

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

openair e R do dia dia

---

Mario Gavidia-Calderón, Rafaela Squizzato, Thiago Nogueira

06/02/2024

Universidade de São Paulo

Média móvel

Combinar data frames

Transformar tipos de objetos

Mapas de estações

Boas praticas do R

Mais recursos.

- Nesta aula vamos falar sobre situações que acontecem quando trabalhamos com dados de qualidade do ar.
- Também falaremos sobre boas praticas de R.
- E resolver as suas dúvidas.

# Média móvel

---

- O padrão qualidade do ar em São Paulo de  $O_3$  é a **média móvel de 8 horas**.
- Outro padrão da **WHO** é a **MDA8** (*average of daily maximum 8-hour*) e também **Peak season**.
- **openair** conta com a função **rollingMean()** para fazer esses padrões.

# Média móvel

- Vamos calcular o padrão de qualidade do ar de  $O_3$  para CETESB no 2021.

```
library(openair)
# Lendo o arquivo
pin <- readRDS('../../data/pin_openair_ex.rds')
pin_2021 <- selectByDate(pin, year = 2021)
# Média móvel
pin_2021 <- rollingMean(
  pin_2021,
  pollutant = 'o3',
  width = 8,
  new.name = "o3_8h",
  data.thresh = 0.75)
```

# Média móvel

- Foi criada a coluna **o3\_8h**.

##		date	aqi	wd	pm25	no	no2	o3	ws	o3_8h
## 1	2021-01-01	00:00:00	Pinheiros	346	NA	0	1	27	1.7	NA
## 2	2021-01-01	01:00:00	Pinheiros	331	NA	0	0	28	1.2	NA
## 3	2021-01-01	02:00:00	Pinheiros	338	NA	2	10	24	1.1	NA
## 4	2021-01-01	03:00:00	Pinheiros	NA	NA	0	11	18	0.0	24.25
## 5	2021-01-01	04:00:00	Pinheiros	272	NA	1	17	13	0.8	22.00
## 6	2021-01-01	05:00:00	Pinheiros	359	NA	1	12	20	1.2	23.25

## Exercício 1

A MDA8 é a média móvel **máxima diária**. Como seria calculada?



## Script do exercício 1: MDA8

```
library(openair)
pin_mda8 <- timeAverage(pin_2021[c("date", "o3_8h")],
                        avg.time = "day", # Diária
                        statistic = "max") # Máxima
head(pin_mda8)
```

```
## # A tibble: 6 x 2
##   date                o3_8h
##   <dtm>              <dbl>
## 1 2021-01-01 00:00:00  52.4
## 2 2021-01-02 00:00:00  49.6
## 3 2021-01-03 00:00:00  62.4
## 4 2021-01-04 00:00:00  67.9
```

## Exercício 2

Quantos dias foi superado o padrão de  $O_3$  na estação Pinheiros?

## Script exercício 2

Podemos usar `subset`. O padrão é  $130 \mu\text{gm}^{-3}$ .

```
dias_o3 <- subset(pin_mda8,  
                  subset = o3_8h >= 130)  
print(  
  paste("O padrão foi superado ", nrow(dias_o3), "dias")  
)
```

```
## [1] "O padrão foi superado  3 dias"
```

## Combinar data frames

---

- Muitas vezes precisamos combinar duas tabelas que tem uma coluna comun. Por exemplo se temos uma tabela com a média anual de  $O_3$  das estações e outra tabela com a média anual de  $PM_{2.5}$ , para isso usamos `merge()`.

## Combinar data frames

```
o3 <- data.frame(aqs = c("pin", "ibu", "usp", "fsp"),
                 o3 = runif(4, 0, 160))
pm25 <- data.frame(aqs = c("pin", "ibu", "usp", "fsp"),
                   pm25 = runif(4, 10, 60))
dados <- merge(o3, pm25)
dados
```

```
##   aqs      o3    pm25
## 1 fsp   5.912787 46.30582
## 2 ibu 157.286383 33.39912
## 3 pin 146.039535 59.98452
## 4 usp   2.972991 57.09195
```

## Completar dados faltantes

- Podemos usar o `merge` para completar com `NA` se um data frame não tem uma linha de dados.
- Precisamos adicionar o argumento `all = TRUE`.
- Serve para detectar quantos dados faltantes existem na nossa base de dados.

