Introdução à Linguagem de Programação em R para tratamento de dados de poluição do ar

Data frames, qualR, e plots

Mario Gavidia-Calderón, Rafaela Squizzato, Thiago Nogueira

06/02/2024

Universidade de São Paulo

Dúvidas aula passada

Pacotes

Do Excel para R

Subset data frames

qualR

Plots

Dúvidas aula passada

Dúvidas aula passada

- Como instalar pacotes que podem não ser compativéis com a versão do R?
- Como leer arquivos `.dbc`?

Dúvidas aula passada

- Você pode instalar desde o código fonte.

```
install.packages(
  "~/Downloads/read.dbc_1.0.6.tar.gz",
  repos = NULL,
  type = "source"
)
```

- O pacote `read.dbc` lê os arquivos `dbc`.

Pacotes

Pacotes

- Pacotes são como as extensões no browser.
- Conjunto de funções específicas para tarefas específicas.

Bibliotecas

- Para instalar pacotes usamos a função
  `install.packages("nome_do_biblioteca")`

```
# Instalando Rmarkdown
install.packages("rmarkdown")
```

Bibliotecas

- Por exemplo, para ler arquivos `.xls` podemos instalar a biblioteca `readxls`

```
# Instalando readxl
install.packages("readxl")
```

Do Excel para R

Do Excel para R

- Um jeito de ler arquivos `.xls` é abrir os arquivos em Excel ou Google Sheet e salvar como `.csv`.
- Depois ler a tabela usando a função `read.table()`
- As vezes vale a pena abrir no Excel ou Google Sheet e mudar o nome das colunas.
- Também podemos usar a função `read_excel()` do pacote `readxl`

Base de dados da WHO

- Vamos usar a base de dados de qualidade do ar da WHO

```
library(readxl)
who <- read_excel("../../data/who_aap_2021_v9_11august2022.xlsx",
                  sheet = 2) # Nome o numero do sheet no arquivo
```

Base de dados da WHO

- Este é um exemplo de nomes de coluna *complicados*.

```
str(who)
```

```
## tibble [32,191 x 15] (S3: tbl_df/tbl/data.frame)
##  $ WHO Region                              : chr [1:32191] "Eastern Me
##  $ ISO3                                    : chr [1:32191] "AFG" "ALB"
##  $ WHO Country Name                        : chr [1:32191] "Afghanista
##  $ City or Locality                        : chr [1:32191] "Kabul" "Du
##  $ Measurement Year                        : num [1:32191] 2019 2015 2
##  $ PM2.5 (µg/m3)                           : num [1:32191] 119.8 NA 14
##  $ PM10 (µg/m3)                            : num [1:32191] NA 17.6 24.
##  $ NO2 (µg/m3)                             : num [1:32191] NA 26.6 24.
##  $ PM25 temporal coverage (%)              : num [1:32191] 18 NA NA NA
```

Subset data frames

Subset

- As vezes precisamos selecionar um sub-conjunto de dados do data frame.
- Um jeito simples é usar a função `subset()`

```
novo_df <- subset(df_original,
                  subset = condição baseado nas filas,
                  select = vetor com as colunas)
```

Subset dados da who para Brasil

- Vamos selecionar os dados do Brasil.

```
who_br <- subset(who,
                 subset = `WHO Country Name` == "Brazil")
```

- Olha como o nome da coluna que tem espaços é escrita usando ‘ ‘.

Subset dados da who para Brasil

- Vamos selecionar os dados do Brasil.

```
# Examinando o novo data frame
nrow(who_br)
```

```
## [1] 479
```

```
head(unique(who_br$`City or Locality`))
```

```
## [1] "Americana"   "Aracatuba"   "Araraquara"  "Araucaria"   "Barra M
## [6] "Bauru"
```

- Olha como o nome da coluna que tem espaços é escrita usando ‘ ‘.

