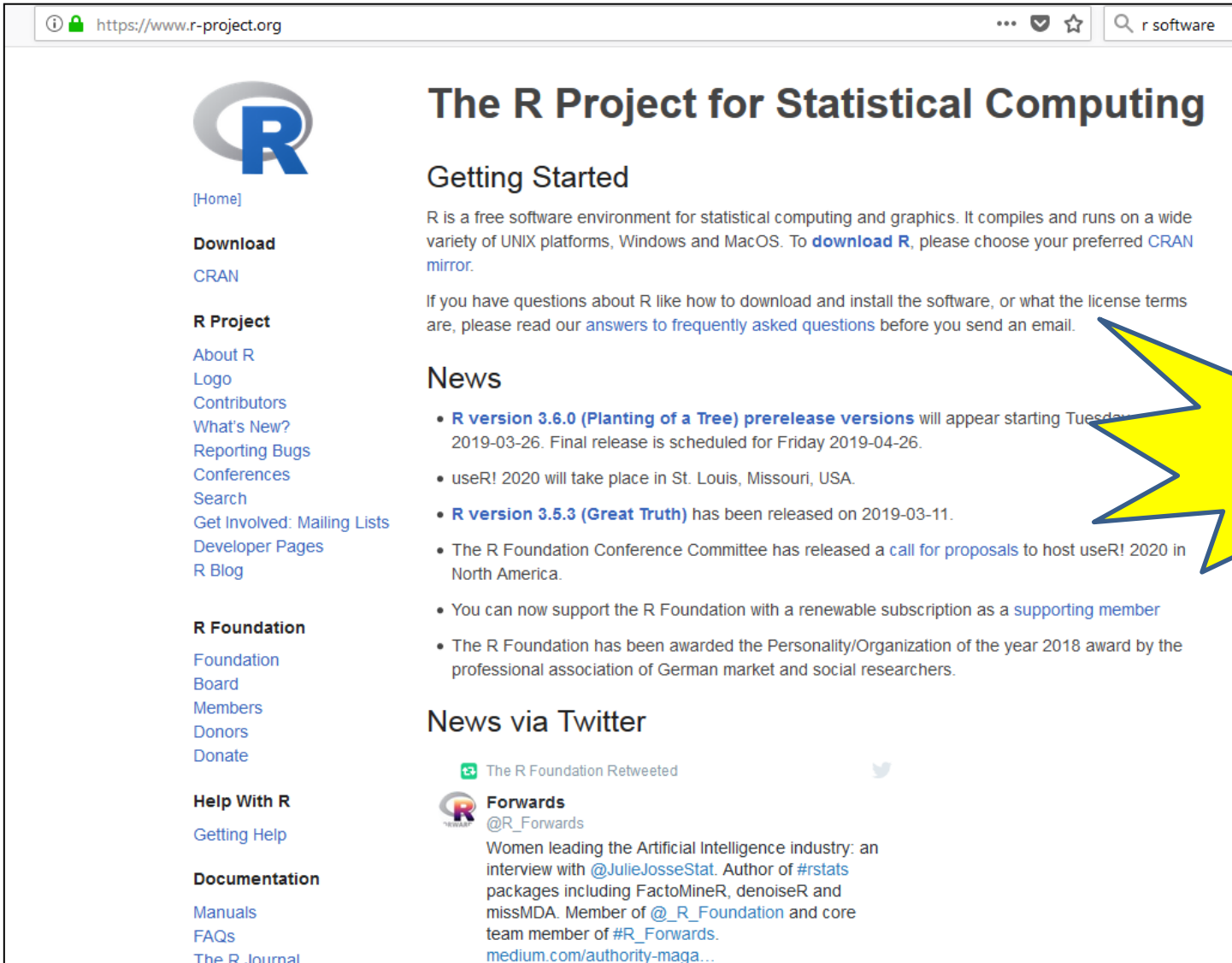


Introduzione alla chemiometria e disegno sperimentale

Modulo 1: Il Software R

Docente: Dr. Sabina Licen (slicen@units.it)

II Software R



https://www.r-project.org

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).


If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.6.0 (Planting of a Tree) prerelease versions** will appear starting Tuesday 2019-03-26. Final release is scheduled for Friday 2019-04-26.
- useR! 2020 will take place in St. Louis, Missouri, USA.
- **R version 3.5.3 (Great Truth)** has been released on 2019-03-11.
- The R Foundation Conference Committee has released a [call for proposals](#) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

News via Twitter

The R Foundation Retweeted

 **Forwards**
@R_Forwards

Women leading the Artificial Intelligence industry: an interview with [@JulieJosseStat](#). Author of [#rstats](#) packages including FactoMineR, denoiseR and missMDA. Member of [@_R_Foundation](#) and core team member of [#R_Forwards](#).
medium.com/authority-maga...

Navigation:
[Home]
Download
CRAN
R Project
About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Developer Pages
R Blog
R Foundation
Foundation Board
Members
Donors
Donate
Help With R
Getting Help
Documentation
Manuals
FAQs
The R Journal

SOFTWARE FREE

<https://www.r-project.org/>



Caratteristiche di R

R è un ambiente integrato di risorse software per:

- ✓ la **manipolazione** dei dati;
 - ✓ l'**elaborazione statistica** dei dati;
 - ✓ la **rappresentazione grafica** dei dati
-
- ❑ E' un linguaggio di programmazione semplice ed efficace che contiene numerose funzioni per l'elaborazione e la rappresentazione dei dati;
 - ❑ E' implementabile con l'aggiunta di "pacchetti" scaricabili dal sito che contengono funzioni dedicate per diversi settori della scienza, dell'economia, ecc...
 - ❑ E' un ambiente interattivo, ossia i comandi producono una risposta immediata, e prevedono una programmazione **object oriented**.
- Contiene funzioni per l'**input** di dati in diversi formati;
- Contiene funzioni di **output** di risultati e grafici in diversi formati

Cioè i dati da elaborare vengono raggruppati in **OGGETTI** e lo sono anche i risultati

Download di R e installazione

<https://cran.r-project.org/>

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

da qui installare la versione "base"

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-03-11, Great Truth) [R-3.5.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

E' installabile su diversi sistemi operativi!
Linus - Mac - Windows

What are R and CRAN?

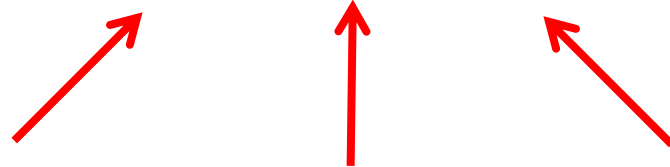
R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to your network load.

Struttura di R

Sistema
base

Alcuni packages contenenti funzioni di vario tipo sono presenti
nella versione base (utils,stats,graphics,grid, tools...)



