

# 1.3\_funzioni&errori

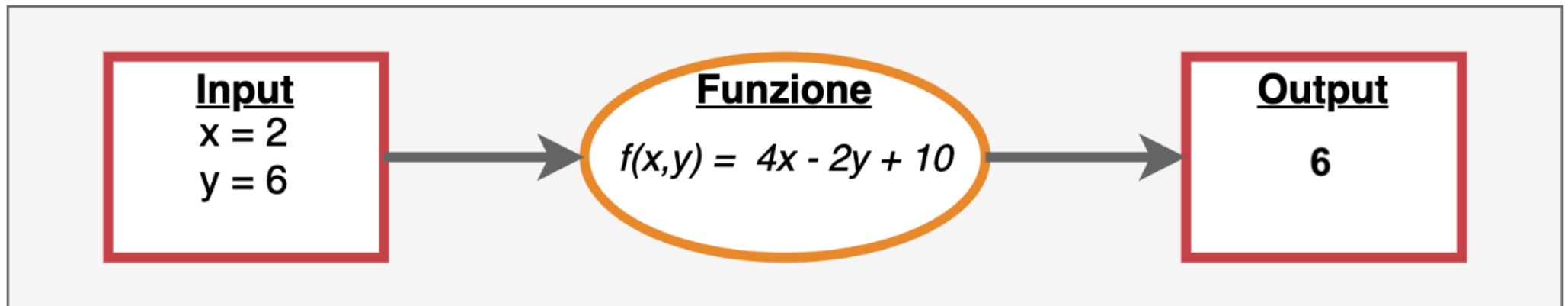
# Funzioni

Le funzioni sono un argomento relativamente complesso ed avanzato. Lo tratteremo più avanti. Siccome le usiamo fin da subito (**es.**, **mean**) è importante avere chiari alcuni aspetti:

- Funzioni come oggetti
- Argomenti obbligatori, opzionali e default
- Ordine degli argomenti
- Documentazione

# Funzioni

Possiamo pensare alle funzioni in R in modo analogo alle classiche funzioni matematiche. Dati dei valori in **input**, le funzioni eseguono dei specifici calcoli e restituiscono in **output** il risultato ottenuto.



# Funzioni come oggetti

Abbiamo già visto che ogni cosa in R è un oggetto. Anche le funzioni seppur molto diverse da altri elementi sono creati e trattati in R come oggetti. In questo esempio, creiamo una funzione che ha come input  $x$ , e che svolge l'operazione di **sommare** ad  $x$  il valore 3.

L'output dipenderà dal valore di  $x$ .

```
1 myfun = function(x) {  
2   return(x + 3)  
3 }  
4  
5 myfun(2)
```

```
[1] 5
```

```
1 myfun(7)
```

```
[1] 10
```

# Funzioni come oggetti

Possiamo crearle, eliminarle o sovrascriverle come un normale oggetto. Vedremo più avanti come crearle ma tenete in considerazione che tutte le funzioni che usiamo sono create come oggetti e salvati nell'ambiente.

# Argomenti

Gli argomenti delle funzioni sono quelli che da *utenti* dobbiamo conoscere ed impostare nel modo corretto per fare in modo che la funzioni faccia quello per cui è stata pensata. Nell'esempio precedente l'unico argomento era `x`. Vediamo invece l'`help` della funzione `mean()`.

# Argomenti

Per impostare questi argomenti ci sono 2 regole:

- l'ordine non conta SE DEFINISCO NOME DELL'ARGOMENTO con `x = vettore, na.rm = TRUE`, etc.
- l'ordine conta SE NON DEFINISCO IL NOME DELL'ARGOMENTO. Posso quindi omettere `argomento = valore` ma devo rispettare l'ordine con cui è stata scritta la funzione

# Argomenti

In questo caso proviamo ad usare la funzione `mean()`:

```
1 #creo un vettore campionando 100 elementi da una distribuzione normale con
2 myvec = rnorm(n = 100, mean = 10, sd = 5)
3 mean(x = myvec) # x definito, trim non definito, na.rm non definito
```

```
[1] 10.69628
```

```
1 mean(x = myvec, trim = 0.10) # x definito, trim definito, na.rm non definit
```

```
[1] 10.58379
```

```
1 mean(x = myvec, na.rm = TRUE) # x definito trim non definito, na.rm definit
```

```
[1] 10.69628
```

```
1 mean(x = myvec, TRUE) # cosa succede?
```

```
Error in mean.default(x = myvec, TRUE): 'trim' must be numeric of length one
```



