

Un curso introductorio de R

openair, series temporales y buenos plots

Mario Gavidia-Calderón

4/9/2021

El menú de hoy

- ▶ `as.POSIXct()`
- ▶ Completando datos faltantes de fecha
- ▶ `openair`
- ▶ Plots (buenos plots)

as.POSIXct()

- ▶ Un dato POSIXct representa **tiempos y fechas de calendarios**.
- ▶ Importante decirle a R que tenemos datos de fechas de calendarios:
 - ▶ Permite nuevas funcionalidades
 - ▶ Integración con otros paquetes/librerías

as.POSIXct()

- ▶ Para decirle a R que tenemos datos de fecha usamos `as.POSIXct` junto con `strptime()`
- ▶ `strptime()`: de character para `POSIXct`

as.POSIXct(): Datos marzo de 2019

```
mar19 <- read.table("../02_data/pm25_0319.csv",  
                    header = T, sep = ",",  
                    dec = ".")  
  
# Datos válidos  
mar19 <- subset(mar19, subset = QC.Name == "Valid")
```

as.POSIXct(): Datos marzo de 2019

```
# Dos primeros fechas
```

```
head(mar19$Date..LT)[1:2]
```

```
## [1] "2019-03-01 12:00 AM" "2019-03-01 01:00 AM"
```

```
class(mar19$Date..LT)
```

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

- ▶ El formato es: AÑO(4 dig)-mes(2 dig)-dia Hora:Min AM/PM
- ▶ En código: %Y-%m-%d %I:%M %p
- ▶ %I y %p cuando tenemos el AM/PM

as.POSIXct(): Datos marzo de 2019

- Transformando para POSIXct

```
# Creamos una nueva columna
mar19$date <- as.POSIXct(
  strptime(
    mar19$Date..LT., format = "%Y-%m-%d %I:%M %p"
  ),
  tz = "America/Lima"
)
class(mar19$date)

## [1] "POSIXct" "POSIXt"
```

as.POSIXct(): Datos marzo de 2019

- Veamos los primeros valores

```
mar19$date..LT.[1:2]
```

```
## [1] "2019-03-01 12:00 AM" "2019-03-01 01:00 AM"
```

```
mar19$date[1:2]
```

```
## [1] "2019-03-01 00:00:00 -05" "2019-03-01 01:00:00 -05"
```


Datos faltantes

- ▶ Veamos nuestros datos faltantes

```
horas_marzo <- 31 * 24  
horas_faltantes <- horas_marzo - nrow(mar19)  
horas_faltantes
```

```
## [1] 85
```

Datos Faltantes

- ▶ Vamos a completar las horas faltantes con NA
- ▶ La estrategia:
 - ▶ Crear un data frame con la fecha completa
 - ▶ Combinarla con nuestro data frame original (`mar19`)

Datos faltantes

- Data frame con fechas completas

```
todas_horas <- data.frame(  
  date = seq(  
    as.POSIXct("2019-03-01 00:00", tz = "America/Lima"),  
    as.POSIXct("2019-03-31 23:00", tz = "America/Lima"),  
    by = "hour")  
)  
head(todas_horas, 5)
```

```
##           date  
## 1 2019-03-01 00:00:00  
## 2 2019-03-01 01:00:00  
## 3 2019-03-01 02:00:00  
## 4 2019-03-01 03:00:00  
## 5 2019-03-01 04:00:00
```

Datos faltantes

```
nrow(mar19)
```

```
## [1] 659
```

```
mar19_all <- merge(todas_horas, mar19, all = T)  
nrow(mar19_all)
```

```
## [1] 745
```

Datos faltantes

- Cuales fueron mis datos faltantes

```
mar19_na <- subset(mar19_all, subset = is.na(mar19_all$Raw  
head(mar19_na, 3)
```

##		date	Site	Parameter	Date..LT.	Year	Mo
## 89		2019-03-04 16:00:00	<NA>	<NA>	<NA>	NA	
## 219		2019-03-10 02:00:00	<NA>	<NA>	<NA>	NA	
## 244		2019-03-11 02:00:00	<NA>	<NA>	<NA>	NA	
##		NowCast.Conc.	AQI	AQI.Category	Raw.Conc.	Conc..Unit	
## 89		NA	NA	<NA>	NA	<NA>	
## 219		NA	NA	<NA>	NA	<NA>	
## 244		NA	NA	<NA>	NA	<NA>	

Datos faltantes

- ▶ Ahora que mis data set está “completo” lo puedo guardar y exportar

```
# Sólo neceito date y pm25
pm25 <- mar19_all[c("date", "Raw.Conc.")]
names(pm25) <- c("date", "pm25")
write.table(pm25, "pm25_mar19_all.csv", sep = ",",
            row.names = F, quote = F)
```