The screenshot shows the CRAN website with the following content:

- CRAN logo
- Navigation links: [CRAN Mirrors](#), [What's new?](#), [Task Views](#), [Search](#)
- About R links: [R Homepage](#), [The R Journal](#)
- Software links: [R Sources](#), [R Binaries](#), [Packages](#), [Other](#)
- Documentation links: [Manuals](#), [FAQs](#), [Contributed](#)
- Available CRAN Packages By Name: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)
- Package list (partially visible):
 - [A3](#)
 - [aaSEA](#)
 - [ABACUS](#)
 - [abbyR](#)
 - [abc](#)
 - [abc.data](#)
 - [ABC.RAP](#)
 - [abcADM](#)
 - [ABCanalysis](#)
 - [abcdeFBA](#)
 - [ABCOptim](#)
 - [ABCp2](#)
 - [abcrf](#)
 - [abcrlda](#)
 - [abctools](#)
 - [abd](#)
 - [abdiv](#)
 - [abe](#)
 - [abf?](#)
 - [ABHgenotypeR](#)
- Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
- Amino Acid Substitution Effect Analyser
- Apps Based Activities for Communicating and Understanding Statistics
- Access to Abby Optical Character Recognition API
- Tools for Approximating Bayesian C
- Data Only: Tools for A
- Array Based CpG Reg
- Fit Accumulated Dama
- Computed An
- ABCDE_FBA: A
- Implementation of
- Approximate Bay
- Approximate P
- Asymptotically Bias-Corre
- Tools for ABC Analyses
- The Analysis of Biological Data
- Alpha and Beta Diversity Measures
- Augmented Backward Elimination
- Load Gap-Free Axon ABF2 Files
- Easy Visualization of ABH Genotypes

A yellow starburst graphic is overlaid on the screenshot, containing the text: **Circa 13000 pacchetti disponibili**

Bioconductor packages

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

[Home](#) [Install](#) [Help](#) [Developers](#) [About](#)

About Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community. Bioconductor is also available as an [AMI](#) (Amazon Machine Image) and a series of [Docker](#) images.

Bioc2020

Get the latest updates on the [Bioc2020 Conference!](#)

- Registration is Now Open! [Register Today!](#)
- Call for Abstracts! If you are interested in presenting a workshop, poster, or talk please [submit your proposal](#). Deadline March 3rd.
- Apply for [Travel Scholarships](#). Deadline March 3rd.

Bioconductor is

Install »

- Discover [1823 software packages](#) available in *Bioconductor* release 3.10.

Get started with *Bioconductor*

- [Install Bioconductor](#)
- [Get support](#)
- [Latest newsletter](#)
- [Follow us on twitter](#)
- [Install R](#)

Learn »

Master *Bioconductor* tools

- [Courses](#)
- [Support site](#)

Più di 1800 pacchetti per l'analisi statistica di dati biologici e di genomica

Use »

Create bioinformatic solutions with *Bioconductor*

- [Software](#), [Annotation](#), and [Experiment](#) packages
- [Docker](#) and [Amazon](#) machine images
- Latest [release announcement](#)
- [Community Slack](#) sign-up
- [Support site](#)

Develop »

Contribute to *Bioconductor*

- [Developer resources](#)
- [Use Bioc 'devel'](#)
- ['Devel' packages](#)
- [Package guidelines](#)
- [New package submission](#)
- [Git source control](#)
- [Build reports](#)

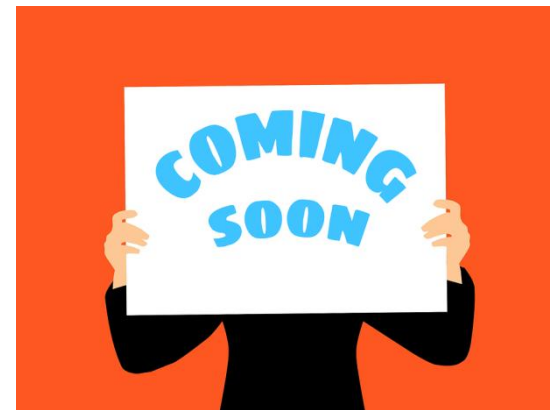
Ma... come è fatto un package di R?

Un pacchetto creato per essere utilizzato in R contiene:

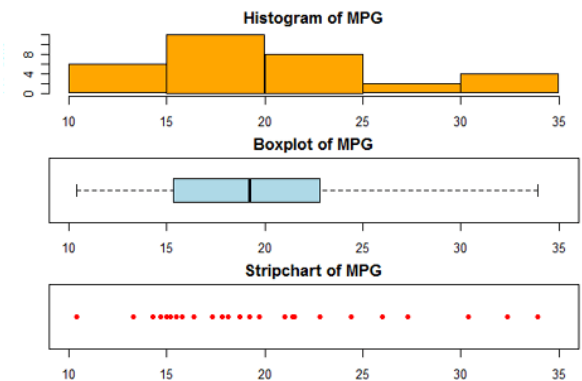
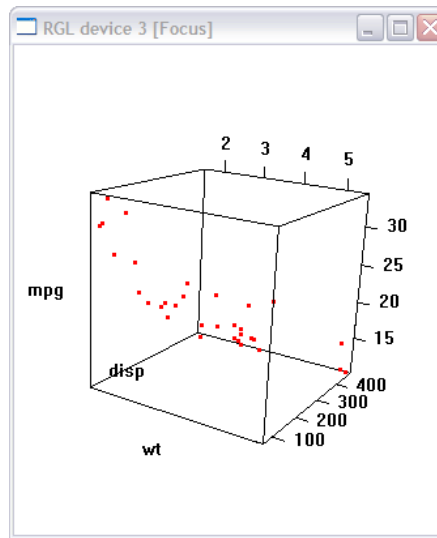
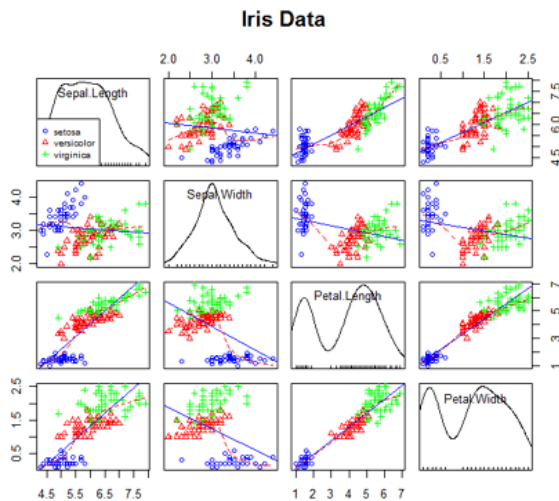
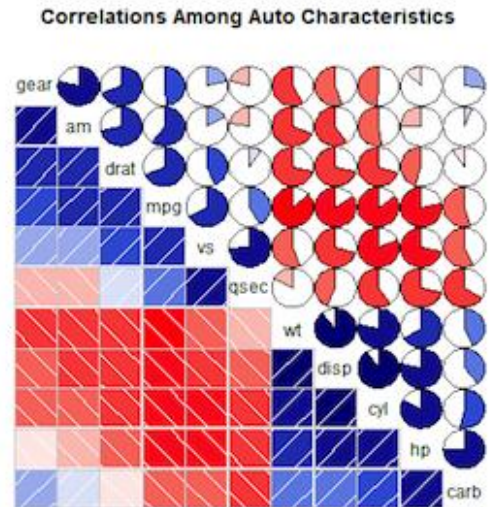
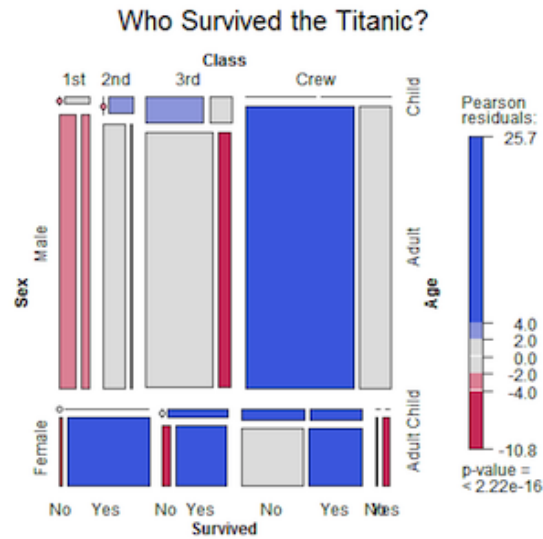
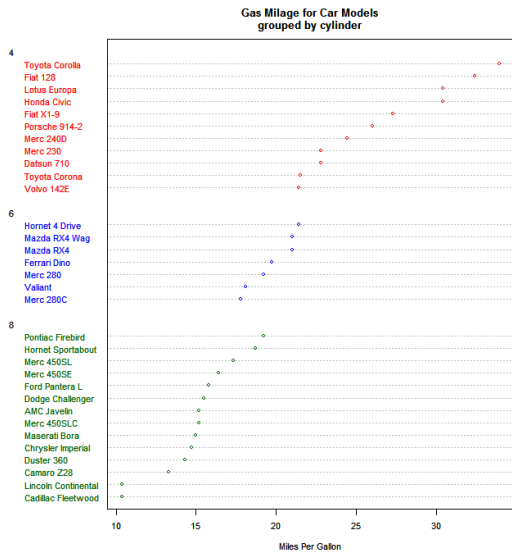
- **una serie di funzioni** che possono essere utilizzate per effettuare determinati tipi di calcoli o per creare grafici;
- **un file pdf** che contiene la descrizione dello scopo generale delle funzioni contenute nel pacchetto e la descrizione delle modalità di utilizzo delle singole funzioni e relativi esempi.

