Minha primeira aula do `openair`

Mario Gavidia-Calderón

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.table("../data/au_lytton_05_07_22.csv",
                 sep = ",", # Separador das colunas
                 dec = ".", # Decimal
                 header = T,
                 na.string = -9999) # Se as colunas têm no
head(au, 5)
```

```
##   locationId location                    city country
## 1       5515   Lytton South East Queensland      AU 2022
## 2       5515   Lytton South East Queensland      AU 2022
## 3       5515   Lytton South East Queensland      AU 2022
## 4       5515   Lytton South East Queensland      AU 2022
## 5       5515   Lytton South East Queensland      AU 2022
##                        local parameter value  unit latitu
## 1 2022-07-31T13:00:00+10:00      pm25   3.3 ug/m3 -27.40
```

Para filtrar usamos `subset()`

- 'df_subset <- **subset**(df, nome_colun == "valor")

```
au_pm25 <- subset(au, parameter == "pm25")
head(au_pm25)
```

```
##    locationId location                    city country
## 1        5515   Lytton South East Queensland      AU 202
## 4        5515   Lytton South East Queensland      AU 202
## 6        5515   Lytton South East Queensland      AU 202
## 8        5515   Lytton South East Queensland      AU 202
## 9        5515   Lytton South East Queensland      AU 202
## 11       5515   Lytton South East Queensland      AU 202
##                          local parameter value  unit latit
## 1  2022-07-31T13:00:00+10:00      pm25   3.3 µg/m³ -27.4
## 4  2022-07-31T12:00:00+10:00      pm25   3.5 µg/m³ -27.4
```

Para filtrar usamos `subset()`

- 'df_subset <- **subset**(df, nome_colun == "valor")

```
au_pm25_high <- subset(au_pm25, value >= median(au_pm25$val
head(au_pm25_high)
```

```
##    locationId location                    city country
## 8        5515   Lytton South East Queensland      AU 202
## 13       5515   Lytton South East Queensland      AU 202
## 16       5515   Lytton South East Queensland      AU 202
## 17       5515   Lytton South East Queensland      AU 202
## 19       5515   Lytton South East Queensland      AU 202
## 25       5515   Lytton South East Queensland      AU 202
##                          local parameter value  unit latit
## 8  2022-07-31T10:00:00+10:00      pm25   3.8 µg/m³ -27.4
## 13 2022-07-31T07:00:00+10:00      pm25   3.8 µg/m³ -27.4
```

Lembre da data: `as.POSIXct()`

```
str(au_pm25)
```

```
## 'data.frame':    2019 obs. of  11 variables:
##  $ locationId: int  5515 5515 5515 5515 5515 5515 5515 5
##  $ location  : chr  "Lytton" "Lytton" "Lytton" "Lytton"
##  $ city      : chr  "South East Queensland" "South East
##  $ country   : chr  "AU" "AU" "AU" "AU" ...
##  $ utc       : chr  "2022-07-31T03:00:00+00:00" "2022-07
##  $ local     : chr  "2022-07-31T13:00:00+10:00" "2022-07
##  $ parameter : chr  "pm25" "pm25" "pm25" "pm25" ...
##  $ value     : num  3.3 3.5 3.6 3.8 3.7 3.6 3.8 4 3.8 4
##  $ unit      : chr  "µg/m³" "µg/m³" "µg/m³" "µg/m³" ...
##  $ latitude  : num  -27.4 -27.4 -27.4 -27.4 -27.4 ...
##  $ longitude : num  153 153 153 153 153 ...
```

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_pm25$date <- as.POSIXct(
  strptime(au_pm25$local,
           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_pm25)
```

```
## 'data.frame':    2019 obs. of  12 variables:
##  $ locationId: int  5515 5515 5515 5515 5515 5515 5515 5
##  $ location  : chr  "Lytton" "Lytton" "Lytton" "Lytton"
##  $ city      : chr  "South East Queensland" "South East
##  $ country   : chr  "AU" "AU" "AU" "AU" ...
##  $ utc       : chr  "2022-07-31T03:00:00+00:00" "2022-07
##  $ local     : chr  "2022-07-31T13:00:00+10:00" "2022-07
##  $ parameter : chr  "pm25" "pm25" "pm25" "pm25" ...
##  $ value     : num  3.3 3.5 3.6 3.8 3.7 3.6 3.8 4 3.8 4
##  $ unit      : chr  "µg/m³" "µg/m³" "µg/m³" "µg/m³" ...
##  $ latitude  : num  -27.4 -27.4 -27.4 -27.4 -27.4 ...
```

Lembre da data: `as.POSIXct()`

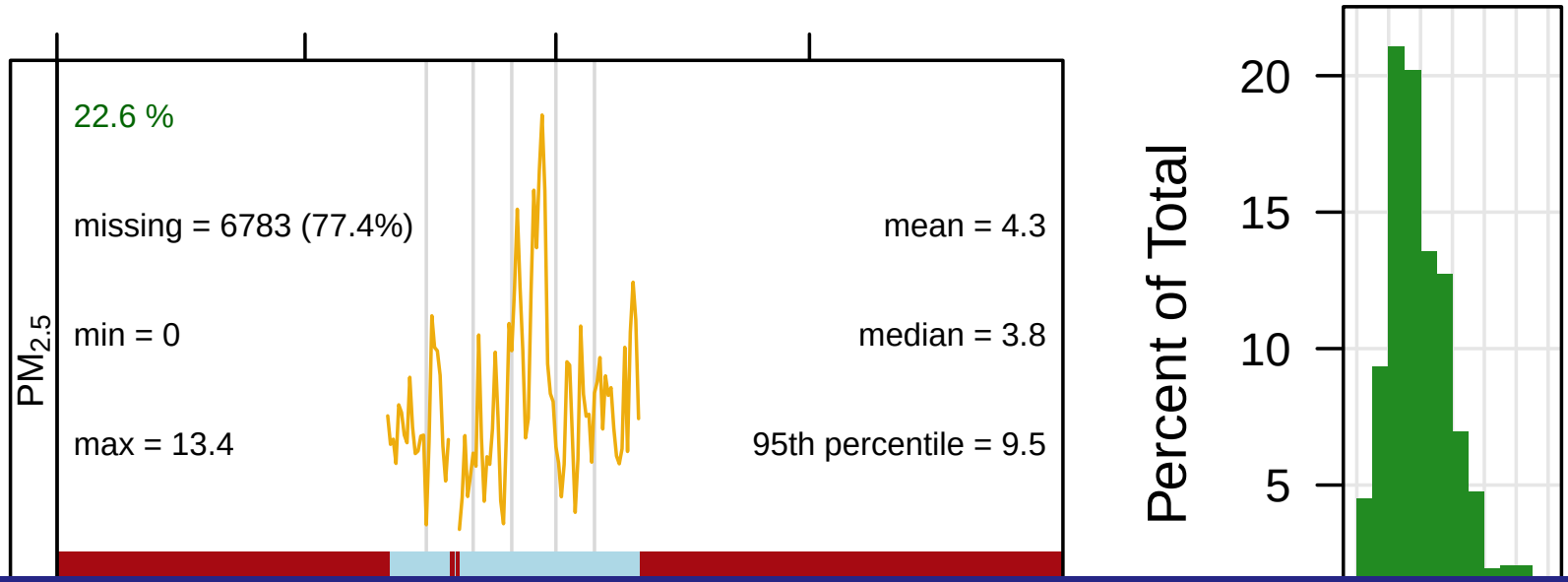
Porém vamos simplificar um pouco as coisas

```
au_pm25_df <- au_pm25[c("date", "value")]
names(au_pm25_df)[2]  <- "pm25"
```