Subset dados da WHO para São Paulo

```
who_sp <-  subset(who,
                  subset = `City or Locality` == "Sao Paulo" ,
                  select = c("Measurement Year",
                             "PM2.5 (µg/m3)",
                             "PM10 (µg/m3)",
                             "NO2 (µg/m3)" ))
```

Subset dados da WHO para São Paulo

- Vamos arrumar um pouco o data frame

```
names(who_sp)
```

```
## [1] "Measurement Year" "PM2.5 (µg/m3)"    "PM10 (µg/m3)"     "NO2 (µ
```

```
names(who_sp) <- c("year", "pm25", "pm10", "no2")
names(who_sp)
```

```
## [1] "year" "pm25" "pm10" "no2"
```

Anos acima do padrão da WHO

```
plot(who_sp$year, who_sp$no2, t = "l", lwd = 1.5,
     xlab = "Anos", ylab = "NO2 (ug/m3)")
abline(h = 40, lwd = 1.5, col = "red")
legend("topright", col = "red", lwd = 1.5, legend = "NO2 WHO AQS")
```

Exercício

- Quais cidades ultrapassaram o Padrão de qualidade do ar de NO2 no ano 2019?

qualR

qualR

- Pacote desenvolvido para baixar os dados da CETESB dentro do R.
- Gera dataset prontos e completos para análise:
  - Horários faltantes preenchidos com `NA`.
  - Coluna `date` tipo `POSIXct` para usar diretamente com `openair`.
- Precisa ter cadastro na plataforma QUALAR da CETESB.
- Referência qualR

Instalação qualR

- O jeito mais fácil é o seguinte:

```
install.packages('qualR',
                 repos = c('https://ropensci.r-universe.dev',
                           'https://cloud.r-project.org'))
```

Instalação qualR

Atenção

Para usar ‘qualR‘ você precisa ter uma conta no sistema QUALAR da CETESB. Você precisa do usuário e da senha.

Códigos dos parâmetros e das estações

- Para saber os códigos das estações usamos `cetesb_aqs` na coluna `code`.

```
library(qualR)
head(cetesb_aqs, 4)
```

```
##         name code       lat       lon      loc
## 1  Americana  290 -22.72425 -47.33955 Interior
## 2  Araçatuba  107 -21.18684 -50.43932 Interior
## 3 Araraquara  106 -21.78252 -48.18583 Interior
## 4      Bauru  108 -22.32661 -49.09276 Interior
```

Códigos dos parâmetros e das estações

- Para saber os códigos dos parârmetros `cetesb_param` na coluna `code`.

```
library(qualR)
tail(cetesb_param, 4)
```

```
##                            name units code
## 17    TEMP (Temperatura do Ar)    ºC   25
## 18               TOL (Tolueno) ug/m3   62
## 19 UR (Umidade Relativa do Ar)     %   28
## 20    VV (Velocidade do Vento)   m/s   24
```

Baixar um poluente de uma estação

- Vamos baixar dados de **ozônio** da **primeira semana de janeiro de 2024** na estação **Pinheiros**. Usamos a função `cetesb_retrieve_param()`.

```
pin_o3 <- cetesb_retrieve_param(
  username = Sys.getenv("QUALAR_USER"), # Trocar pelo seu usuário
  password = Sys.getenv("QUALAR_PASS"), # Trocar pela sua senha
  parameters = 63, # de cetesb_param
  aqs_code = 99, # de cetesb_aqs,
  start_date = "01/01/2024", # Formato dd/mm/yyyy
  end_date = "07/01/2024" # Formato dd/mm/yyyy
)
```

Baixar um poluente de uma estação

- Também aceita o nome da estação (igual em `cetesb_aqs`) e o nome do parâmetro (igual em `cetesb_param`).

```
pin_o3 <- cetesb_retrieve_param(
  username = Sys.getenv("QUALAR_USER"), # Trocar pelo seu usuário
  password = Sys.getenv("QUALAR_PASS"), # Trocar pela sua senha
  parameters = "O3", # de cetesb_param
  aqs_code = "Pinheiros", # de cetesb_aqs,
  start_date = "01/01/2024", # Formato dd/mm/yyyy
  end_date = "07/01/2024" # Formato dd/mm/yyyy
)
```

