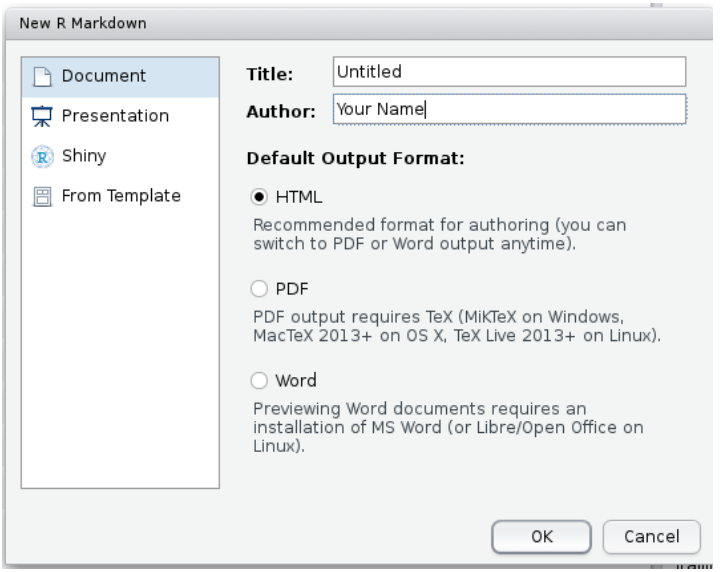
Vantagens de trabalhar com o R e o Rmarkdown

É possível elaborar relatórios no Rmarkdown. Porque é interessante?

- É possível escrever os códigos que geram as tabelas e gráficos.
- Quando o banco de dados for atualizado não será necessário refazer os gráficos. Apenas necessário compilar novamente.
- Facilita o compartilhamento dos dados e análises com outros pesquisadores.
- É possível exportar em diversos formatos, inclusive transformar o relatório em apresentações.

Já temos um projeto, agora vamos criar um arquivo .Rmd





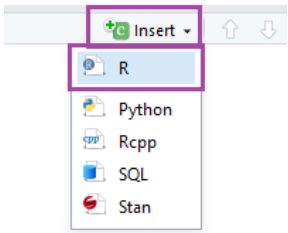
yaml - metadados do documento:

- Controla muitas das configurações do documento.

```
---
title: "Untitled"
author: "Beatriz MIlz"
date: "27 de novembro de 2018"
output: html_document
---
```

- title: título
- subtitle: subtítulo
- author: autor
- date: data
- output: formato (Ex: html_document, pdf_document, etc).
- Cuidado com a **identação**

Adicionando campos de código - chunks



- Aceita código em R, Python, SQL.
- O que for código -> Colocar dentro de um campo de código (chunk).
- O que for texto (também aceita markdown, html, css, javascript) -> Colocar fora do campo de código (chunk).
- Para colocar comentários dentro do chunk, usar o #. Ex:

```
# Se colocar um hashtag no inicio da linha, o que estiver a seguir nao será co
```

Dica: Atalho para criar chunk: `Ctrl + ALt + I`.

Classes Básicas ou Atômicas do R

- **Character**: texto
- **Integer**: números inteiros
- **Numeric**: números racionais
- **Complex**: números complexos (raramente usados para Análise de Dados)
- **Logical**: TRUE, FALSE
- **Factor**: variavéis categóricas

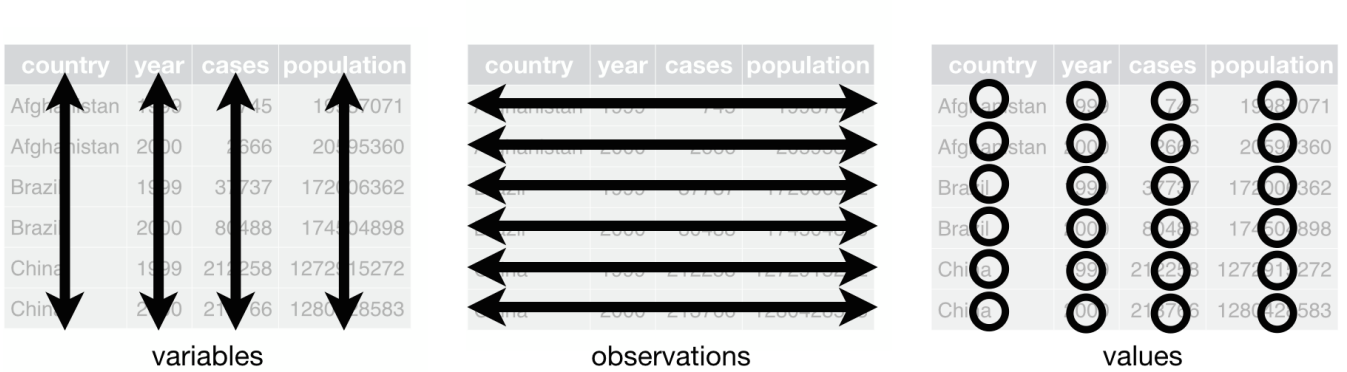
Fonte: Maria Marinho

Tipos de objetos:

- **Vector**: armazena elementos de mesma classe.
- **Matrix**: vetores de duas dimensões que armazenam elementos de mesma classe.
- **List**: tipo especial de vetor que aceita elementos de classes diferentes.
- **Data.frame**: são tabelas de dados com linhas e colunas, como uma tabela do Excel. Como são listas, essas colunas podem ser de classes diferentes.