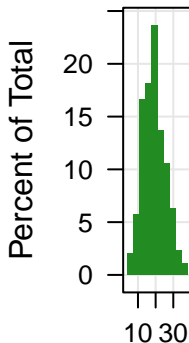
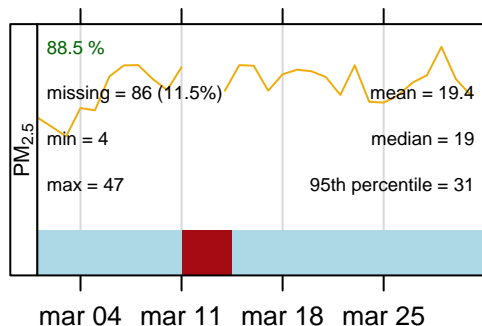
openair

- ▶ openair es una librería para el análisis de datos de contaminación atmosférica
- ▶ Requiere de una columna con datos de fecha llamada date
- ▶ Para instalar:
 - ▶ `install.package("openair")`
- ▶ Para usarla:
 - ▶ `library(openair)`
- ▶ Vamos a ver algunas de sus funcionalidades

openair : summaryplot

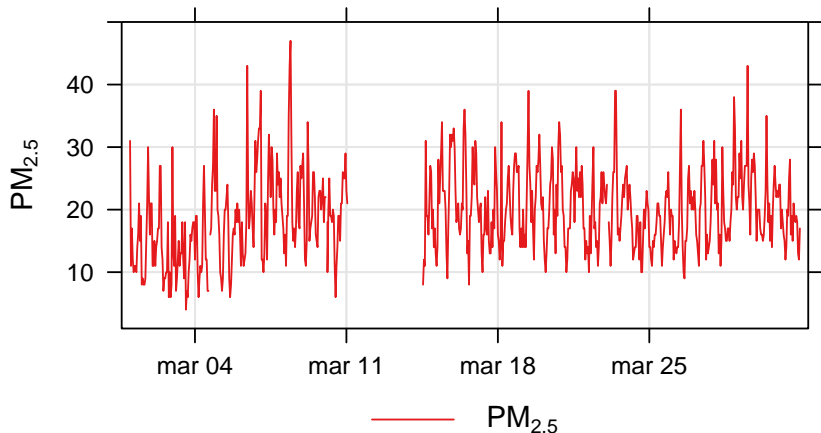
```
library(openair)
summaryPlot(pm25, period = "months")
```

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



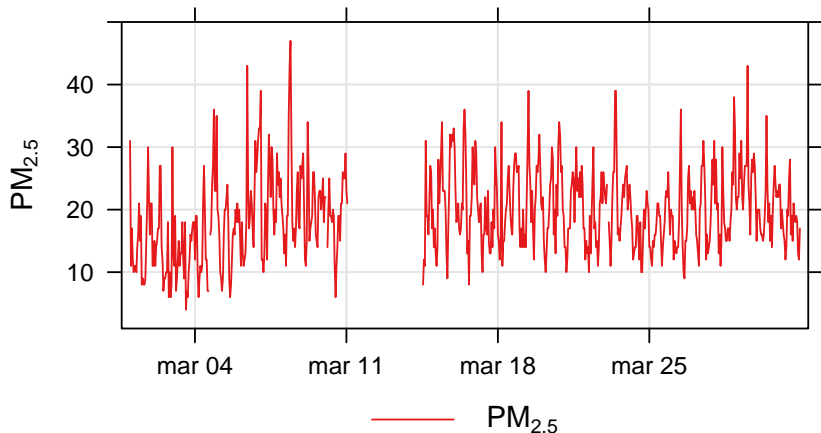
openair: timePlot

```
library(openair)  
openair::timePlot(pm25, pollutant = "pm25")
```



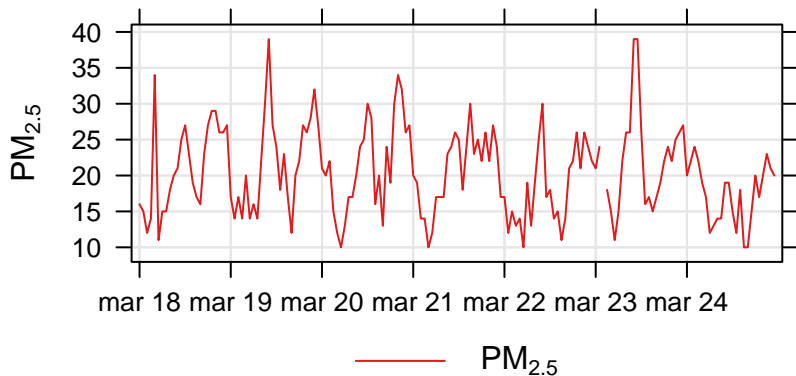
openair: timePlot

```
library(openair)
timePlot(pm25, pollutant = "pm25", avg.time = "hour")
```



openair: selectByDate

```
a_week <- selectByDate(pm25,  
                        start="18/03/2019", end="24/03/2019")  
timePlot(a_week, pollutant = "pm25")
```