Può contenere anche uno o più set di dati utilizzati per illustrare gli output che si ottengono utilizzando le funzioni.

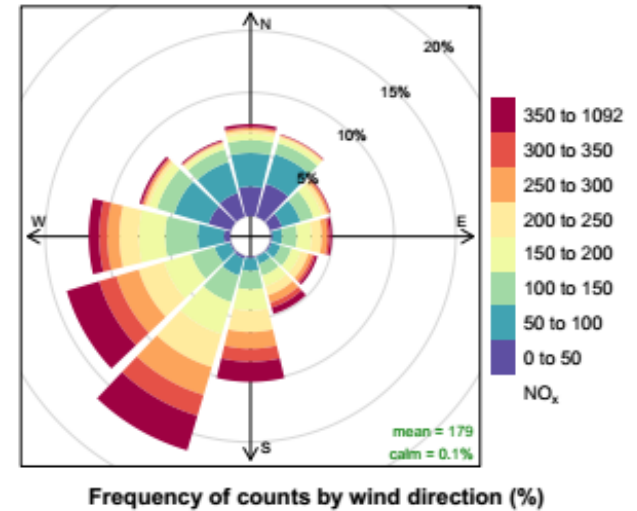
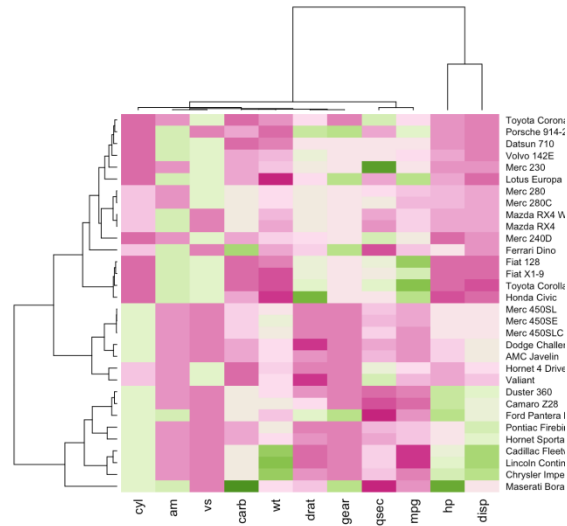
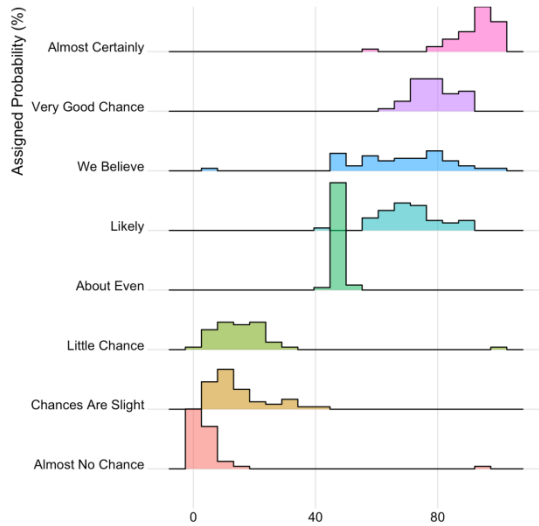
**... E cosa si intende per
funzione in R?**