Baixar vários poluentes de uma estação

- Também podemos baixar vários poluentes de uma estação. Só precisamos definir os poluentes para baixar em um `vetor`. Vamos baixar O_3, $PM_{2.5}$, e NO_X.

```
pols <- c("O3", "MP2.5", "NOx") # Olhar usando cetesb_aqs
pin_o3 <- cetesb_retrieve_param(
  username = Sys.getenv("QUALAR_USER"), # Trocar pelo seu usuário
  password = Sys.getenv("QUALAR_PASS"), # Trocar pela sua senha
  parameters = pols, # vetor com poluentes
  aqs_code = "Pinheiros", # de cetesb_aqs,
  start_date = "01/01/2024", # Formato dd/mm/yyyy
  end_date = "07/01/2024" # Formato dd/mm/yyyy
)
```

Baixar vários poluentes e meteorologia de uma estação

- Só precisamos definir os parâmetros para baixar em um vetor. Vamos baixar $PM_{2.5}$ e velocidade e direção do vento.

```
params <- c("MP2.5", "VV", "DV") # Olhar usando cetesb_aqs
pin_pm25 <- cetesb_retrieve_param(
  username = Sys.getenv("QUALAR_USER"), # Trocar pelo seu usuário
  password = Sys.getenv("QUALAR_PASS"), # Trocar pela sua senha
  parameters = params, # vetor com poluentes
  aqs_code = "Pinheiros", # de cetesb_aqs,
  start_date = "01/01/2024", # Formato dd/mm/yyyy
  end_date = "07/01/2024" # Formato dd/mm/yyyy
)
```

Baixar vários poluentes e meteorologia de uma estação

- Você não precisa colocar os argumentos das funções mas precisa seguir a ordem dos argumentos.

```
params <- c("MP2.5", "VV", "DV")
pin_pm25 <- cetesb_retrieve_param(
  Sys.getenv("QUALAR_USER"),  # username
  Sys.getenv("QUALAR_PASS"), # password
  params, # parameters
  "Pinheiros", # aqs_code
  "01/01/2024", # start_date
  "07/01/2024" # end_date
)
```

Salvar dados em `.csv`

- Pode ser que você precise usar outro software (e.g. PMF). Então você pode exportar os dados em `.csv`. Só adicionar o argumento `to_csv = TRUE`.

```
params <- c("MP2.5", "VV", "DV") # Olhar usando cetesb_aqs
pin_pm25 <- cetesb_retrieve_param(
  username = Sys.getenv("QUALAR_USER"), # Trocar pelo seu usuário
  password = Sys.getenv("QUALAR_PASS"), # Trocar pela sua senha
  parameters = params, # vetor com poluentes
  aqs_code = "Pinheiros", # de cetesb_aqs,
  start_date = "01/01/2024", # Formato dd/mm/yyyy
  end_date = "07/01/2024", # Formato dd/mm/yyyy
  to_csv = TRUE
)
```

Salvar dados em `.csv`

- O arquivo salvo tem o nome `Pinheiros_MP2.5_VV_DV_01-01-2024_07-01-2024.csv` e ficará na pasta de trabalho (conferir usando `getwd()`).

Um mesmo poluente de várias estações

- As vezes você precisa comparar valores de várias estações. Neste exemplo vamos baixar valores de NO_X da estação Ibirapuera e Pinheiros.

```
aqs <- c(99, 83) # de cetesb_aqs, Pinheiros é 99 e Ibirapuera 83
nox_pin_ibi <- lapply(
  aqs,
  cetesb_retrieve_param,
  username = Sys.getenv("QUALAR_USER"),
  password = Sys.getenv("QUALAR_PASS"),
  parameters = "NOx",
  start_date = "01/01/2024",
  end_date = "07/07/2024"
)
```

Um mesmo poluente de várias estações

- O resultado de usar `lapply` é uma lista. Vamos transformar em um `data.frame`.

```
nox_all <- do.call(rbind, nox_pin_ibi)
```

Um mesmo poluente de várias estações

- Usando subset podemos separar os data frames.

```
nox_pin <- subset(nox_all, subset = aqs == "Pinheiros")
nox_ibi <- subset(nox_all, subset = aqs == "Ibirapuera")
```

Um mesmo poluente de várias estações

- Vamos comparar as estações

```
mean(nox_pin$nox, na.rm = TRUE)
```

```
## [1] 25.81125
```

```
mean(nox_ibi$nox, na.rm = TRUE)
```

```
## [1] 14.09988
```

- Pinheiros tem maior concentração de NOX do que Ibirapuera. Por que?

