# Un curso introductorio de R

## Scripts, funciones y loops

Mario Gavidia-Calderón

3/31/2021

# El Menú de hoy

- Operaciones lógicas
- `subset()`
- Scripts
- Funciones
- Loops
- `packages`

# Operaciones lógicas

# Operaciones lógicas

- `TRUE` $= 1$
- `FALSE` $= 0$
- Conjunción es **y** , en R es `&`, equivale a `*`
- Disjunción es **o** , en R es `|` , equivale a $+$

# Operaciones lógicas

- Entonces:
  - `TRUE & TRUE`
  - `TRUE & FALSE`
  - `FALSE & TRUE`
  - `FALSE & FALSE`

# Operaciones lógicas

- Entonces:

```
TRUE & TRUE
```

```
## [1] TRUE
```

```
TRUE & FALSE
```

```
## [1] FALSE
```

```
FALSE & TRUE
```

```
## [1] FALSE
```

```
FALSE & FALSE
```

```
## [1] FALSE
```

# Operaciones lógicas

- Entonces:
  - `TRUE | TRUE`
  - `TRUE | FALSE`
  - `FALSE | TRUE`
  - `FALSE | FALSE`

# Operaciones lógicas

- Entonces:

```
TRUE | TRUE
```

```
## [1] TRUE
```

```
TRUE | FALSE
```

```
## [1] TRUE
```

```
FALSE | TRUE
```

```
## [1] TRUE
```

```
FALSE | FALSE
```

```
## [1] FALSE
```

# Operaciones lógicas

- Mayor o menor que: >, <
- Mayor o menor igaul que: >=, <=
- Es igual que: ==
- No es igual que: !=

# subset()

# subset()

- De ?subset:
  - Retorna un subconjunto de vectores, matrices o data frames que **satisfacen las condiciones**
- **condiciones = operaciones lógicas**
- `subset(data, subset = _condiciones_)`

# subset(): Leyendo encuesta

```
survey <- read.table(
  "../02_data/respuestas27.csv",
  header = T,
  sep = ","
)
```

# subset(): Leyendo encuesta

```
names(survey) <- c("date", "name", "last.name",
                   "age", "district", "molinero",
                   "faculty", "year", "program",
                   "prog.lang", "os", "labs",
                   "Excel", "R", "why")
names(survey)
```

```
##  [1] "date"      "name"      "last.name" "age"          "di
##  [7] "faculty"   "year"      "program"   "prog.lang"    "os
## [13] "Excel"     "R"         "why"
```

# subset(): Leyendo encuesta

- Cuántos estudian ing. ambiental

```
ing_amb <- subset(
  survey,
  subset = faculty == "Ingeniería ambiental")
class(ing_amb)
```

```
## [1] "data.frame"
```

```
nrow(ing_amb)
```

```
## [1] 9
```

# subset(): Leyendo encuesta

- Cuántos son menores que la edad média

```
menores_edad_media <- subset(
  survey,
  subset = age < mean(survey$age)
)

nrow(menores_edad_media)
```

```
## [1] 12
```

# subset(): Leyendo encuesta

- Qué estudia el más valiente

```
par_fin <- "No hago las prácticas me defiendo en el parcial
temerario <- subset(
  survey,
  subset = labs == par_fin
)

temerario$faculty
```

```
## [1] "Meteorología"
```

# subset(): Leyendo encuesta

- Programan **y** saben Python

```
program_py <- subset(
  survey,
  subset = (program == "Sí" | program == "Algo" &
              prog.lang == "Python")
)

head(program_py[, c("program", "prog.lang")], 5)
```

```
##    program prog.lang
## 1     Algo    Python
## 3     Algo    Python
## 4     Algo    Python
## 9     Algo    Python
## 16      Sí    Python
```

# Scripts

# Qué es un script?

- Una **receta**: *Script para analisar resultados de encuesta*.
  - Serie de operaciones (`statements`) en un archivo de texto (`*.R`)
- Consola de R para **experimentar**: *Qué hace está función?*
- Editor de texto[1]:
  - **Espacio** para trabajar
  - Código que **funciona** y nos **importa**

[1] Wickham & Grolemund, 2016

Curso_R_UNALM_ciencias - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Source