summaryPlot()

```
library(openair)
summaryPlot(au_pm25_df)
```

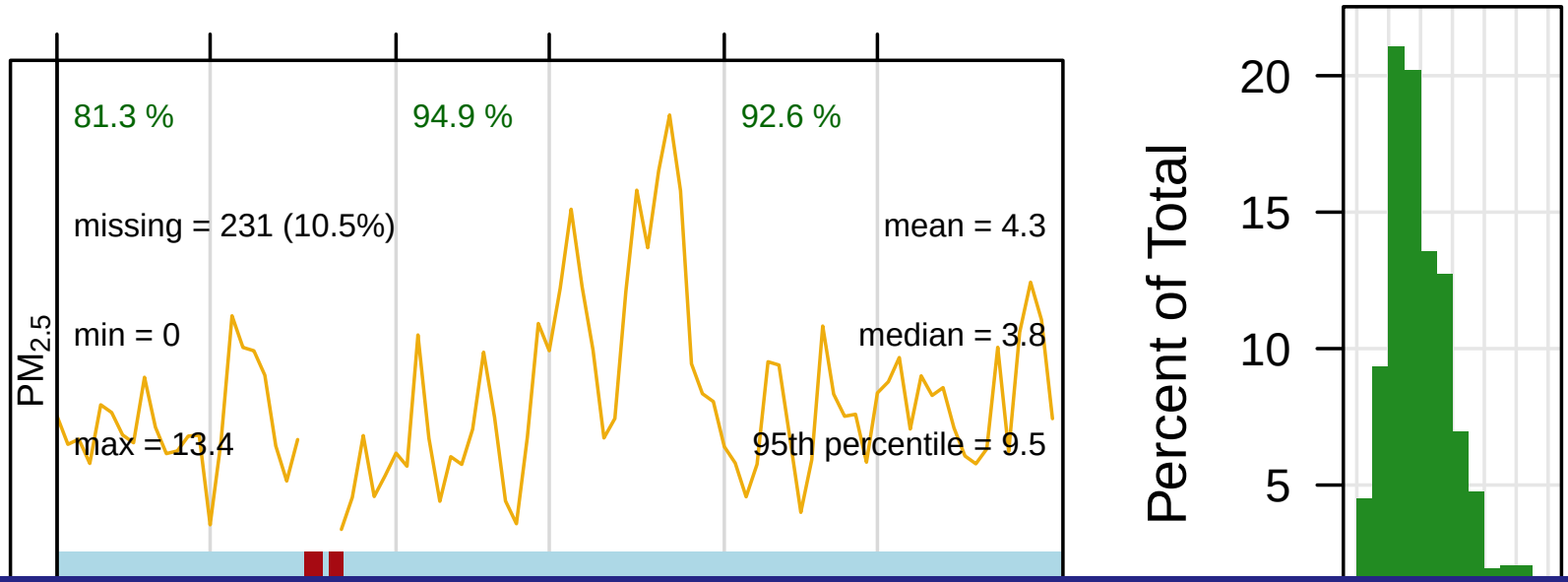
```
##      date1      date2       pm25
## "POSIXct"  "POSIXt" "numeric"
```



summaryPlot()

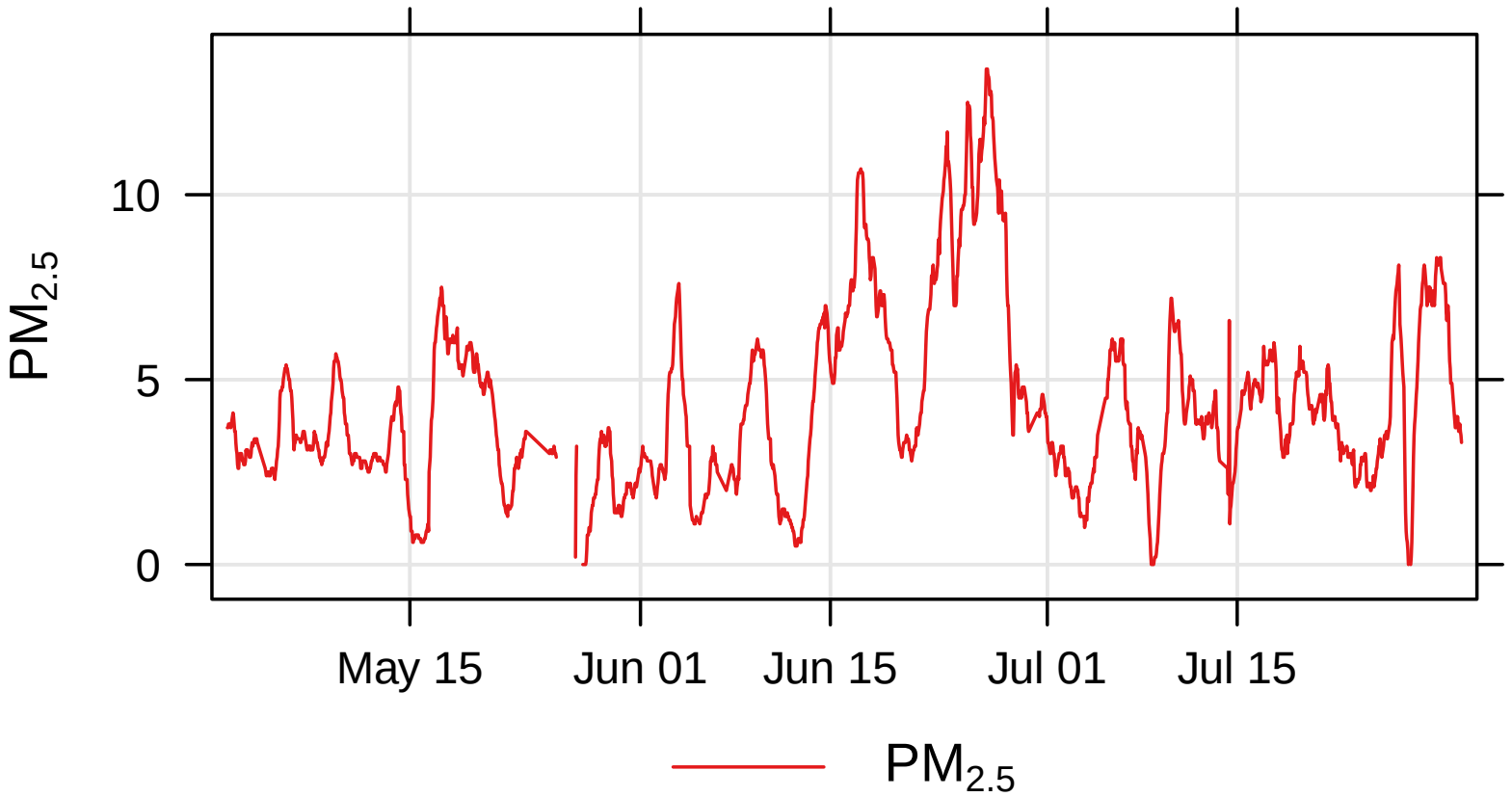
```
library(openair)
summaryPlot(au_pm25_df, period = "months")
```

```
##      date1      date2       pm25
## "POSIXct"  "POSIXt" "numeric"
```