## Completar dados faltantes

```
o3 <- data.frame(aqs = c("pin", "ibu", "usp", "fsp"),
                 o3 = runif(4, 0, 160))
pm25 <- data.frame(aqs = c("pin", "ibu", "usp"),
                   pm25 = runif(3, 10, 60))
dados <- merge(o3, pm25, all = TRUE)
dados
```

```
##   aqs      o3    pm25
## 1 fsp  61.78832    NA
## 2 ibu 119.38911 40.71834
## 3 pin 142.50880 10.68583
## 4 usp 106.35466 58.47673
```



# Transformar tipos de objetos

---

## character para numeric

- Às vezes quando usamos `read.table` ou `read_excel`, se uma coluna tem o dado faltante como um character especial (e.g “M”). Para poder operar precisamos forçar a transformação para `numeric`.

## character para numeric

```
o3 <- c(90, 89, 76, 83, "M")  
class(o3)
```

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

```
o3 <- as.numeric(o3) # Atualizamos o valor do vetor o3
```

```
## Warning: NAs introduced by coercion
```

```
class(o3)
```

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

```
o3
```

## Transformar zona horária

- Fontes de dados globais podem ter a zona horária em UTC. Você precisa trazer para horário local.

```
# Criamos uma base de dados
```

```
date <- seq(as.POSIXct("2024-01-01 00:00", tz = "UTC"),  
            length.out = 30 * 24, by = "hour")
```

```
o3 <- runif(length(date), 100, 160)
```

```
df <- data.frame(date, o3)
```

```
head(df$date)
```

```
## [1] "2024-01-01 00:00:00 UTC" "2024-01-01 01:00:00 UTC"
```

```
## [3] "2024-01-01 02:00:00 UTC" "2024-01-01 03:00:00 UTC"
```

```
## [5] "2024-01-01 04:00:00 UTC" "2024-01-01 05:00:00 UTC"
```

## Transformar zona horária

```
attributes(df$date)$tzone <- "America/Sao_Paulo"  
head(df$date)
```

```
## [1] "2023-12-31 21:00:00 -03" "2023-12-31 22:00:00 -03"  
## [3] "2023-12-31 23:00:00 -03" "2024-01-01 00:00:00 -03"  
## [5] "2024-01-01 01:00:00 -03" "2024-01-01 02:00:00 -03"
```

## Mapas de estações

---

Vamos plotar a média mensal das estações de qualidade do ar num mapa.

## Calculando as médias

```
# Lendo o arquivo
```

```
aqs_o3_pm25 <- read.table("../..//data/o3_pm25_pin_ibi_usp_pico.csv", he  
                        sep = ",")
```

```
aqs_o3_pm25$date <- as.POSIXct(aqs_o3_pm25$date)
```



## Calculando as médias

```
aqs_o3_pm25_mean <- aggregate(aqs_o3_pm25[c("o3", "pm25")],  
                               aqs_o3_pm25["aqs"],  
                               mean,  
                               na.rm = TRUE)
```

```
aqs_o3_pm25_mean
```

```
##           aqs      o3      pm25  
## 1 Cid.Universitaria-USP-Ipen 52.42589  9.997187  
## 2           Ibirapuera 51.17599 10.530137  
## 3           Pico do Jaragua 53.35484  7.469477  
## 4           Pinheiros 33.43814  8.991859
```