Console Terminal R Markdown Jobs

~/R_tests/Curso_R_UNALM_ciencias/

```
> mar19 <- read.table("02_data/pm25_0319.csv", header = T, sep = ",", dec = ".")
> head(mar19)
  Site         Parameter           Date..LT. Year Month Day Hour NowCast.Conc. AQI AQI.Category Raw.Conc.
1 Lima PM2.5 - Principal 2019-03-01 12:00 AM 2019     3   1    0          27.9  84     Moderate        31
2 Lima PM2.5 - Principal 2019-03-01 01:00 AM 2019     3   1    1          19.4  66     Moderate        11
3 Lima PM2.5 - Principal 2019-03-01 02:00 AM 2019     3   1    2          18.2  64     Moderate        17
4 Lima PM2.5 - Principal 2019-03-01 03:00 AM 2019     3   1    3          15.1  57     Moderate        12
5 Lima PM2.5 - Principal 2019-03-01 04:00 AM 2019     3   1    4          12.5  52     Moderate        10
6 Lima PM2.5 - Principal 2019-03-01 05:00 AM 2019     3   1    5          11.7  49         Good        11
  Conc..Unit Duration QC.Name
1      UG/M3     1 Hr   Valid
2      UG/M3     1 Hr   Valid
3      UG/M3     1 Hr   Valid
4      UG/M3     1 Hr   Valid
5      UG/M3     1 Hr   Valid
6      UG/M3     1 Hr   Valid
> horas_marzo <- 24 * 31
> nrow(mar19)
[1] 660
> no_data_marzo <- horas_marzo - nrow(mar19)
> no_data_marzo
[1] 84
> max(mar19$Raw.Conc.)
[1] 96
> mar19[mar19$Raw.Conc. == max(mar19$Raw.Conc.), ]
    Site         Parameter           Date..LT. Year Month Day Hour NowCast.Conc. AQI AQI.Category Raw.Conc.
447 Lima PM2.5 - Principal 2019-03-23 02:00 AM 2019     3  23    2            23  74     Moderate        96
    Conc..Unit Duration QC.Name
447      UG/M3     1 Hr Invalid
> mean(mar19$Raw.Conc.)
[1] 19.55
> plot(mar19$Raw.Conc., t = "l", ylab = "PM2.5 (ug/m3)", xlab = "Marzo 2019 (Horas)")
>
```

Environment History Connections Git Tutorial

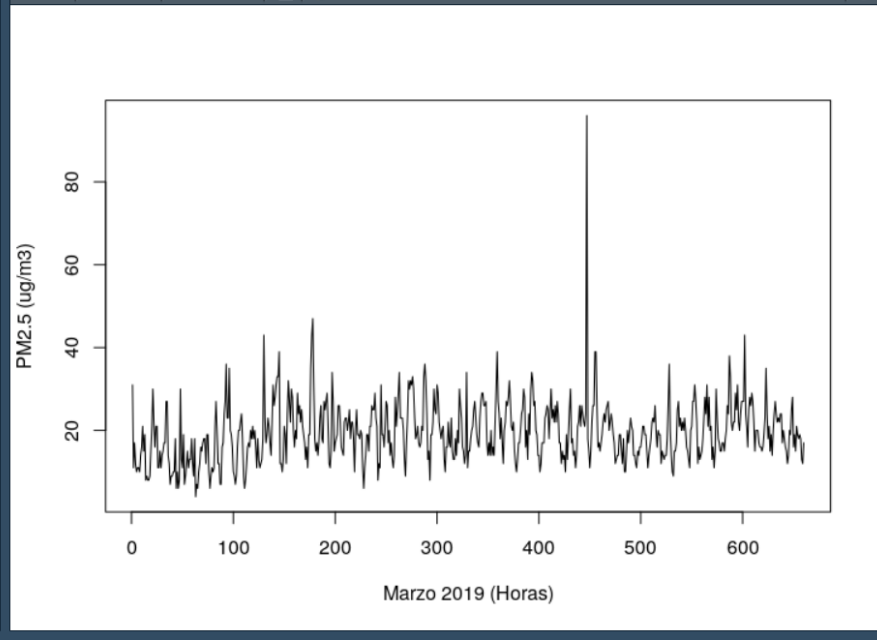
Import Dataset List

Global Environment

| Data | |
|---|---|
| mar19 | 660 obs. of 14 variables |
| **Values** | |
| horas_marzo | 744 |
| no_data_marzo | 84 |

Files Plots Packages Help Viewer

Zoom Export Publish



Curso_R_UNALM_ciencias - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Curso_R_UNALM_ciencias

04_dudas_pendientes_clase1.Rmd | Clase_2_Scripts_funciones_loops.... | Untitled1*

Source on Save | Run | Source

```
1  # Script para leer data horaria por mes de PM2.5
2  # medido en la embajada de los EEUU en Perú
3
4  ## Dirección del archivo
5  pm25_file <- "02_data/pm25_0319.csv"
6
7  ## Importando datos
8  mar19 <- read.table(pm25_file, header = T, sep = ",", dec = ".")
9
10 ## Los datos están completos?
11 horas_marzo <- 31 * 24
12 horas_faltantes <- horas_marzo - nrow(mar19)
13
14 ## Máxima concentración horaria
15 max_mar19 <- max(mar19$Raw.Conc.)
16
17 ## Qué día y hora?
18 day_max  <- mar19$Day[mar19$Raw.Conc. == max_mar19]
19 hour_max <- mar19$Hour[mar19$Raw.Conc. == max_mar19]
20
21 # Media mensuak
22 mean_mar19 <- mean(mar19$Raw.Conc.)
23
24 # Solución
25 ans <- data.frame(miss_hour = horas_faltantes,
26                   max_pm25 = max_mar19,
27                   max_day = day_max,
28                   max_hour = hour_max,
29                   mean = mean_mar19)
```

