

# Research Methods for Economics and Policy Data and Methods

Lorenzo Neri<sup>1</sup>

<sup>1</sup> University of Rome "Tor Vergata", CESifo and IZA

April-May 2025

# Programming in R

- ▶ First, R is an interpreted language, not a compiled one, meaning that all commands typed on the keyboard are directly executed
- ▶ Language is case sensitive; Commands are separated either by a semi-colon ; or by a newline
- ▶ Comments can be put almost anywhere, starting with a hashmark #, everything to the end of the line is a comment
- ▶ When R is running, variables, data, functions, results, etc, are stored in the active memory of the computer in the form of objects which have a name
- ▶ The collection of objects currently stored is called the workspace

# Variables

- ▶ Variables are one of the basic data types in R that store values and are an important component in R programming
- ▶ A variable in R can store a number, an object, a statistical result, vector, dataset, a model prediction basically anything R outputs.
- ▶ We can use that variable later simply by calling the name of the variable.
- ▶ To declare variable data structures in R, we need to assign a variable name. The name should not have space
- ▶ To add a value to the variable in data types in R programming, use `<-` or `=`.

# The pipe operator

- ▶ The pipe operator is %>%
- ▶ It takes the output of one statement and makes it the input of the next statement. When describing it, you can think of it as a "THEN".

```
84 df.sum <- mtcars %>%  
85   select(mpg, cyl, vs, am, gear, carb) %>% # select variables to summarise  
86   summarise_each(funs(min = min,  
87                       q25 = quantile(., 0.25),  
88                       median = median,  
89                       q75 = quantile(., 0.75),  
90                       max = max,  
91                       mean = mean,  
92                       sd = sd))  
93
```

- ▶ Similar to the + operator that we will use with *ggplot*

## Basic commands

- ▶ `objects()` or `ls()`: display the names of (most of) the objects which are currently stored within R.
- ▶ `rm()`: remove objects
- ▶ `print()`: display the content of an object (can also type the object itself)
- ▶ `help()`: display the online help file of a function (can also use `?` in front of the function or `help("")`)

# Functions

- ▶ All arithmetic and statistical functions are available; for example,
  - `log`, `exp`, `sin`, `cos`, `tan`, `sqrt`
  - `max()`, `min()`, `length()`
  - `mean()`, `var()`
  - ...

# Objects

- ▶ Objects are characterized by their names and their content, but also by attributes which specify the kind of data represented by an object
- ▶ All objects have two intrinsic attributes: `mode` and `length`:
  - Four main modes: numeric, character, complex, and logical (FALSE or TRUE)
  - The length is the number of elements of the object
- ▶ To display the mode and the length: use the functions `mode` and `length`
- ▶ A value of mode character is input with double (") or single (') quotes

# Objects

object	modes	several modes possible in the same object?
vector	numeric, character, complex <i>or</i> logical	No
factor	numeric <i>or</i> character	No
array	numeric, character, complex <i>or</i> logical	No
matrix	numeric, character, complex <i>or</i> logical	No
data frame	numeric, character, complex <i>or</i> logical	Yes
ts	numeric, character, complex <i>or</i> logical	No
list	numeric, character, complex, logical, function, expression, ...	Yes

## Reading and saving data (1)

- ▶ First of all, need to set the directory (i.e. the path to your files): use `setwd("")`
- ▶ R can read data stored in text (ASCII) files with `read.table` (which has several variants, see below) `scan` and `read.fwf`
- ▶ But R can read anything you want... - Excel, SAS, SPSS,... - and access SQLtype databases
- ▶ You can create a data frame using:  
`mydata <- read.table("")`  
and access individual variables `V1`, `V2`, etc using `mydata$V1`, `mydata$V2` or `mydata["V1"]`, `mydata["V2"]`

## Reading data (2)

```
read.table(file, header = FALSE, sep = "", quote = "\"'", dec = ".", row.names, col.names,  
as.is = FALSE, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE,  
fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#")
```

## Saving data

- ▶ The function `write.table` writes in a file an object (see `help(write.table)`)
- ▶ To write in a simpler way an object in a file, the command `write(x, file="data.txt")` can be used, where `x` is the name of the object (which can be a vector, a matrix, or an array). Two options:
  - `ncol`: defines the number of columns
  - `append`: (logical) add the data without deleting those possibly already in the file (TRUE) or deleting them if the file already exists (FALSE, the default)
- ▶ To record a group of objects of any type, you can use `save(x, y, z, file= "xyz.RData")`.

# References

- ▶ Emmanuel Paradis. "R for beginners"
- ▶ W. N. Venables, D. M. Smith and the R Core Team. "An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics".