

# 00-IntRoduzione

Ottavia M. Epifania, Ph.D

Lezione di Dottorato @Università Cattolica del Sacro Cuore (MI)

8-9 Giugno 2023

# Table of contents

---

- 1 PeRché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi
- 5 L'ambiente e l'ordine
- 6 Working directories

# Materiale del corso

---



# Table of Contents

---

- 1 PeRché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi
- 5 L'ambiente e l'ordine
- 6 Working directories

# Le cose belle

---

- Open-source: R è gratuito, RStudio (la IDE di R) no, ma la versione free è più che sufficiente
- Permette la replicabilità dei risultati → è gratis
- *R community is the beast feature of R*
- Aiuto online e gratuito

# Le cose un po' meno belle

---

- Difficile da imparare all'inizio
- Non è intuitivo (all'inizio) → se non si ha già una vaga idea di dove partire non si riesce a fare nulla

# Le cose cose belle nelle cose brutte

---

- Più dimestichezza nell'analisi dei dati, più conoscenza del dato, modelli più complessi
- Permette di addentrarsi sempre di più nei linguaggi di programmazione
- Imparare un altro linguaggio di programmazione dopo aver imparato R è (quasi) una passeggiata

# Table of Contents

---

- 1 Perché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi
- 5 L'ambiente e l'ordine
- 6 Working directories



Tol - master - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

```

1 # DATA PREPARATION TOL
2 # OTTAVIA, MAGGIO 2023
3 #
4 rm(list = ls())
5 library(lubridate)
6 library(dplyr)
7
8 setwd("H:/./shortcut-targets-by-id/1740KGGGxe6z7MnN0OcT3W23DXIim")
9
10 name.data.43 = paste0("AdapTo1_43/", # do not change
11                       "to143_2023_05_08.csv") # change according t
12 name.data.52 = paste0("AdapTo1_52/", # do not change
13                       "to152_2023_05_08.csv") # change according t
14 name.data.45 = paste0("AdapTo1_45/", # do not change
15                       "to145_2023_05_08.csv") # change according t
16
17 name.env = ls()
18
19 for (i in 1:length(name.env)) {
20   assign(gsub("name.", "", name.env[i]),
21         read.csv(get(name.env[i]), header = T, sep = ","))
22 }
23
24 total_time

```

```

R R 4.2.3 - H:/shortcut-targets-by-id/1740KGGGxe6z7MnN0OcT3W23DXIim/V/PyqAssist/
> ggplot(a11.data, aes(x = n_moves, y = as.numeric(preplanning))) + geom_
point()
warning message:
Removed 8 rows containing missing values (geom_point).
>

```

Environment History Connections Tutorial

R - Global Environment

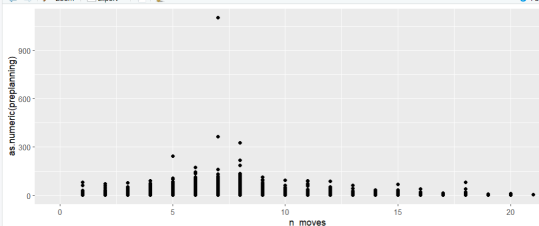
Object	Value
total_time43	378 obs. of 29 variables
total_time45	166 obs. of 44 variables
total_time52	412 obs. of 36 variables
wide.acc	412 obs. of 28 variables

Values

Object	Value
alleged.trials	num [1:3] 20 35 27
i	12L
name_desc	chr [1:3] "sbj_char43" "sbj_char45" "sbj_char52"
name.data.43	"AdapTo1_43/to143_2023_05_08.csv"
name.data.45	"AdapTo1_45/to145_2023_05_08.csv"
name.data.52	"AdapTo1_52/to152_2023_05_08.csv"
name.env	chr [1:3] "data.43" "data.45" "data.52"
names.print	chr [1:12] "accuracy43" "accuracy45" "accuracy52" "execution43" "execution45" "execution52"

Files Plots Packages Help Git Viewer Presentation

Zoom Export



# console vs. script

## Console

I comandi nella console vengono eseguiti e non salvati. Si può accedere ai comandi eseguiti in precedenza cliccando sulla freccina in alto.

Per eseguire il comando → Invio (o Enter)

L'output è immediato ed appare nella console

# console vs. script

## Console

I comandi nella console vengono eseguiti e non salvati. Si può accedere ai comandi eseguiti in precedenza cliccando sulla freccina in alto.