10:30 (Top Level) R Script

Console

Environment History Connections Git Tutorial

Import Dataset | List

Global Environment

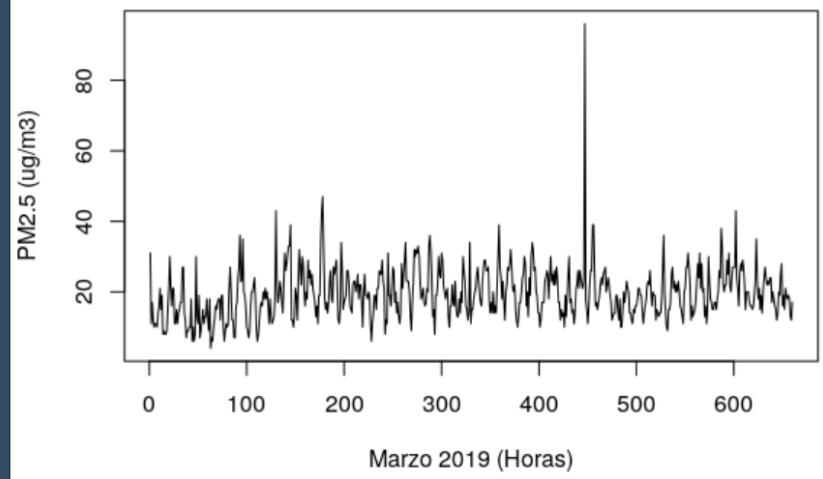
Data

| | |
|---|---|
| ans | 1 obs. of 5 variables |
| mar19 | 660 obs. of 14 variables |

Values

Files Plots Packages Help Viewer

Zoom | Export | Publish



# Scripts

- Un proceso ordenado
- Suceptible a mejoras
- Más fácil de compartir

# Funciones

# Funciones

```
functionName <- function(arg1, arg2, ...){
  statements1
  statements2
  ...
  return(object)
}
```

# Funciones: Ejemplos prácticos

- Cubo de un número

```
CuboNumero <- function(x){
  cubo <- x * x * x
  return(cubo)
}

CuboNumero(3)
```

```
## [1] 27
```

# Funciones: Ejemplos prácticos

- Más argumentos

```
MyAge <- function(this_year, born_year){
  my_age <- this_year - born_year
  return(my_age)
}

MyAge(2020, 1988)
```

```
## [1] 32
```

# Funciones: Ejemplos prácticos

```
MediaVector <- function(vec){
  suma <- sum(vec)
  n <- length(vec)
  media <- suma / n
  return(media)
}

mi_vector <- c(1, 4, 5, 6, 15)
MediaVector(mi_vector)
```

```
## [1] 6.2
```

# Scripts y funciones

Usualmente un **script** puede transformarse en una **función**.

Curso_R_UNALM_ciencias - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Curso_R_UNALM_ciencias

04_dudas_pendientes_clase1.Rmd | Clase_2_Scripts_funciones_loops.... | Untitled1*

Source on Save | Run | Source

```
1  # Script para leer data horaria por mes de PM2.5
2  # medido en la embajada de los EEUU en Perú
3
4  ## Dirección del archivo
5  pm25_file <- "02_data/pm25_0319.csv"
6
7  ## Importando datos
8  mar19 <- read.table(pm25_file, header = T, sep = ",", dec = ".")
9
10 ## Los datos están completos?
11 horas_marzo <- 31 * 24
12 horas_faltantes <- horas_marzo - nrow(mar19)
13
14 ## Máxima concentración horaria
15 max_mar19 <- max(mar19$Raw.Conc.)
16
17 ## Qué día y hora?
18 day_max  <- mar19$Day[mar19$Raw.Conc. == max_mar19]
19 hour_max <- mar19$Hour[mar19$Raw.Conc. == max_mar19]
20
21 # Media mensuak
22 mean_mar19 <- mean(mar19$Raw.Conc.)
23
24 # Solución
25 ans <- data.frame(miss_hour = horas_faltantes,
26                   max_pm25 = max_mar19,
27                   max_day = day_max,
28                   max_hour = hour_max,
29                   mean = mean_mar19)
```