Outras funções.

- `cetesb_retrieve_param()` é a função mais importante e a mais usada.
- Existem outras funçãoes que não precisam do argumento `parameters` pois foram desenvolvidas para baixar parâmetros específicos:
  - `cetesb_retrieve_pol()`: Baixa todos os poluentes.
  - `cetesb_retrieve_met()`: Baixa todos os parâmetros meteorológicos.
  - `cetesb_retrieve_met_pol()`: Baixa todos os parâmetros da estação.

Plots

Plots

- *Uma figura vale mais do que 100 palavras.*
- Vamos aprofundar como usar **R Base Graphics** para criar os plots.
- Seguir as *10 simple rules for better figures* do **Rougier et al. (2014)**

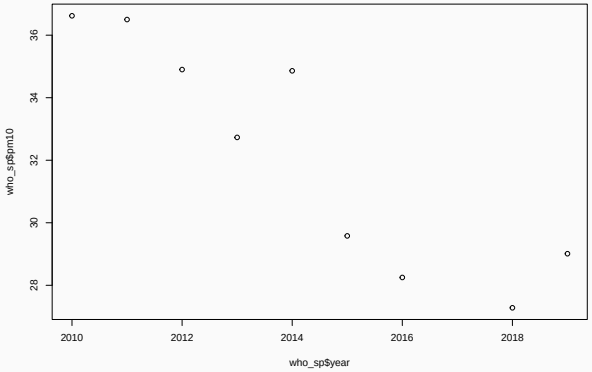
Plots

- A principal função é **`plot()`**. Podemos usar `plot()` para criar séries de tempo e gráfico de dispersão.
- Outras funções são:
  - `hist()`: Cria histogramas.
  - `barplot()`: Diagrama de barras.
  - `boxplot()`: Diagrama de caixas.

Séries temporais

- Vamos plotar a série temporal da concentrações de pm10.

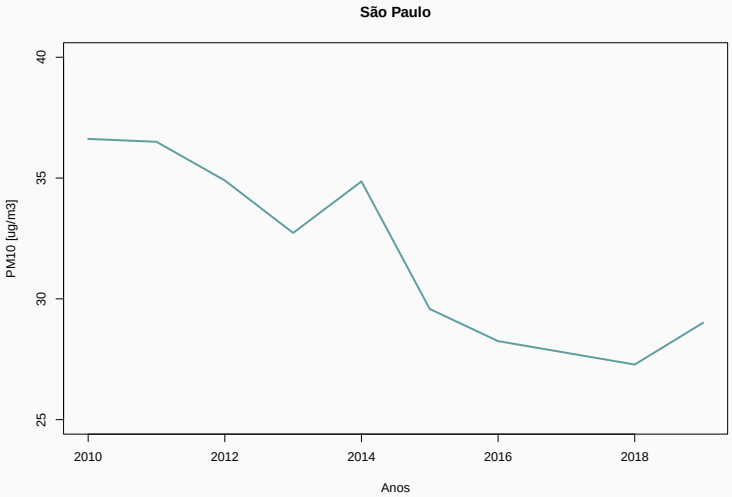
```
plot(who_sp$year, who_sp$pm10)
```



Séries temporais

```
plot(who_sp$year, who_sp$pm10,  # Dado eixo x, dado eixo y
     t = "l", # tipo linha
     lwd = 2.5, # largura da linha
     col = "cadetblue", # cor da linha
     ylim = c(25, 40), # limite do eixo x
     xlab = "Anos",  # nome eixo x
     ylab = "PM10 [ug/m3]", # nome eixo y
     main = "São Paulo" # título do plot
     )
```