TimePlot()

```
timePlot(au_pm25_df, pollutant = "pm25")
```



Um parentesis

Melhor usar dados de São Paulo

Um parentesis

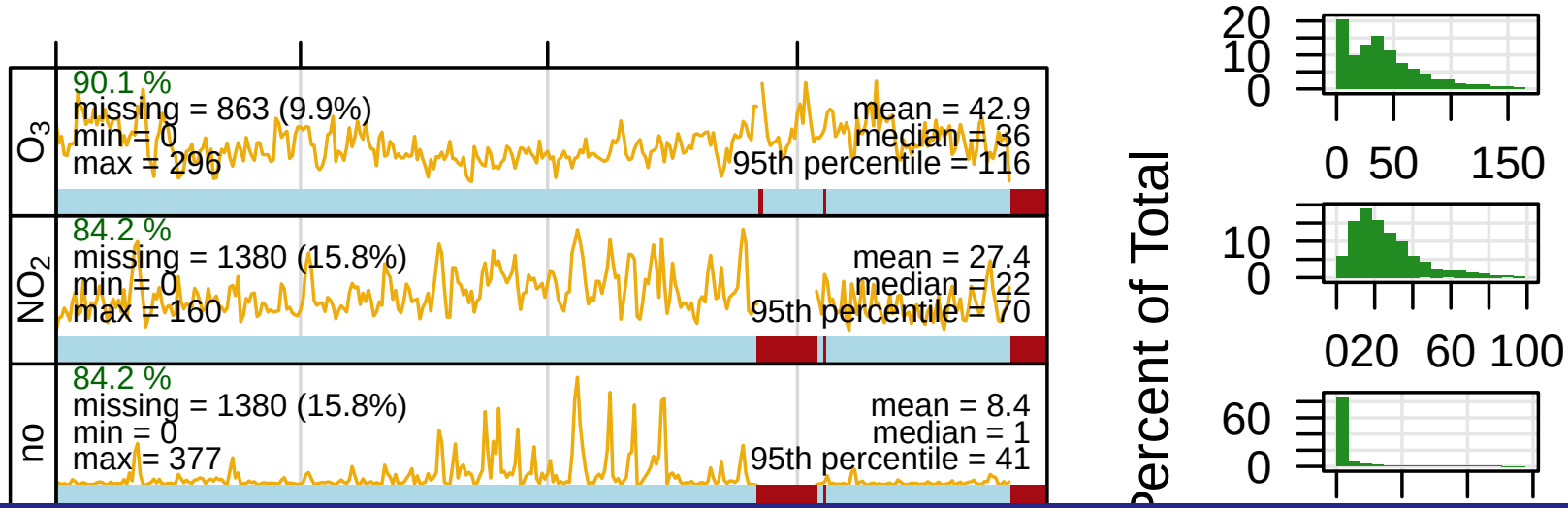
Lendo dados de Ibirapuera 2019 anos

```
ibi <- readRDS("../results/ibi_2019.rds")
```

summaryPlot()

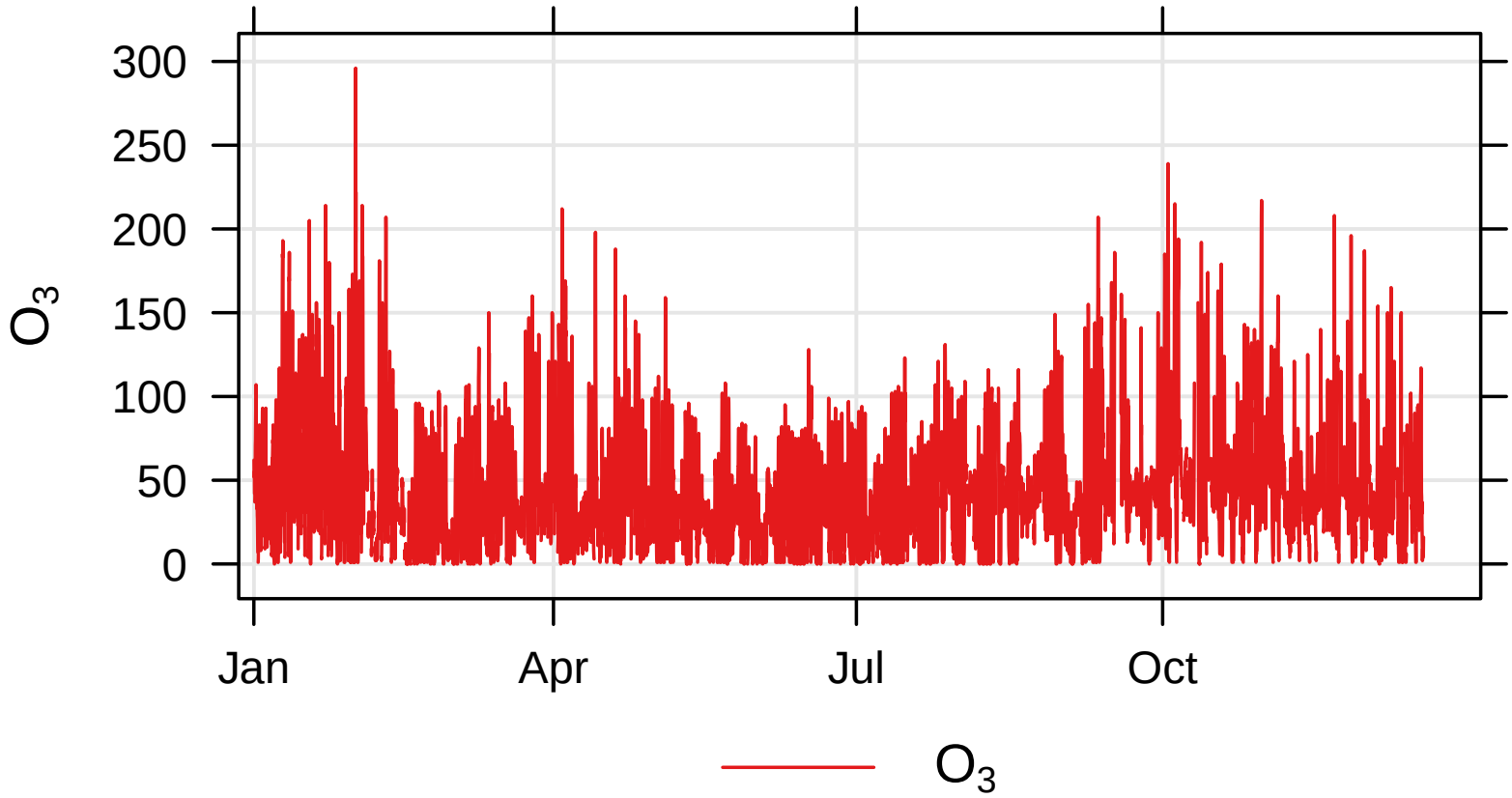
```
summaryPlot(ibi)
```

```
##       date1       date2         aqs          wd          p
##   "POSIXct"    "POSIXt" "character"   "numeric"   "numer
##         no2          o3          ws
##   "numeric"   "numeric"   "numeric"
```



