

Attività di allineamento

Laboratorio Informatico-Statistico



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Corso di Laurea Magistrale in Statistica Economia e Impresa

Dipartimento di Scienze Statistiche Paolo Fortunati

Anno Accademico 2021/2022



Who am I?

raffaele.anselmo2@unibo.it

raffaeleanselmo@gmail.com



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Laurea in **Scienze Statistiche**,
curriculum Economia e Impresa



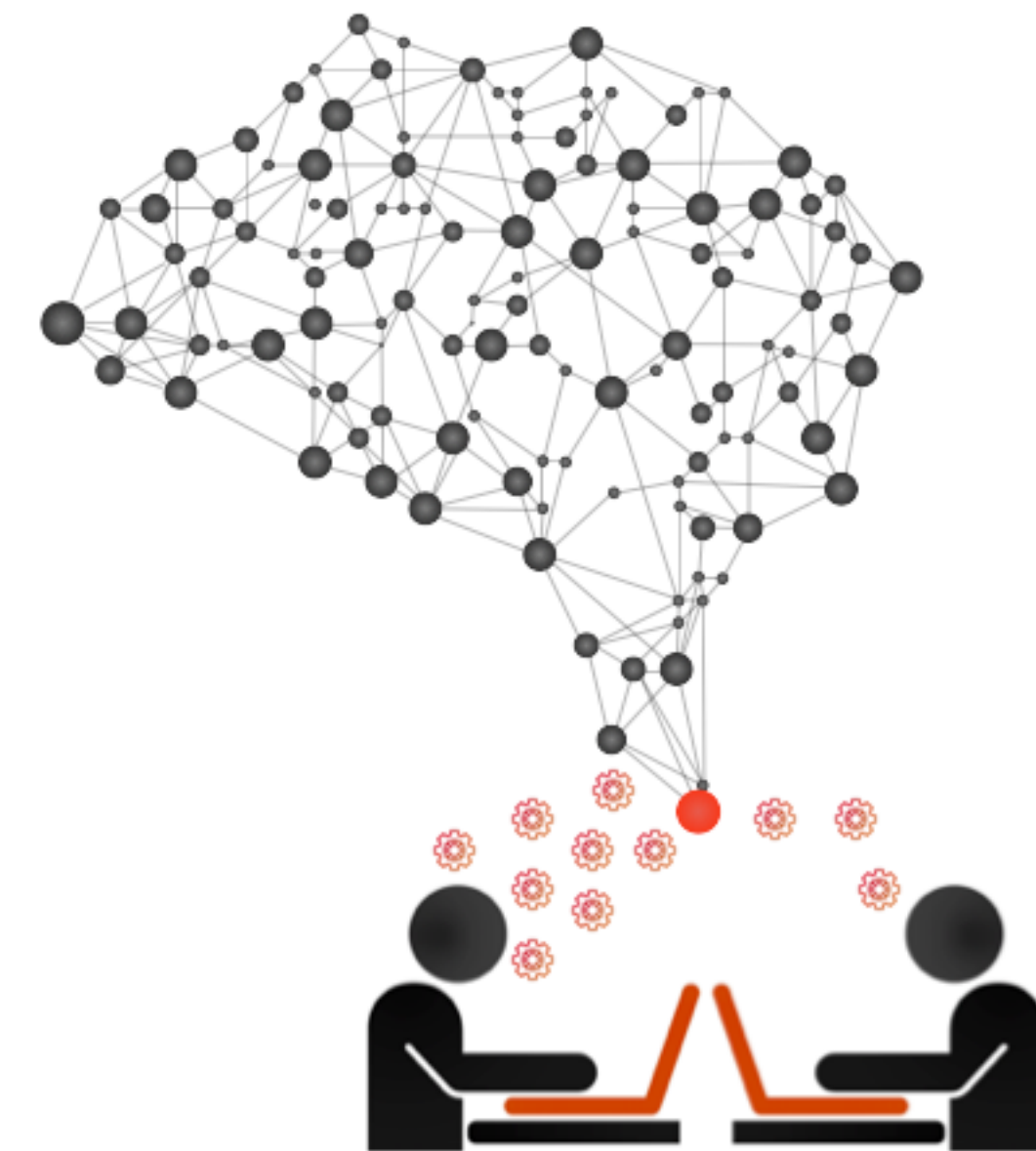
Master Degree in
Data Science



Tensorflow
Certified developer



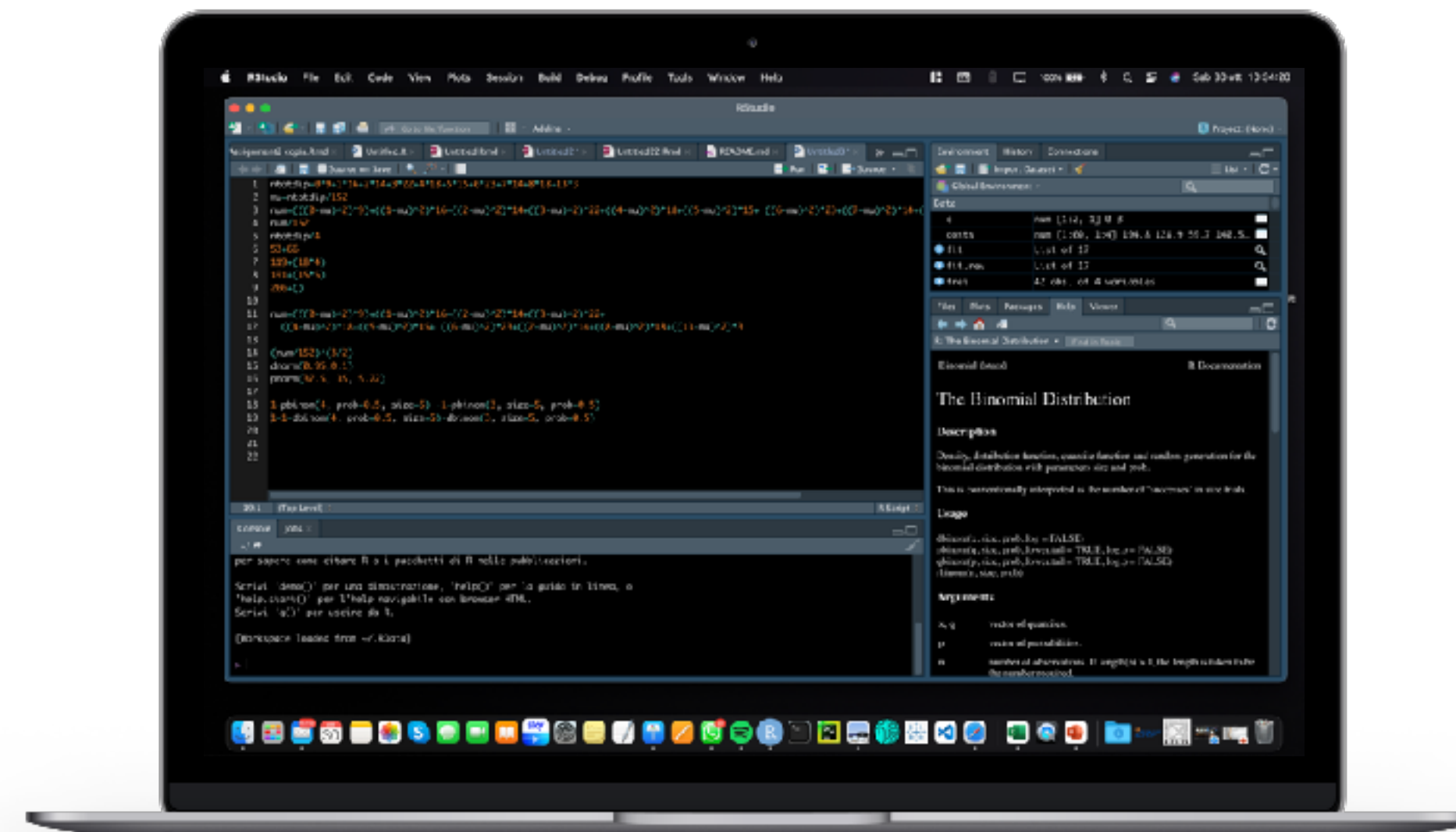
Data Scientist
@CRIF



What is this course about?

21
3²4⁷5
6 9 Il mondo *Data* e gli
strumenti per gestirlo

R Linguaggio di