Séries temporais



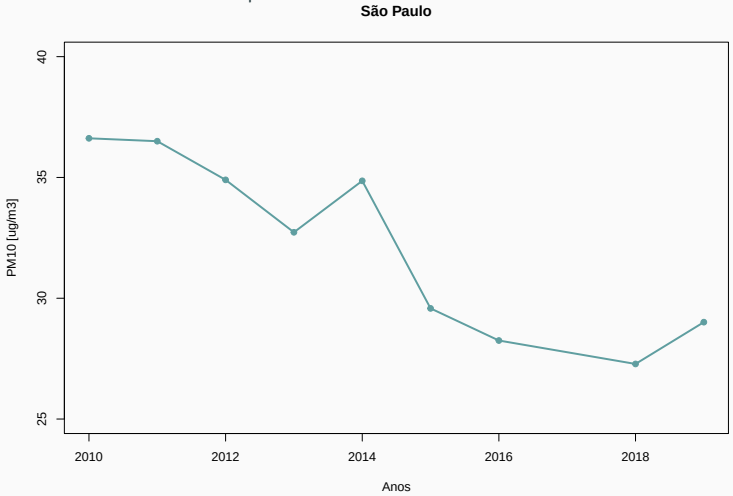
Séries temporais

- Podemos adicionar pontos

```
plot(who_sp$year, who_sp$pm10,  # Dado eixo x, dado eixo y
     t = "l", # tipo linha
     lwd = 2.5, # largura da linha
     col = "cadetblue", # cor da linha
     ylim = c(25, 40), # limite do eixo x
     xlab = "Anos",  # nome eixo x
     ylab = "PM10 [ug/m3]",#
     main = "São Paulo"
     )
points(who_sp$year, who_sp$pm10,
       pch = 19, # tipo do ponto
       col = "cadetblue") # cor do ponto
```

Séries temporais

- Podemos adicionar pontos



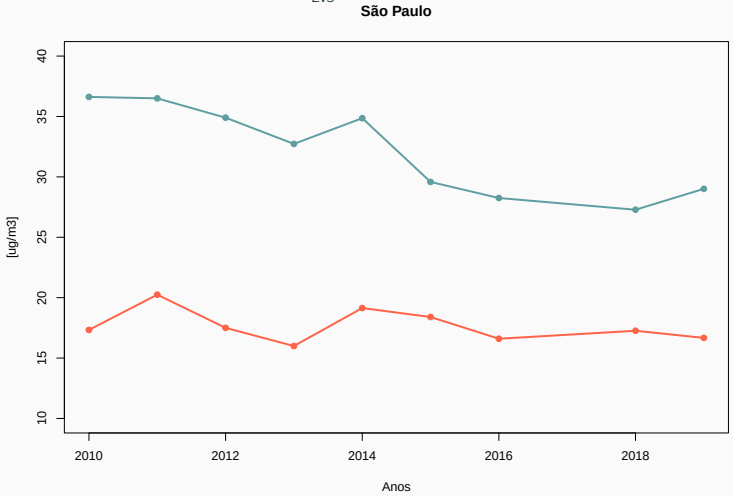
Séries temporais

- Adicionemos também $PM_{2.5}$

```
plot(who_sp$year, who_sp$pm10,  # Dado eixo x, dado eixo y
     t = "l", lwd = 2.5, col = "cadetblue",
     ylim = c(10, 40), # para visualizar valores de PM2.5
     xlab = "Anos", main = "São Paulo",
     ylab = "[ug/m3]") # So unidades
points(who_sp$year, who_sp$pm10, col = "cadetblue", pch = 19 )
# Adicionamos PM.25
lines(who_sp$year, who_sp$pm25, col = "tomato", lwd = 2.5)
points(who_sp$year, who_sp$pm25, col = "tomato", pch = 19 )
```

Séries temporais

- Adicionemos também $PM_{2.5}$



Séries temporais

- Adicionemos também $PM_{2.5}$

```
plot(who_sp$year, who_sp$pm10,t = "l", lwd = 2.5, col = "cadetblue",
     ylim = c(10, 40), xlab = "Anos", main = "São Paulo", ylab = "[ug/m
points(who_sp$year, who_sp$pm10, col = "cadetblue", pch = 19 )
lines(who_sp$year, who_sp$pm25, col = "tomato", lwd = 2.5)
points(who_sp$year, who_sp$pm25, col = "tomato", pch = 19 )
legend("topright", col = c("cadetblue", "tomato"), lwd = 2.5,
       legend = c("PM10", "PM25"))
```