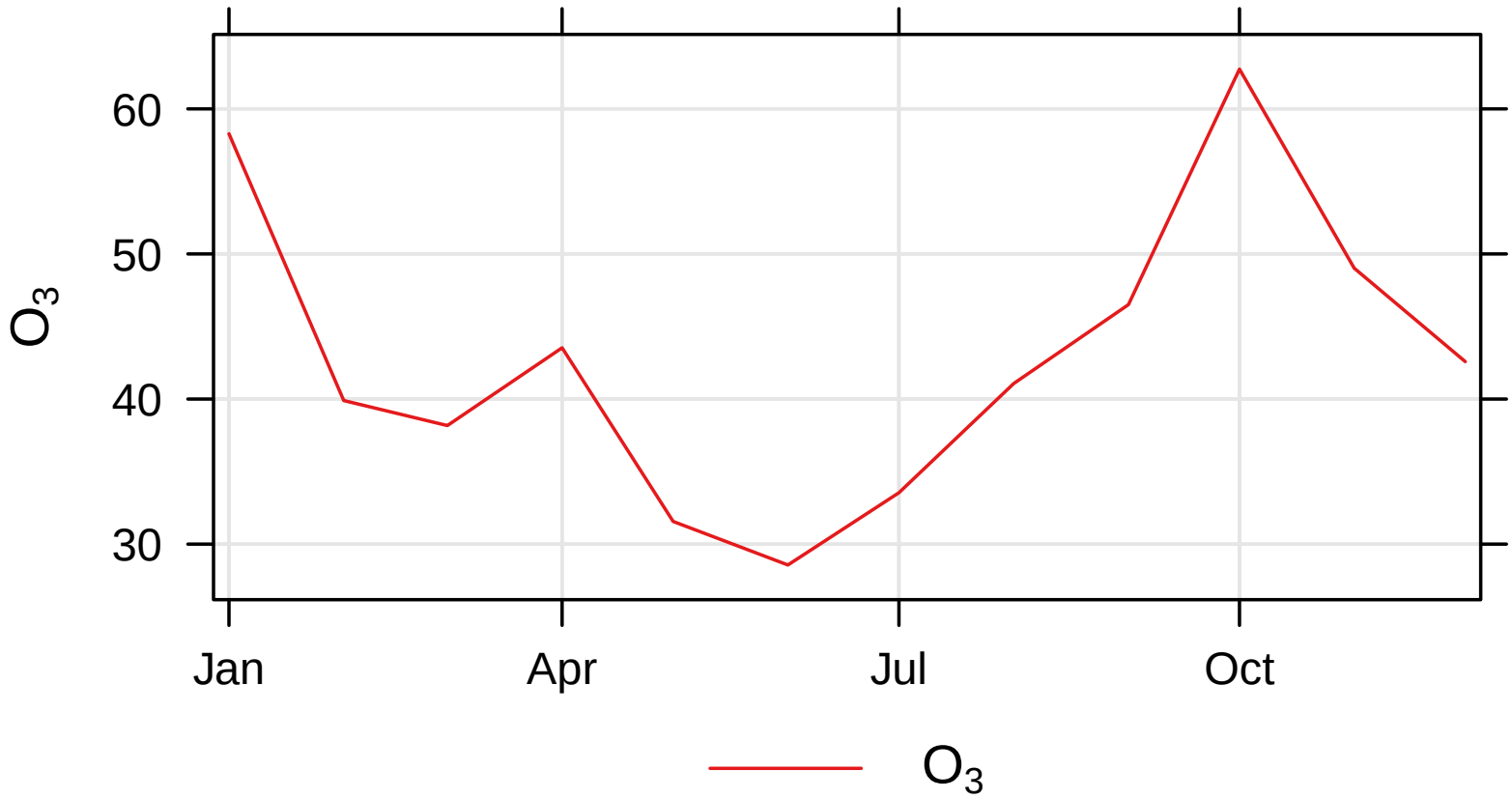
timePlot()

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



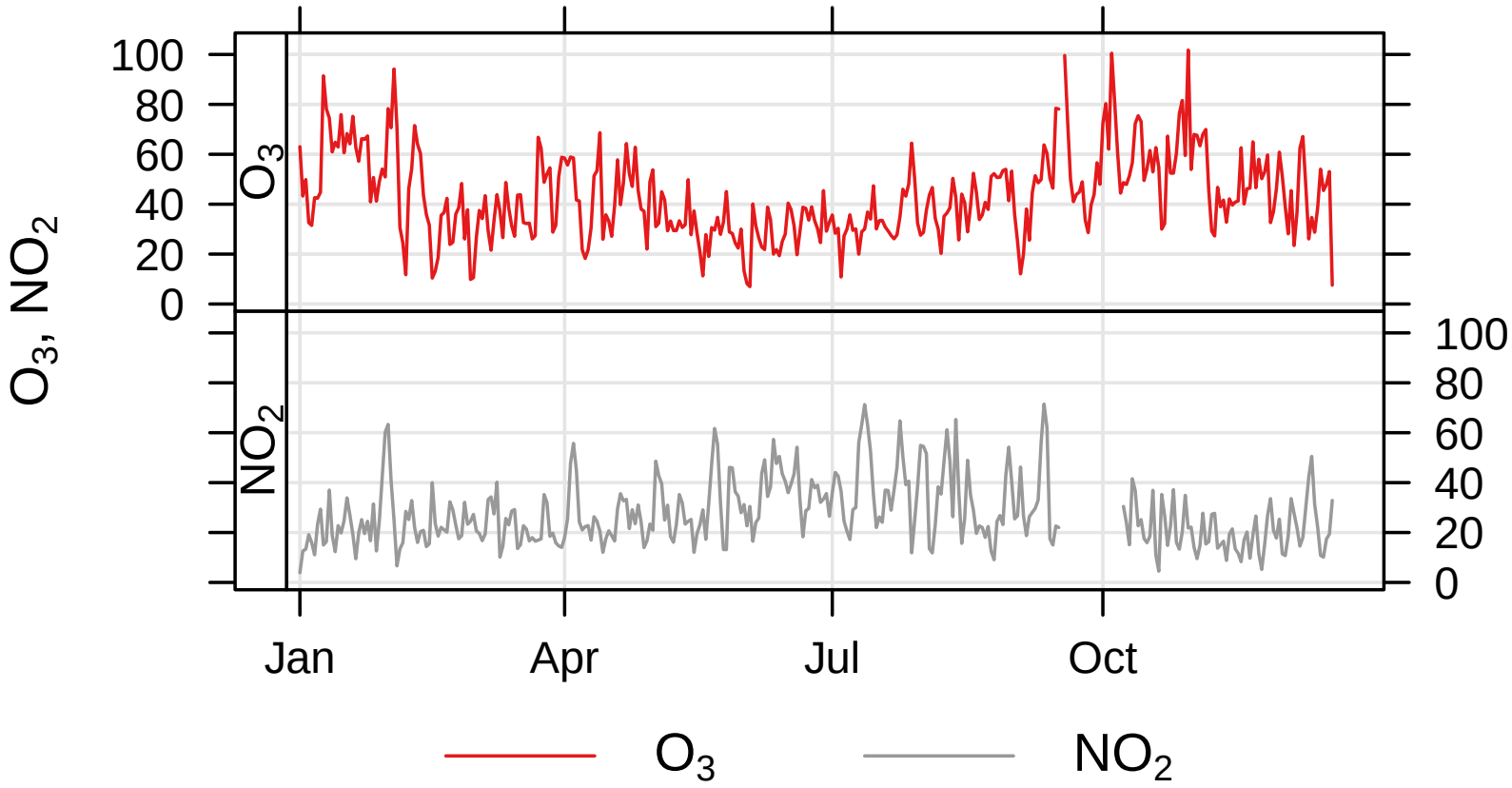
timePlot()