programmazione per
analisi statistiche



How is it structured?

1 Intro

Nozioni di base sulla programmazione ed R

2 Data Exploration

Matrici,
Dataframes,
statistiche e grafici

3 Next Steps

Funzioni,
Approfondimenti e
librerie utili

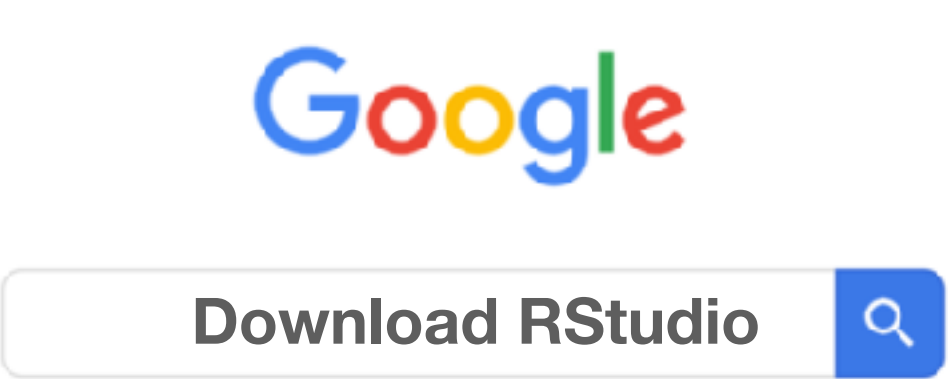
Introduzione

First Steps

Download



<https://cran.r-project.org/bin/windows/base/>



<https://www.rstudio.com/products/rstudio/download/>

RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Workbench ⓘ
Open Source License	Commercial License	Open Source License	Commercial License
Free	\$995 /year	Free	\$4,975 /year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY
Learn more	Learn more	Learn more	Evaluation Learn more