Fonte: Maria Marinho

Dataframes - Tidy data



Fonte: Data Science with R by Garrett Grolemund

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

```
8 ^ 2    # potenciação
```

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

```
sqrt(1024) # radiciação
```

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

A ordem matemática das operações também vale no R.

Fonte: Maria Marinho

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
```

Criando objetos no R

- Para atribuir um valor a um objeto no R, utilizamos o operador <-
- O atalho ALT + - gera o operador <-
- Todas as declarações R onde são criados objetos atribuindo valores a eles, tem a mesma forma:

nome_do_objeto <- valor

Fonte: **Maria Marinho**

Nomes de objetos e variáveis

- Os nomes devem começar com uma letra. Podem conter letras, números, _ e .

- Recomendação do autor do livro R For Data Science: **usar_snake_case**, ou seja, palavras escritas em minúsculo separadas pelo underscore (_).

- O R é *case sensitive*, isto é, faz a diferenciação entre as letras minúsculas e maiúsculas. Portanto, um objeto chamado *teste* é diferente de um outro objeto chamado *Teste*.

Fonte: Maria Marinho

Exemplos de objetos e atribuição de valores

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

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

```
horas_pesquisa <- 160
horas_pesquisa
```

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

```
bolsa <- 1500.00
bolsa
```

```
## [1] 1500
```

```
ativo <- TRUE
```

Operadores Lógicos

- Igual a: `==`
- Diferente de: `!=`
- Maior que: `>`
- Maior ou igual: `>=`
- Menor que: `<`
- Menor ou igual: `<=`

Fonte: Maria Marinho

Exemplos

```
bolsa_fapesp_mestrado <-2043
bolsa_capes_mestrado <- 1500

bolsa_capes_mestrado == bolsa_fapesp_mestrado
```

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

```
bolsa_capes_mestrado >= bolsa_fapesp_mestrado
```

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

```
bolsa_capes_mestrado <= bolsa_fapesp_mestrado
```

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

Exemplos

- Comparação com vetores

```
bolsas_fapesp <- c(695.70, 2043,  3010.80, 7373.10)

bolsas_fapesp >= bolsa_capes_mestrado
```

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

Exemplos

- Operações com vetores

```
aluguel_kitnet_barata <- 806
aluguel_kitnet_cara <- 1743

bolsas_fapesp - aluguel_kitnet_barata
```

```
## [1] -110.3 1237.0 2204.8 6567.1
```

```
bolsas_fapesp - aluguel_kitnet_cara
```

```
## [1] -1047.3   300.0  1267.8  5630.1
```

Fonte dos valores: http://www.custodevida.com.br/sp/sao-paulo/

Operadores Lógicos

- Negação: **!**
- E: **&**
- OU: |

NA

Uma característica importante do R que pode dificultar a comparação são os valores ausentes ou **NAs** (não disponíveis).

NA representa um valor desconhecido.

Fonte: Maria Marinho

NA

Operações envolvendo um valor desconhecido também será desconhecido:

```
NA > 10
```

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

```
10 == NA
```

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

```
NA + 10
```

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

```
NA / 2
```

NA

```
NA == NA
```

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

is.na() é a função que testa se um objeto é NA.

Fonte: Maria Marinho

O Tidyverse

É uma coleção de pacotes R projetados para a ciência de dados. Todos os pacotes compartilham uma mesma filosofia de desenvolvimento, sintaxe e estruturas de dados. https://www.tidyverse.org/