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



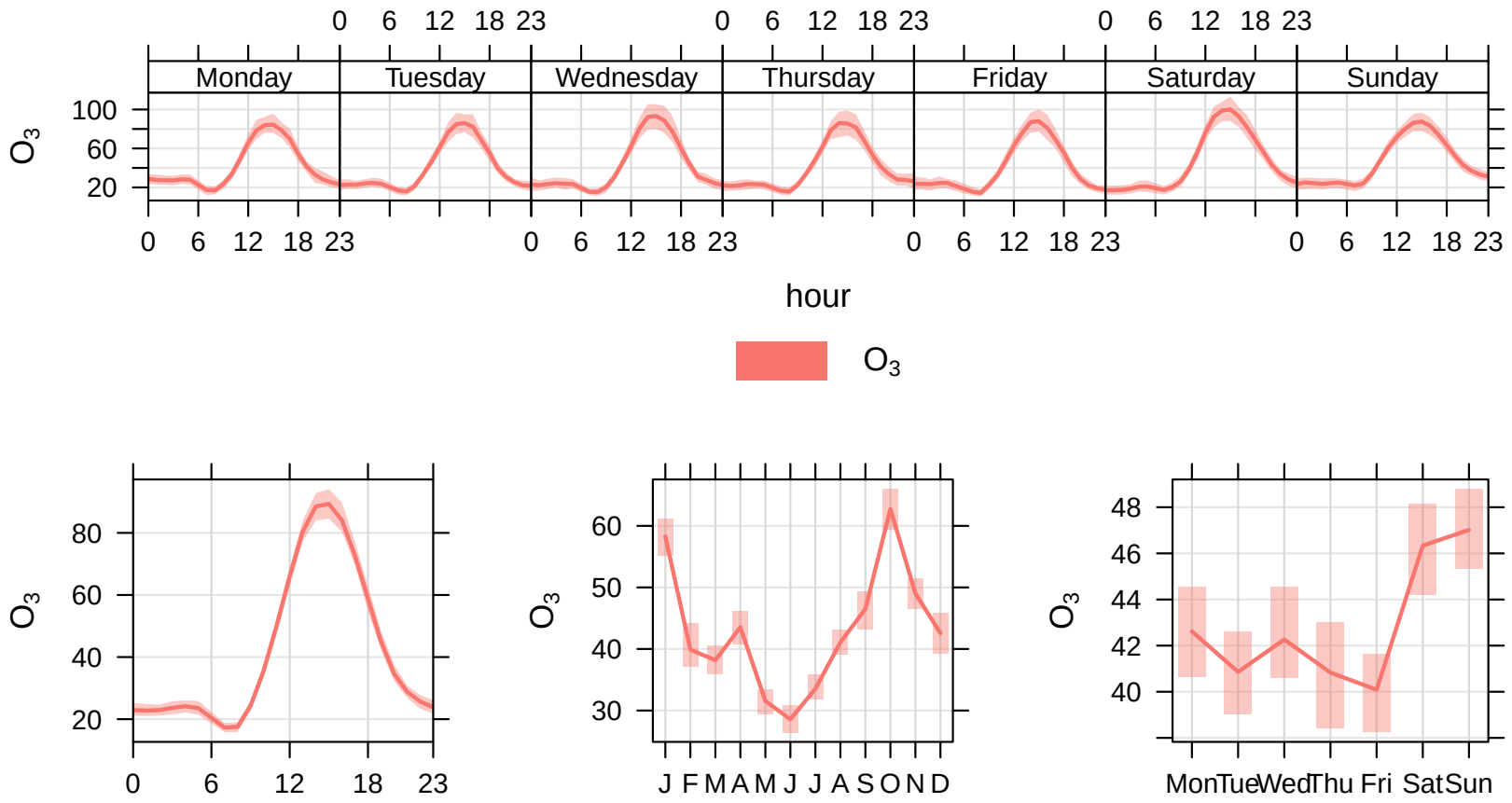
timePlot()

```
timePlot(ibi, pol = c("o3", "no2"), avg.time = "day")
```



timeVariation()

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



WindRose

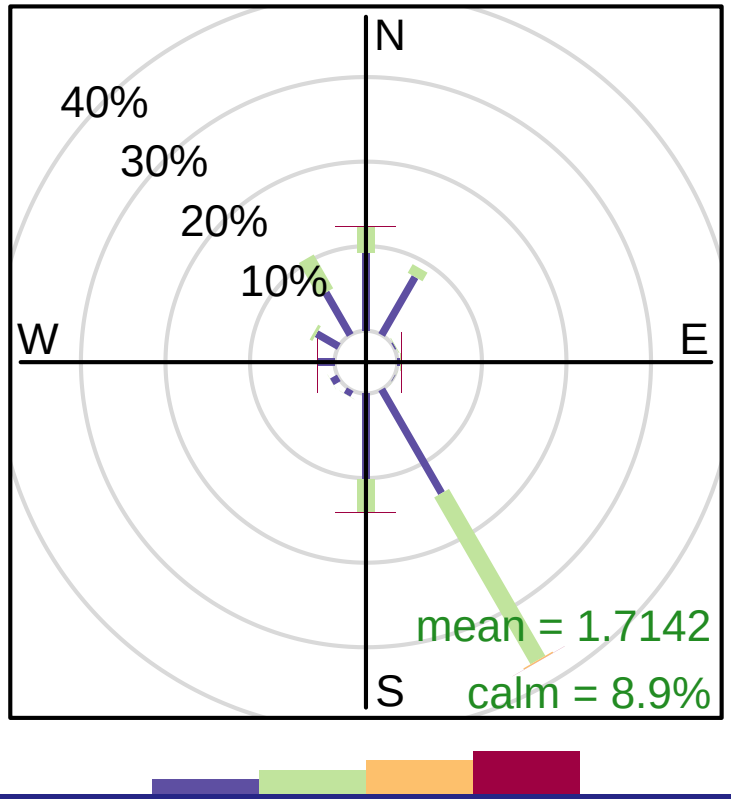
```
library(qualR)

params <- c("MP10", "MP2.5", "NOx", "VV", "DV")
pin_march <- cetesb_retrieve_param(Sys.getenv("QUALAR_USER"
                                   Sys.getenv("QUALAR_PASS"),
                                   params,
                                   "Pinheiros",
                                   "01/03/2020",
                                   "31/03/2020")

saveRDS(pin_march, file=paste0(getwd(), "/results/pin_pol_w
```

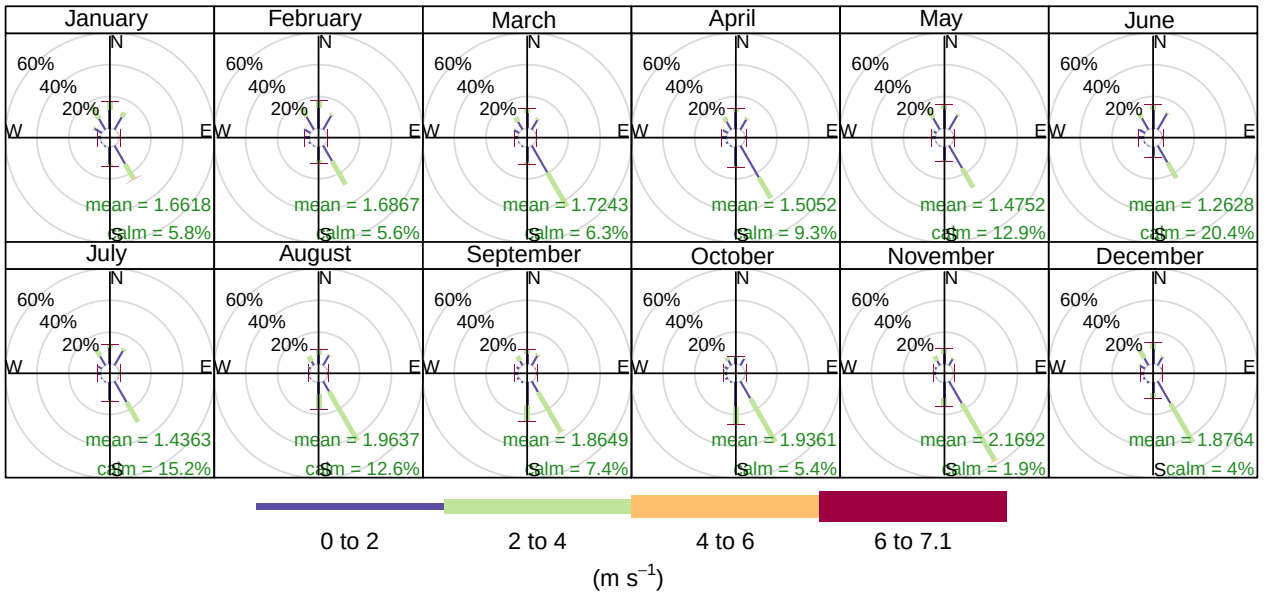
windRose()

```
pin <- readRDS("../results/pin_2019.rds")
wr <- windRose(pin, ws="ws", wd="wd")
```