First Steps

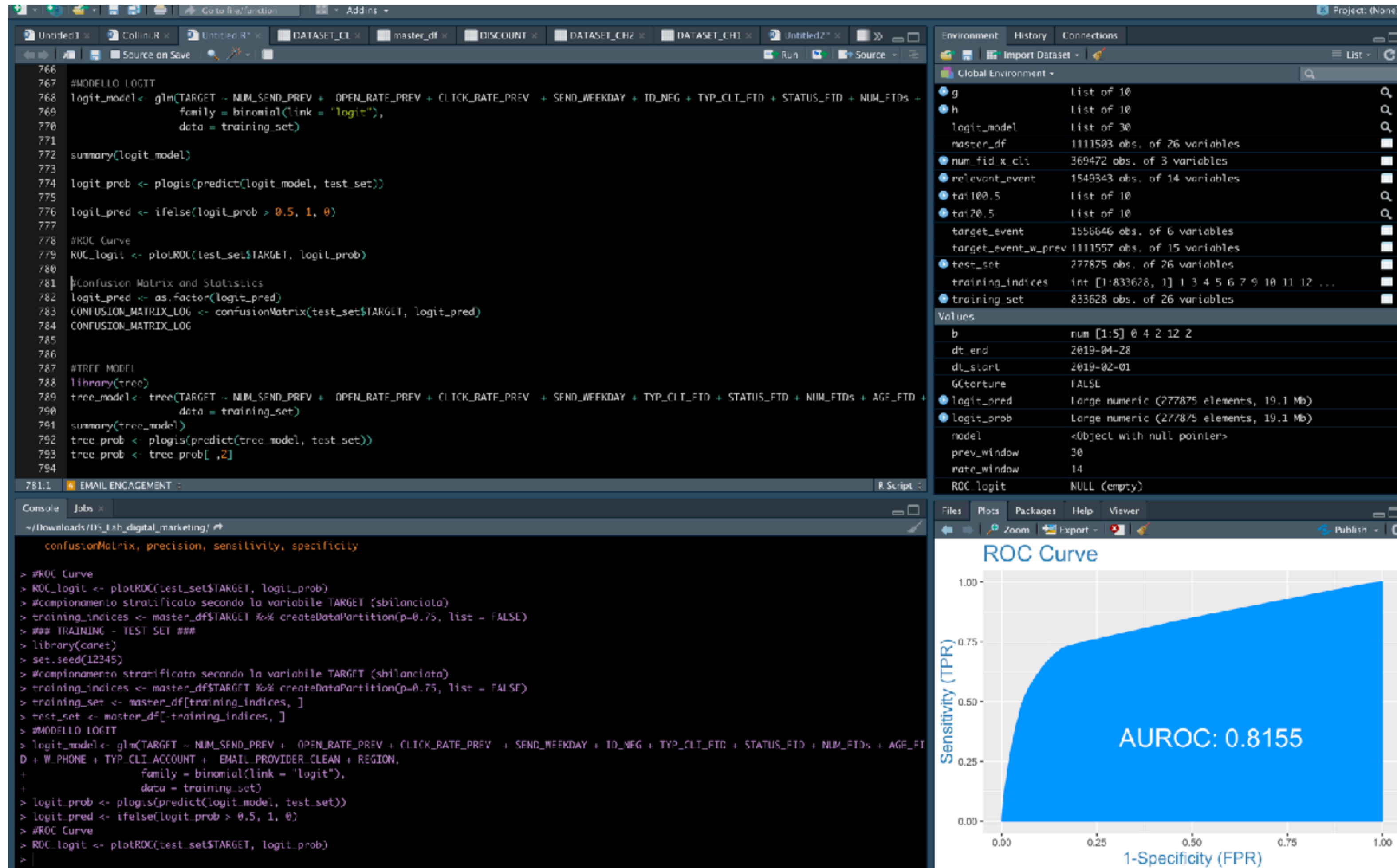
La finestra RStudio

Code Editor

Workspace

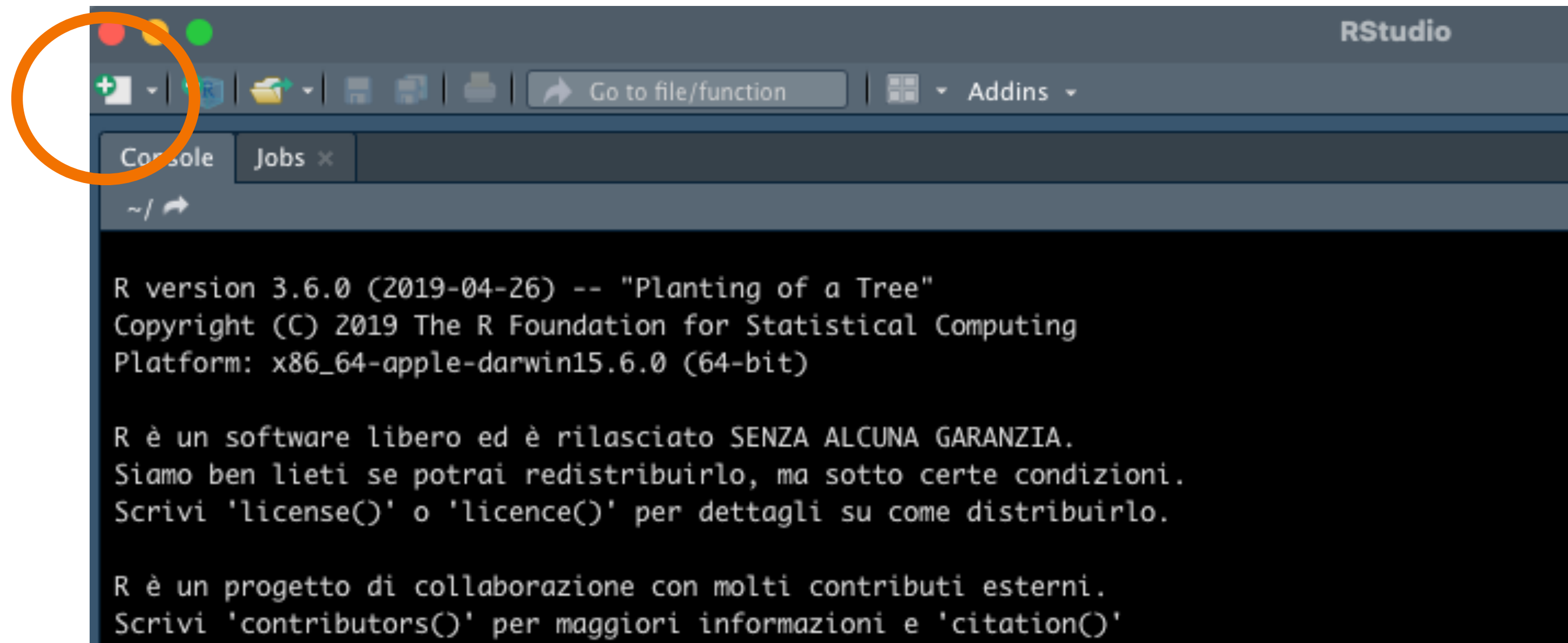
Console

Plot/Help



First Steps

“Hello world”