10:30 (Top Level) R Script

Console

Environment History Connections Git Tutorial

Import Dataset | List

Global Environment

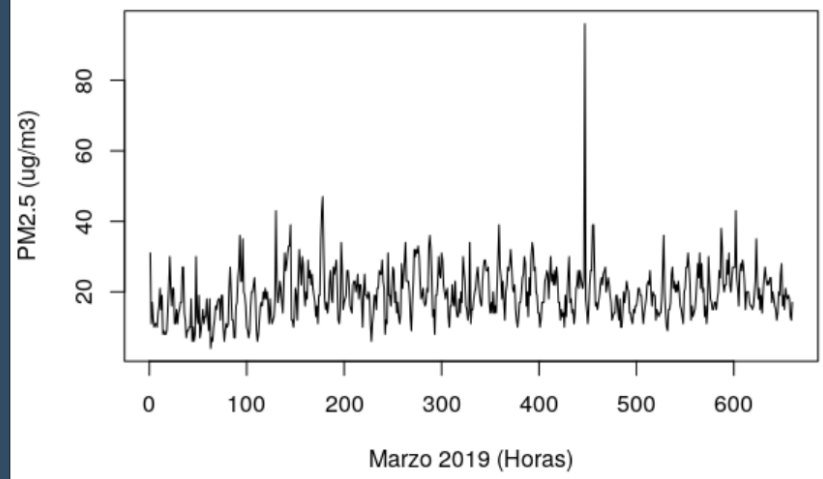
Data

| | |
|---|---|
| ans | 1 obs. of 5 variables |
| mar19 | 660 obs. of 14 variables |

Values

Files Plots Packages Help Viewer

Zoom | Export | Publish



Curso_R_UNALM_ciencias - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Clase_2_Scripts_funciones_loops.... | clase_2_primer_script.R | class_2_DRY.R

```
1  # Don't repeat yourself
2
3  ReadPM25Data <- function(file_path){
4    mes <- read.table(file_path, header = T, sep = ",", dec = ".")
5    horas_faltantes <- 31*24 - nrow(mes)
6    max_value <- max(mes$Raw.Conc.)
7    day_max  <- mes$Day[mes$Raw.Conc. == max_value]
8    hour_max <- mes$Hour[mes$Raw.Conc. == max_value]
9    mean_mes <- mean(mes$Raw.Conc.)
10   ans <-  data.frame(miss_hour = horas_faltantes,
11                      max_pm25 = max_value,
12                      max_day = day_max,
13                      max_hour = hour_max,
14                      mean = mean_mes)
15   return(ans)
16 }
17
18 mar19_file <- "02_data/pm25_0319.csv"
19 mar20_file <- "02_data/pm25_0320.csv"
20 mar21_file <- "02_data/pm25_0321.csv"
21
22 ReadPM25Data(mar19_file)
23 ReadPM25Data(mar20_file)
24 ReadPM25Data(mar21_file)
```

11:43 ReadPM25Data(file_path) R Script

Console

Environment History Connections Git Tutorial

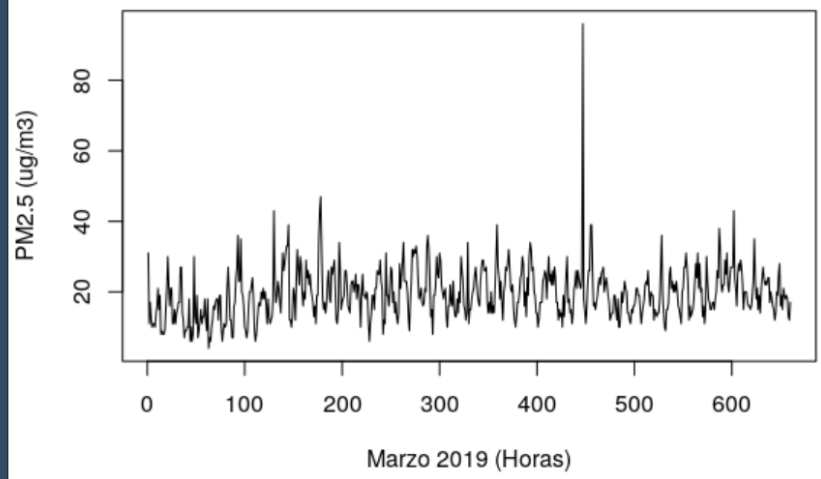
Import Dataset List

Global Environment

| | |
|---|---|
| horas_marzo | 744 |
| hour_max | 2L |
| mar19_file | "02_data/pm25_0319.csv" |
| mar20_file | "02_data/pm25_0320.csv" |
| mar21_file | "02_data/pm25_0321.csv" |

Files Plots Packages Help Viewer

Zoom Export Publish



Curso_R_UNALM_ciencias - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Curso_R_UNALM_ciencias

Clase_2_Scripts_funciones_loops.... | clase_2_primer_script.R | class_2_DRY.R

Source on Save | Run | Source

```
1  # Don't repeat yourself
2
3  ReadPM25Data <- function(file_path){
4    mes <- read.table(file_path, header = T, sep = ",", dec = ".")
5    horas_faltantes <- 31*24 - nrow(mes)
6    max_value <- max(mes$Raw.Conc.)
7    day_max  <- mes$Day[mes$Raw.Conc. == max_value]
8    hour_max <- mes$Hour[mes$Raw.Conc. == max_value]
9    mean_mes <- mean(mes$Raw.Conc.)
10   ans <-  data.frame(miss_hour = horas_faltantes,
11                      max_pm25 = max_value,
12                      max_day = day_max,
13                      max_hour = hour_max,
14                      mean = mean_mes)
15   return(ans)
```