Per eseguire il comando → Invio (o Enter)

L'output è immediato ed appare nella console

## Script

è possibile salvare gli script con tutti i comandi salvati

Per eseguire il comando → Ctrl + Invio (o ctrl + Enter o cmd + Enter)

L'output è restituito nella console

Si possono scrivere delle righe di codice non eseguite → righe di codice commentate con # all'inizio

Per passare alla console → ctrl + 2

Per passare allo script → ctr + 1

# Table of Contents

---

- 1 PeRché?
- 2 Come è fatto
- 3 Con cosa lavoriamo**
- 4 Le basi
- 5 L'ambiente e l'ordine
- 6 Working directories

La gestione delle working directory è una delle cose meno intuitive di R

Rischiate di perdere dei pezzi e di non avere tutto ordinato nelle cartelle

Per risolvere questo problema → si possono usare gli R project (progetti R)

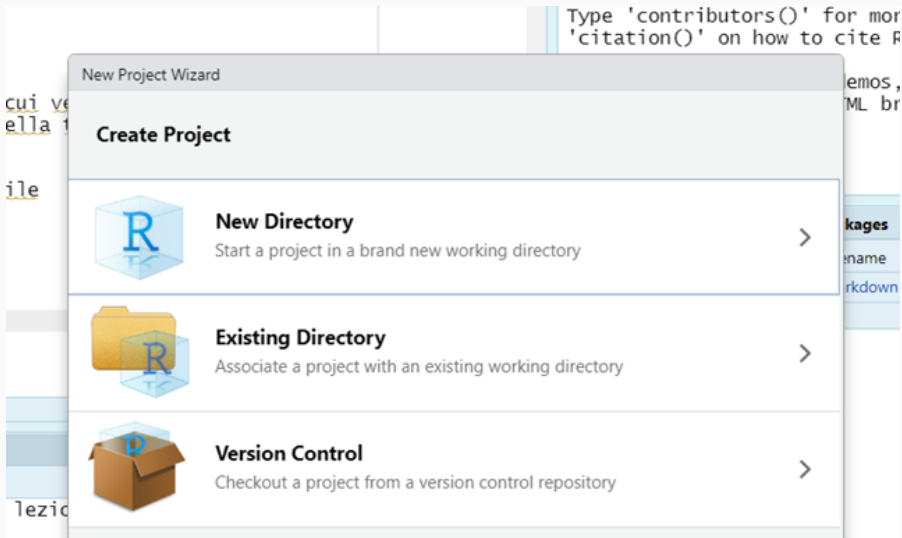
Un progetto R crea una sua directory, tutti i file che vengono salvati al suo interno sono sempre accessibile senza bisogno di settare *a mano* tutte le volte la directory

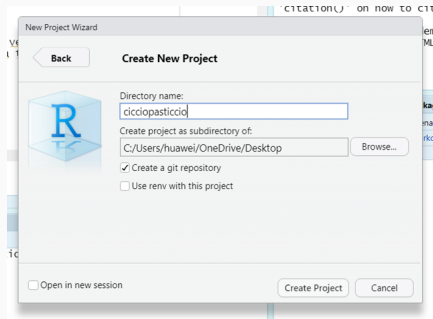
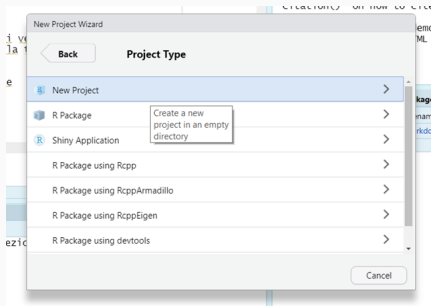
Sono molto comodi perché:

- 1 Permettono di avere più istanze R aperte contemporaneamente → è possibile lavorare su più progetti contemporaneamente
- 2 Tenendo tutti i file ordinati vi permettono di poter risalire a cosa avete fatto mesi prima (utile quando vi arrivano le revisioni di un paper)

# Creare un progetto R

File → New project:





# Table of Contents

---

- 1 Perché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi**
- 5 L'ambiente e l'ordine
- 6 Working directories



# Simboli, variabili, funzioni

---

Simboli   variabili   funzioni

# Simboli

## Operatori:

```
3 + 2    # più
3 - 2    # meno
3 * 2    # per
3 / 2    # diviso
5 > 2    # maggiore di
2 < 5    # minore di
5 == 3 + 2 # uguale
5 != 3    # diverso
```

## Parentesi e altri simboli

() [] {} " " : ; ,

Operatori e parentesi si possono combinare insieme per risolvere equazioni:

```
(15 + 22)/(13 * 4)
```

```
[1] 0.7115385
```

# Your turn!

---



- Create un progetto R per questo corso (sarà il progetto che useremo sempre!)
- Create un nuovo script (`shift + ctrl + n`)
- Provate a usare R come una calcolatrice
  - ① Usando lo script
  - ② Usando la console
- Salvate lo script

# Variabili

---

La variabile (in senso statistico) viene registrata in una variabile (in senso informatico)

## Variabile (Informatica)

è un oggetto che contiene informazione

Si distinguono in base al tipo di informazione che contengono, che in R può essere:

- **int**: numeriche (discrete, 1, 2, 3 ....)
- **num**: numeriche (continue, 0.01, 0.02, 0.03, ...)
- **logi**: logiche (TRUE, FALSE)
- **chr**: caratteri ("a", "B", "c", ...)
- **factor**: fattori distinti da diversi livelli

# Nomi delle variabili

Alle variabili si può dare il nome che si vuole → non può iniziare con un numero

`modello_1` → Sì , `1Modello` → No

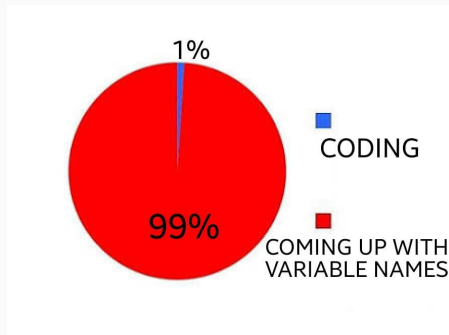
## Warning!