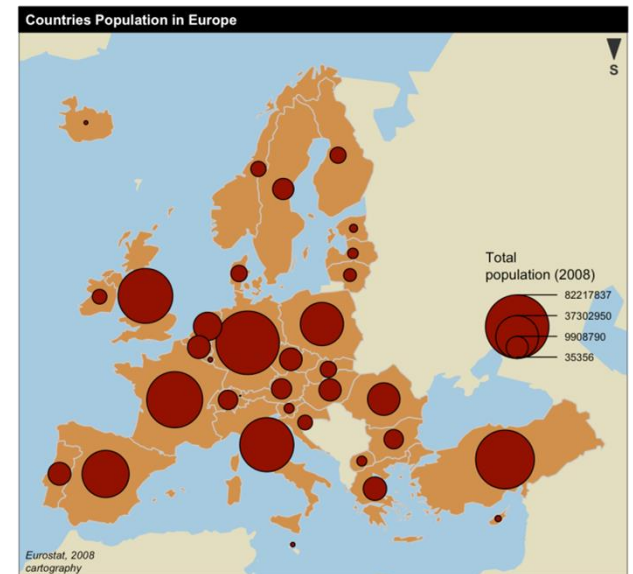
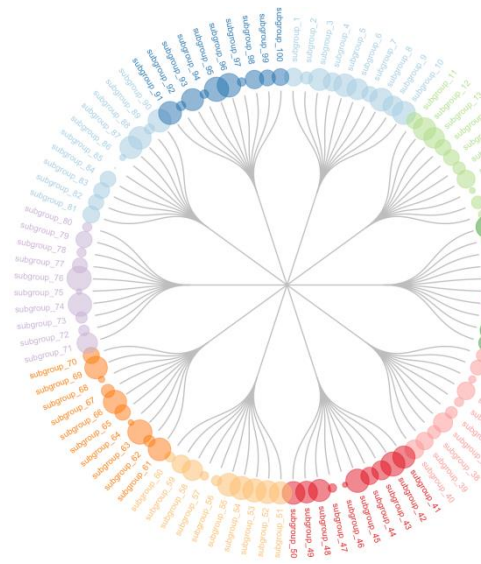
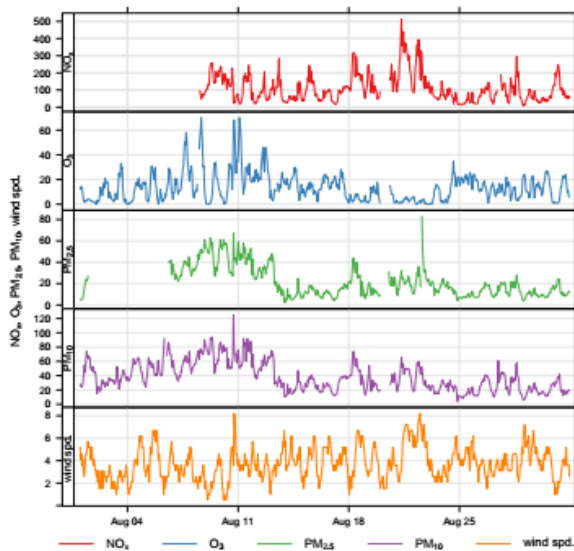
Esempi di rappresentazioni grafiche in R



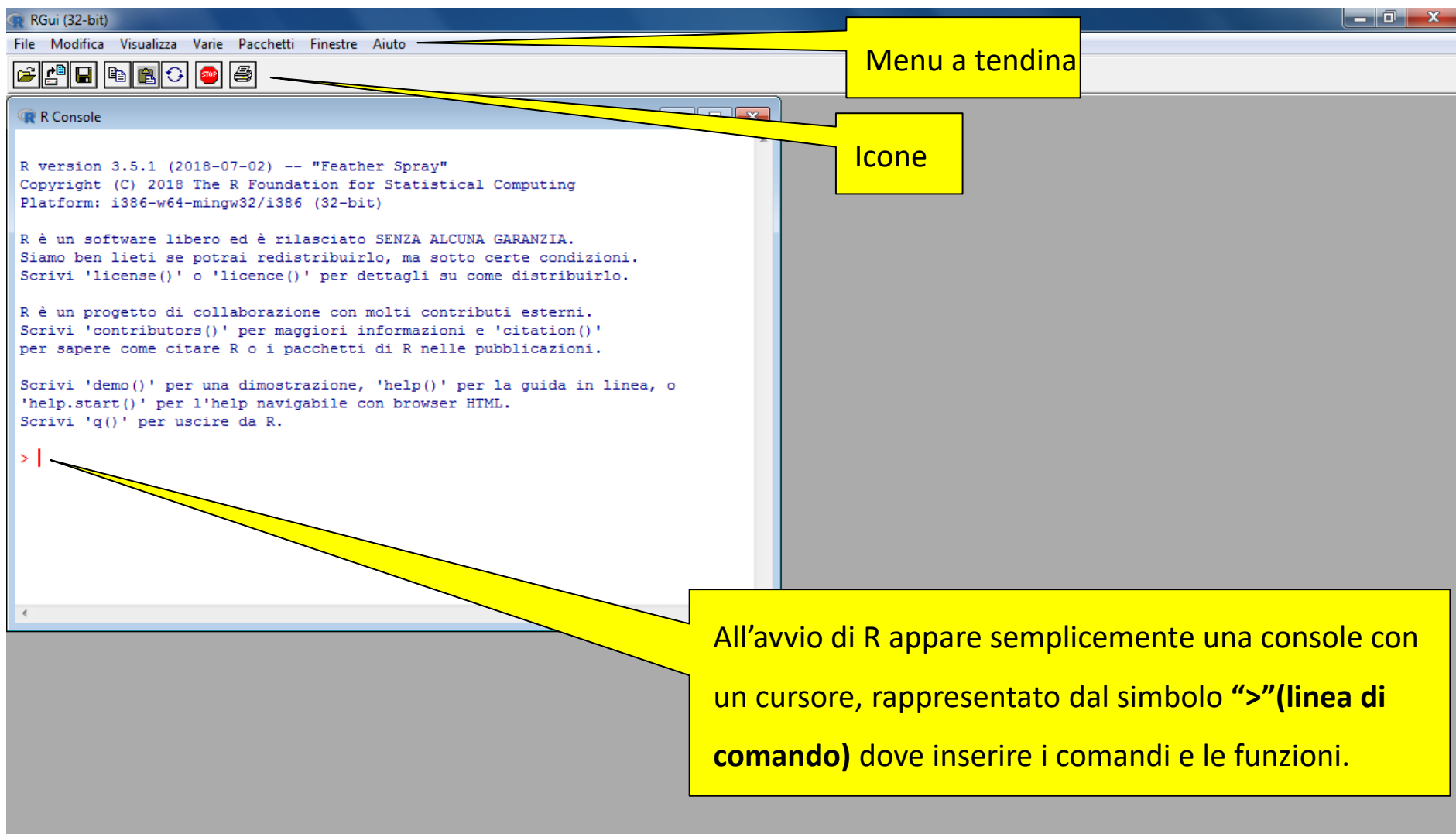
Esempi di rappresentazioni grafiche in R



Frequency of counts by wind direction (%)



Interfaccia di R all'avvio



Interfaccia di R

The image shows the R GUI interface with three main windows: R Console, Editor di R, and Finestra di grafico. A red box highlights the menu bar and toolbar at the top. A yellow callout points to the Editor di R window, and a pink callout points to the Finestra di grafico window.