# Packages

In R è possibile installare e caricare pacchetti aggiuntivi che non fanno altro che rendere disponibili librerie di funzioni create da altri utenti. Per utilizzare un pacchetto:

- Installare il pacchetto con **`install.packages("nomepacchetto")`**
- Caricare il pacchetto con **`library(nomepacchetto)`**
- Accedere ad una funzione senza caricare il pacchetto **`nomepacchetto::nomefunzione()`**. Utile se serve solo una funzione o ci sono conflitti)

# Packages

Files

Plots

Packages

Help

Viewer

Presentation

⬇

Install

🔄

Update

🔍

↺

Name

Description

Version

System Library

☐

abind

Combine Multidimensional Arrays

1.4-5

🌐

✖

☐

afex

Analysis of Factorial Experiments

1.3-1

🌐

✖

☐

AlgDesign

Algorithmic Experimental Design

1.2.1

🌐

✖

☐

antiword

Extract Text from Microsoft Word Documents

1.3.4

🌐

✖

☐

apaTables

Create American Psychological Association (APA) Style Tables

2.0.8

🌐

✖

☐

ape

Analyses of Phylogenetics and Evolution

5.7-1

🌐

✖

☐

appRiori

Code and Obtain Customized Planned Comparisons with 'appRiori'

0.0.1

🌐

✖

☐

arm

Data Analysis Using Regression and Multilevel/Hierarchical Models

1.13-1

🌐

✖

☐

arrayhelpers

Convenience Functions for Arrays

1.1-0

🌐

✖

☐

AsioHeaders

'Asio' C++ Header Files

1.22.1-2

🌐

✖

☐

askpass

Safe Password Entry for R, Git, and SSH

1.1

🌐

✖

☐

assertthat

Easy Pre and Post Assertions

0.2.1

🌐

✖

☐

audio

Audio Interface for R

0.1-11

🌐

✖

☐

backports

Reimplementations of Functions Introduced Since R-3.0.0

1.4.1

🌐

✖

☒

base

The R Base Package

4.2.3

☐

base64enc

Tools for base64 encoding

0.1-3

🌐

✖

☐

BayesFactor

Computation of Bayes Factors for Common Designs

0.9.12-4.7

🌐

✖

☐

bayesplot

Plotting for Bayesian Models

1.10.0

🌐

✖

# Come risolvere i problemi in R

In R gli errori sono:

- inevitabili
- parte del codice stesso
- educativi

# R ed errori

Ci sono diversi livelli di **allerta** quando scriviamo codice:

- **messaggi**: la funzione ci restituisce qualcosa che è utile sapere, ma tutto liscio
- **warnings**: la funzione ci informa di qualcosa di *potenzialmente* problematico, ma (circa) tutto liscio
- **error**: la funzione non solo ci informa di un **errore** ma le operazioni richieste non sono state eseguite

# Come risolvere?

- Capire il messaggio
- Leggere la documentazione della funzione
- Cercare il messaggio su internet
- Chiedere aiuto nei forum dedicati

# La documentazione della funzione

- Ogni funzione ha una pagina di documentazione accessibile con `?nomefunzione`, `??nomefunzione` oppure `help(nomefunzione)`
- Possiamo cercare anche la documentazione del pacchetto
- Possiamo cercare su internet il nome della funzione o l'eventuale messaggio che riceviamo

# Stack overflow

**Stack overflow** è un forum di discussione riguardo qualsiasi cosa coinvolga codice (statistica, programmazione, etc.). E' pieno di errori comuni, *How to do ...* e di risposte/soluzioni estremamente utili. Nel 90% dei casi il problema che avete è comune ed è già presente una soluzione.

stackoverflow About Products OverflowAI Search...

Home Questions Tags Users Companies LABS Jobs Discussions COLLECTIVES TEAMS

## R: mean function not working? [closed]

Asked 9 years, 6 months ago Modified 9 years, 6 months ago Viewed 4k times Part of R Language Collective

**Closed.** This question is [not reproducible or was caused by typos](#). It is not currently accepting answers.

This question was caused by a typo or a problem that can no longer be reproduced. While similar questions may be [on-topic](#) here, this one was resolved in a way less likely to help future readers. Closed 9 years ago.

[Improve this question](#)

I'm trying to do a simple calculation with my `mean` function. When I input `mean(1, 10, 100)`, R returns `1`, which is obviously not the correct average. It always returns the first entry of my vector. What went wrong?

R r

Share Improve this question Follow

Asked May 19, 2015 at 8:43

# ChatGPT

## Attenzione

- Ponete bene la domanda
- Sinceratevi di aver capito la risposta
- Controllate!