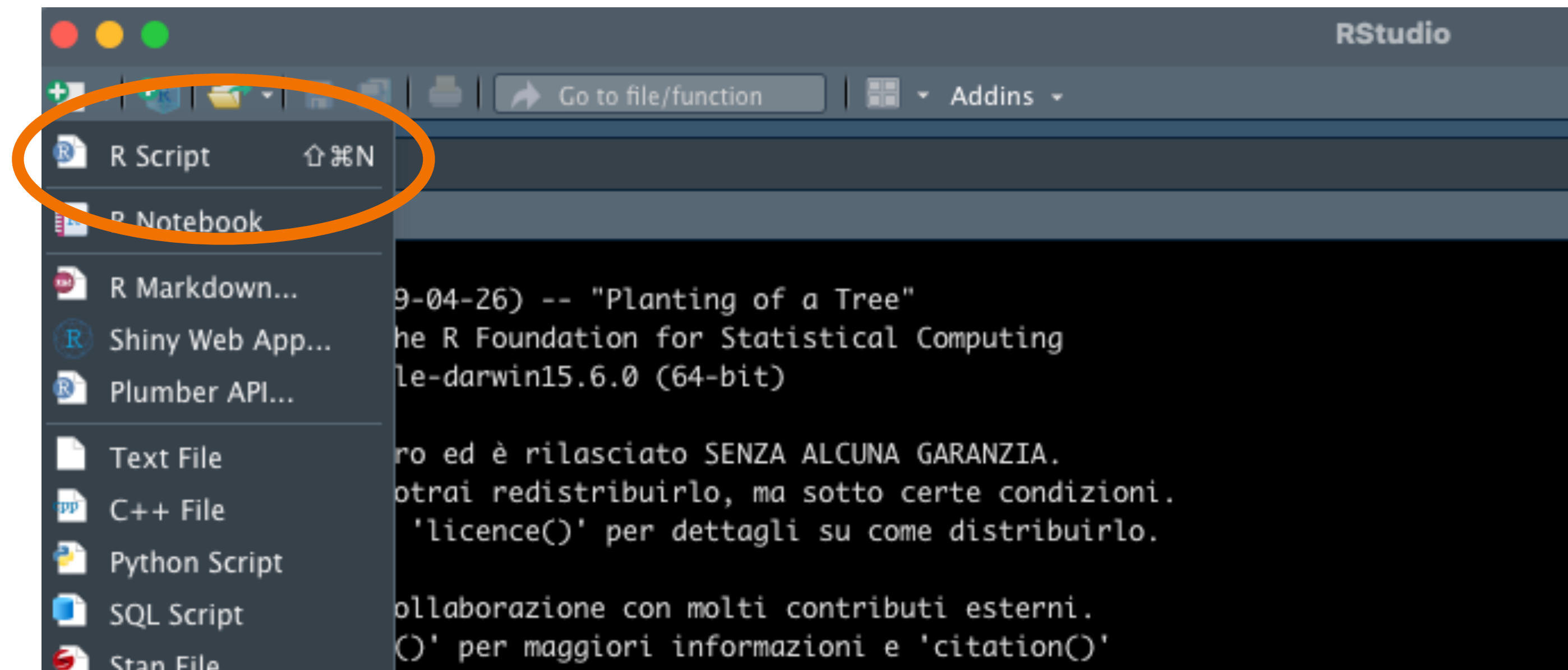
```
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R è un software libero ed è rilasciato SENZA ALCUNA GARANZIA.
Siamo ben lieti se potrai redistribuirlo, ma sotto certe condizioni.
Scrivi 'license()' o 'licence()' per dettagli su come distribuirlo.

R è un progetto di collaborazione con molti contributi esterni.
Scrivi 'contributors()' per maggiori informazioni e 'citation()'
```