Séries temporais

- Adicionemos também $PM_{2.5}$

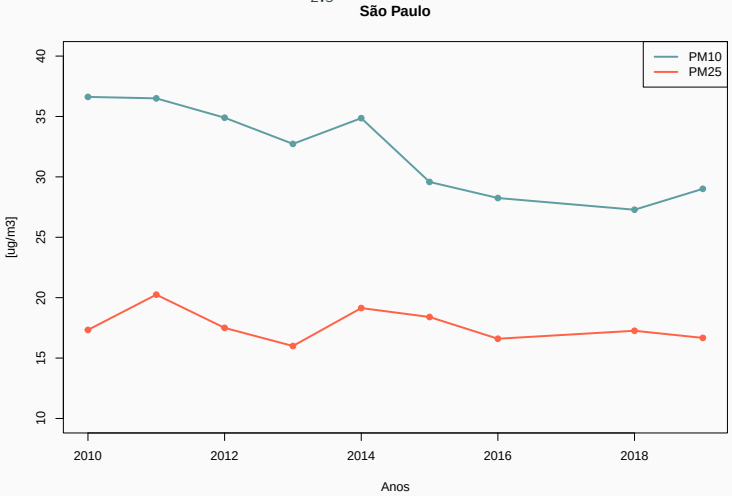


Diagrama de barras.

- Usamos a função `barplot()`.

```
barplot(who_sp$pm10, names.arg = who_sp$year)
```

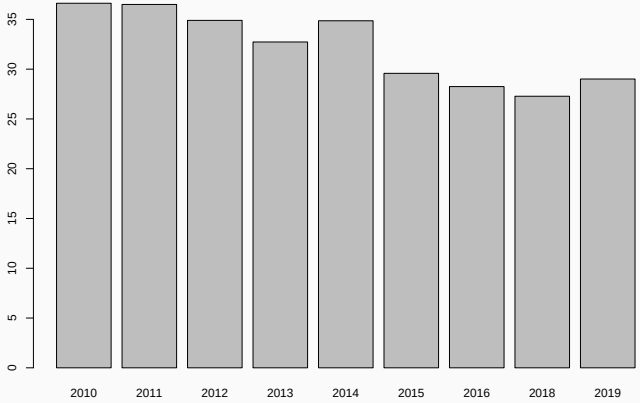


Diagrama de barras.

- Usamos a função `barplot()`.

```
barplot(who_sp$pm10,
        names.arg = who_sp$year, # Nome das barras
        col = "cadetblue", # Color das barras
        border = FALSE,  # Sem borde das barras
        main = "Concentração de PM10 em São Paulo")
```

Diagrama de barras.

- Usamos a função `barplot()`.

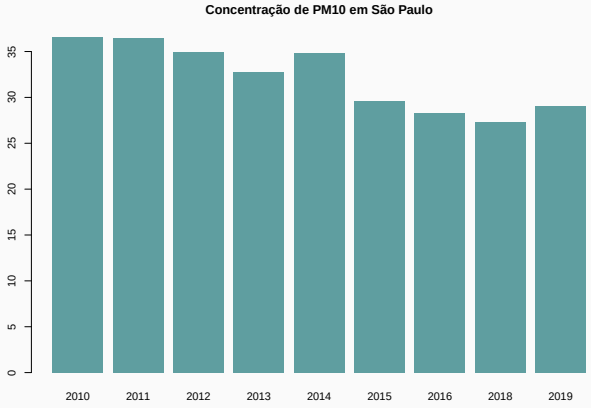


Diagrama de barras.

- Comparemos com $PM_{2.5}$.
- Primeiro temos que transformar nosso dado em matriz.

```
pm <- subset(who_sp, select = c("pm10", "pm25"))
pm_m  <- t(as.matrix(pm))
```

Diagrama de barras.

- Comparemos com $PM_{2.5}$.
- Primeiro temos que transformar nosso dado em matriz.

```
barplot(pm_m, beside = TRUE, border = TRUE,
        col = c("cadetblue", "tomato"),
        names.arg = who_sp$year,
        main = "Concentrações de PM em São Paulo")
legend("topright", legend = c("PM10", "PM2.5"),
       pch = 15, col = c("cadetblue", "tomato"))
```

Diagrama de barras.

- Comparemos com $PM_{2.5}$.
- Primeiro temos que transformar nosso dado em matriz.