```
library(tidyverse)
```

Fonte: Maria Marinho

Pacotes do Tidyverse

- **ggplot2**: cria gráficos
- **dplyr**: manipulação de dados
- **tidyr**: arrumar os dados
- **readr**: leitura dos dados
- **purrr**: ferramentas para programação funcional, trabalha com funções e vetores
- **tibble**: dataframes modernos, mais simples de manipular
- **magrittr**: facilita a escrita e leitura do código

Importar arquivos

Pacote **readr**: funções para ler arquivos texto

- read_csv
- **read_csv2**
- read_delim
- read_log
- read_rds

Pacote **readxl**: função para ler arquivo Excel

- **read_excel**

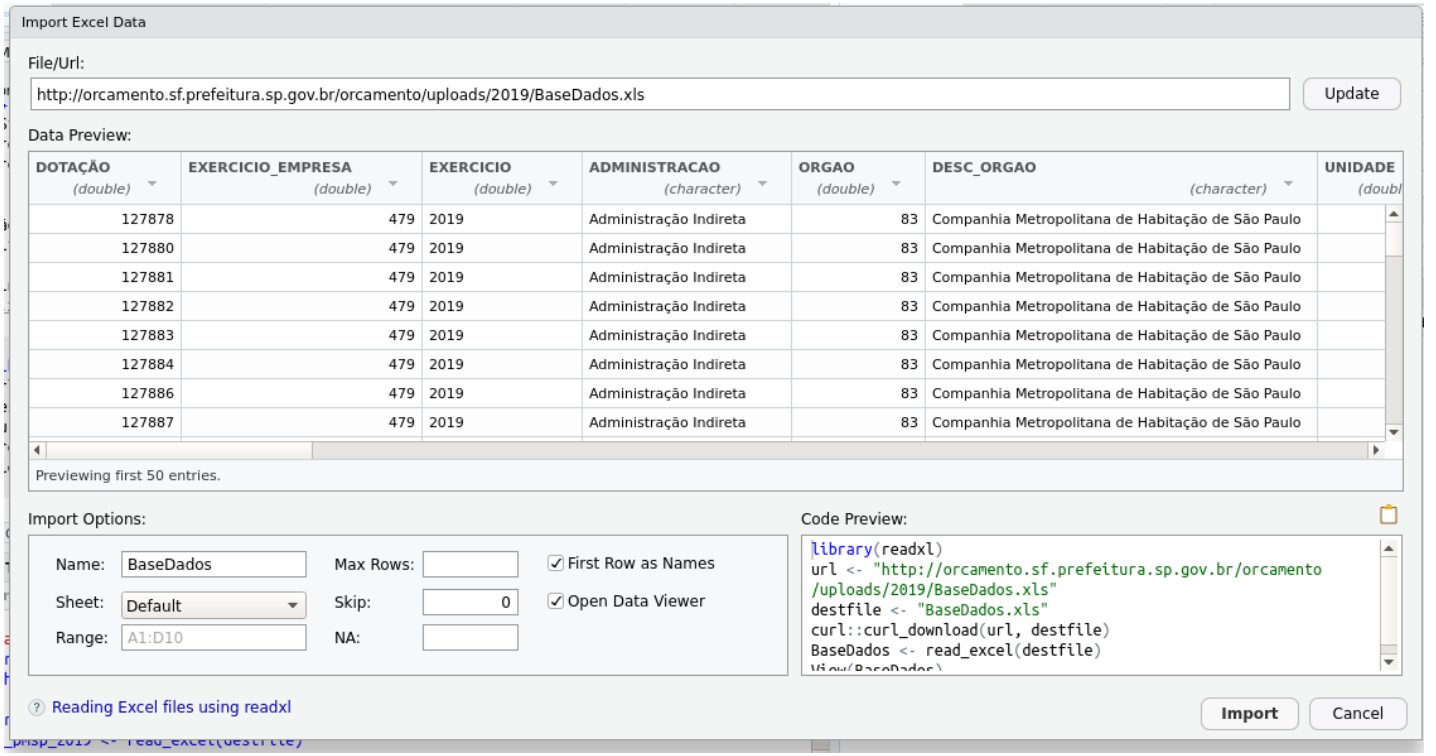
Fonte: Maria Marinho

Importar arquivos:

- Via **código (próximo slide)**, ou via **"Import Dataset"**
- RStudio -> Environment -> **Import Dataset**
  - From Excel -> arquivos xls
  - From text (readr) -> csv
  - File/URL - Colocar o link da tabela (se estiver online), ou colocar o caminho da tabela no seu projeto
  - Update - Se selecionar URL, após colar a URL clique em UPDATE. O R irá procurar essa tabela, e caso encontre, apresentará uma "amostra" da sua tabela.
  - **SEMPRE copie o Code Preview**, e guarde no seu código para que você tenha o registro das etapas realizadas (lembre-se da reprodutibilidade).

Importar arquivos:

- Exercício: importar a planilha de proposta de orçamento de 2019 da Prefeitura Municipal de São Paulo: Site e Link para a tabela.



Code preview

```
library(readxl)
url <- "http://orcamento.sf.prefeitura.sp.gov.br/orcamento/uploads/2019/BaseDa
destfile <- "proposta_pmsp_2019.xls"
curl::curl_download(url, destfile)
proposta_pmsp_2019 <- read_excel(destfile)
View(proposta_pmsp_2019)
```

View e Summary

- Para visualizar um objeto: **View**(nome-do-objeto)
- **summary()**: mostra informações sobre as colunas do dataframe:

```
summary(proposta_pmsp_2019)
```

```
##     DOTAÇÃO       EXERCICIO_EMPRESA   EXERCICIO    ADMINISTRACAO
##  Min.   :127823   Min.   :448.0     Min.   :2019   Length:4207
##  1st Qu.:129074   1st Qu.:448.0     1st Qu.:2019   Class :character
##  Median :130475   Median :448.0     Median :2019   Mode  :character
##  Mean   :130475   Mean   :449.5     Mean   :2019
##  3rd Qu.:131854   3rd Qu.:448.0     3rd Qu.:2019
##  Max.   :133265   Max.   :484.0     Max.   :2019
##      ORGAO        DESC_ORGAO           UNIDADE      DESC_UNIDADE
##  Min.   : 1.00   Length:4207        Min.   :10.00   Length:4207
##  1st Qu.:16.00   Class :character   1st Qu.:10.00   Class :character
##  Median :37.00   Mode  :character   Median :10.00   Mode  :character
```

