Minha primeira aula do `openair`

Mario Gavidia-Calderón
Rafaela Squizzato
Alejandro Delgado

7/12/2021

O cardápio

1. Do Excel para o R
2. **Lembre da data:** `as.POSIXct()`
3. Figuras legais:
   - `summaryPlot()`
   - `timePlot()`
   - `windRose()`
   - `PollutionRose()`
   - `timevariation()`
4. Funções para alegrar a vida:
   - `timeAverage`
   - `selectByDate`
   - `splitByDate`

Do Excel para o R

- Existe o pacote **`readxl`** para ler os `.xls` dentro do R.
- Eu não recomendo >:(
- **Melhor salvar o arquivo como `.csv` e ler o `.csv` no R. :)**
- Aproveitar para trocar nomes das columnas:
  - tc > T (ºC)
  - rh > Umidade relativa (%)
- Serve para Excel, Google Sheet, Libre Office, etc

Do Excel para o R

- Exemplo com dados de Southport - Australia.
- Dados de **openaq.org**
- Procesados para nosso curso.
  - Maior informação em `02_preparing_example.R`

Do Excel para o R

```
au <- read.csv("../03_output/au_df_example.csv",
               sep = ",", dec = ".", header = T,
               na.strings = -9999)
head(au, 5)
```

```
##                       date        id    o3 pm10 pm25
## 1 2021-07-10T11:00:00+10:00 Southport 0.023  4.5  3.1
## 2 2021-07-10T11:00:00+10:00 Southport 0.019  4.5  3.1
## 3 2021-07-10T11:00:00+10:00 Southport 0.018  4.6  3.2
## 4 2021-07-10T10:00:00+10:00 Southport 0.018  4.7  3.4
## 5 2021-07-10T10:00:00+10:00 Southport 0.018  4.7  3.4
```

Lembre da data: `as.POSIXct()`

```
str(au)
```

```
## 'data.frame':    3096 obs. of  5 variables:
##  $ date: chr  "2021-07-10T11:00:00+10:00" "2021-07-10T11
##  $ id  : chr  "Southport" "Southport" "Southport" "South
##  $ o3  : num  0.023 0.019 0.018 0.018 0.018 0.016 0.019
##  $ pm10: num  4.5 4.5 4.6 4.7 4.7 4.7 4.5 4.4 4.5 4.5 .
##  $ pm25: num  3.1 3.1 3.2 3.4 3.4 3.4 3.2 3.2 3.2 3.2 .
```

Lembre da data: `as.POSIXct()`

- Temos que dizer para o R que a coluna `date` não é `chr` senão dado de data
- **Dica: `openair` sempre vai procurar a coluna `date`**

```
au$date <- as.POSIXct(
  strptime(au$date,
           format = "%Y-%m-%dT%H:%M:%S+10:00"),
  tz = "Etc/GMT+10"
  )
```

Lembre da data: `as.POSIXct()`

Agora sim, já podemos usar `openair`

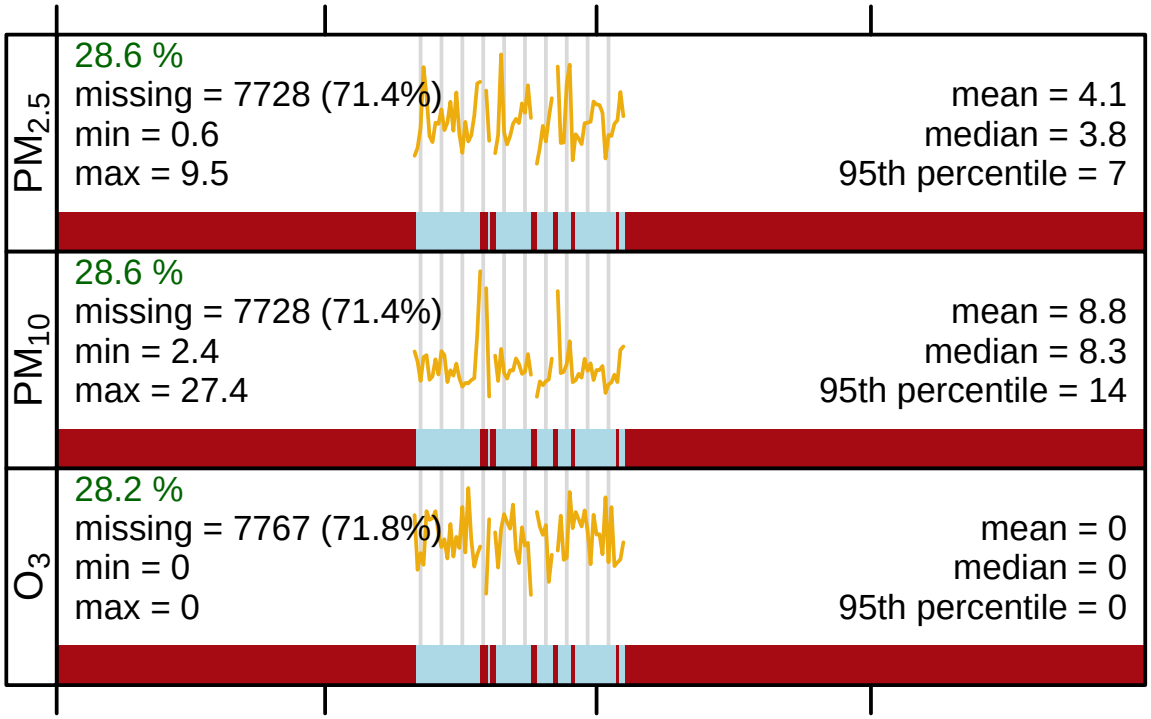
```
str(au)
```

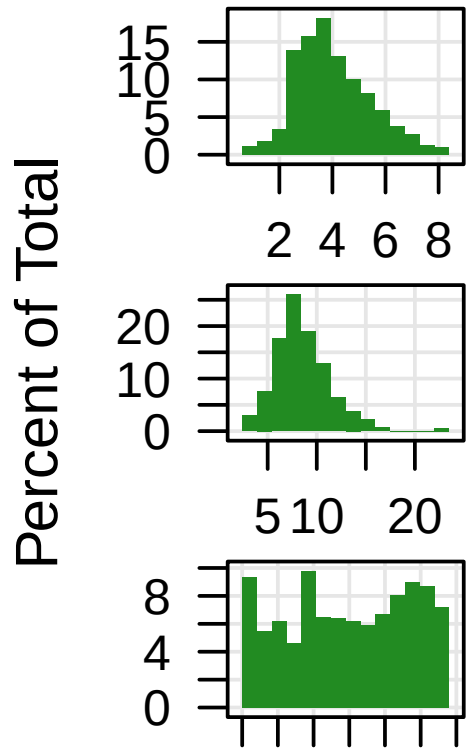
```
## 'data.frame':    3096 obs. of  5 variables:
##  $ date: POSIXct, format: "2021-07-10 11:00:00" "2021-07
##  $ id  : chr  "Southport" "Southport" "Southport" "South
##  $ o3  : num  0.023 0.019 0.018 0.018 0.018 0.016 0.019
##  $ pm10: num  4.5 4.5 4.6 4.7 4.7 4.7 4.5 4.4 4.5 4.5 .
##  $ pm25: num  3.1 3.1 3.2 3.4 3.4 3.4 3.2 3.2 3.2 3.2 .
```

summaryPlot()

```
library(openair)
summaryPlot(au)
```

```
##       date1       date2          id          o3          p
##   "POSIXct"   "POSIXt" "character"   "numeric"   "numer
```

