Introduction to **R**

Fuertehack 2013
Beatriz Martínez
@maritrinez

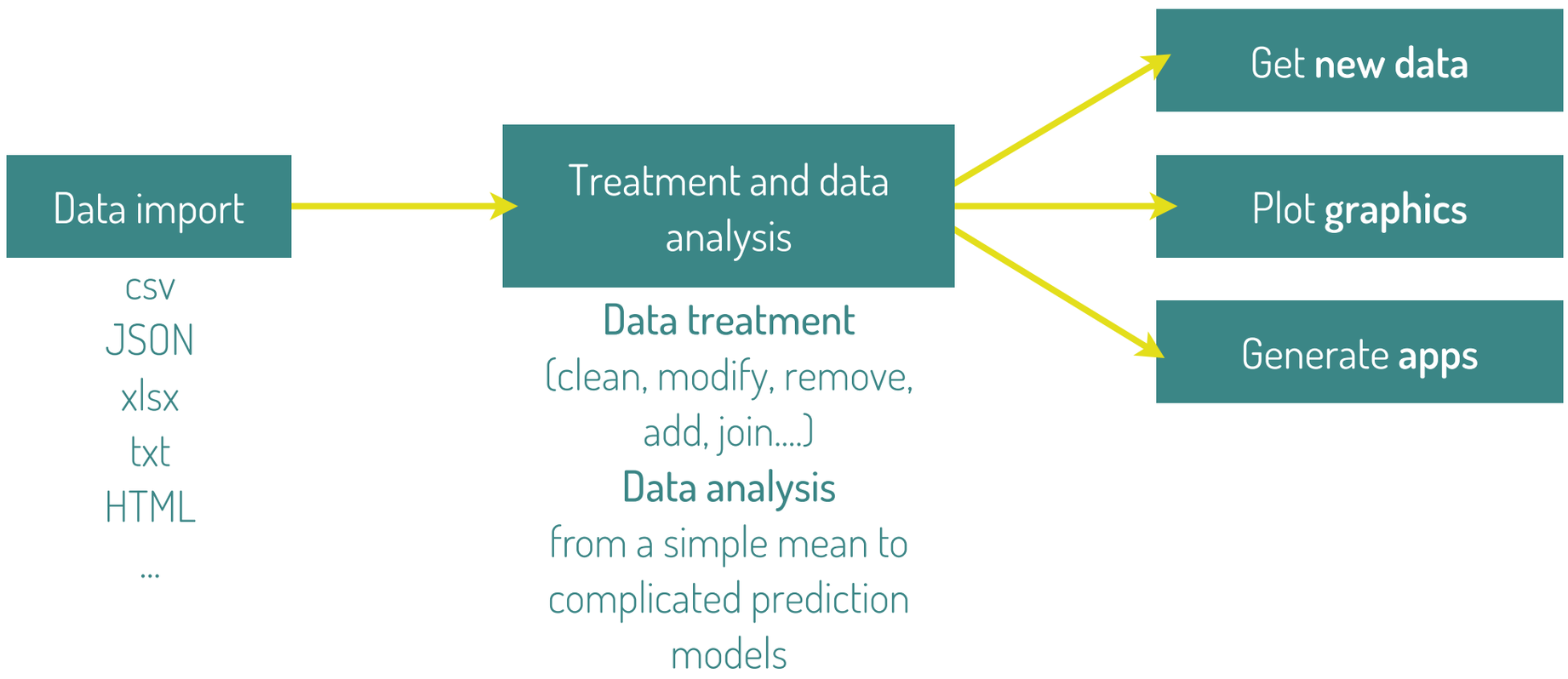
What is **R**

R is a programming language
(object-oriented)

free and Open Source for
statistical computing
and graphics.

Since
★ 1993 ★

What can be done with **R**



With the **advantages** of being a **programming language**:

- Reproducible
- It is easily shared with others
- Almost infinite

How does **R** work?