```
sum(proposta_pmsp_2019$SALDO_ORÇ) #soma do orçamento (em reais)
```

```
## [1] 60137660056
```

```
sum(proposta_pmsp_2019$SALDO_ORÇ) / 10^9 #transformar em bilhões
```

```
## [1] 60.13766
```

Média

```
mean(proposta_pmsp_2019$SALDO_ORÇ)
```

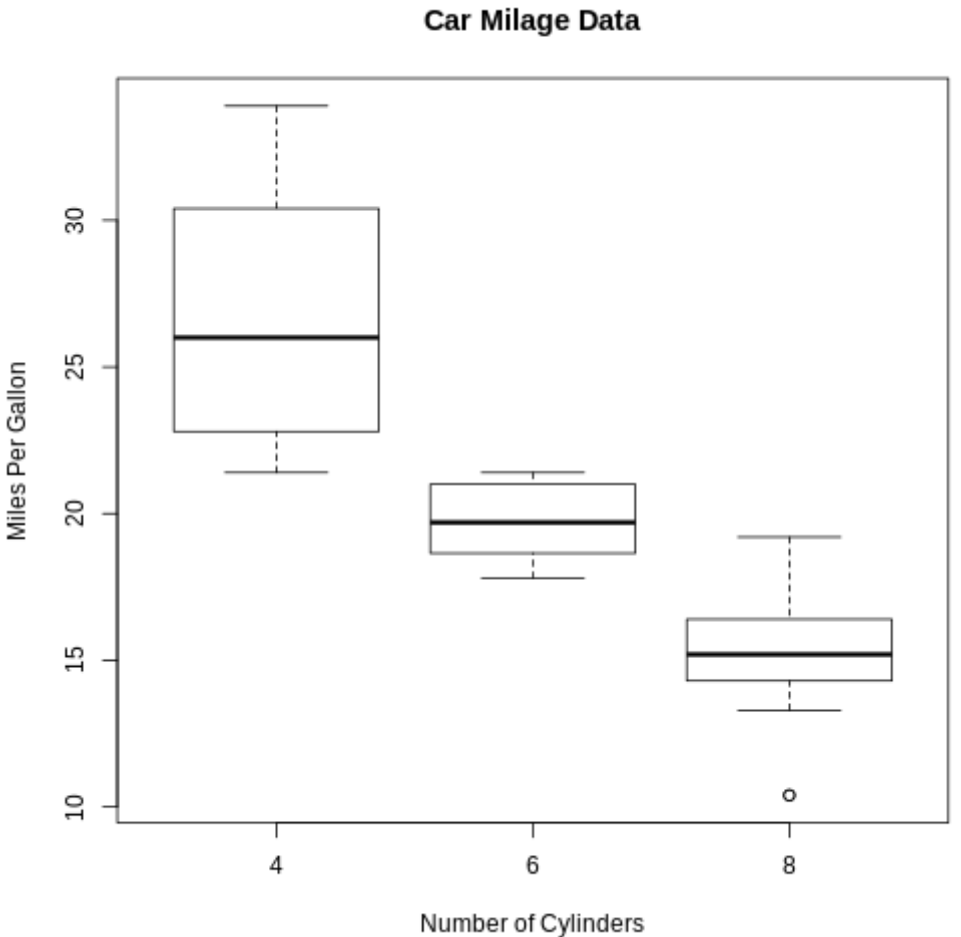
```
## [1] 14294666
```

Desvio Padrão

```
sd(proposta_pmsp_2019$SALDO_ORÇ)
```

```
## [1] 144843747
```

```
# Boxplot of MPG by Car Cylinders
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data",
   xlab="Number of Cylinders", ylab="Miles Per Gallon")
```



O operador %>%, o Pipe

Imagine uma receita que tenha as instruções: junte os ingredientes, misture e leve ao forno. Na forma usual do R, essas instruções provavelmente seriam assim:

forno(misture(junte(ingredientes)))

Dessa forma temos que pensar “de dentro para fora”. O primeiro comando que lemos é forno, sendo que essa é a última operação que será realizada.

Com o operador pipe seria algo assim:

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

É mais intuitivo!

Fonte: Maria Marinho

O operador %>%, o Pipe

Para ficar mais fácil: pense no Pipe %>% como um operador que efetua as operações à direita nos valores que estão à esquerda.

Ou ainda, o operador %>% passa o que está à esquerda como argumento para a operação da direita.

Atalho: CTRL + SHIFT + M

Fonte: Maria Marinho

dplyr

- A ideia do pacote **dplyr** é tornar a **manipulação de dados** explícita utilizando verbos que indicam a ação a ser realizada.
- O encadeamento dos verbos com o banco de dados é realizado com o operador **pipe**: **%>%**
- O dplyr foi desenhado para trabalhar com o operator pipe **%>%** do pacote magritrr.

Fonte: Maria Marinho

Os 6 verbos do dplyr

- **filter**(): seleciona linhas
- **arrange**(): ordena de acordo com uma ou mais colunas
- **select**(): seleciona colunas
- **mutate**(): cria/modifica colunas
- **summarise**(): sumariza/agrega colunas
- **group_by**(): agrupa colunas

Fonte: Maria Marinho

Exemplo dplyr

- A partir da proposta de orçamento da Prefeitura Municipal de São Paulo para 2019, filtre os dados referentes à função Gestão Ambiental e organize o saldo orçado de forma decrescente.

```
proposta_pmsp_2019_ga <-
  proposta_pmsp_2019 %>%
  filter(DESC_FUNCAO == "Gestão Ambiental")%>%
  arrange(desc(SALDO_ORÇ))

head(proposta_pmsp_2019_ga,5)
```

```
## # A tibble: 5 x 28
##   DOTAÇÃO EXERCICIO_EMPRE… EXERCICIO ADMINISTRACAO ORGAO DESC_ORGAO UNIDADE
##     <dbl>            <dbl>     <dbl> <chr>         <dbl> <chr>        <dbl>
## 1  129793              448      2019 Administraçã…    27 Secretari…      10
## 2  129808              448      2019 Administraçã…    27 Secretari…      10
## 3  132180              448      2019 Administraçã…    86 Fundo Mun…      27
```

Exemplo dplyr - DESAFIO!

- A partir da proposta de orçamento da Prefeitura Municipal de São Paulo para 2019:

1) filtre os dados referrentes à função Urbanismo;

2) organize o saldo orçado de forma decrescente.

Resposta

```
proposta_pmsp_2019_desafio <-proposta_pmsp_2019 %>%
  filter(DESC_FUNCAO == "Urbanismo")%>%
  arrange(desc(SALDO_ORÇ))
head(proposta_pmsp_2019_desafio)
```