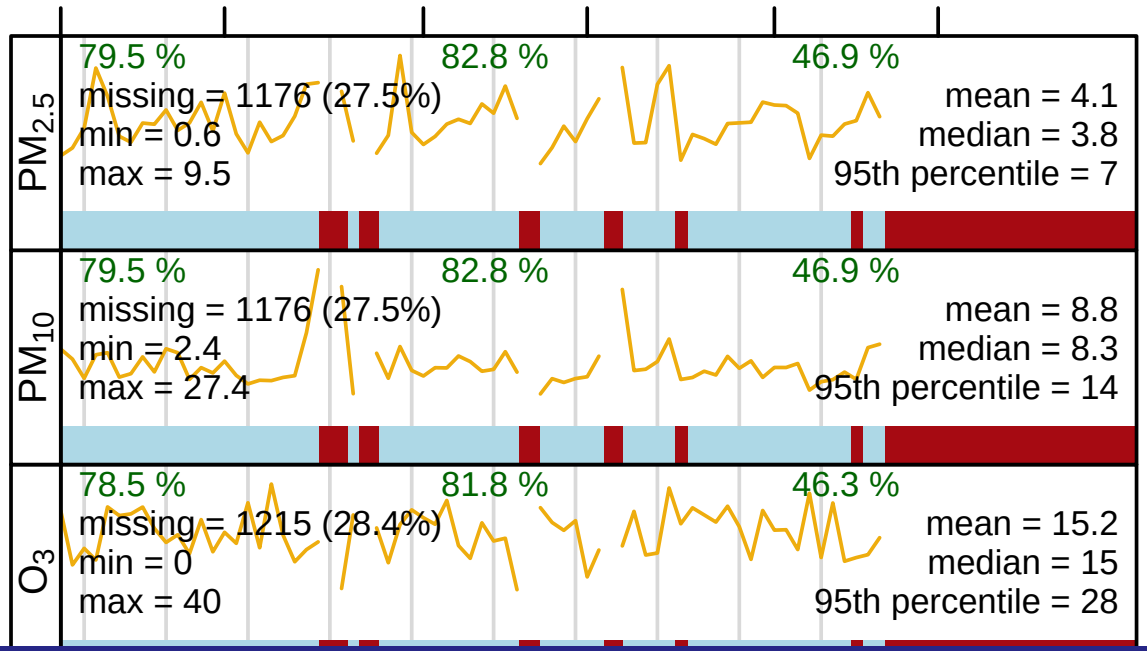


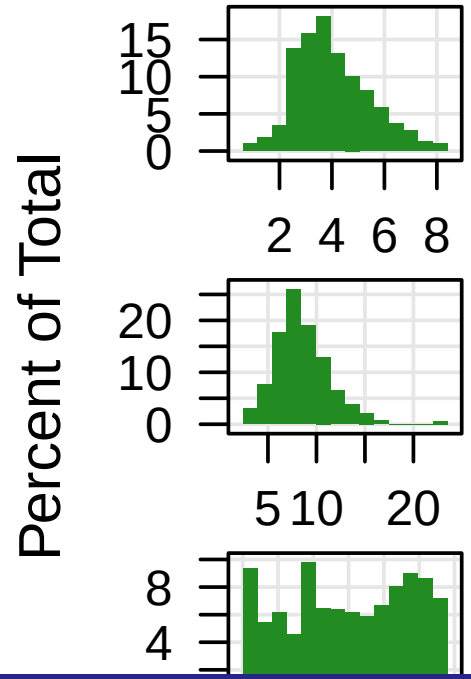


summaryPlot()

```
library(openair)
au$o3 <- au$o3 * 1000
summaryPlot(au, period="months")
```

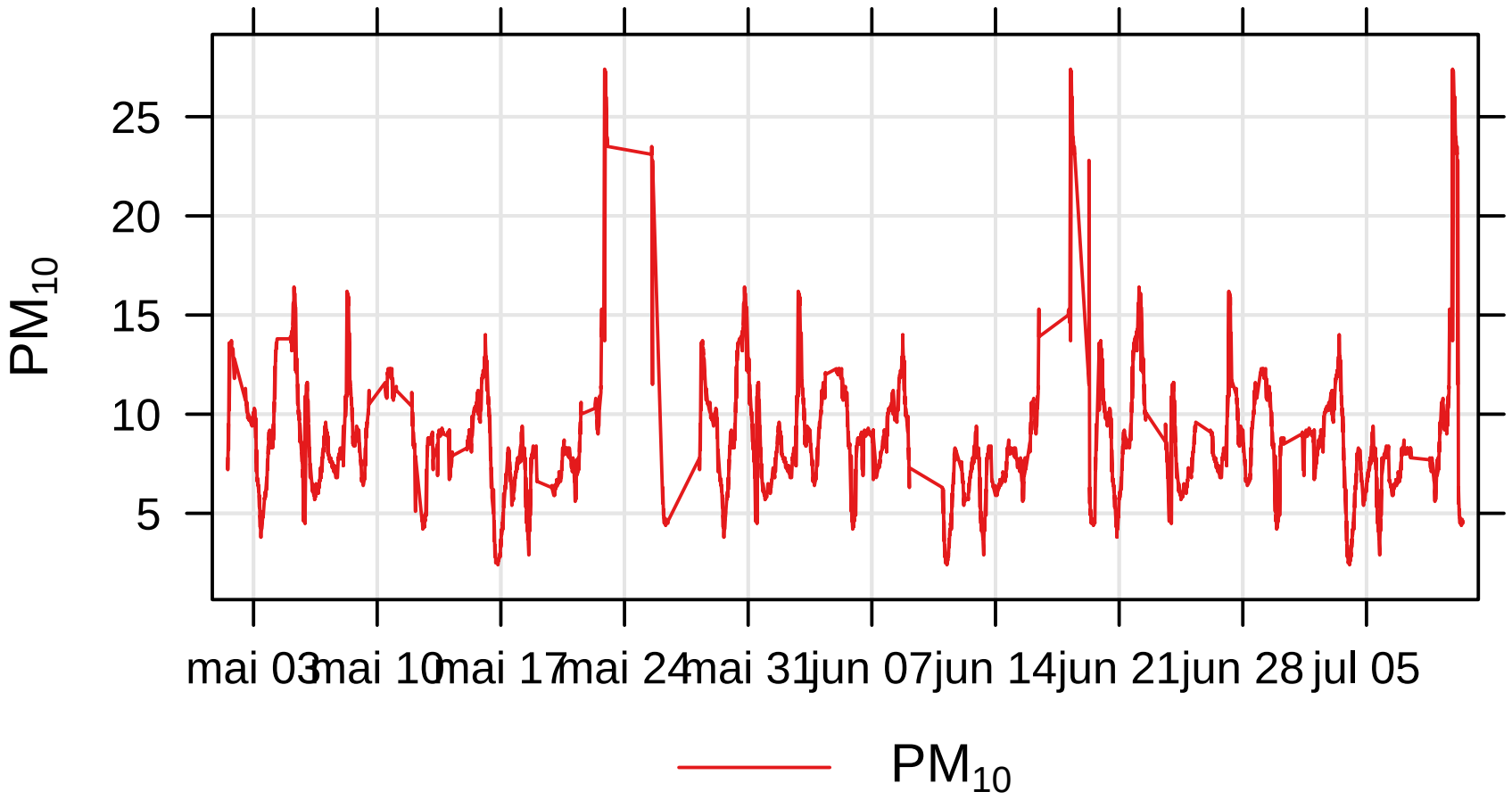
```
##       date1       date2          id          o3           p
##   "POSIXct"   "POSIXt" "character"   "numeric"   "numer
```





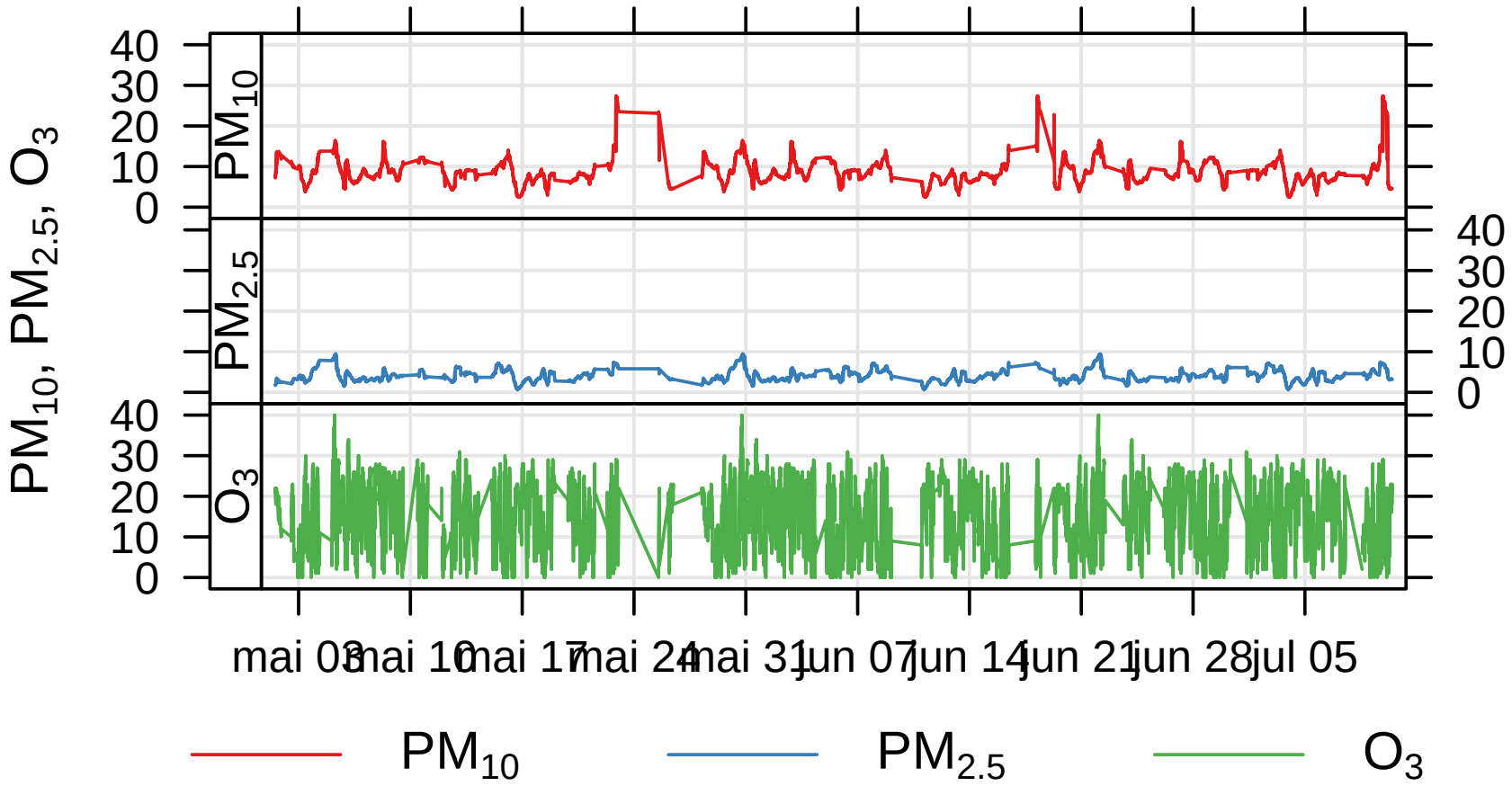
TimePlot()

```
timePlot(au, pollutant="pm10")
```