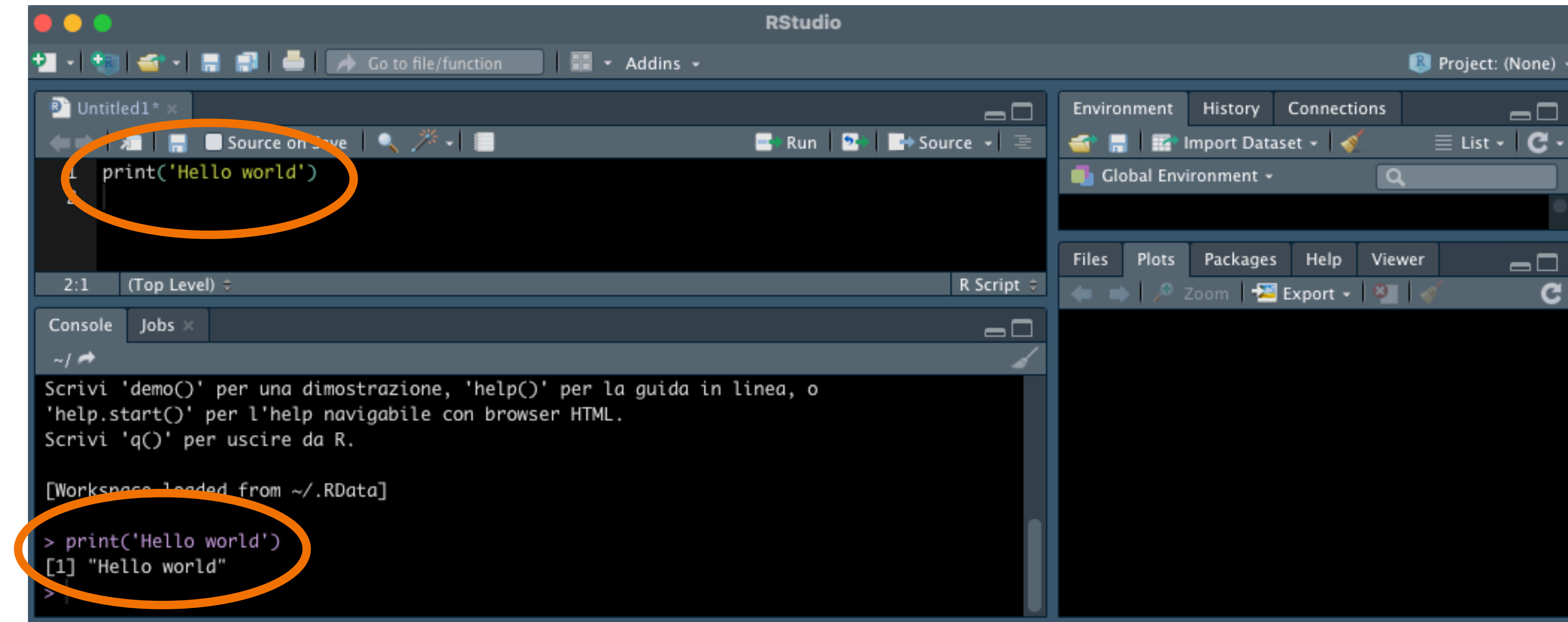

First Steps

“Hello world”



First Steps

“Hello world”



First Steps

Gestione dei pacchetti

R contiene poche semplici funzioni. Gran parte delle funzioni più *potenti* sono rese disponibili tramite *librerie (packages)* direttamente installabili da R.

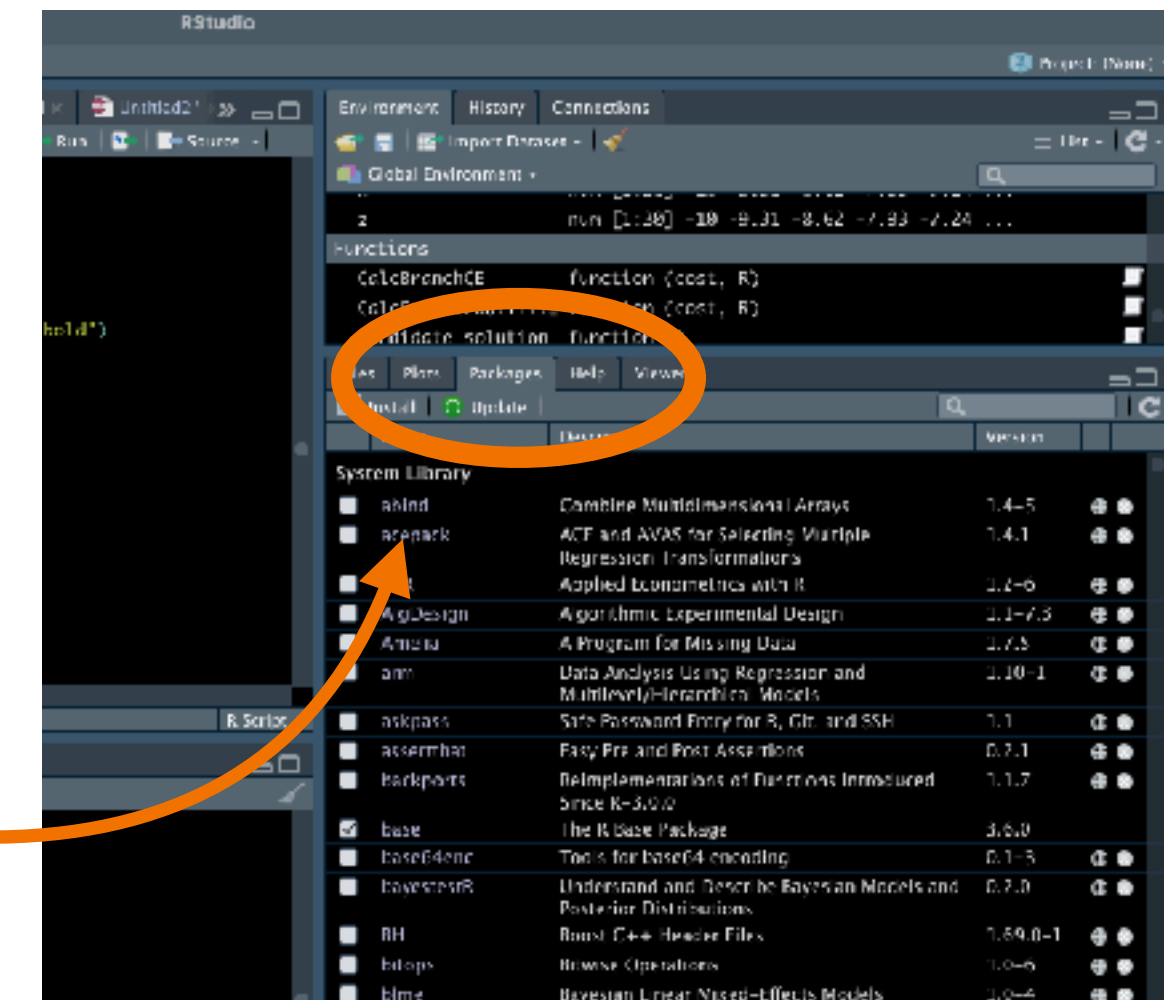
Per installare una libreria si utilizza la funzione:

```
>install.packages("nomelibreria")
```