```
## # A tibble: 6 x 28
##   DOTAÇÃO EXERCICIO_EMPRE… EXERCICIO ADMINISTRACAO ORGAO DESC_ORGAO UNIDADE
##     <dbl>            <dbl>     <dbl> <chr>         <dbl> <chr>        <dbl>
## 1  128041              462      2019 Administraçã…    81 Autoridad…      10
## 2  128036              462      2019 Administraçã…    81 Autoridad…      10
## 3  131649              448      2019 Administraçã…    99 Fundo Mun…      10
## 4  128069              462      2019 Administraçã…    81 Autoridad…      20
## 5  128040              462      2019 Administraçã…    81 Autoridad…      10
## 6  131304              448      2019 Administraçã…    87 Fundo Mun…      10
## # ... with 21 more variables: DESC_UNIDADE <chr>, FUNCAO <dbl>,
## #   DESC_FUNCAO <chr>, SUBFUNCAO <dbl>, DESC_SUBFUNCAO <chr>,
```

- A partir da proposta de orçamento da Prefeitura Municipal de São Paulo para 2019: filtre os dados referrentes à função Urbanismo; Apresente a soma de saldo orçado para as subfunções de urbanismo.

```
proposta_pmsp_2019_desafio2 <-proposta_pmsp_2019 %>%
  filter(DESC_FUNCAO == "Urbanismo")%>%
  group_by(DESC_SUBFUNCAO) %>%
  summarise(soma = sum(SALDO_ORÇ))
head(proposta_pmsp_2019_desafio2)
```

```
## # A tibble: 6 x 2
##   DESC_SUBFUNCAO                                        soma
##   <chr>                                                <dbl>
## 1 Administração Geral                                862542129
## 2 Defesa do Interesse Público no Processo Judiciário      1000
## 3 Informação e Inteligência                               1000
## 4 Infra-Estrutura Urbana                             587899420
## 5 Lazer                                                100000
## 6 Ordenamento Territorial                                 1000
```

ggplot2

- É um pacote usado para visualização de dados.
- É uma implementação da Grammar of Graphics de Leland Wilkinson - um esquema geral para visualização de dados que divide gráficos em componentes semânticos como escalas e camadas.
- Cheatsheet em português
- Material interessante: Code Your Graph - A Workshop on Visualizing Your Data with ggplotby Alison Presmanes Hill & Julianne Myers

Atalhos importantes

Os atalhos facilitam. Veja os principais:

- CTRL + ENTER: roda a linha selecionada no script.
- ALT + -: (<-) sinal de atribuição.
- CTRL + SHIFT + M: (%>%) operador pipe.
- CTRL + ALT + I: cria um chunk do RMarkdown.

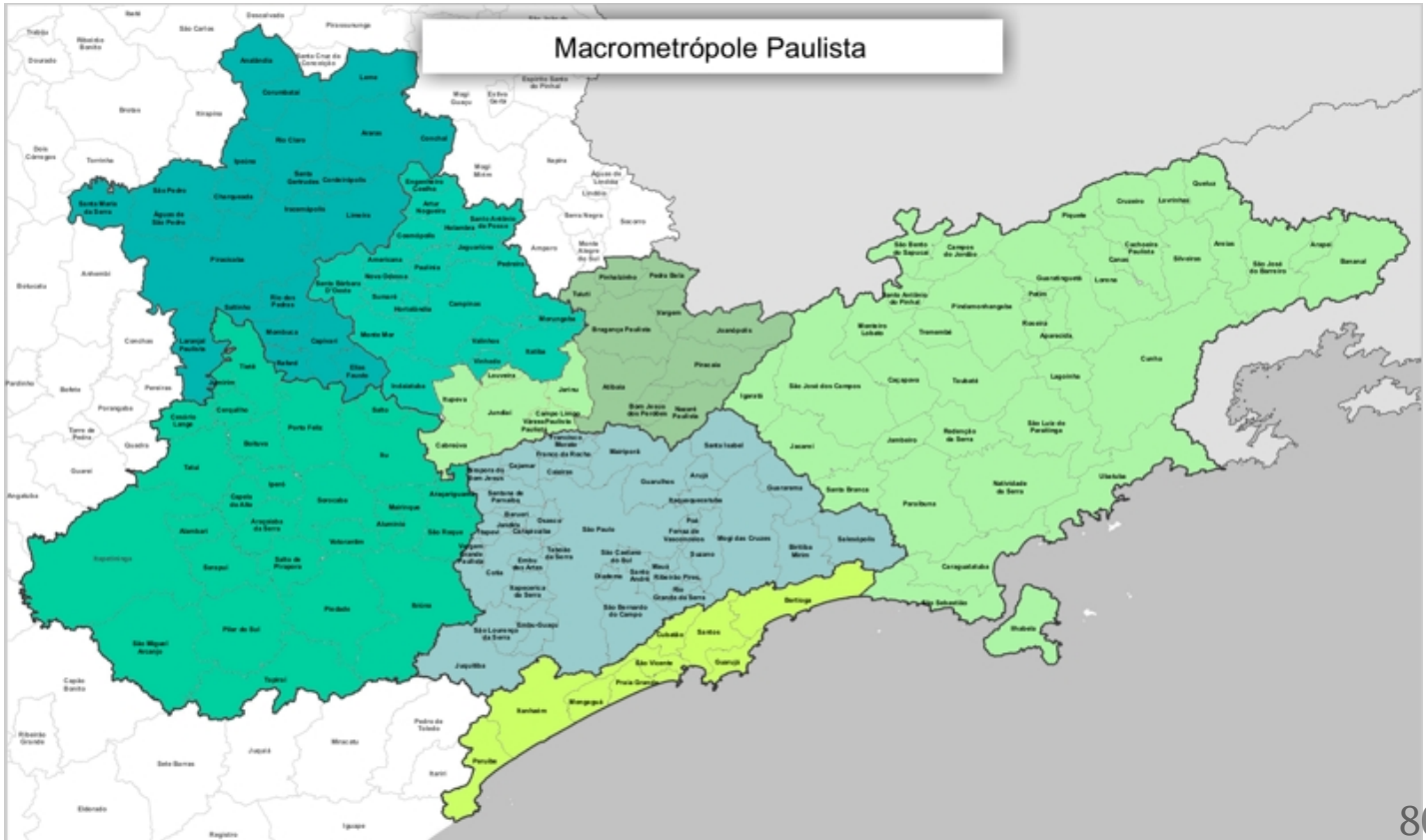
Fonte: Maria Marinho

Estudo de caso Parte 1 - Municípios da MMP

Objetivos: Preparar os dados referentes à MMP.

Macrometrópole Paulista

Projeto Temático Fapesp: Governança ambiental da macrometrópole paulista face à variabilidade climática



Delimitação territorial da Macrometrópole Paulista (MMP)

- Diferentes delimitações:

- Empresa Paulista de Planejamento Metropolitano (EMPLASA): 174 municípios
- Departamento de Águas e Energia Elétrica (DAEE): 180 municípios

Pacotes:

Instalar pacotes:

Obs: apenas se necessário! Se você tentar carregar e aparecer a mensagem `there is no package called 'nome_do_pacote'`, é só apagar o # no início da linha e rodar esse código para instalar.

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

Carregar os pacotes necessários:

```
library(tidyverse)
```

Estudo de caso

Fonte: Tabela elaborada pelo LAPLAN - UFABC Atualizado em outubro/2018

```
#Carregar a tabela de divisão administrativa
divisao_adm <- readxl::read_xlsx("dados/divisao_adm.xlsx")
summary(divisao_adm)
```

```
##  Código IBGE         Município         Região Administrativa
##  Length:192         Length:192         Length:192
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##  Região de Governo  Região Metropolitana Aglomerações Urbanas
##  Length:192         Length:192           Length:192
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##     rm_au
```

Estudo de caso

```
#Renomear as colunas
names(divisao_adm) <- c("codigo_ibge", "municipio", "regiao_administrativa", "
```

```
summary(divisao_adm)
```

```
##  codigo_ibge         municipio         regiao_administrativa
##  Length:192         Length:192         Length:192
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##  regiao_de_governo  regiao_metropolitana aglomeracoes_urbanas
##  Length:192         Length:192           Length:192
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##     rm_au
##  Length:192
```

Estudo de caso

```
#Transformar a coluna do código IBGE em número
divisao_adm$codigo_ibge <- as.integer(divisao_adm$codigo_ibge)
```

```
summary(divisao_adm)
```

```
##   codigo_ibge       municipio         regiao_administrativa
##  Min.   :3110509   Length:192         Length:192
##  1st Qu.:3512630   Class :character   Class :character
##  Median :3526803   Mode  :character   Mode  :character
##  Mean   :3520847
##  3rd Qu.:3545040
##  Max.   :3557006
##  regiao_de_governo  regiao_metropolitana aglomeracoes_urbanas
##  Length:192         Length:192           Length:192
##  Class :character   Class :character     Class :character
```

Estudo de caso

```
# Importar a tabela para o R
mmp_emplasa <- readxl::read_xlsx("dados/delimitacaoterritorialMMP_Emplasa.xlsx
summary(mmp_emplasa)
```

```
##  Municípios EMPLASA Código IBGE
##  Length:174         Length:174
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

Estudo de caso

```r
# Renomear as colunas para padronizar os nomes
names(mmp_emplasa) <- c("municipio","codigo_ibge")

summary(mmp_emplasa)
```

```
##    municipio         codigo_ibge
##  Length:174         Length:174
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

Estudo de caso

```
# Padronizar e deixar o código do IBGE configurado como número
mmp_emplasa$codigo_ibge <- as.integer(mmp_emplasa$codigo_ibge)

summary(mmp_emplasa)
```

```
##   municipio          codigo_ibge
##  Length:174         Min.   :3500600
##  Class :character   1st Qu.:3513529
##  Mode  :character   Median :3526803
##                     Mean   :3529095
##                     3rd Qu.:3545196
##                     Max.   :3557006
```

Estudo de caso - Dplyr Join

```
#Juntar os dados de divisão administrativa baseados nos municípios
#contidos na tabela da Emplasa
mmp_emplasa2 <- left_join(mmp_emplasa, divisao_adm, by="codigo_ibge")
```

```
# Deletar a coluna que fica repetida
mmp_emplasa2$municipio.y <- NULL

summary(mmp_emplasa)
```

```
##   municipio          codigo_ibge
##  Length:174         Min.   :3500600
##  Class :character   1st Qu.:3513529
##  Mode  :character   Median :3526803
##                     Mean   :3529095
##                     3rd Qu.:3545196
##                     Max.   :3557006
```

Podemos cruzar com outros dados!

Dados utilizados no estudo de caso:

- Portal de Estatísticas do Estado de São Paulo - Informações dos Municípios Paulistas - http://www.imp.seade.gov.br/frontend/
- Data do download dos dados: 18/11/2018.