TimePlot()

```
timePlot(au, pollutant=c("pm10", "pm25", "o3"))
```



Um parentesis

Melhor usar dados de São Paulo

Um parentesis

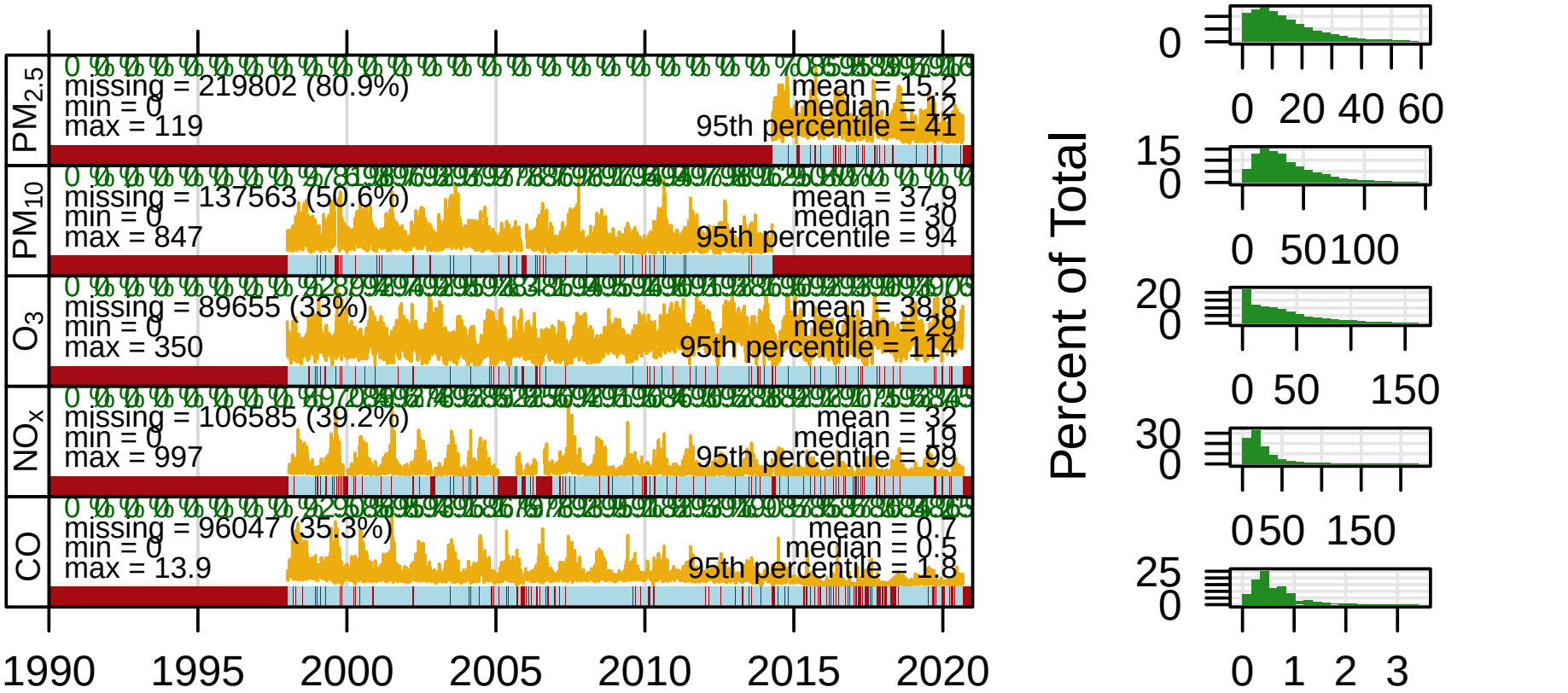
Lendo dados de Ibirapuera 30 anos

```
ibi <- readRDS("../02_data/ibi_30_year_df.RDS")
```

summaryPlot()

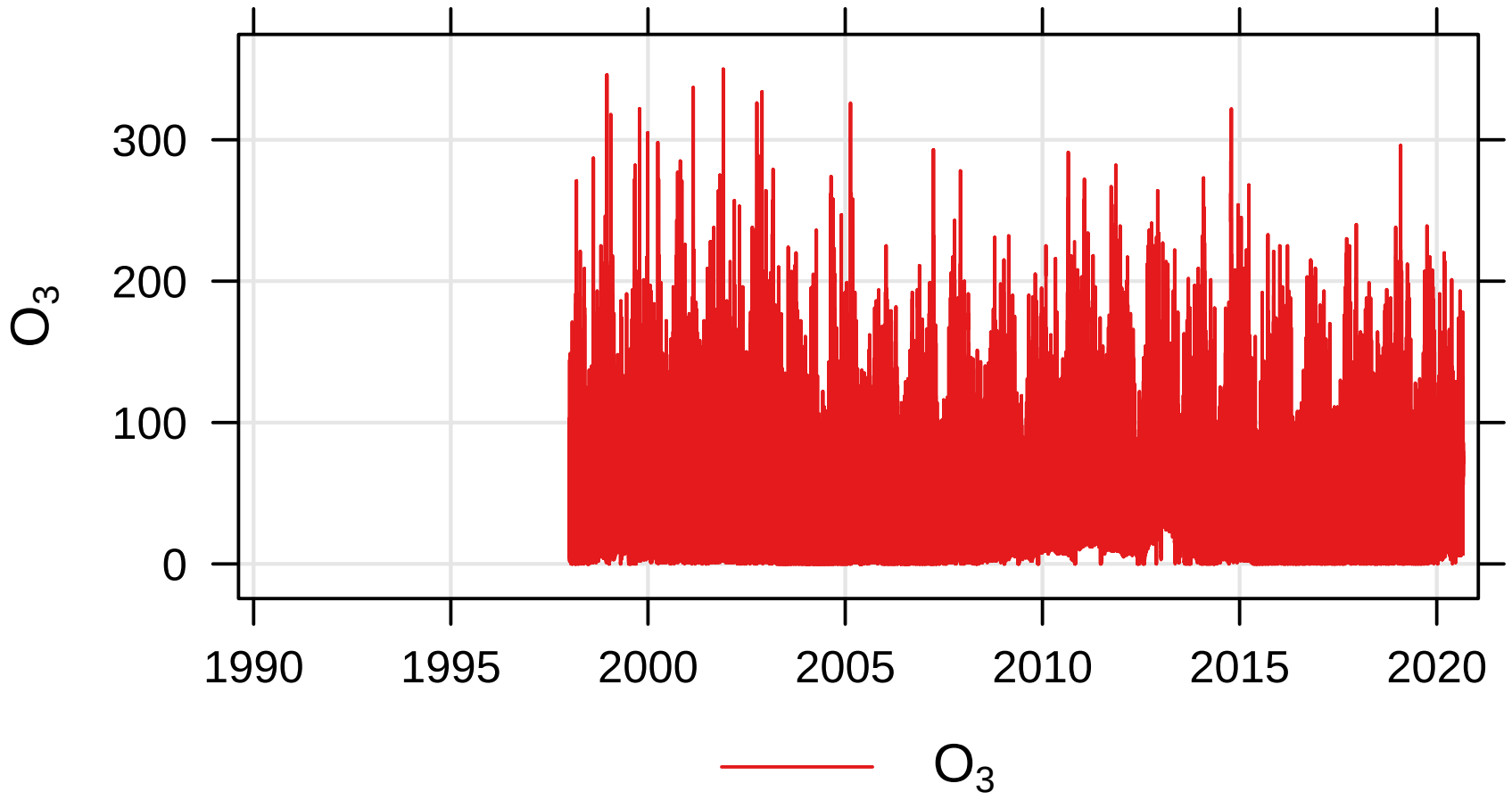
```
summaryPlot(ibi)
```

```
##      date1      date2         co        nox         o3          p
## "POSIXct"  "POSIXt" "numeric" "numeric" "numeric" "numeric" "numer
```