R Console

```
> a<-1:6  
> b<-(-1):4  
>  
> plot(a,b,pch=16,cex=1.5,col="red")  
> |
```

Editor di R

```
a<-1:6  
b<-(-1):4  
  
plot(a,b,pch=16,cex=1.5,col="red")
```

Finestra di grafico

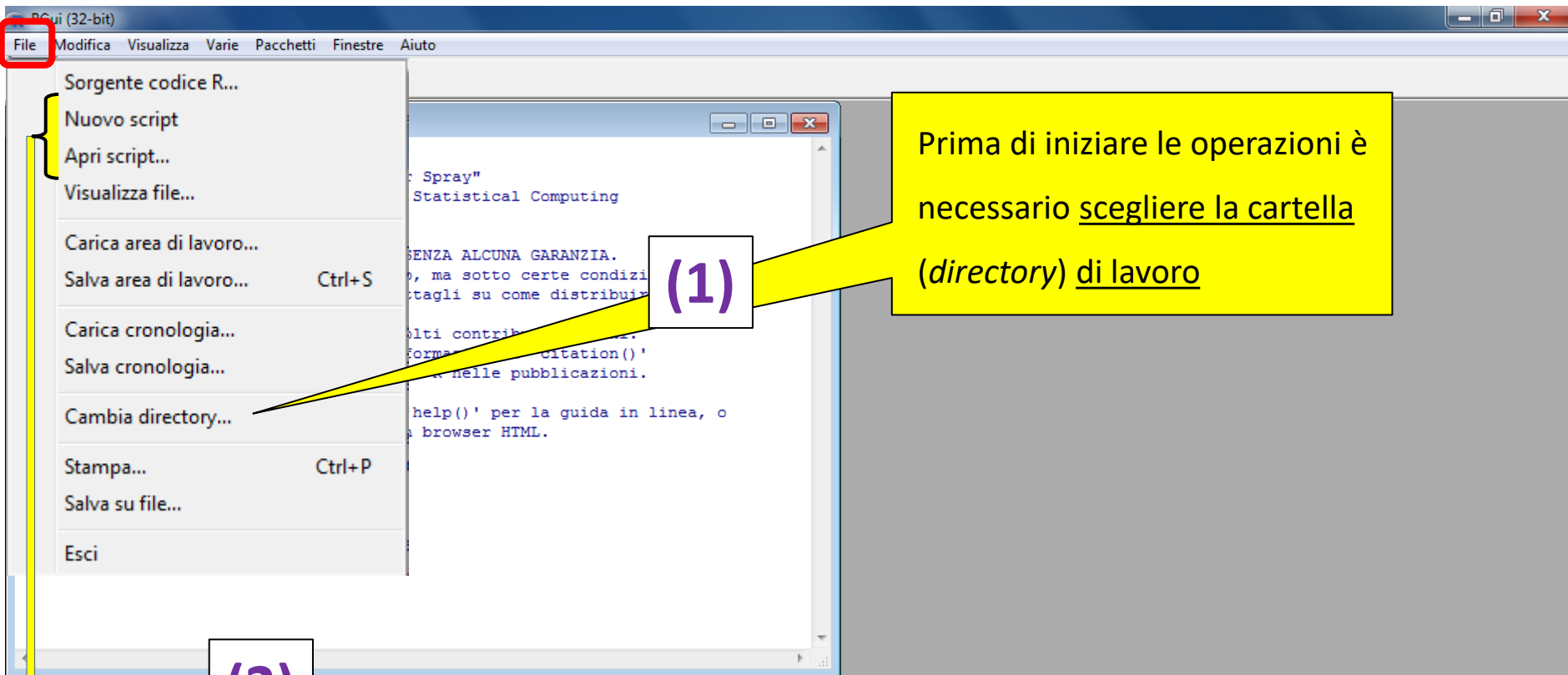
The plot shows a scatter plot with 4 red points. The y-axis is labeled with 3 and 4. The x-axis has 4 tick marks.

x	y
1	2
2	3
3	4
4	5

Attenzione!!!
I menu a tendina e le icone si modificano a seconda della finestra attiva in cui si sta lavorando

Si possono scrivere tutte le operazioni qui e poi eseguirle tramite il comando `Ctrl+Enter` (Editor contiene lo **SCRIPT**)

Menu file - scegliere la cartella di lavoro



Prima di iniziare le operazioni è necessario scegliere la cartella (directory) di lavoro

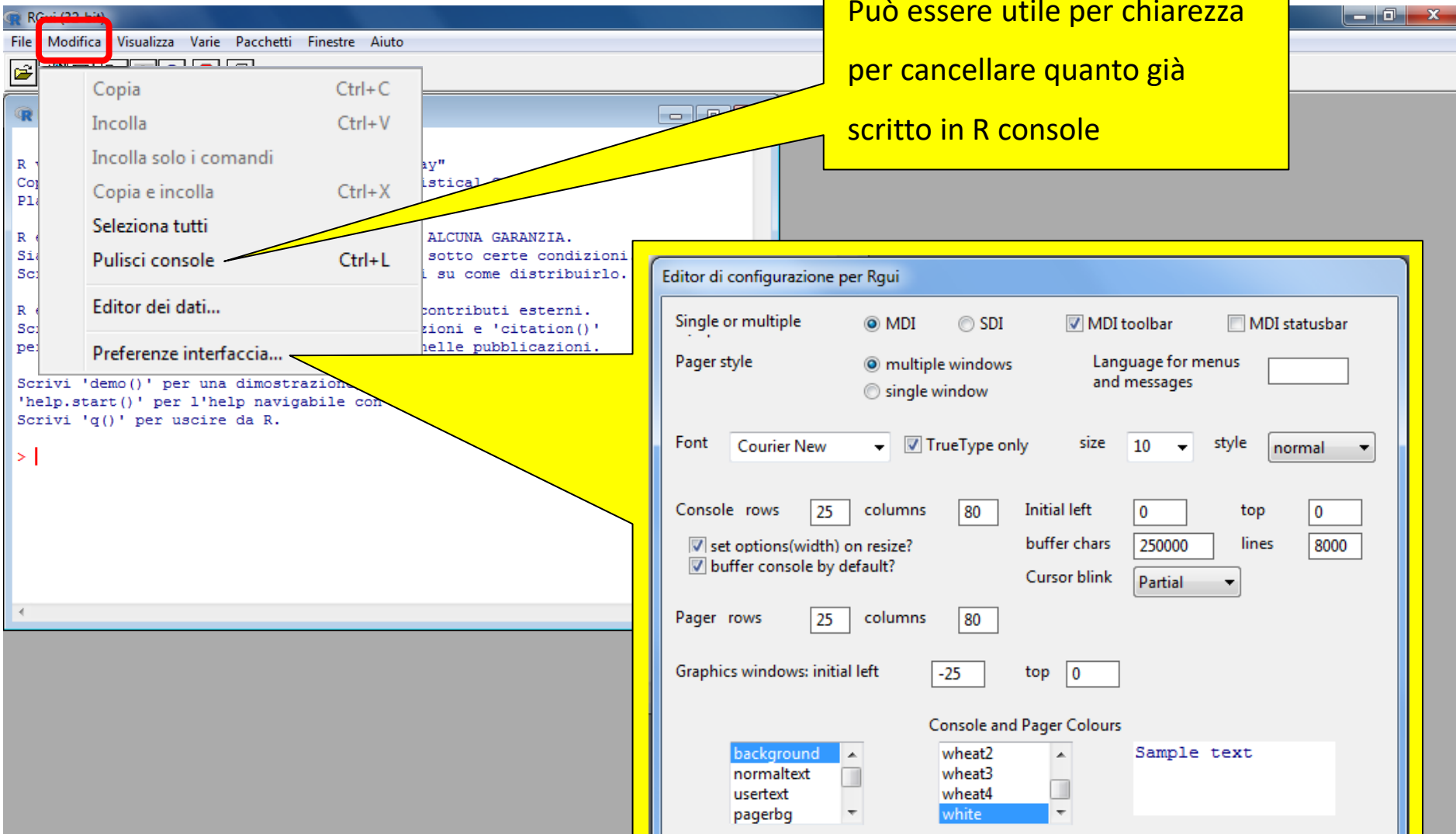
(1)