11:43 ReadPM25Data(file_path) R Script

Console | Terminal | R Markdown | Jobs

~/R_tests/Curso_R_UNALM_ciencias/

```
> mar19_file <- "02_data/pm25_0319.csv"
> mar20_file <- "02_data/pm25_0320.csv"
> mar21_file <- "02_data/pm25_0321.csv"
>
> ReadPM25Data(mar19_file)
  miss_hour max_pm25 max_day max_hour  mean
1        84       96      23        2 19.55
> ReadPM25Data(mar20_file)
  miss_hour max_pm25 max_day max_hour     mean
1        74       33       5       20 13.90746
> ReadPM25Data(mar21_file)
  miss_hour max_pm25 max_day max_hour     mean
1       234       36      20       12 14.38627
2       234       36      20       21 14.38627
>
```

Environment | History | Connections | Git | Tutorial

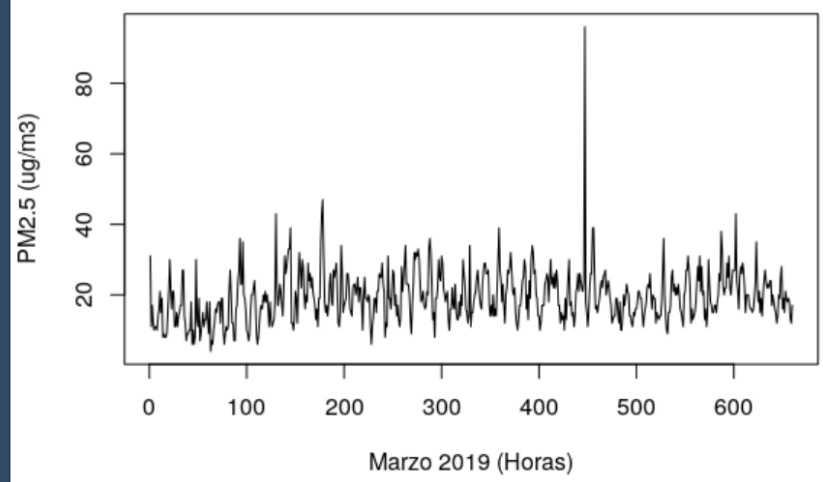
Import Dataset | List

Global Environment

| | |
|---|---|
| horas_marzo | 744 |
| hour_max | 2L |
| mar19_file | "02_data/pm25_0319.csv" |
| mar20_file | "02_data/pm25_0320.csv" |
| mar21_file | "02_data/pm25_0321.csv" |

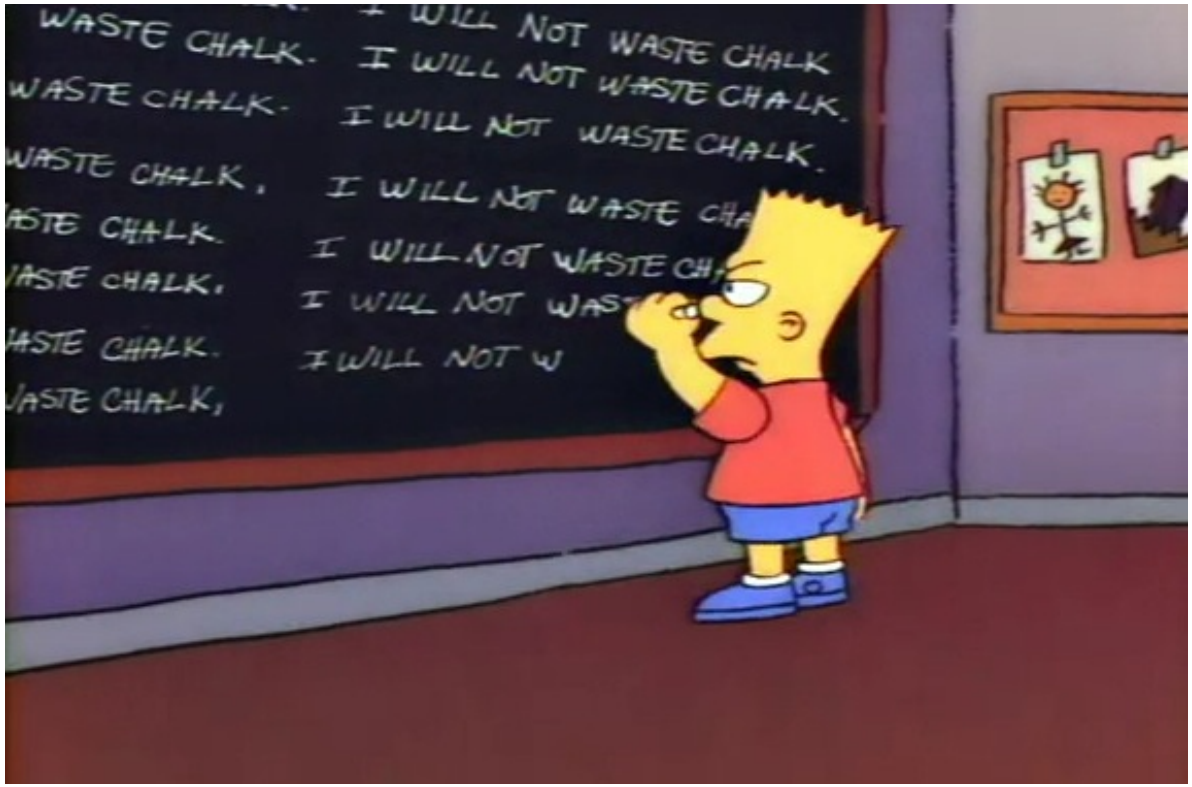
Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