Per utilizzare le funzioni di una libreria già installata è necessario importare la libreria nell'ambiente di lavoro (*workspace*) tramite la funzione:

```
>library("nomelibreria")
```

Alternativamente, i pacchetti possono essere direttamente gestiti tramite il tab *packages* in basso a destra



First Steps

I primi comandi

In R tutto ciò che esiste è un oggetto. Per assegnare qualcosa (il numero 5) ad un oggetto (a) si possono utilizzare gli operatori “=” e “<-”

```
> a = 5
```

```
> a
```

```
[1] 5
```

```
> a <- 5
```

```
> a
```

```
[1] 5
```

Gli oggetti possono essere di diverso tipo. Per controllare la tipologia dell'oggetto si usa la funzione:

```
> class(a)
```

```
[1] "numeric"
```

First Steps

I primi comandi

Se il valore è compreso tra apici, questo sarà considerato di tipo *character*:

```
> a = '5'
```

```
> class(a)
```

```
[1] "character"
```

Nota bene che abbiamo assegnato il carattere '5' all'oggetto a che precedentemente conteneva il valore numerico 5. In questo caso l'oggetto numerico 5 è stato sostituito con l'oggetto carattere '5'. Per vedere la lista degli oggetti in memoria (per oggetti si intendono sia le variabili che le funzioni) si utilizza la funzione:

```
> ls()
```

```
[1] "a"
```

Per rimuovere un oggetto si usa la funzione:

```
> rm('a')
```

```
> ls()
```

```
character(0)
```

First Steps

I primi comandi

Le funzioni vengono utilizzate nella seguente forma:

```
> nomefunzione(arg_1, arg_2, ... , arg_n)
```

Dove *arg_i* sono gli argomenti della funzione.

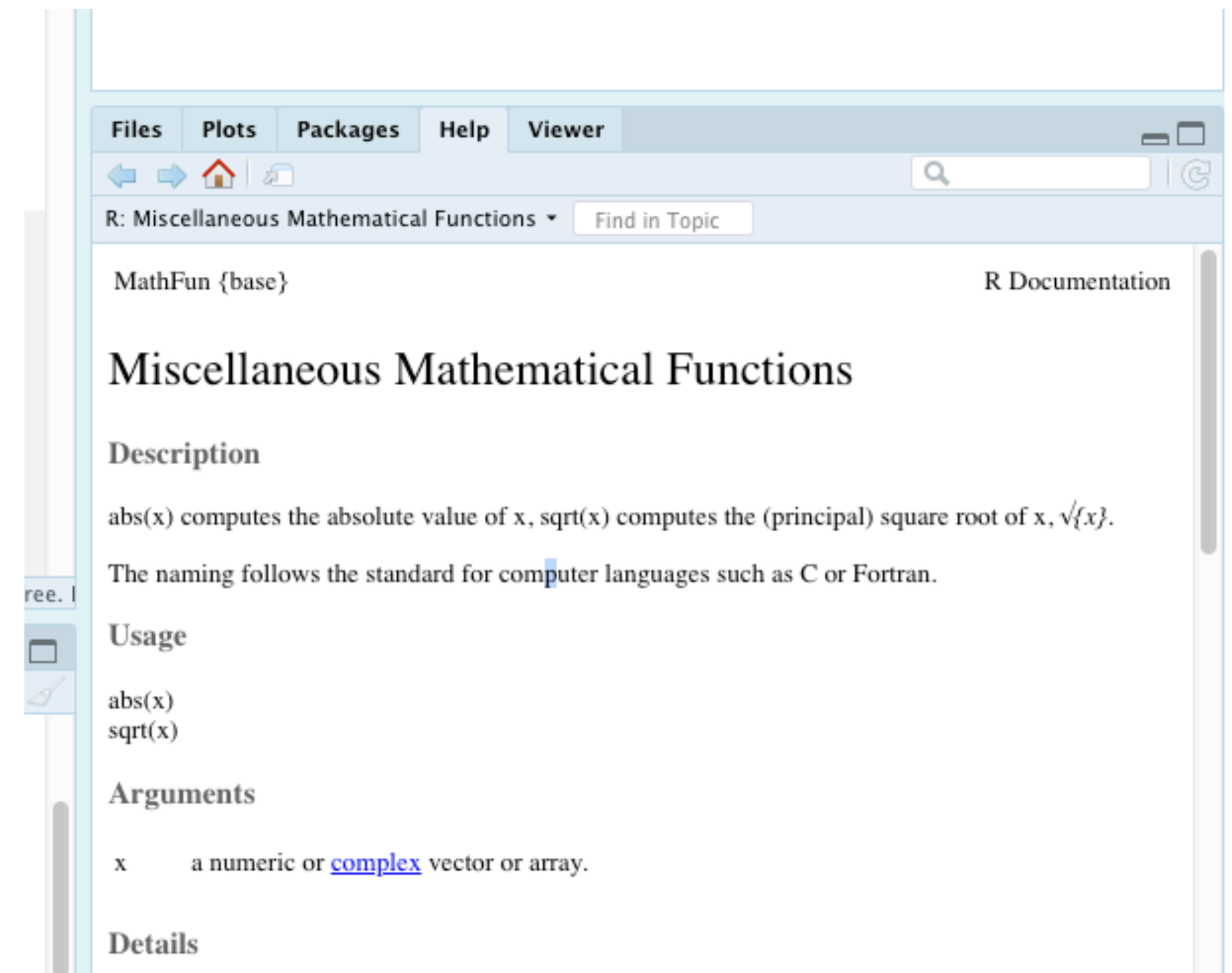
Ogni funzione è accompagnata da una documentazione racchiusa nel comando *help*. Ad esempio per conoscere la funzione *sqrt* utilizziamo il comando:

```
> help(sqrt)
```