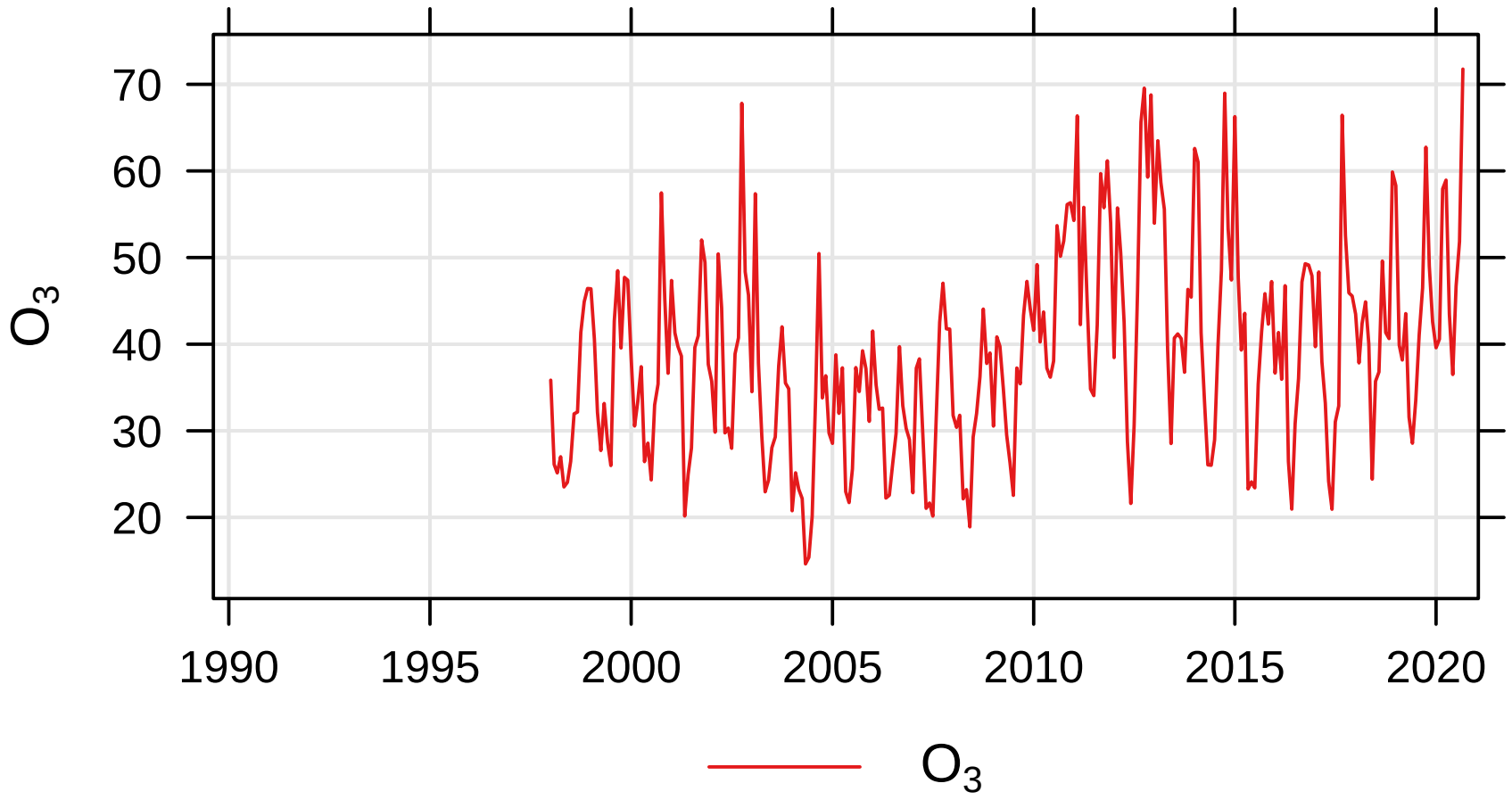
timePlot()

```
timePlot(ibi, pol = "o3")
```



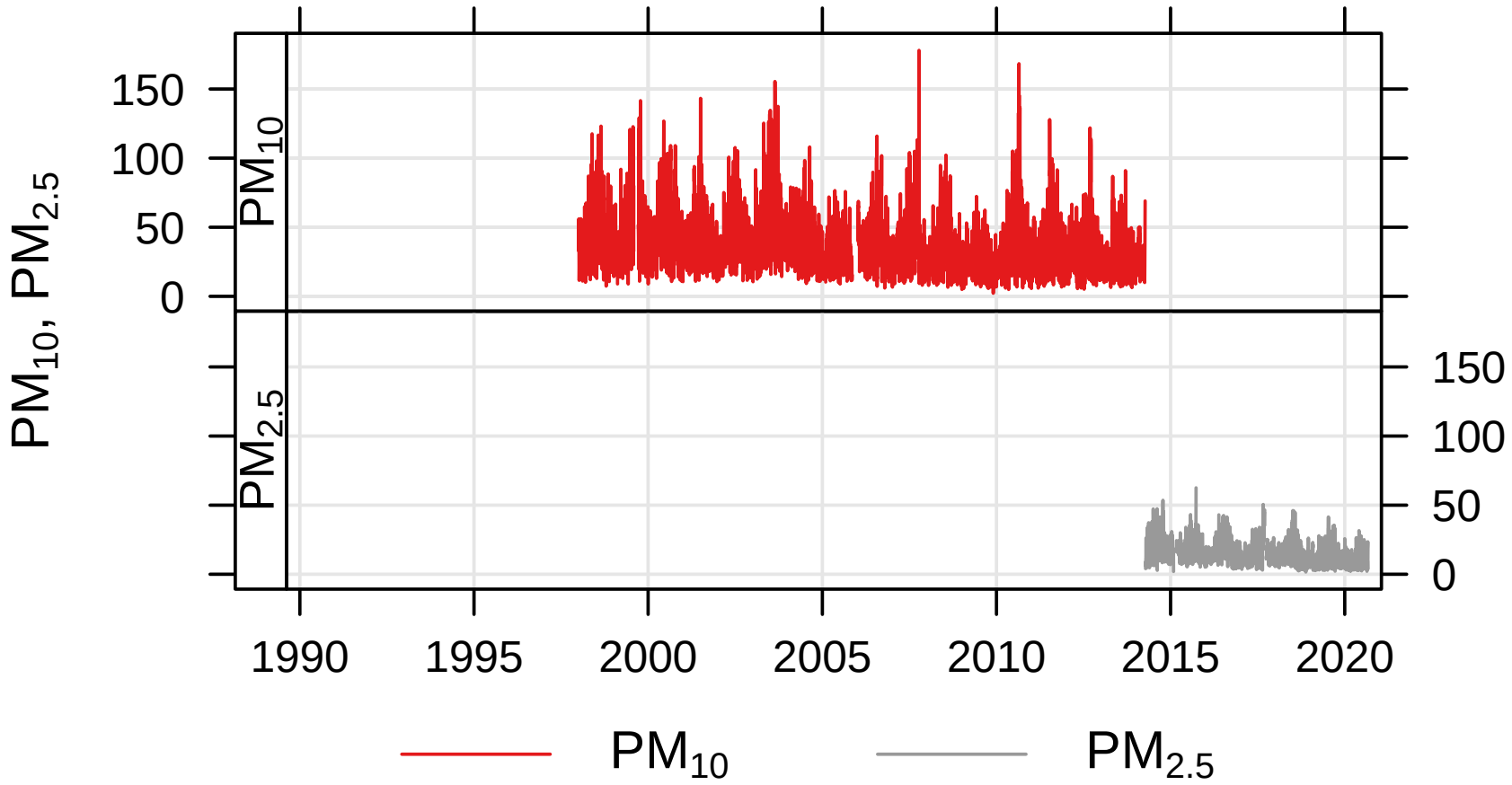
timePlot()

```
timePlot(ibi, pol = "o3", avg.time = "month")
```