# Loops

# Loops




From: Bart the Genius

Season 1, Episode 2

In: http://bartsblackboard.com/tag/chalk/

# Loops

- Sintaxis

```
for (var in sequence){
  statements
}
```

# Loops

- Haciendo la tarea de Bart

```
for (i in seq(1, 10)){
  print("Ya no gastaré más tiza")
}
```

```
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
## [1] "Ya no gastaré más tiza"
```

# Loops: Ejemplos prácticos

- Por **posición**

```
tempC <- c(20, 22, 23, 25)
for (i in seq(1, length(tempC))){
  print(tempC[i] + 273.15)
}
```

```
## [1] 293.15
## [1] 295.15
## [1] 296.15
## [1] 298.15
```

# Loops: Ejemplos prácticos

- Por **elemento**

```
tempC <- c(20, 22, 23, 25)
for (t in tempC){
  print(t + 273.15)
}
```

```
## [1] 293.15
## [1] 295.15
## [1] 296.15
## [1] 298.15
```

# Loops: Ejemplos prácticos

- Una función a un vector

```
for (i in mi_vector){
  print(CuboNumero(i))
}
```

```
## [1] 1
## [1] 64
## [1] 125
## [1] 216
## [1] 3375
```

# Loops: Advertencia

- Obviamente es más sencillo:

```
tempC + 273.15
```

```
## [1] 293.15 295.15 296.15 298.15
```

```
CuboNumero(mi_vector)
```

```
## [1]    1   64  125  216 3375
```

- Loops son importantes para *automatizar*

# Don't repeat yourself (DRY)

- Es un principio de programación
- Si estás reescribiendo código es momento de:
  - Crear una **función**
  - Hacer un **loop**

# Packages

# Packages

- Conjunto de funciones desarrolladas para tareas ***ad hoc***:
  - Análisis de datas contaminación de aire: `openair`
  - Crear aplicaciones: `shiny`
  - GIS: `sf` y `raster`
  - Webscrapping: `RCurl`
  - *Etcetera*

# Packages

- Instalación

```
install.packages("openair") # Simpre colocar las " "
```

- Utilizar

```
library(openair) # No es neceario las " "
```

- Buena práctica: Colocarlas al inicio de cada script

# Otrosí digo

# Otros objetos

- Existen otros tipos de objetos:
  - listas: `list()`
  - matrizes: `matrix()`
  - factores: `as.factor()`
  - **fechas:** `as.POSIXct()`

# as.POSIXct

- Le dice a R que estos son datos de fechas
- Importante en análisis de series temporales

```
mar19 <- read.table(
  "../02_data/pm25_0319.csv",
  header = T,
  sep = ",",
  dec = "."
)

head(mar19)
```

```
##   Site          Parameter            Date..LT. Year Month
## 1 Lima PM2.5 - Principal 2019-03-01 12:00 AM 2019     3
## 2 Lima PM2.5 - Principal 2019-03-01 01:00 AM 2019     3
## 3 Lima PM2.5 - Principal 2019-03-01 02:00 AM 2019     3
```