Oppure:

```
> ?sqrt
```

L'output dell'help appare nel tab in basso a destra



First Steps

Le prime operazioni

La console può essere utilizzata come una vera e propria calcolatrice:

```
> 3+2*10/5-6
```

```
[1] 1
```

R segue l'ordine logico delle operazioni. Le operazioni di base sono:

- + : addizione
- - : sottrazione
- * : moltiplicazione
- / : divisione
- ^ : elevamento a potenza

First Steps

Le prime operazioni

Oltre agli operatori matematici, vi sono gli operatori logici:

- $>$: maggiore
- $<$: minore
- $>=$: maggiore o uguale
- $<=$: minore o uguale
- $==$: identico
- $!=$: diverso (! Indica la negazione)
- $\&$: intersezione (e)
- $|$: unione (o)

L'output di un operatore logico è a sua volta un valore logico, ovvero *TRUE*, *FALSE* e *NA*, che indicale “risposta non disponibile”.

```
> 3>2
```

```
[1] TRUE
```

```
> 3<2
```

```
[1] FALSE
```

```
> A=7
```

```
> A==7
```

```
[1] TRUE
```

```
> A!=3
```

```
[1] TRUE
```

First Steps

Gli oggetti

Variabili



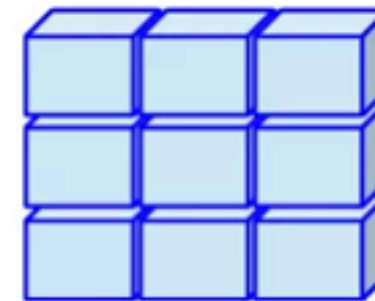
Dato in forma atomica

Vettori



Insieme lineare di
elementi omogenei per
tipologia

Matrici



Vettori bidimensionali

Fattori



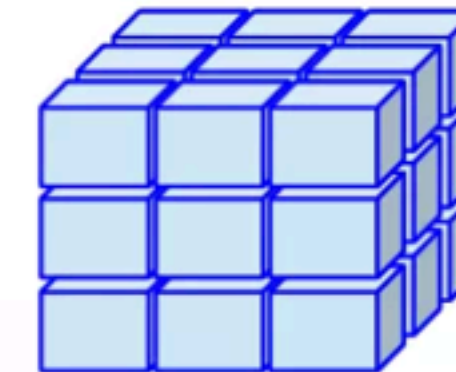
Classifica o suddivide in
livelli gli elementi di un
altro vettore

Dataframe



Matrici con vettori
eterogenei per tipo

Array



Matrici n-dimensionali

First Steps

Gli oggetti - Variabili e Vettori

Le variabili sono oggetti atomici come quelli già visti in precedenza:

```
> a = 5
```

```
> a
```

Un vettore è invece un insieme di dati omogenei. Un vettore può essere creato utilizzando la funzione `c`, che combina gli argomenti al suo interno:

```
> vec.1 = c(18,2,3,6,6,4,3)
```

Se i vettori contengono valori numerici si possono applicare le funzioni statistiche come:

```
> min(vec.1) #valore minimo
```

```
> max(vec.1) #valore massimo
```

```
> length(vec.1) #numero di valori
```