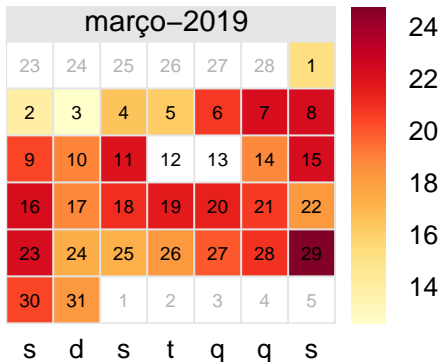
openair: timeAverage

```
hour_prof <- timeAverage(pm25, avg.time = "hour")  
head(hour_prof)
```

```
## # A tibble: 6 x 2  
##   date                pm25  
##   <dtm>              <dbl>  
## 1 2019-03-01 00:00:00    31  
## 2 2019-03-01 01:00:00    11  
## 3 2019-03-01 02:00:00    17  
## 4 2019-03-01 03:00:00    12  
## 5 2019-03-01 04:00:00    10  
## 6 2019-03-01 05:00:00    11
```

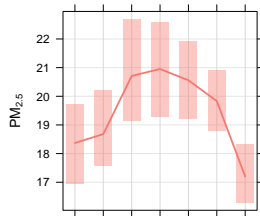
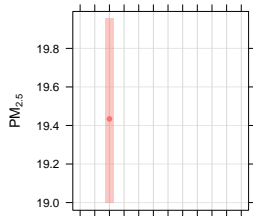
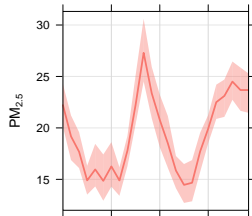
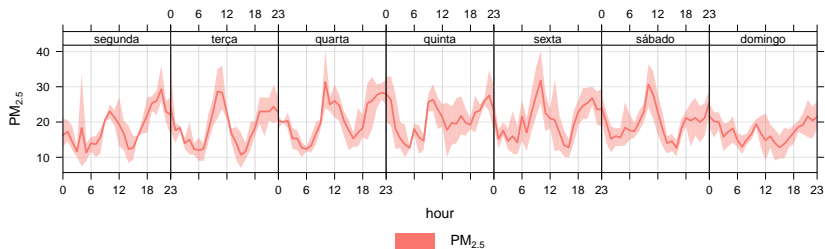
openair: calendarPlot

```
calendarPlot(pm25, pollutant = "pm25")
```



openair: timeVariation

```
timeVariation(pm25, pollutant = "pm25")
```



Plots: Cómo hacer buenas figuras?

1. Conoce a tu audiencia.
2. Identifica el mensaje.
3. Adapta la figura al medio de soporte.
4. Las etiquetas no son opcionales.
5. No confíes en la configuración por defecto. ¹

¹Rougier et al. 2014

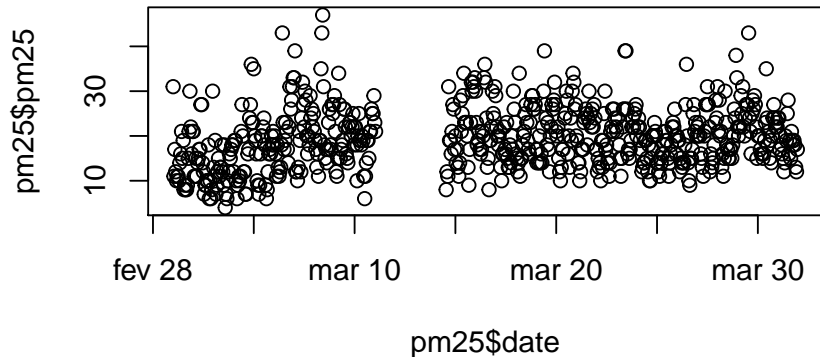
Plots: Cómo hacer buenas figuras?