```
# Importar a tabela para o R
seade_imp <- read.csv2("dados/imp_2018-11-18_00-43.csv", header=TRUE, stringsA
View(seade_imp)
```

Estudo de caso

```
#Renomear as colunas
names(seade_imp) <- c("municipio", "ano", "populacao", "populacao_masculina",
```

```
summary(seade_imp)
```

```
##   municipio              ano         populacao        populacao_masculina
##  Length:3225        Min.   :2014   Min.   :     808   Min.   :    422
##  Class :character   1st Qu.:2015   1st Qu.:    5402   1st Qu.:   2751
##  Mode  :character   Median :2016   Median :   13013   Median :   6684
##                     Mean   :2016   Mean   :   67208   Mean   :  32707
##                     3rd Qu.:2017   3rd Qu.:   39896   3rd Qu.:  19712
##                     Max.   :2018   Max.   :11753659   Max.   :5590397
##
##  populacao_feminina razao_de_sexos   populacao_urbana   populacao_rural
##  Min.   :    385    Min.   : 84.88   Min.   :     629   Length:3225
```

Left Join - Emplasa

```
# Cria a tabela e faz o left join baseado no código IBGE
seade_mmp_emplasa <- left_join(mmp_emplasa2, seade_imp, by="codigo_ibge")
# A coluna de município está repetida, então apago uma delas.
seade_mmp_emplasa$municipio.y <- NULL
```

```
write_csv(seade_mmp_emplasa,path = "dados/seade_mmp_emplasa.csv")
```

Estudo de caso Parte 2 - Municípios da Macrometrópole Paulista

Objetivos: Explorar os dados referentes à Macrometrópole Paulista.

Estudo de caso - Utilidade com relatórios dinâmicos

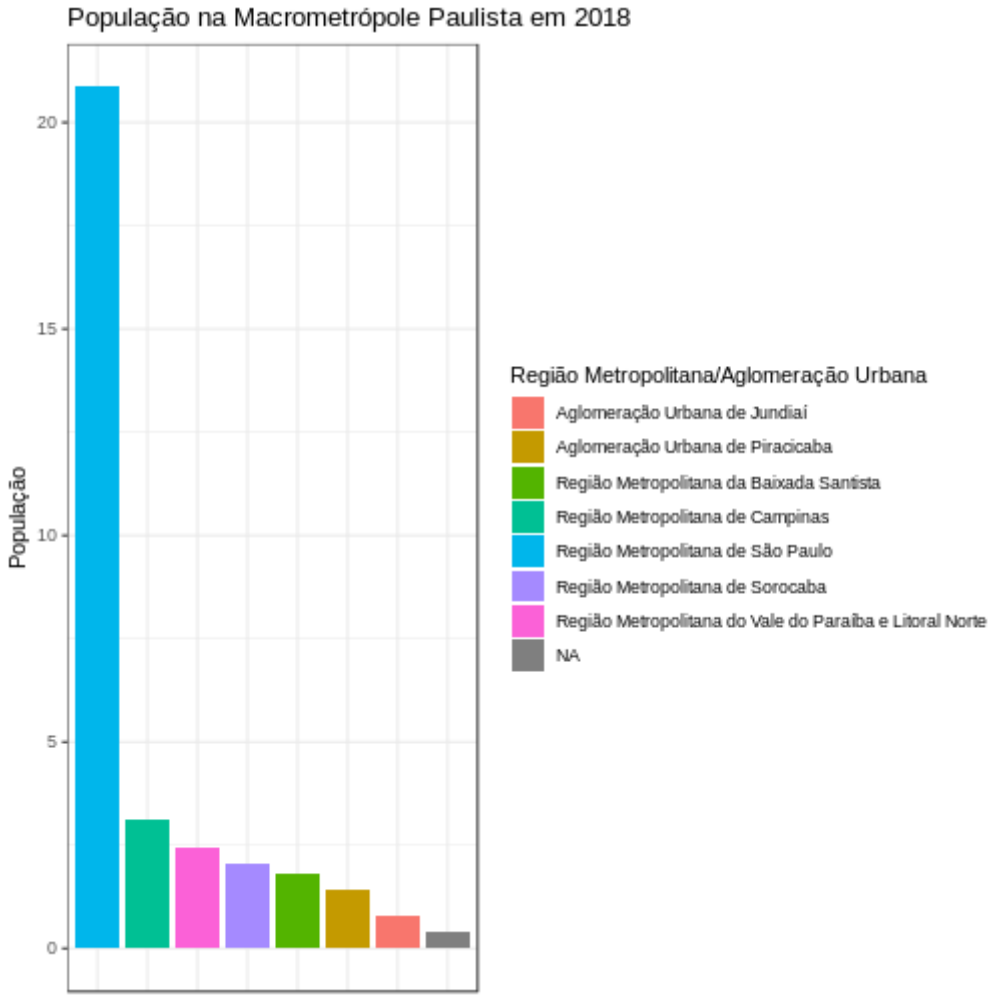
```
pop_mmp <- seade_mmp_emplasa %>%
group_by(ano) %>%
  summarise(PopulacaoTotal = sum(populacao)) %>%
  mutate(PopulacaoTotal = PopulacaoTotal/1000000)  #transformar em milhões

pop_mmp <- round(pop_mmp, 2)


# Segundo os dados da SEADE, atualmente (em 2018) a MMP tem  `r pop_mmp[5,2]`
```

Segundo os dados da SEADE, atualmente (em 2018) a MMP tem 32.88 milhões de habitantes.

População na MMP em 2018, por RM/AU, segundo os dados da SEADE:

```
seade_mmp_emplasa %>%
  mutate(populacao = populacao/1000000) %>%  #transformar em milhões
  filter(ano =="2018") %>%
  group_by(rm_au) %>% summarise(PopulacaoTotal = sum(populacao)) %>%
arrange (desc(PopulacaoTotal)) %>%
  ggplot()+
  geom_col(aes(x = reorder( rm_au ,-PopulacaoTotal), y = PopulacaoTotal, fill=
   labs(
  x = "Região Metropolitana/Aglomeração Urbana",
  y = "População",
  fill = "Região Metropolitana/Aglomeração Urbana",
  title = "População na Macrometrópole Paulista em 2018"
)+
  theme_bw()+
 theme(axis.title.x=element_blank(),
       axis.text.x=element_blank(),
       axis.ticks.x=element_blank())
```

População na MMP em 2018, por RM/AU, segundo os dados da SEADE:



Conclusão

Referências e materiais de estudo

Maria Marinho

R for Reproducible Scientific Analysis - SW Carpentry

(mais referências nos próximos slides)

Onde estudar mais?

- Repositório de materias de estudo com R
- Curso R
  - Curso-R - Cursos presenciais sobre R.
  - Material Curso-R - Material disponibilizado pela Curso-R para estudo.
  - Blog Curso-R - Blog do Curso-R com diversos conteúdos interessantes.
- Datacamp: Introdução ao R/Introduction to R - Conteúdo: introdução, tipos de dados no R: vetores, matrizes, fatores, data frames, listas.

Onde estudar mais?

- Software Carpentry - Programming with R
- Software Carpentry - R for Reproducible Scientific Analysis

Disciplinas

- **Disciplina FLS6397:** Introdução à Programação e Ferramentas Computacionais para as Ciências Sociais
- **USP - FFLCH**

- O material da disciplina é disponibilizado no GITHUB pelo Leonardo Barone. Link: https://github.com/leobarone/FLS6397_2018

- **Conteúdo**: 1 - Básico da Programação em R 2 - Estruturas de dados e manipulação de bases em R 3 - Tabelas e Gráficos em R 4 – R e integração com SQL 5 - Git básico 6 - Markdown básico 7 - LaTex básico 8 - Captura de dados na internet 9 - Textos, corpus e processamento de linguagem natural 10 - Mapas e GIS 11 - Redes e grafos

Disciplinas

- **Disciplina EPI5713** - Introdução ao R para a Análise de Dados
- **USP - FSP**
- **Conteúdo**: 1 - Introdução (instalação do R, RStudio, estrutura dos dados, pacotes e abertura de bancos de dados de outras fontes). 2 – Limpeza de bancos de dados (criação, renomeação e exclusão de variáveis, alteração de valores, identificação de valores missing, alteração do tipo de variável, fusão de bancos de dados). 3 – Análise descritiva (descrição das variáveis, análise de frequência e tabelas bivariadas). 4 – Análise bivariada (Chi-quadrado, teste-t e correlação). 5 – Regressão (regressão linear e logística). 6 – Mapas no R (shapefiles e visualização de dados em mapas). 7 – Gráficos com o ggplot2 (gráficos de dispersão, pirâmides populacionais, boxplots e histogramas).

Disciplinas

- **Disciplina BIE5782** - Uso da Linguagem R para Análise de Dados em Ecologia
- **USP - IB**
- **Conteúdo**: 1. Introdução: histórico e filosofia de trabalho do R, breve histórico da linguagem S e do R, exemplo de uma sessão de trabalho no R. 2. Funções no R e sua aplicação: matemáticas, lógicas, e de distribuições de probabilidade. 3. Leitura e manipulação de dados: tipos de variáveis e estrutura dos dados, transformações de dados, operações vetoriais e matriciais, indexação, agregação e transformação de estrutura de dados. 4. Análise exploratória de dados: estatísticas descritivas, gráficos exploratórios. 5. Modelos lineares: lógica geral em R, regressão linear, análise de variância, verificação das pressuposições dos modelos lineares. 6. Simulações e reamostragem. 7. Noções de programação em linguagem S: fundamentos de programação orientada a objetos; lógica e controle de fluxo em linguagem S; procedimentos vetoriais de programação.

Livros

- Hands-On Programming with R - Livro em inglês, disponibilizado gratuitamente online. Conteúdo: básico do R e RStudio, Pacotes, objetos, tipos de dados, funções, etc.
- R for Data Science - Livro em inglês, disponibilizado gratuitamente online. Conteúdo muito bom sobre R.
- R Cookbook
- Ciência de Dados com R - Instituto Brasileiro de Pesquisa e Análise de Dados
- Biblioteca do R-Project, livros em diversas línguas: https://www.r-project.org/doc/bib/R-books.html

Cheatsheets

- Ambiente de desenvolvimento RSTUDIO: Português, Inglês
- Importar dados/Data import - Inglês
- Visualização de dados - Português
- Manipular dados - Português
- Data transformation - Inglês
- Trabalhar com textos/ String manipulation - Inglês
- Comunicar seus resultados - RMarkdown - Inglês
- Criar gráficos com Ggplot2 - Inglês
- LaTeX - Inglês
- Expressões regulares - Inglês
- Pode conferir também outras cheatsheets em inglês no site: https://www.rstudio.com/resources/cheatsheets/

Links interessantes

- Dados de execução orçamentária da PMSP
- Rpollution - Blog onde é publicado análise de dados sobre poluição do ar;

Obrigada!

Apresentação elaborada com Xaringan