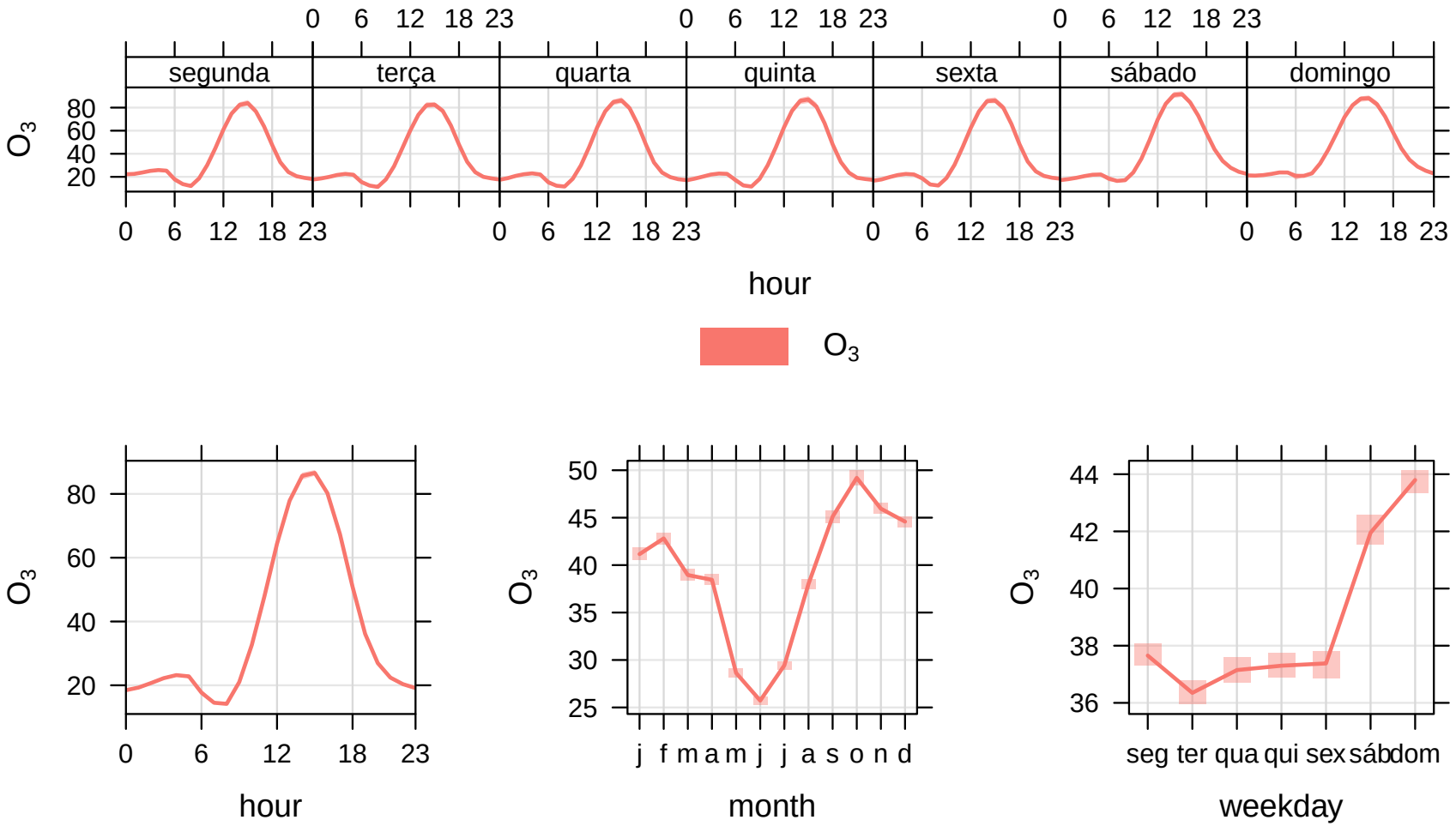
timePlot()

```
timePlot(ibi, pol = c("pm10", "pm25"), avg.time = "day")
```



timeVariation()

```
timeVariation(ibi, pol = "o3")
```



WindRose

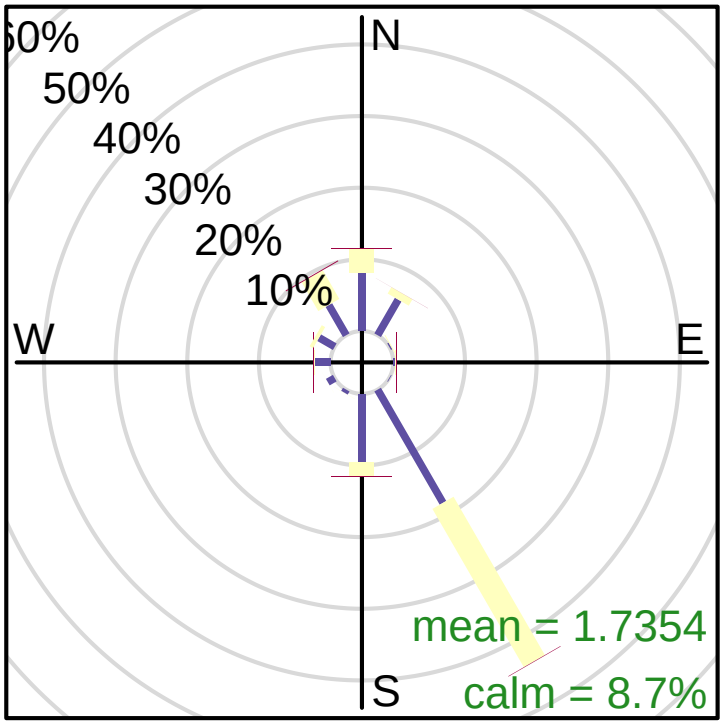
```
library(qualR)

params <- c("MP10", "MP2.5", "NOx", "VV", "DV")
pin <- CetesbRetrieveParam(Sys.getenv("QUALAR_USER"),
                           Sys.getenv("QUALAR_PASS"),
                           params,
                           "Pinheiros",
                           "01/01/2020",
                           "31/12/2020")

saveRDS(pin, file="../02_data/pin_pol_with_wind.Rds")
```

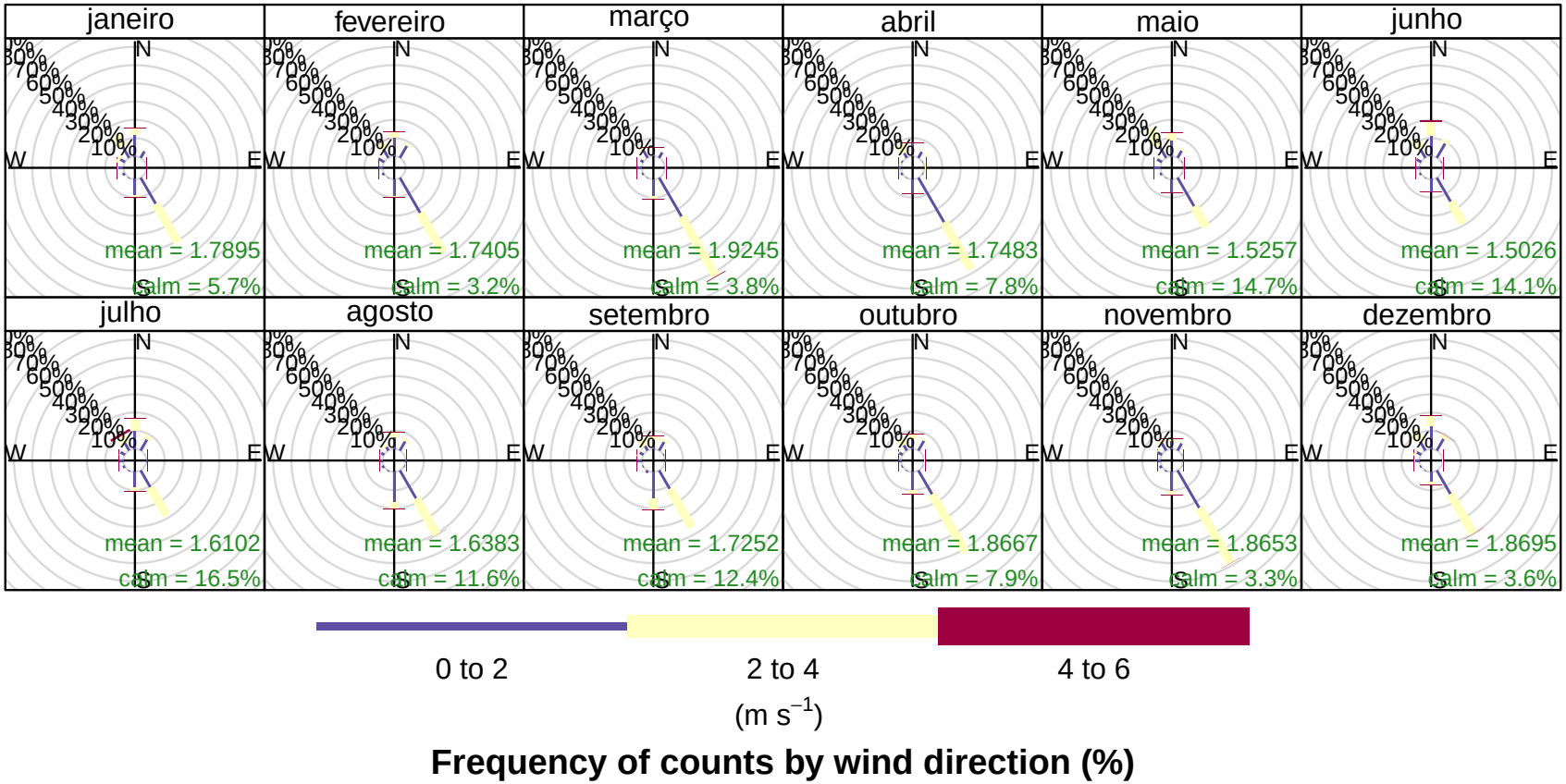
windRose()

```
pin <- readRDS("../02_data/pin_pol_with_wind.Rds")
wr <- windRose(pin, ws="ws", wd="wd")
```



