# Data management with R and RStudio

**Javier Sánchez Bachiller**

**Higher School of Economics**

*jsbachiller.work@gmail.com*

## Why R?

- Easy-to-learn language
- Specially popular in economics and health sciences
- Scalable to big-data ecosystems like Spark
- Large community behind it, constant and steady development
- Used in leading research institutions and universities around the world
- Available in all platforms
- Free

## Why R...

### instead of Stata?

- Freely available
- No OS limitations

### instead of Python?

- More common among finance and economics topics
- Easier to install and set-up (out-of-the-box)

## RStudio

- Best IDE (integrated development environment) for R
- All-in-one: Editor, console, viewer, data and document manager
- Perfect integration with Git
- Great debugging

## Basic R

- We assign variables with `<-`, `->` or `=`. Doing `x = 1`, `x <- 1` or `1 -> x` are exactly equivalent. You can define several variables at once: `x <- y <- z <- 1`.
- Vectors are defined with `c(val1, val2, ...)`
- Boolean operators are `==`, `>`, `<`, `|`. They can be combined: `>=`, `<=`. To negate any of the former, we use `!`.
- Functions follow this structure: `some_function(arg1 = val1, arg2 = val2, ...)`
- Conditional statements can be done by using: `if(conditions) {do this}`. Similar syntax applies for loops: `for(i in set) {do this}`, where `i` is a variable inside the loop that will still exist when it ends.
- Packages are loaded by running `library(package_name)`, and installed by `install.packages("package_name")` (note the quotes!)

## The 'tidyverse'

The tidyverse environment is a set of packages that has become one of the most complete and powerful solutions available in R to do anything related to data cleaning and visualization. It is based on:

- A concrete *grammar* of data manipulation with which code can be written in a sentence-like manner, making it easily readible and very intuitive.
- The concept of *tidy* data: All columns are variables, are rows are observations.
- *Vectorised* operations: No more time-consuming ugly-looking loops.

**The basic idea is to concatenate a series of verbs (functions) to manipulate the data by creating a `pipe`, using the *pipe operator* `%>%`**

## Load the data

When loading the tidyverse, the `readr` package will be automatically loaded too. It allows us to read csv files by using the command `read_csv("file.csv")`.

Should we have other formats of data, additional packages are included, such as `readxl` for excel spreadsheets and `haven` for Stata or SPSS files. They just need to be loaded and then can be used the same way, that is, by using the command `read_*("file.*")` and replacing `*` by the desired extension.

**Good news:** We can export the resulting data using the same command but replacing `read` by `write`.

## Load the data

```r
library(tidyverse)
library(haven)
library(readxl)



read_csv("exampleData.csv") -> csv_data
read_csv("exampleData.dta") -> dta_data
read_csv("exampleData.xls") -> excel_data
```

# Manage the data

The `dplyr` package from the tidyverse is the main workhorse regarding data manipulation.

## Data browsing

Columns can be easily subsetted with `select`, rows by using `filter`. Sorting the data is carried out by `arrange` and retrieving the unique values of a column can be achieved with `distinct`.

## Data transformation

There are two kind of functions to be used: *summary* functions (which retrieves information from all data points) or *vectorised* functions (which applies a transformation to each data point) to be used and then the operation to be performed.

## Cheatsheets

You can find cheatsheets for the `tidyverse` (and some other useful packages) with all relevant functions and a quick summary of how to use them in here.

More concretely, the ones we have been mostly touching upon and will be useful to deal with messy data are the ones for the dplyr package (data tranformation) and for the stringr package (string manipulation).

Thanks for your attention!

Спасибо за внимание!