## Calculando as médias

- Adicionando as latitude e longitudes

```
library(qualR)
# Colocando latitude e longitude no noso dataframe
aq5 <- c(99, 83, 95, 284)
lats_lons <- cetesb_aqs[cetesb_aqs$code %in% aq5, ]
aq5_o3_pm25_mean$lon <- lats_lons$lon
aq5_o3_pm25_mean$lat <- lats_lons$lat
```

## Ler mapa da Região Metropolitana de São Paulo

```
library(rgdal) # Para ler shapefiles
library(broom) # Para transformar shapefile to data frame
library(ggplot2) # Para fazer o mapa

rmisp <- readOGR("../../data/rmisp_shp/rmisp_ll.shp")

## OGR data source with driver: ESRI Shapefile
## Source: "/home/quishqa/R_tests/Curso_R_openair_FSP_USP/data/rmisp_shp"
## with 39 features
## It has 9 fields

rmisp_df <- tidy(rmisp, data = "name_mn") # Transforma o shp em dataframe

## Regions defined for each Polygon
```

## Fazendo o mapa

```
# Making the plot
```

```
ggplot() +
```

```
  geom_path(data = rmsp_df, aes(x = long, y = lat, group = group)) +
```

```
  geom_point(data = aqs_o3_pm25_mean, aes(x = lon, y = lat, color = o3)) +
```

```
  scale_color_viridis_c(option = "viridis", direction = -1,  
                        name=expression("(" * mu * "g m" ^{-3} * ")")) +
```

```
  theme_bw() + # Remove grid and remove that grey background
```

```
  coord_quickmap() + # Scale X and Y based on lon and lat
```

```
  labs(title = "Concentração de Ozônio em Janeiro 2024",
```

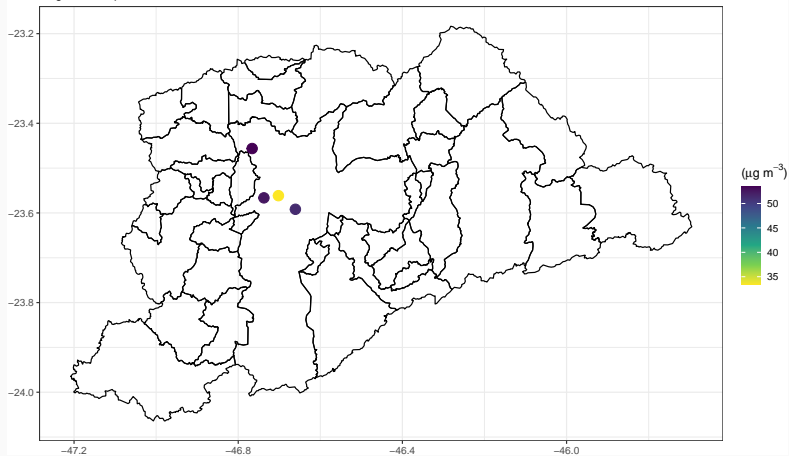
```
        subtitle = "Região Metropolitana de São Paulo",
```

```
        x = "", y = "")
```

# Fazendo o mapa

Concentração de Ozônio em Janeiro 2024

Região Metropolitana de São Paulo



## Boas praticas do R

---

# Boas praticas do R

- Para cada trabalho usar RStudio Projects:
  - Mais fácil de compartilhar. Não precisa definir o caminho dos inputs.
- É bom olhar guias de estilo do R.
  - Google R guide:
    - Usar `<-` para definir variáveis.
    - Usar `=` para os argumentos.
    - Usar espaços para separar argumentos e `=`.
    - Usar nome significativos das variáveis.

Mais recursos.

---



- Dados dos aeroportos: **riem** package
- Dados do brasil geobr **geobr** package
- Dados qualidade do ar São Paulo: **qualR** package
- Dados DATASUS: **DATASUS** package
- Dados DATASUS: **microdatasus** package
- Dados de qualidade do ar: *openaq*

- Galeria de figuras no R: **The R Graph Gallery**
- RStudio folha de dicas: **RStudio cheat sheet**

Muito Obrigado!