Attenzione perché R è case sensitive:

`A`  $\neq$  `a`

Altro vincolo: Non si possono usare nomi che possono essere confusi con funzioni/oggetti interni di R:

‘FALSE’ è un oggetto logico di R, non possiamo usarlo come nome di una nostra variabile.



Molto bene

`media_maschi, modello1,`  
`statistiche_descrittive`

Molto male

`x3, x2, x1`

# Assign

---

Le variabili vengono create “assegnando” loro i risultati delle operazioni

Esistono fondamentalmente due comandi di assegnazione

- 1  $x = \exp(2^2) =$  ( non piace molto ai puristi di R ma è uguale a Python
- 2  $X \leftarrow \log(2^2) =$ ) Fate contenti i puristi di R

La logica è che l'oggetto a destra (l'operazione e quindi il risultato che se ne ottiene) viene assegnato (= oppure  $\leftarrow$ ) alla variabile (o oggetto) a destra

**Attenzione!** Siccome R è case sensitive:  $x$  e  $X$  sono due oggetti differenti!

# Le funzioni

---

```
my.function(arg1, arg2, arg3 = default)
```

Sono definite dal loro nome (che solitamente riflette cosa fanno – la loro funzione), un paio di parentesi tonde

AL loro interno si possono settare diversi argomenti:

- Alcuni hanno dei valori di default → vengono applicati a meno che l'utente non li cambi
- Altri argomenti sono obbligatori e vanno settati dall'utente



# Un esempio

---

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

- `mean`: è il nome della funzione per calcolare la media
- `x`: è il primo argomento della funzione, indica che va passato alla funzione un oggetto `x` (una variabile) su cui calcolare la media
- `trim`: secondo argomento della funzione con un default (`trim = 0`), specifica se applicare il trimming sui dati prima del calcolo della media
- `na.rm`: terzo argomento della funzione con un default (`na.rm = FALSE`), determina il trattamento dei dati mancanti
- `...` indica che si possono passare altri argomenti alla funzione

## concatenate

---

è la funzione più usata su R

`c()`

Serve per concatenare diversi oggetti (variabili) per combinarli in un unico oggetto →

```
x = c(1, 2, 3) # crea un vettore con tre numeri  
x
```

```
[1] 1 2 3
```

```
X = 1:3 # crea lo stesso identico vettore  
X
```

```
[1] 1 2 3
```

```
x == X
```

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

# Aiuto

---

R community is the best feature of R

Copia & Incolla l'errore su google

Se non si sa fare qualcosa “how to **[something]** in r”

Chiedere a R: Nella consolle si può scrivere `?nome.funzione()` per accedere alla documentazione sulla funzione:

`?mean()`

Apri la documentazione sulla funzione `mean()`

# Your turn!

---



- Create un nuovo script nel vostro progetto
- Assegnate un vettore composto da numeri a una variabile
- Calcolate la media del vettore con la funzione `mean()`
- Calcolate la media del vettore usando la funzione `sum()`
- Trovate il minimo e massimo del vettore

# Table of Contents

---

- 1 PeRché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi
- 5 L'ambiente e l'ordine**
- 6 Working directories

# Take out the trash

---

L'ambiente di R dovrebbe essere sempre ordinato

Gli oggetti che non servono più andrebbero eliminati

MA si può anche togliere tutto insieme

```
ls() # lista gli oggetti che sono nell'environment  
rm(A) # rimuove l'oggetto A all'environment  
rm(list=ls()) # rimuove tutto dall'environment
```

# Save the environment

---

A volte conviene salvare non solo lo script di R ma anche i risultati che avete ottenuto con quello script

```
save.image("my-computations.RData")
```

Questo ambiente può essere ricaricato nell'ambiente di R:

```
load("my-Computations.RData")
```

# Table of Contents

---

- 1 Perché?
- 2 Come è fatto
- 3 Con cosa lavoriamo
- 4 Le basi
- 5 L'ambiente e l'ordine
- 6 Working directories



Se decidete di non usare i progetti di R (non è una buona idea, choose your own adventure) dovete conoscere le vostre directory:

```
getwd() # restituisce la working directory in cui siete
```

```
dir() # restituisce tutti gli oggetti che sono nella wd
```

Per cambiare la working directory:

```
setwd("C:/Users/huawei/OneDrive/Documenti/GitHub/coRso")
```