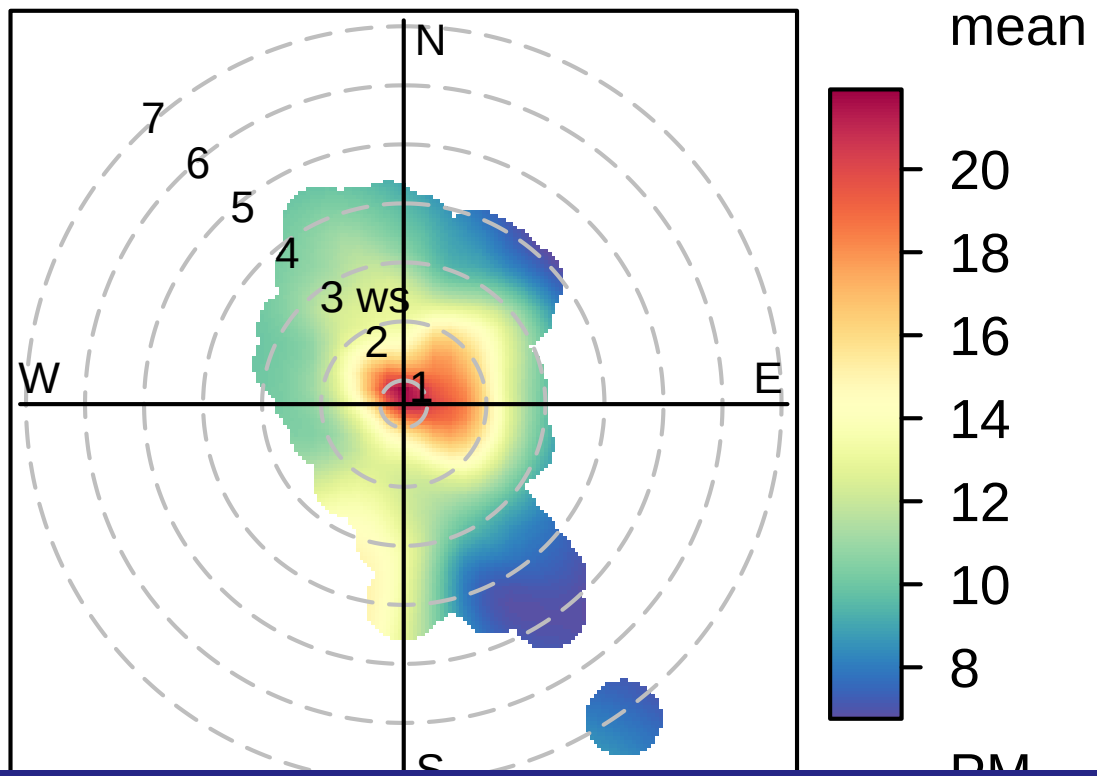
windRose()

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



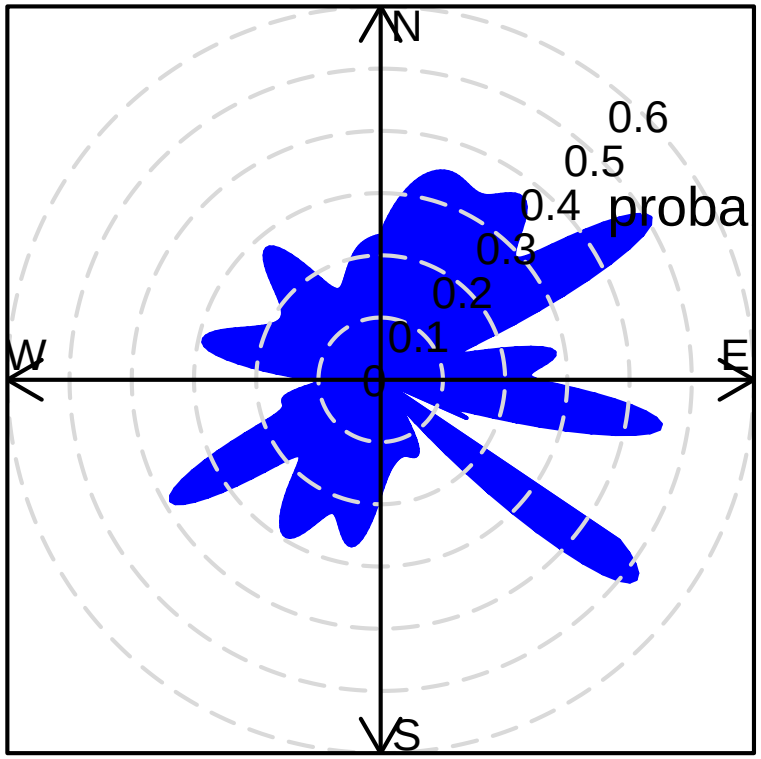
polarPlot()

```
pp <- polarPlot(pin, pollutant = "pm25")
```



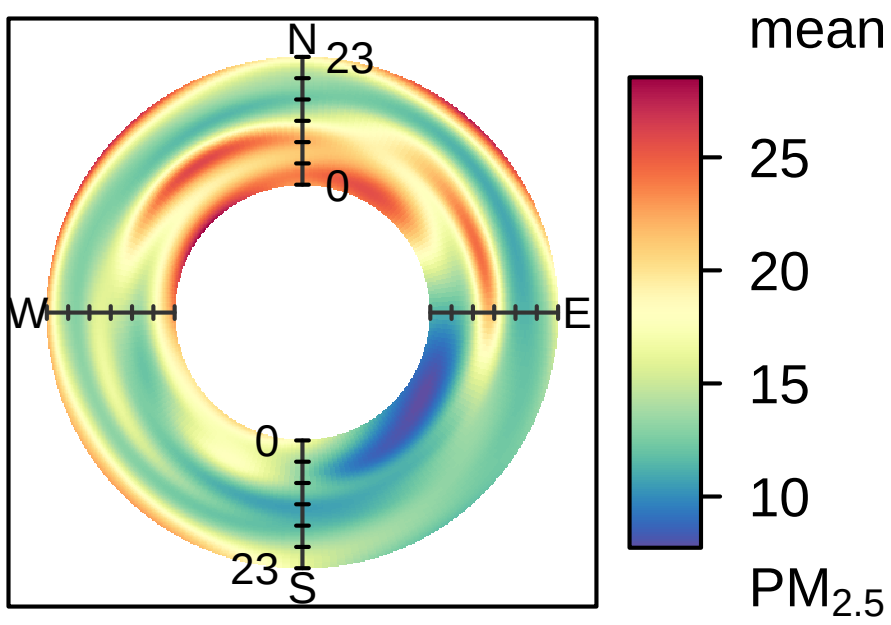
percentileRose()

```
pr <- percentileRose(pin, pollutant = "pm25", percentile=75
                     method = "cpf", col = "blue", smooth =
```



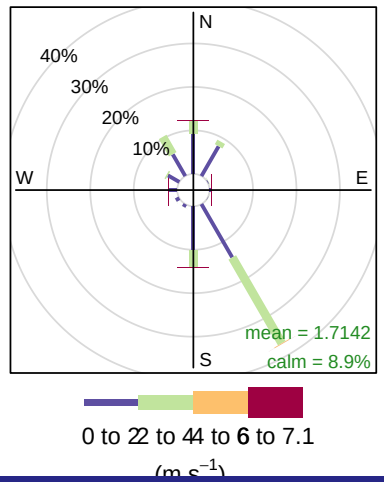
‘polarAnnulus()

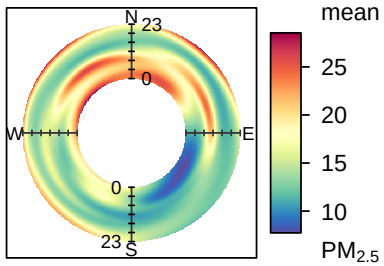
```
pa <- polarAnnulus(pin, pollutant = "pm25", period="hour",
                   exclude.missing=F)
```

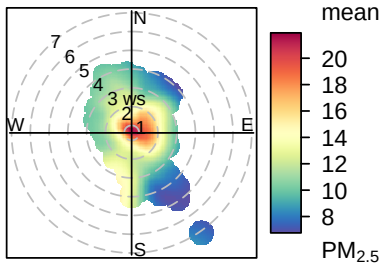


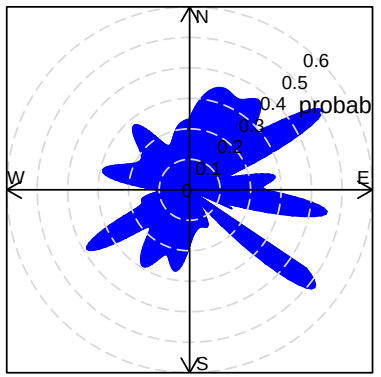
multiplots

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









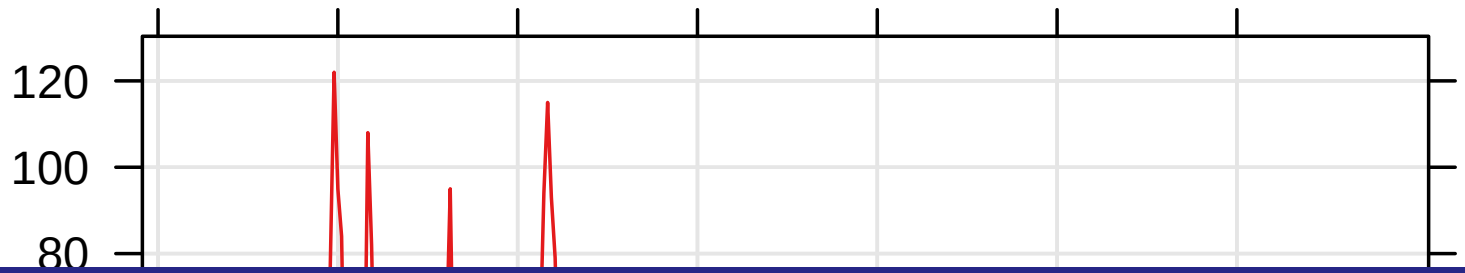
CPF at the 75th percentile (=19)

SelectByDate

```
pin_march <- readRDS("../results/pin_pol_with_wind.Rds")
covid <- selectByDate(pin_march,
                      start="15/3/2020",
                      end="28/3/2020")
```

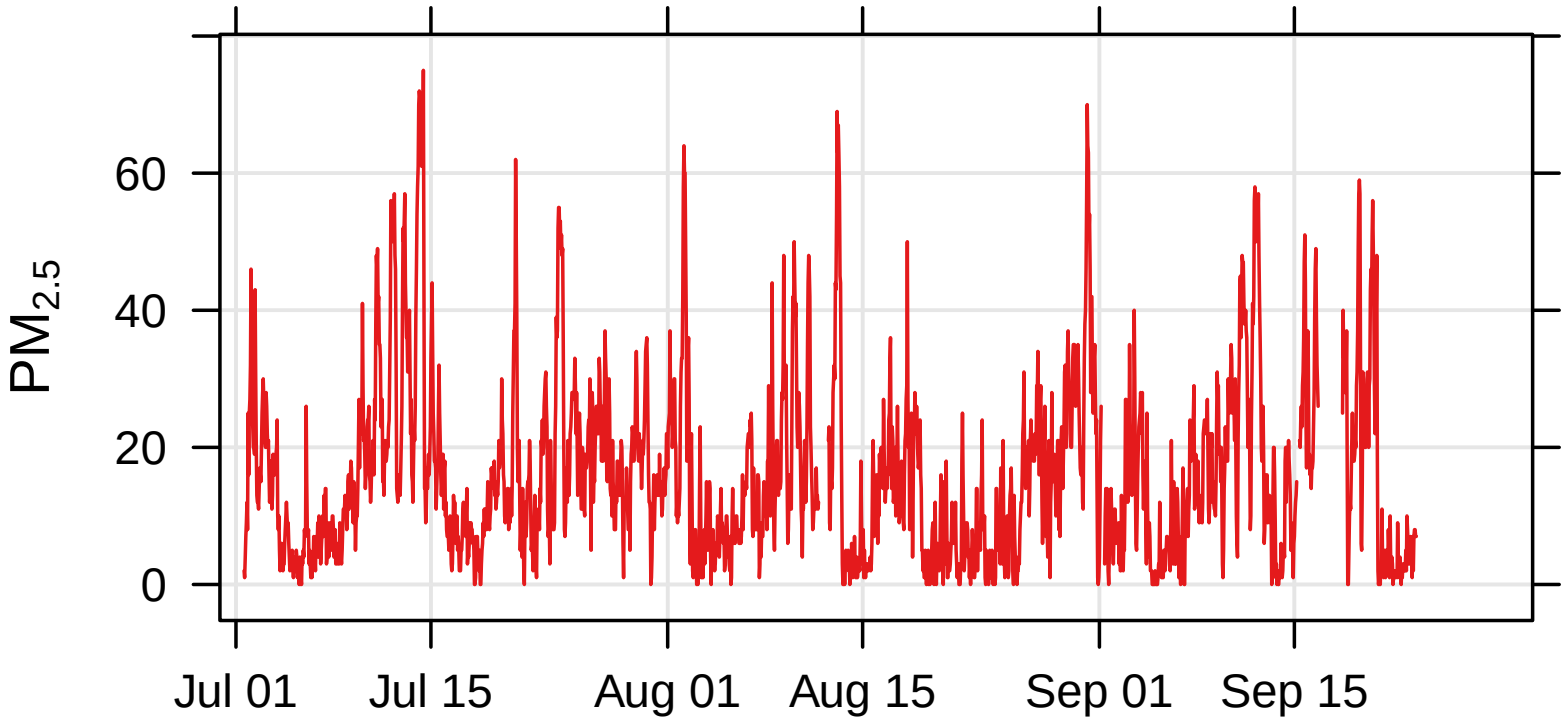
```
## Warning in checkPrep(mydata, vars, "default", remove.cal
## FALSE): Wind direction < 0 or > 360; removing these data
```

```
timePlot(covid, pol = "nox")
```