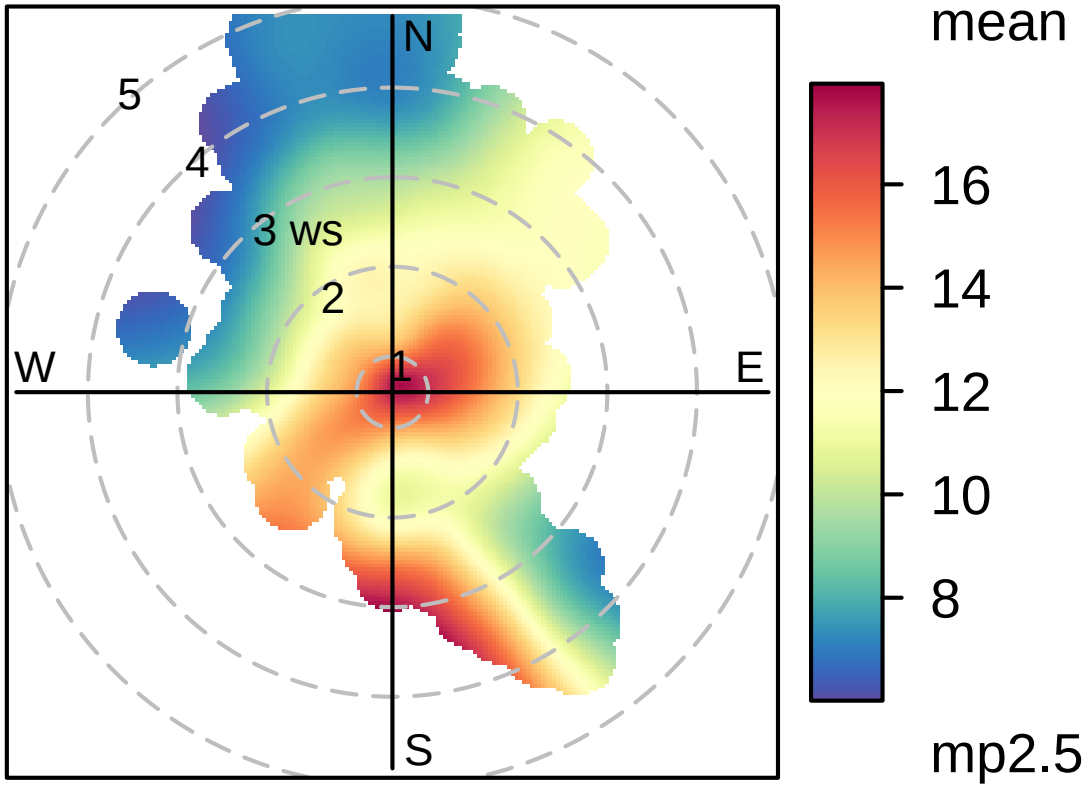
windRose()

```
windRose(pin, ws="ws", wd="wd", type = "month",
                layout = c(6, 2))
```



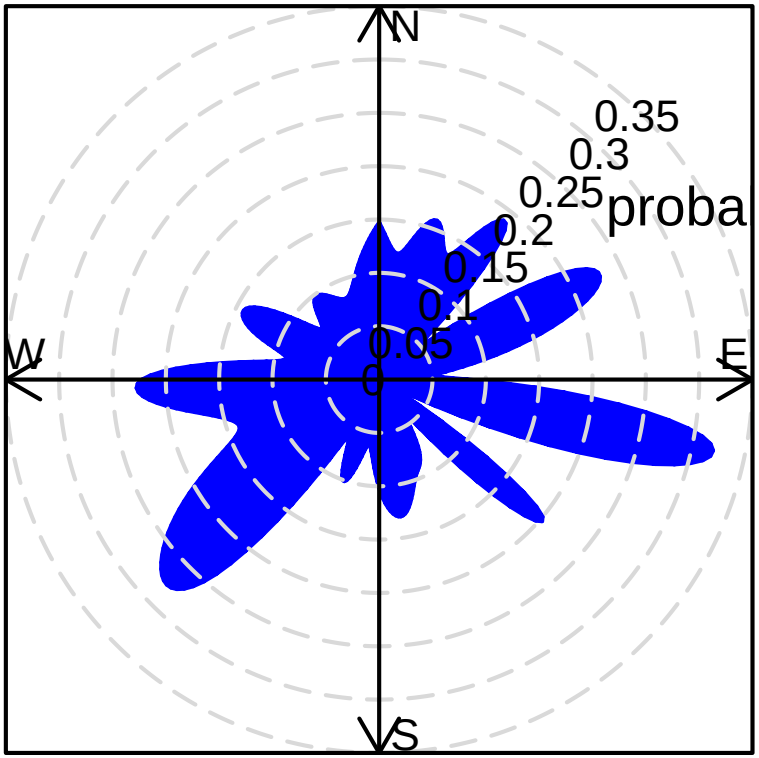
polarPlot()

```
pp <- polarPlot(pin, pollutant = "mp2.5")
```



percentileRose()

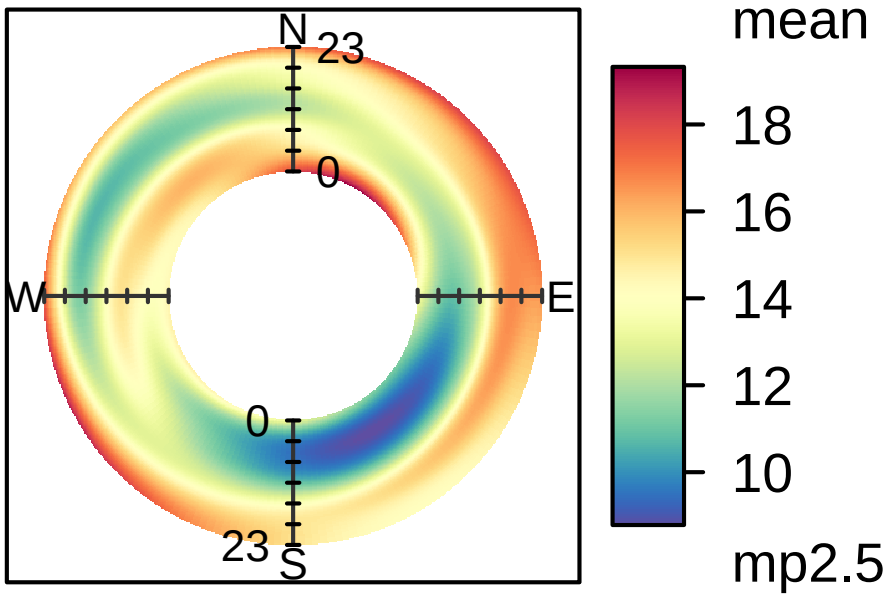
```
pr <- percentileRose(pin, pollutant = "mp2.5", percentile=7
                     method = "cpf", col = "blue", smooth =
```



CPF at the 75th percentile (=18)

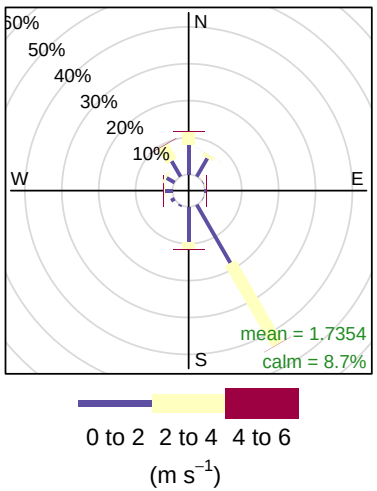
‘polarAnnulus()

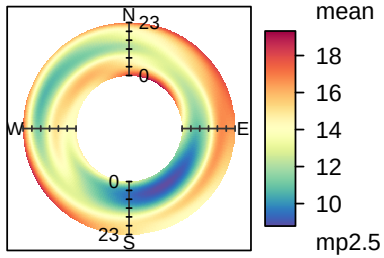
```
pa <- polarAnnulus(pin, pollutant = "mp2.5", period="hour",
                   exclude.missing=F)
```