With functions.

```
data <- read.csv("nombre_archivo.csv", header = TRUE, sep = ",")
```

- `data <-` — object which the result is assigned to
- `read.csv` — function name
- `"nombre_archivo.csv"` — object which the function is passed to
- `header = TRUE, sep = ","` — arguments (with their default values)

- Functions **Self-help**: `?read.csv`

- There are a huge number of functions grouped in '**packages**'. In case you need any function that is not in the default package (named 'base')

```
install.packages("maps")  # Download the package which contains the function.
library(maps)  # Load it into R.
map("world")  # Run the function.
```

Objects

Anything is an object in **R**

Every **object** has a **class** that describes what the object has inside and what every function does with it.

Types of '**atomic**' **objects**:

- character

```
> x <- 'Hola'
> class(x)
[1] "character"
```

- **numeric** (with decimals)

```
> z <- 23.5
> class(z)
[1] "numeric"
```

- integer

```
> v <- 3L  # To be recognized as an integer, it must be written the number followed by an 'L'.
> class(v)
[1] "integer"
```

- complex

```
u <- -13+0i
> class(u)
[1] "complex"
```

- logic (True/False)

```
> w <- TRUE
> class(w)
[1] "logical"
```

Objects

There are objects that are a **combination of objects**.

- **vector** it cannot contain elements of different classes.

```
> character vector  x <- c("ayer","hoy",
"siempre")
> numeric vector  x <- c(3,6,9)
> logic vector  x <- c(TRUE, FALSE, TRUE)
```

- **list** it can contain elements of different classes; it is in fact a list of vectors.

```
> x<-list(1,"a", TRUE, 1+4i)
> class(x)
[1] "list"
> x
[[1]]
[1] 1

[[2]]
[1] "a"

[[3]]
[1] TRUE

[[4]]
[1] 1+4i
```

- **factor** specific type of vector, which are used to represent categorical data (male/female; users/non users....)

```
> x<-factor(c("yes","yes","no", "yes",
"no"))
> x
[1] yes yes no  yes no
Levels: no yes
```

- **matrix** it is a vector with a **dimension** attribute.

```
> x <- c(1:24)
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24
> dim(x) <- c(4,6)
> x
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    5    9   13   17   21
[2,]    2    6   10   14   18   22
[3,]    3    7   11   15   19   23
[4,]    4    8   12   16   20   24
```

Objects

There are objects that are a **combination of objects**.

- **data.frame** it is the star of the **R** objects.
  - a list of vectors where **every element** of the list has to have the **same length**.
  - a kind of table in which is possible to store **objects of different classes**.

```
> x <- data.frame("days" = c("lun", "mar", "mie", "jue", "vie"), "drinks" = c(2,3,0,1,8))
> x
  days drinks
1  lun      2
2  mar      3
3  mie      0
4  jue      1
5  vie      8

> x$days
[1] lun mar mie jue vie
Levels: jue lun mar mie vie

> x$drinks
[1] 2 3 0 1 8
```

Check out the **IntroductionR.R** file at
github.com/maritrinez/fuertehack

Resources

- Great and complete **'Introduction to R Programming'** presentation

  https://dl.dropboxusercontent.com/u/1811289/RBootcamp/INTRO_TO_R_PROGRAMMING_SECTOR_67.html#(1)

- Easy to understand tutorial web with the basics **'Quick-R'**

  http://www.statmethods.net/

- Videos from the Coursera course 'Computing for Data Analysis' (4 weeks)

  http://www.youtube.com/user/rdpeng/videos?view=1&flow=grid

- Two minutes R tutorials

  http://www.twotorials.com/

- **R** Users meetings

  Madrid http://r-es.org/Grupo+de+Inter%C3%A9s+Local+de+Madrid+-+GIL+Madrid&structure=Comunidad

  Barcelona http://rugbcn.wordpress.com/