(2)

Poi è possibile aprire uno script o crearne uno nuovo che vengono aperti nell' Editor di R

Menu modifica

Può essere utile per chiarezza
per cancellare quanto già
scritto in R console



Classi di valori in R

R contempla 5 differenti possibili classi che assegna ai "valori":

```
> class(1)
[1] "numeric"
> class(1L)
[1] "integer"
> class("a")
[1] "character"
> class(TRUE)
[1] "logical"
```

- ***Numeric*** (numeri reali , anche con segno)
- ***Integer*** (numeri interi, anche con segno)
- ***Character*** (caratteri, cioè lettere o parole)
- **Logical** (TRUE/FALSE)
- **Complex**

La funzione **class()** restituisce la classe di un oggetto ("**definizione funzioni**" ..to be continued..)

Gli Oggetti in R

Ogni entità che il programma crea e manipola è definita **oggetto**.

Le strutture (tipologie) di dati contemplate in R sono:

- **scalari**: contengono un singolo dato
- **vettori**: insieme lineare di elementi omogenei per tipologia (tutti numeri, tutte stringhe, ecc.)
- **fattori**: vettore in cui gli elementi presenti sono classificati anche come "livelli"
- **array e matrici**: sono vettori multidimensionali, le matrici hanno due dimensioni: righe e colonne. Gli array sono insiemi di numeri con p dimensioni
- **dataframe**: sono matrici bidimensionali dove ogni colonna può avere un tipo di dato diverso dalle altre
- **liste**: insieme di elementi non omogenei tra loro

Nomi degli oggetti

```
>
> #Si crea un oggetto di nome "a" con un elemento
> a<-5
>
> #Per vedere il "contenuto" dell'oggetto basta digitarlo e premere invio
> a
[1] 5
>
> #R è case-sensitive, l'oggetto "A" non esiste
> A
Errore: oggetto "A" non trovato
>
> #Gli oggetti possono anche contenere testo
> A<-"mela"
>
> #Vedere il "contenuto" dell'oggetto
> A
[1] "mela"
>
> #Attenzione che il contenuto degli oggetti può essere sovrascritto!!!
> a<-10.6
>
> #Vedere il nuovo "contenuto" dell'oggetto
> a
[1] 10.6
>
> #I nomi degli oggetti possono essere composti da più lettere, numeri e altri caratteri
> Ogg1<-45.6
>
> #Vedere il nuovo "contenuto" dell'oggetto
> Ogg1
[1] 45.6
> |
```

Un oggetto viene assegnato ad un nome tramite l'utilizzo dei simboli appaiati <- ("minore" e "meno")

I nomi degli oggetti non devono contenere spazi vuoti, simboli matematici, non devono iniziare con un numero e bisogna evitare caratteri già utilizzati da R (es. T per TRUE o F per FALSE)

R è *case sensitive*, cioè considera caratteri differenti lettere maiuscole e minuscole.

Operazioni matematiche con R

```
>
> #somma
> 10.3+6.5
[1] 16.8
>
>
> #moltiplicazione
> 3*2
[1] 6
>
>
> #sottrazione e divisione
> 5.3-6.8
[1] -1.5
>
>
> #sottrazione e divisione
> 5/2.3
[1] 2.173913
>
>
> #elevamento a potenza
> 2^6
[1] 64
>
>
> #combinazione di operazioni
> ((14-3.2)/8)^4
[1] 3.321506
>
> |
```

```
>
> #radice quadrata
> sqrt(144)
[1] 12
>
>
> #logaritmo in base 10
> log10(1000)
[1] 3
>
>
> #e(numero naturale) elevato ad esponente
> exp(4)
[1] 54.59815
>
>
> |
```

Testi di **commento** possono essere aggiunti nello script dopo il **simbolo cancelletto (“#”)**, tutto ciò che si trova dopo questo simbolo e fino alla riga successiva viene ignorato da R.

E possibile **richiamare i comandi** impartiti in precedenza utilizzando la **freccia "su" (↑)** della tastiera.

Valori mancanti, "non numeri" e infinito

```
> x<-NA
> x
[1] NA
>
> x*3
[1] NA
>
> 0/0
[1] NaN
>
> 5/0
[1] Inf
>
> log10(0)
[1] -Inf
>
> |
```

In R i valori mancanti sono rappresentati dal valore **NA** (“*Not Available*”). Le operazioni logiche eseguite su valori mancanti o comunque non utilizzabili, restituiscono ancora il valore NA.

Operazioni aritmetiche “impossibili”, restituiscono il valore **NaN** che significa che “non è un numero” (“*Not a Number*”). **Inf** e **-Inf** rappresentano l’infinito positivo e negativo

I vettori

```
> #Un vettore contiene una serie di elementi
> b<-c(1,2,5,7)
>
> #Vedere il "contenuto" dell'oggetto
> b
[1] 1 2 5 7
>
> #Un vettore può contenere anche elementi di testo
> tex<-c("pera","banana","kiwi")
>
> #Vedere il "contenuto" dell'oggetto
> tex
[1] "pera" "banana" "kiwi"
>
> #I vettori possono essere concatenati con altri vettori
> #con medesimo tipo di contenuto
> d<-c(a,b)
>
> #Vedere il "contenuto" dell'oggetto
> d
[1] 10.6 1.0 2.0 5.0 7.0
>
> #I vettori possono essere concatenati con altri vettori
> #con medesimo tipo di contenuto
> vex<-c(tex,A)
>
> #Vedere il "contenuto" dell'oggetto
> vex
[1] "pera" "banana" "kiwi" "mela"
>
> |
```



e_i = i-esimo elemento

Per creare un vettore esistono molti modi, il più comune dei quali è la **funzione c()** che combina i valori presenti nel suo elenco in un vettore.