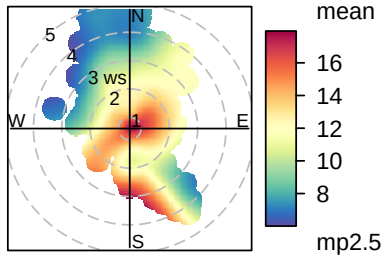


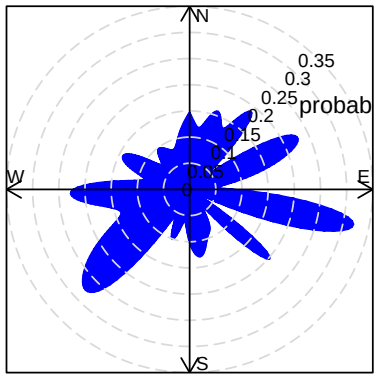
multiplots

```
library(gridExtra)
grid.arrange(wr$plot, pa$plot, pp$plot, pr$plot,
             nrow= 1, ncol = 4)
```







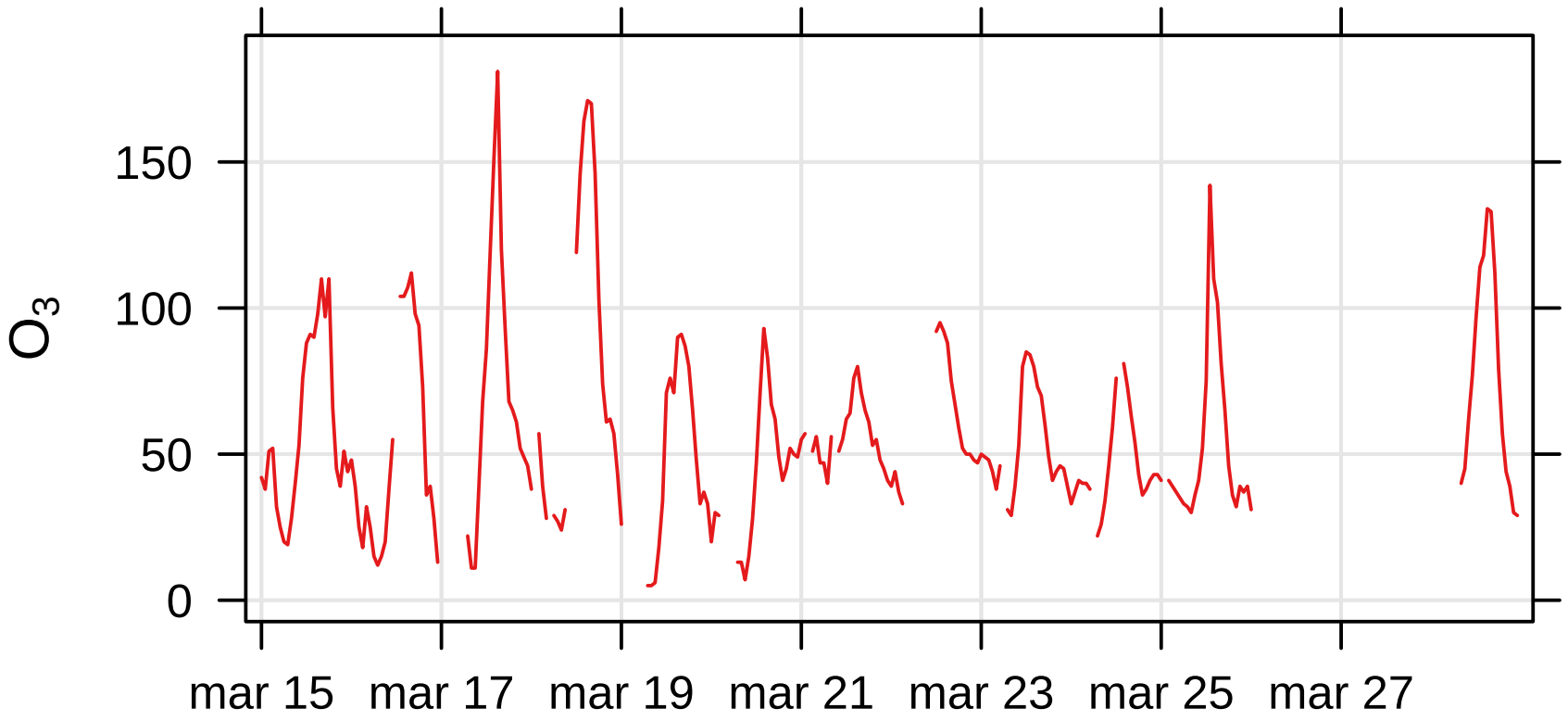


CPF at the 75th percentile (=18)

ıency of counts by wind direction (%)

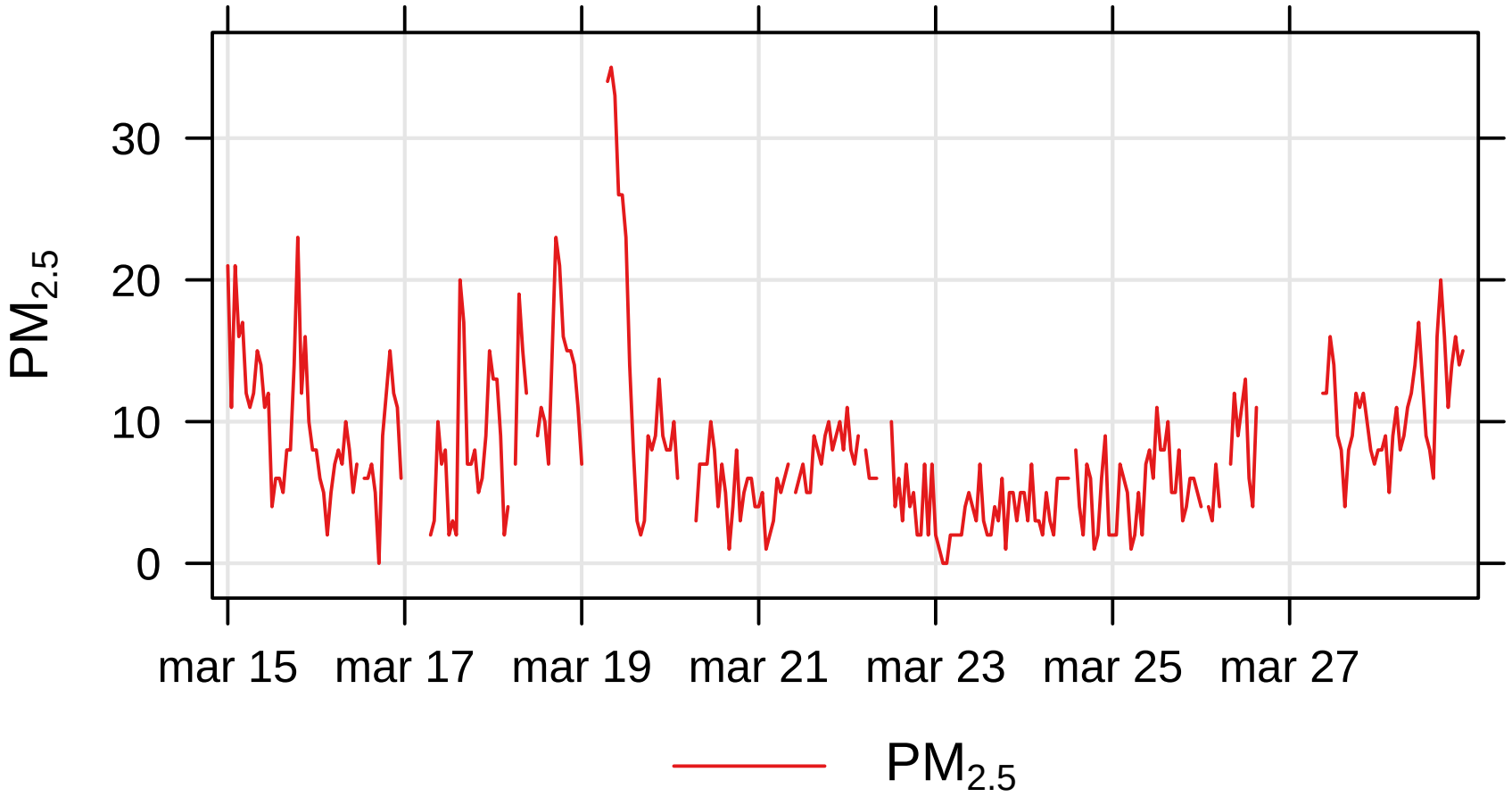
SelectByDate

```
covid <- selectByDate(ibi,
                      start="15/3/2020",
                      end="28/3/2020")
timePlot(covid, pol = "o3")
```



SelectByDate

```
ibi19 <- selectByDate(ibi, year = 2019)
timePlot(covid, pol = "pm25")
```



SelectByDate

```
ibi19_weekday <- selectByDate(ibi19, day="weekday")
ibi19_weekend <- selectByDate(ibi19, day="weekend")
mean(ibi19_weekday$o3, na.rm = T)
```

```
## [1] 41.33666
```

```
mean(ibi19_weekend$o3, na.rm = T)
```