# as.POSIXct

- Creamos una columna date como as.POSIXct

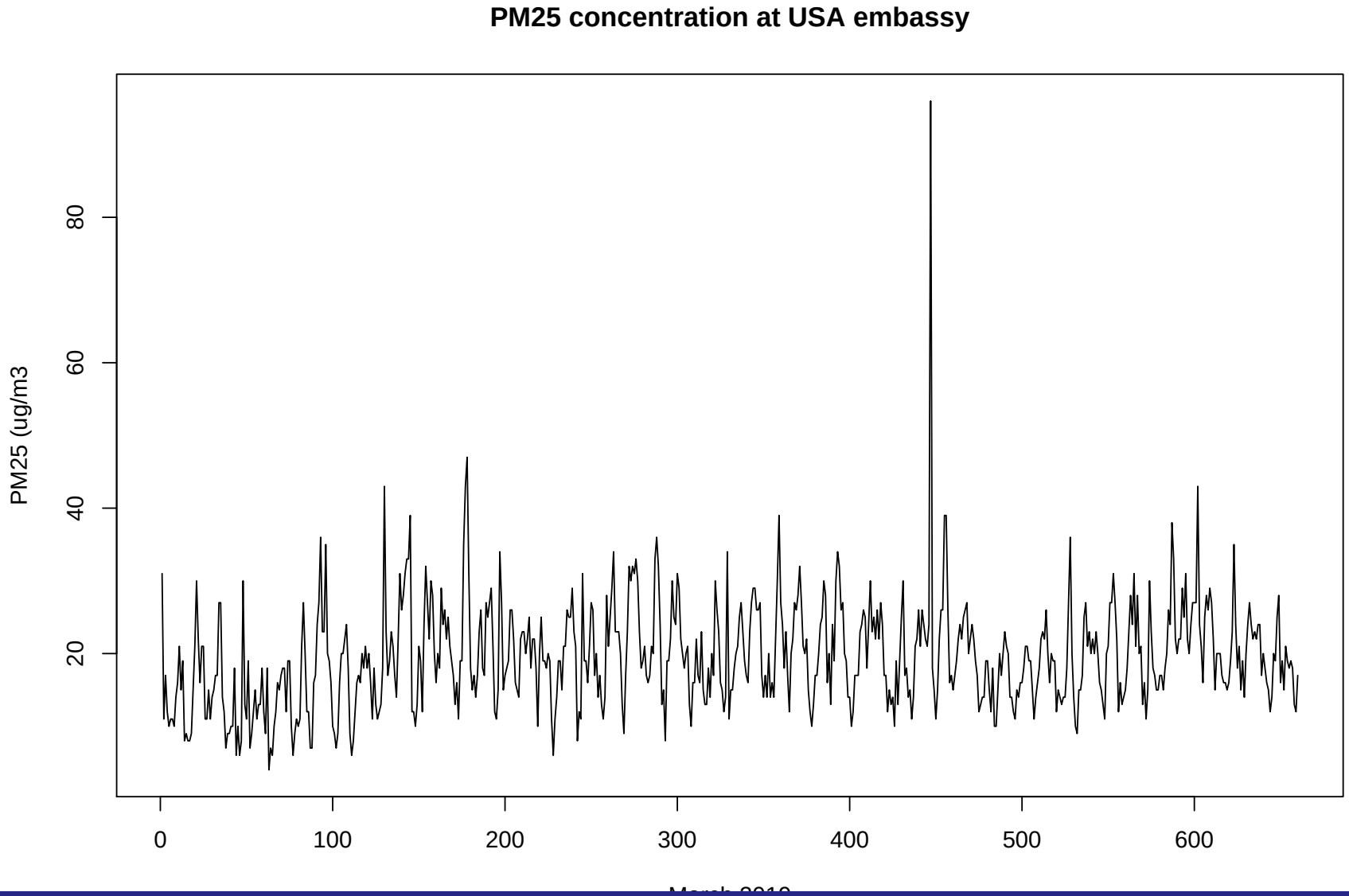
```
mar19$date <- as.POSIXct(
  strptime(
    mar19$Date..LT., format = "%Y-%m-%d %I:%M %p"
  ),
  tz = "America/Lima"
)

class(mar19$date)
```

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

# as.POSIXct

```
plot(mar19$Raw.Conc., t = "l",
     main = "PM25 concentration at USA embassy",
     ylab = "PM25 (ug/m3",
     xlab = "March 2019")
```