Uno **scalare** è considerato anche come un vettore monoelemento



SEGUE

I vettori

```
> #Ci sono altri modi per generare un vettore (1)
> AA<-c(18:11)
>
> #Vedere il "contenuto" dell'oggetto
> AA
[1] 18 17 16 15 14 13 12 11
>
> #Ci sono altri modi per generare un vettore (2)
> BB<-rep(9.2,8)
>
> #Vedere il "contenuto" dell'oggetto
> BB
[1] 9.2 9.2 9.2 9.2 9.2 9.2 9.2 9.2
>
> #Ci sono altri modi per generare un vettore (3)
> Cx<-seq(-2.5,8.5,0.5)
>
> #Vedere il "contenuto" dell'oggetto
> Cx
[1] -2.5 -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5  3.0  3.5  4.0
[15]  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0  8.5
>
> #Tutti questi modi possono essere utilizzati anche insieme
> #Osservare l'uso delle parentesi (!)
> Dx<-c(rep(c(1,3),2),2.5,seq(-1,1,0.2),4:2)
>
> #Vedere il "contenuto" dell'oggetto
> Dx
[1]  1.0  3.0  1.0  3.0  2.5 -1.0 -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4  0.6
[15]  0.8  1.0  4.0  3.0  2.0
> |
```

Altri modi per definire un vettore:

- $c(n_1:n_2)$ genera un vettore con elementi a partire dal numero n_1 fino al numero n_2 a passo 1;
- $rep(n_1,n_2)$ genera un vettore con n_2 elementi di valore n_1 ;
- $seq(n_1,n_2,n_3)$ genera un vettore con numeri che vanno da n_1 a n_2 a passo n_3 .

rep e seq sono funzioni di R

Attributi di un vettore

```
> #Vedere il "contenuto" dell'oggetto
> d
[1] 10.6  1.0  2.0  5.0  7.0
>
> #numero di valori (lunghezza del vettore)
> length(d)
[1] 5
>
> #tipo di contenuto del vettore
> mode(d)
[1] "numeric"
>
> #Vedere il "contenuto" dell'oggetto
> vex
[1] "pera"   "banana" "kiwi"   "mela"
>
> #tipo di contenuto del vettore
> mode(vex)
[1] "character"
>
> |
```

length(nomeoggetto) indica il numero di elementi del vettore
mode(nomeoggetto) o **class(nomeoggetto)** indica il tipo di contenuto del vettore (numeri, testo, ecc...)

Individuare elementi nei vettori

```
> #Vedere il "contenuto" dell'oggetto
```

```
> d  
[1] 10.6 1.0 2.0 5.0 7.0
```

```
> #Individuare un elemento
```

```
> d[5]  
[1] 7
```

```
> #Individuare tutti gli elementi tranne quello indicato
```

```
> d[-5]  
[1] 10.6 1.0 2.0 5.0
```

```
> #Individuare più di un elemento
```

```
> d[c(1,4)]  
[1] 10.6 5.0
```

```
> #Individuare tutti gli elementi tranne quelli indicati
```

```
> d[-c(1,4)]  
[1] 1 2 7
```

```
> #Individuare un sottoinsieme di elementi
```

```
> d[3:5]  
[1] 2 5 7
```

```
> #Individuare un sottoinsieme di elementi esclusi quelli indicati
```

```
> d[-(3:5)]  
[1] 10.6 1.0
```

```
> #Individuare tutti gli elementi "maggiori di"
```

```
> d[d>3]  
[1] 10.6 5.0 7.0
```

```
> #Individuare tutti gli elementi "maggiori di" e "minori di"
```

```
> d[d>3&d<7]  
[1] 5
```

```
> #Individuare tutti gli elementi "minori di" o "maggiori di"
```

```
> d[d<3|d>7]  
[1] 10.6 1.0 2.0
```

Si usano dei criteri all'interno di parentesi quadre **nomeoggetto[criterio]**

In taluni casi nella scrittura del criterio viene "citato" il nome dell'oggetto

Generare nuovi vettori da vettori esistenti

Per generare nuovi vettori si possono utilizzare anche i vettori già creati in varie combinazioni, assegnando un **nuovo nome all'oggetto**

```
> #Attenzione!!! Prima di usare dei nomi di oggetti verificare
> #se esistono già, altrimenti si sovrascrive l'oggetto!!!
> CC
Errore: oggetto "CC" non trovato
>
> #Nuovo vettore da sottoinsieme di elementi di altri:
> CC<-c(BB[c(1,3)],AA[5:8],Dx[Dx>1])
>
> #Vedere il "contenuto" dell'oggetto
> CC
[1]  9.2  9.2 14.0 13.0 12.0 11.0  3.0  3.0  2.5  4.0  3.0  2.0
> |
```

I vettori - riassunto

Insieme lineare di elementi omogenei per tipologia (tutti numeri, tutti caratteri, tutti logici)

Per generare un vettore esistono molti modi, il più comune dei quali è la funzione
c(elemento1, elemento2, ecc..)
che combina i valori presenti nel suo elenco in un vettore.

Altri modi per creare un vettore:

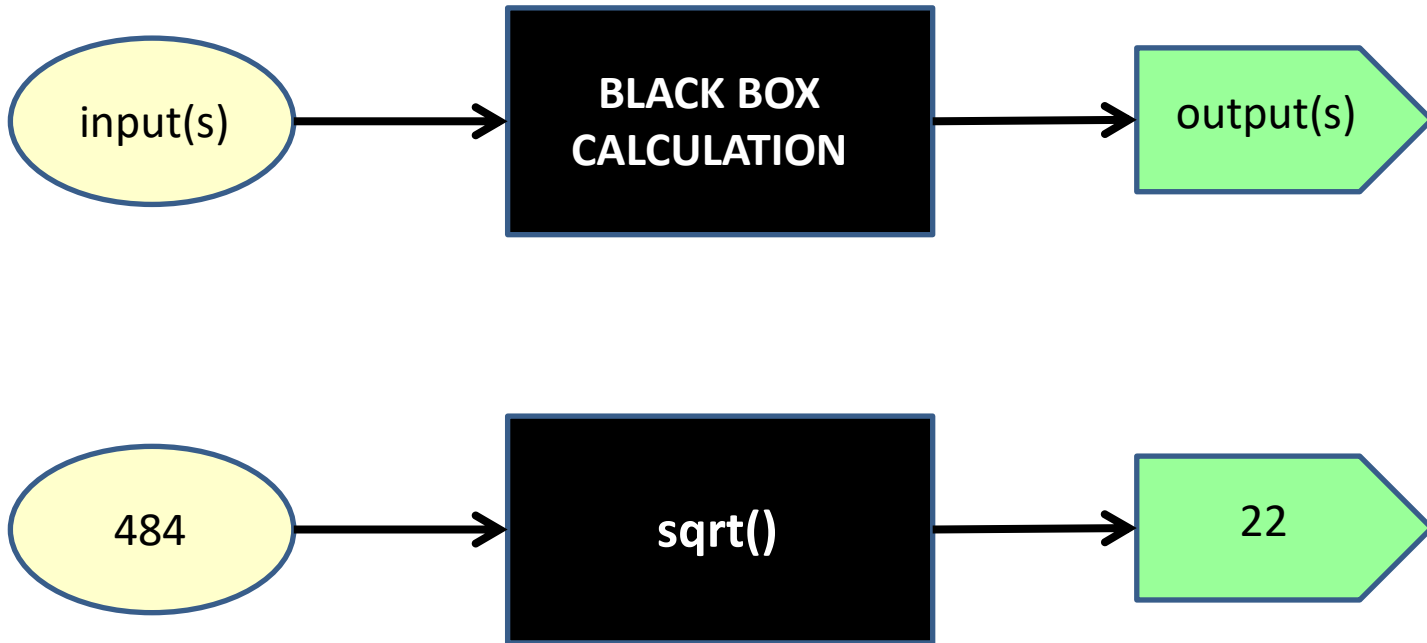
- **c(n₁:n₂)** genera un vettore con elementi a partire dal numero n₁ fino al numero n₂ a passo 1;
- **rep(n₁, n₂)** genera un vettore con n₂ elementi di valore n₁;
- **seq(n₁, n₂, n₃)** genera un vettore con numeri che vanno da n₁ a n₂ a passo n₃.