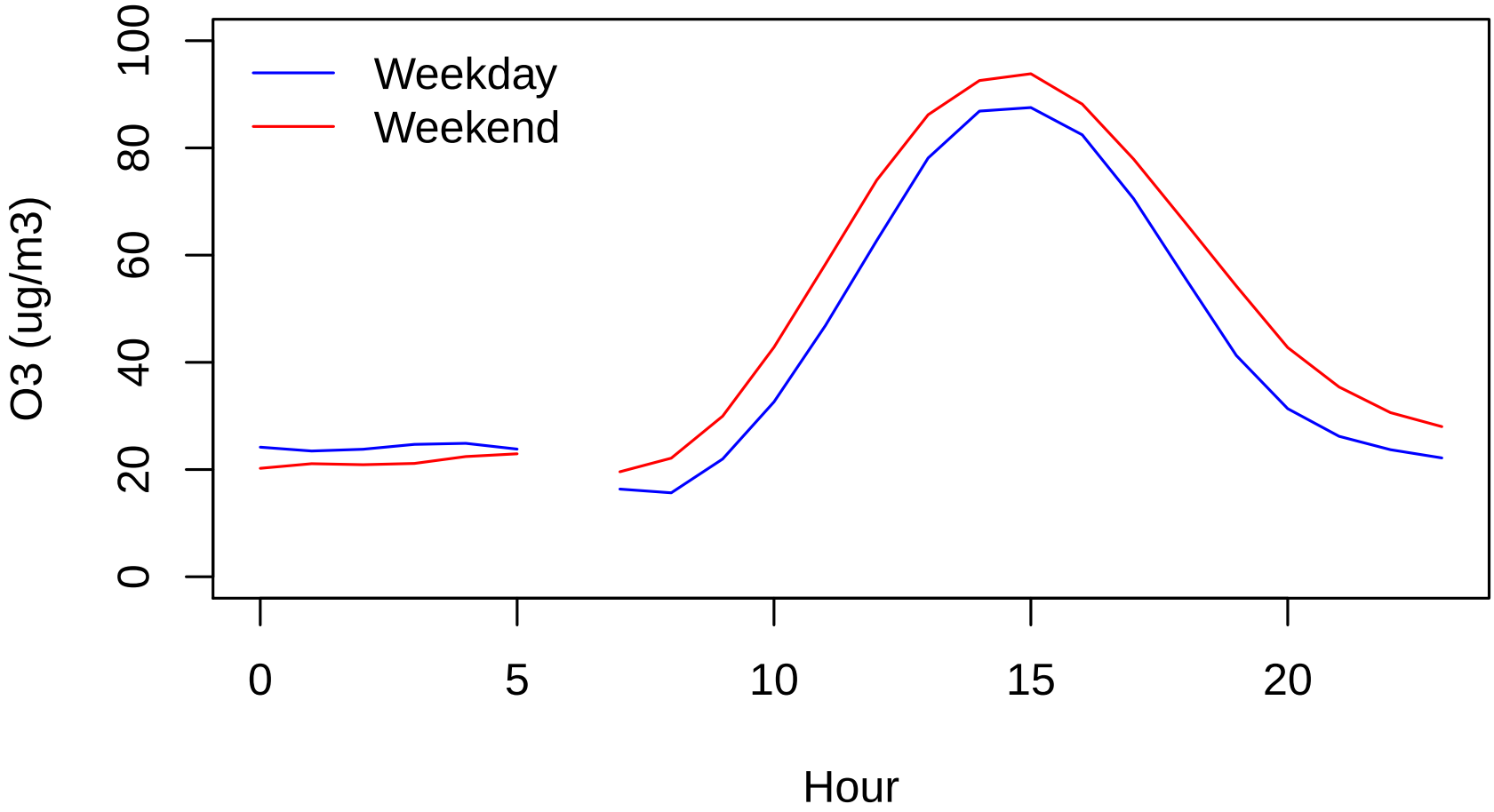
```
## [1] 46.67555
```

SelectByDate

```
weekday_hour <- aggregate(ibi19_weekday["o3"],
                          format(ibi19_weekday["date"], "%H
                          mean, na.rm = TRUE)
weekend_hour <- aggregate(ibi19_weekend["o3"],
                          format(ibi19_weekend["date"], "%H
                          mean, na.rm = TRUE)

plot(weekend_hour$date, weekend_hour$o3, t = "l", col="red"
     ylim = c(0, 100), ylab = "O3 (ug/m3)",
     xlab = "Hour")
lines(weekday_hour$date, weekday_hour$o3, t = "l",
      col = "blue")
legend("topleft", legend=c("Weekday", "Weekend"),
       col=c("blue", "red"),
       lty= c(1,1), bty="n")
```

SelectByDate

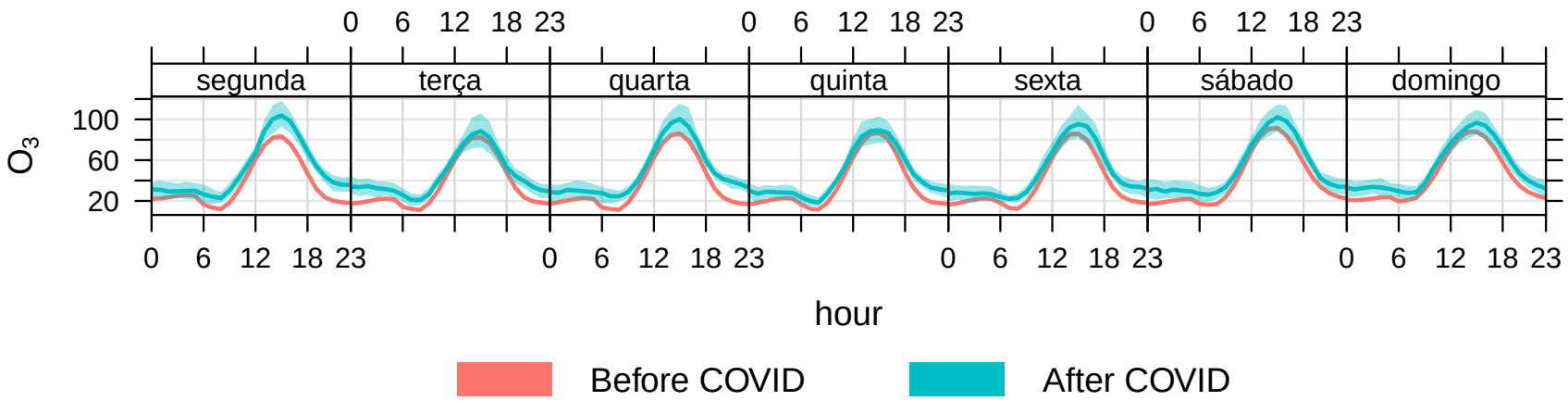


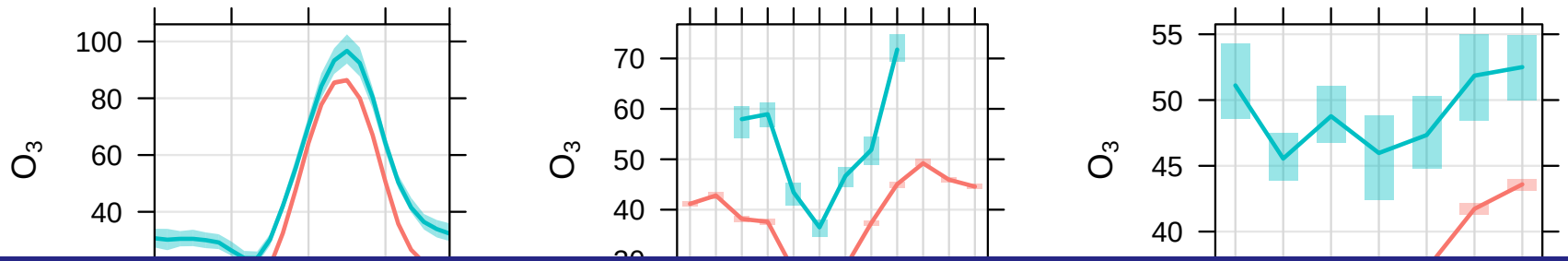
timeAverage

```r
# Média diaria
ibi19_day <- timeAverage(ibi19, avg.time = "day")
# Média mensal
ibi19_month <- timeAverage(ibi19, avg.time = "month")
# Média sasonal
ibi19_season <- timeAverage(ibi19, avg.time = "season")
```

splitByDate

```
ibi_cov <- splitByDate(ibi, dates = "1/3/2020",
                       name = "situation",
                       labels = c("Before COVID", "After C
timeVariation(ibi_cov, pol = "o3", group = "situation")
```





Exemplos mais legais

- Esta presentação foi feita seguindo estes exemplos
- Exemplos SP

Maior informação

- `openair` github repository
- `openair` on-line book
- `openair` paper
- CBPF paper

SOS

- mario.calderon@iag.usp.br
- rafaela.squizzato@iag.usp.br
- adelgado@iag.usp.br