SelectByDate

```
ibi_m <- selectByDate(ibi, month = 7:9)
timePlot(ibi_m, pol = "pm25")
```



SelectByDate

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

```
## [1] 41.33155
```

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

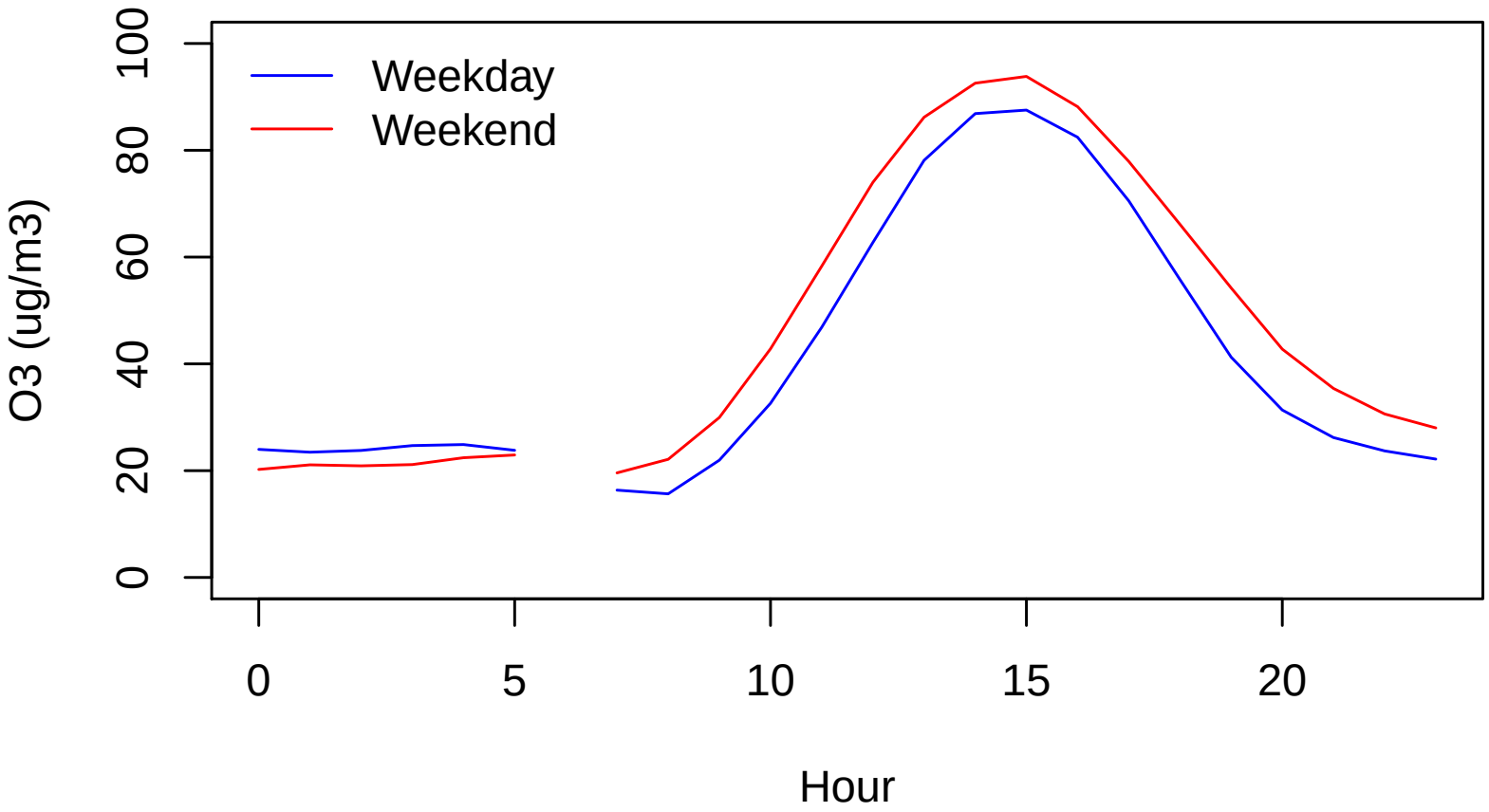
```
## [1] 46.67555
```

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

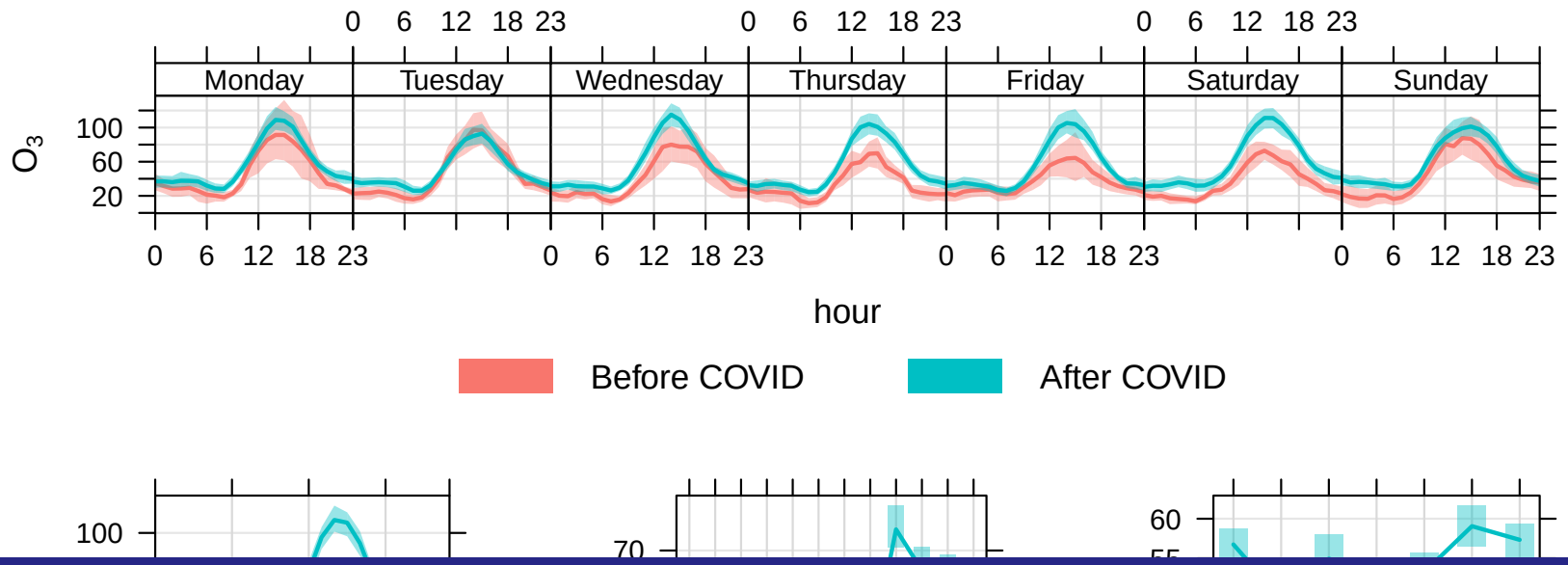


timeAverage

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

splitByDate

```
ibi20 <- readRDS("../results/ibi_2020.rds")
ibi_cov <- splitByDate(ibi20, 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