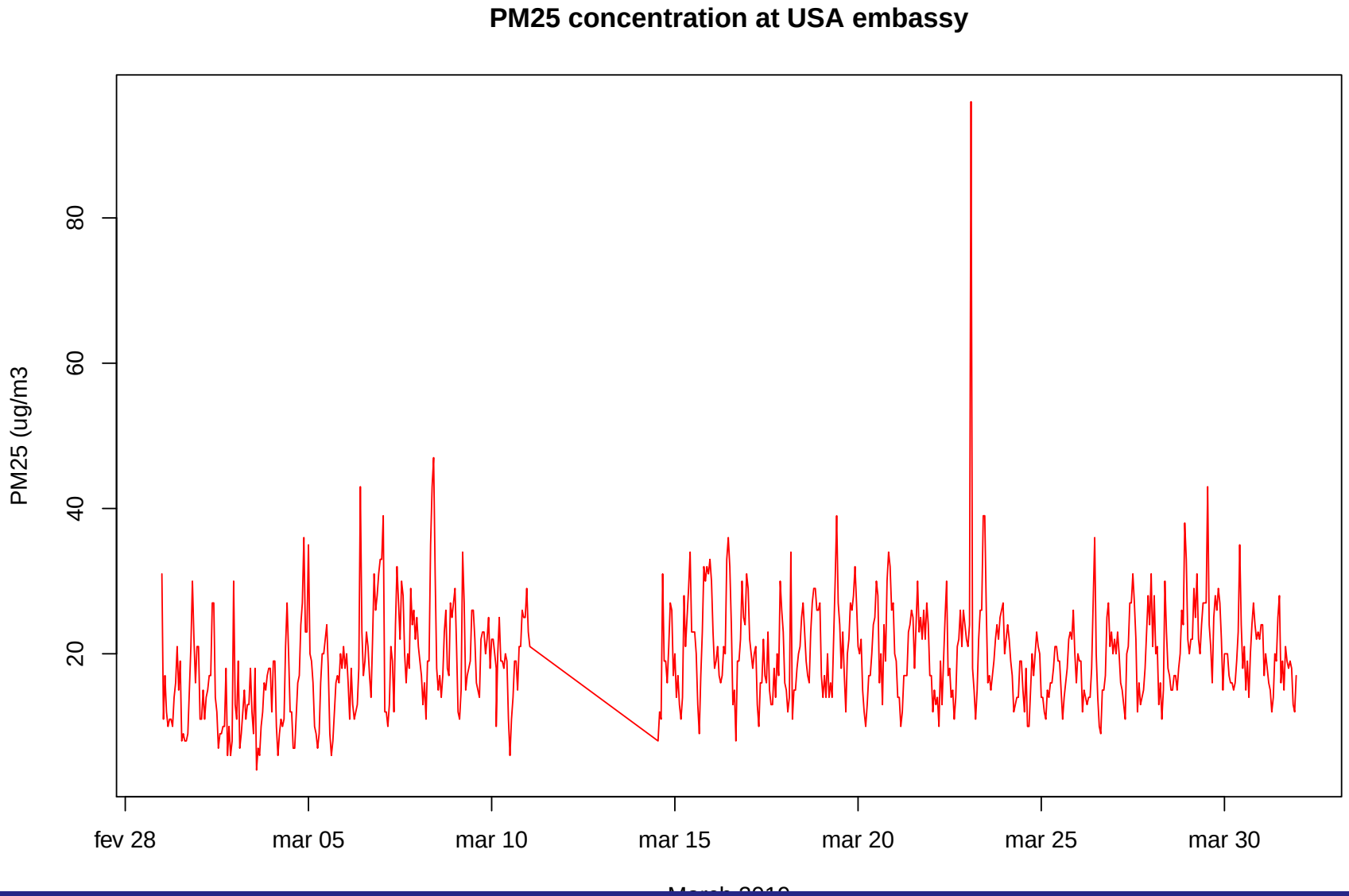
# as.POSIXct


PM25 concentration at USA embassy
PM25 (ug/m3
20
40
60
80
0
100
200
300
400
500
600

# as.POSIXct

```
plot(mar19$date, mar19$Raw.Conc., t = "l", col = "red",
     main = "PM25 concentration at USA embassy",
     ylab = "PM25 (ug/m3",
     xlab = "March 2019")
```

# as.POSIXct


PM25 concentration at USA embassy
PM25 (ug/m3
20
40
60
80
fev 28
mar 05
mar 10
mar 15
mar 20
mar 25
mar 30

# Otras funciones importantes: paste()

- ?paste
  - Concatena vectores luego de convertirlos a character

```
my_name <- "Mario"
my_last_name <- "Gavidia-Calderón"
paste(my_name, my_last_name)
```

```
## [1] "Mario Gavidia-Calderón"
```

# Otras funciones importantes: `paste()`

```
temp <- c(32, 30, 29, 32) + 273.15
paste(temp, "K")
```

```
## [1] "305.15 K" "303.15 K" "302.15 K" "305.15 K"
```

# Otras funciones importantes: as.something()

- Donde something puede ser: character, numeric, data.frame, etc

```
a_character <- "5"
a_character
```

```
## [1] "5"
```

```
class(a_character)
```

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

```
a_number <- as.numeric(a_character)
a_number
```

```
## [1] 5
```

```
class(a_number)
```

# Otras funciones importantes: `write.table()`

```
aire <- data.frame(gas = c("N2", "O2", "Ar", "CO2"),
                   w   = c(28, 32, 40, 12 + 2 * 16),
                   per = c(78.08, 20.95, 0.9, 0.04))
aire
```

```
##   gas  w   per
## 1  N2 28 78.08
## 2  O2 32 20.95
## 3  Ar 40  0.90
## 4 CO2 44  0.04
```

# Otras funciones importantes: `write.table()`

- Exporta en un archivo de texto.

```
aire <- data.frame(gas = c("N2", "O2", "Ar", "CO2"),
                   w   = c(28, 32, 40, 12 + 2 * 16),
                   per = c(78.08, 20.95, 0.9, 0.04))
write.table(aire, "./aire_composition.csv", sep = ",", row
```