6. Usa los colores efectivamente.
7. No confundas al lector.
8. Evita el *chartjunk* (Menos es más)
9. Mensaje > belleza
10. Usa la herramienta correcta. ²

²Rougier et al. 2014

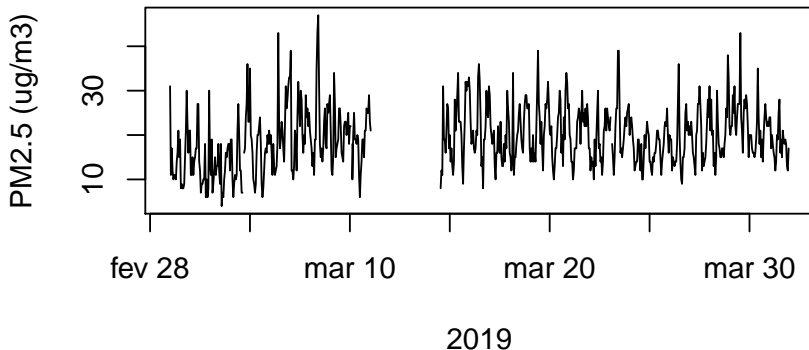
Plots: plot()

```
# Sintáxis plot(x , y)  
plot(pm25$date, pm25$pm25)
```



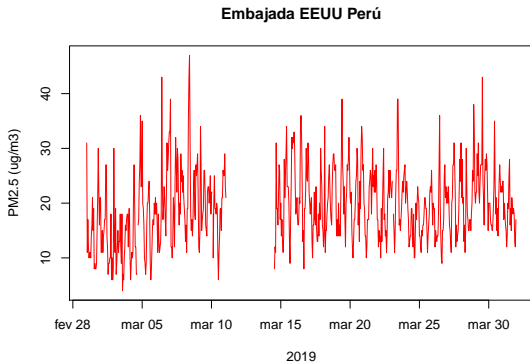
Plots: plot()

```
# No confiar en default, Etiquetas obligatorias  
plot(pm25$date, pm25$pm25, t = "l",  
      ylab = "PM2.5 (ug/m3)", xlab = "2019")
```



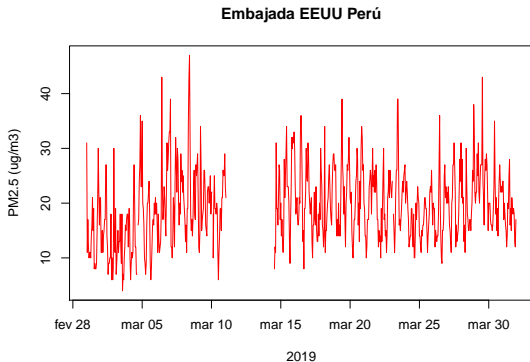
Plots: plot()

```
# No confiar en default, Etiquetas obligatorias  
plot(pm25$date, pm25$pm25, t = "l", col = "red",  
     ylab = "PM2.5 (ug/m3)", xlab = "2019",  
     main = "Embajada EEUU Perú")
```



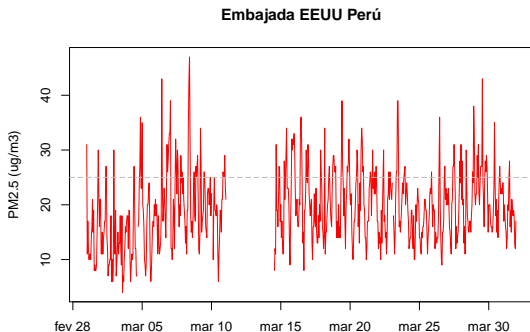
Plots: plot()

```
# No confiar en default, Etiquetas obligatorias  
plot(pm25$date, pm25$pm25, t = "l", col = "red",  
     ylab = "PM2.5 (ug/m3)", xlab = "2019",  
     main = "Embajada EEUU Perú")
```



Plots: plot()

```
# No confiar en default, Etiquetas obligatorias  
plot(pm25$date, pm25$pm25, t = "l", col = "red",  
     ylab = "PM2.5 (ug/m3)", xlab = "2019",  
     main = "Embajada EEUU Perú")  
abline(h = 25, col = "grey", lty = "longdash")
```



Vamos a trabajar con un proyecto.

Palabras finales

1. El mejor comando de R es ?
2. Si no funciona usen Google:
 - ▶ En inglés es mejor
 - ▶ En StackOverflow está la gente más brava
3. El camino está lleno de errores pero así se aprende

Palabras finales

- ▶ Con esta pequeña introducción espero que:
 - ▶ Puedas usar R para tus laboratorios
 - ▶ Para tu tesis
 - ▶ Para automatizar procesor
 - ▶ **Estar más preparado para los cursos on-line.**
 - ▶ Datacamp
 - ▶ Dataquest