Attributi di un vettore:

length(nomeoggetto) indica il numero di elementi del vettore

mode(nomeoggetto) o **class(nomeoggetto)** indica il tipo di contenuto del vettore
(numeri, testo, ecc...)

Le funzioni in R



In realtà non è un *black box* "reale" perché è possibile accedere a tutti gli script che le funzioni eseguono e anche al codice sorgente di R
(non possibile per i software a pagamento!!!)



La documentazione delle funzioni

Per accedere alla documentazione relativa ad una funzione bisogna digitare

help(nomefunzione)

Nel browser si apre
la documentazione
della funzione
(non serve essere
collegati in rete!)

The screenshot shows the R help page for the `mean` function. Annotations include:

- A yellow box labeled "Nome della funzione e pacchetto di appartenenza" pointing to the title "mean (base)" and "Arithmetic Mean".
- A yellow box labeled "Altre informazioni utili" pointing to the "Value" section, which describes the return value of the function.
- A red box highlights the "Arguments" section, which lists parameters like `x`, `trim`, and `na.rm`.
- Pink boxes highlight the "Description", "Usage", and "Arguments" sections.

```
mean (base)
Arithmetic Mean

Description
Generic function for the (trimmed) arithmetic mean.

Usage
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments
x
  An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for trim = 0, only.

trim
  the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

na.rm
  a logical value indicating whether NA values should be stripped before the computation proceeds.

...
  further arguments passed to or from other methods.

Value
If trim is zero (the default), the arithmetic mean of the values in x is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), numeric (in warning.
If trim is non-zero, symmetrically trimmed observations deleted from each end before the mean is computed.

References
Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

See Also
```

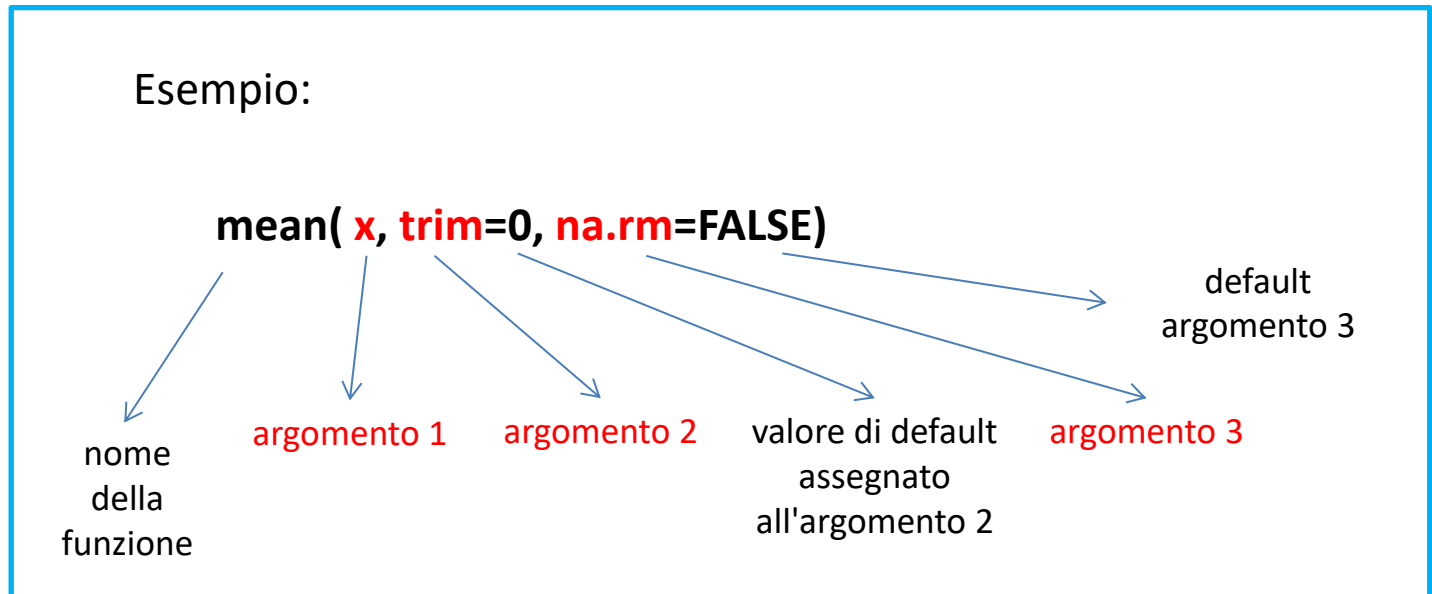


Gli argomenti delle funzioni

Gli argomenti sono gli input che la funzione richiede, alcuni sono obbligatori altri sono opzionali, ad alcuni è assegnato un valore di default

Gli argomenti possono essere inseriti in base

- alla **posizione** indicata nelle istruzioni
- **esplicitando l'assegnazione** all'argomento con il simbolo =



args(nomefunzione) consente di visualizzare gli argomenti previsti dalla funzione

SEGUE

Gli argomenti delle funzioni (esempio)

```
> A<-c(1,6,8,9,5)
> A
[1] 1 6 8 9 5
> B<-c(5:10,NA)
> B
[1] 5 6 7 8 9 10 NA
>
> mean()
Error in mean.default() :
  l'argomento "x" non è specificato e non ha un valore predefinito
> mean(x=A)
[1] 5.8
> mean(A)
[1] 5.8

> mean(B)
[1] NA
> mean(B,na.rm=TRUE)
[1] 7.5
> mean(TRUE,B)
Error in mean.default(TRUE, B) : 'trim' must be numeric of length one
> mean(na.rm=TRUE,x=B)
[1] 7.5
>

> C<-c("mela","pera","banana")
> C
[1] "mela" "pera" "banana"
> mean(C)
[1] NA
Warning message:
In mean.default(C) : l'argomento non è numerico o logico: restituisco NA

> D<-c(TRUE,FALSE,TRUE,TRUE,FALSE)
> D
[1] TRUE FALSE TRUE TRUE FALSE
> mean(D)
[1] 0.6
```

Nelle operazioni **TRUE** è considerato =1 e **FALSE**=0

Elencare e cancellare oggetti

```
> #Visualizzare l'elenco degli oggetti creati
> ls()
 [1] "a"      "A"      "AA"     "b"      "BB"     "Cx"     "d"      "Dx"     "Ogg1"
[10] "tex"   "vex"
>
> #Cancellare un oggetto
> rm(Ogg1)
>
> #Verificare la cancellazione
> Ogg1
Errore: oggetto "Ogg1" non trovato
> |
```

ls() serve a visualizzare tutti gli oggetti creati

rm(nomeoggetto) serve a cancellare un oggetto specifico

Menu varie