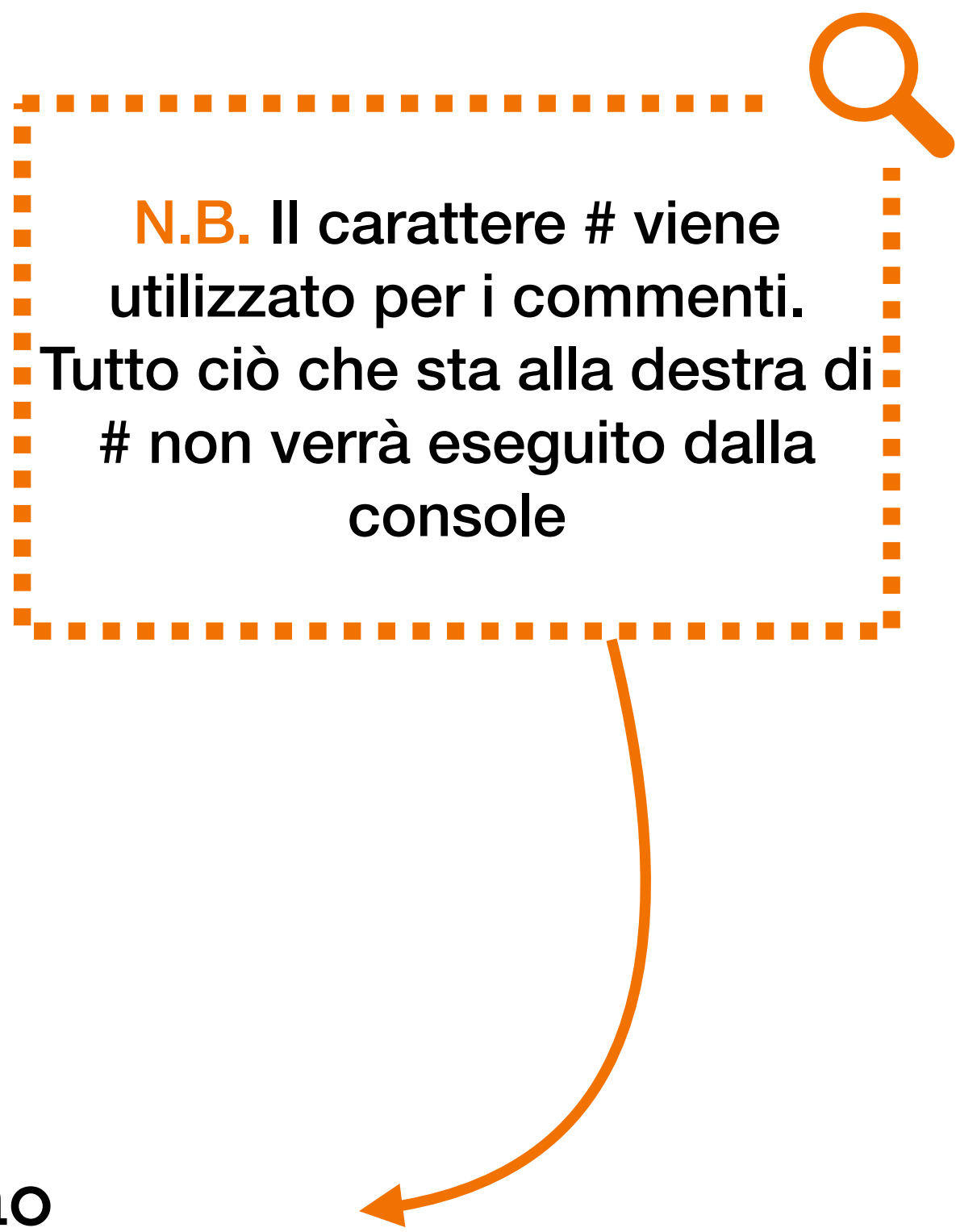
```
> mean(vec.1) #media
```

```
> median(vec.1) #valore mediano
```

```
> range(vec.1) #restituisce un vettore con min e max
```

```
> sd(vec.1) #deviazione standard
```

```
> var(vec.1) #varianza
```



N.B. Il carattere `#` viene utilizzato per i commenti. Tutto ciò che sta alla destra di `#` non verrà eseguito dalla console

First Steps

Gli oggetti - Variabili e Vettori

```
> vec.1 = c(18,2,3,6,6,4,3)
```

Si può accedere agli elementi del vettore tramite **posizione**:

> vec.1[2] #secondo elemento	[1] 2
> vec.1[-3] #tutto tranne il terzo elemento	[1] 18 2 6 6 4 3
> vec.1[2:4] #elementi dal secondo al quarto	[1] 2 3 6
> vec.1[-(2:4)] #tutto tranne gli elementi dal secondo al quarto	[1] 18 6 4 3
> vec.1[c(1,5)] #primo e quinto elemento	[1] 18 6

First Steps

Gli oggetti - Variabili e Vettori

```
> vec.1 = c(18,2,3,6,6,4,3)
```

o tramite **valore**:

<pre>> vec.1[vec.1==6] #elementi uguali a 6</pre>	<pre>[1] 6 6</pre>
<pre>> vec.1[vec.1<4] #elementi minori di 4</pre>	<pre>[1] 2 3 3</pre>
<pre>> vec.1[vec.1 %in% c(2,18)] #elementi contenuti nel set (2,18)</pre>	<pre>[1] 18 2</pre>

First Steps

Gli oggetti - Fattori

I Fattori vengono utilizzati per i vettori che contengono variabili categoriche in quanto suddividono in *livelli* gli elementi del vettore:

```
> vec.2 = c('blu', 'giallo', 'blu', 'verde', 'verde', 'blu', 'blu')
```

```
> fattore.2 = factor(vec.2)
```

```
> fattore.2
```

```
[1] blu   giallo blu   verde verde blu   blu
```

```
Levels: blu giallo verde
```

First Steps

Gli oggetti - Array e Matrici

Le matrici sono vettori bidimensionali che possono essere costruiti partendo da uno o più vettori di dati.

```
> Matrix.1 = matrix(data=
c(vec.1, vec.2), nrow=7, ncols=2)
```

```
> Matrix.1
```

	[,1]	[,2]
[1,]	"18"	"blu"
[2,]	"2"	"giallo"
[3,]	"3"	"blu"
[4,]	"6"	"verde"
[5,]	"6"	"verde"
[6,]	"4"	"blu"
[7,]	"3"	"blu"

Gli array possono invece avere anche più di due dimensioni.

```
> vec.3 = c(1,20,8,5,6,1,5)
```

```
> vec.4 = c('giallo', 'verde', 'verde', 'blu', 'verde', 'giallo',
'verde')
```

```
> Array.1 = array(data= c(vec.1, vec.2, vec.3, vec.4),
dim=c(7,2,2))
```

```
> Array.1
```

		[,1]	[,2]			[,1]	[,2]
[1,]	"18"	"blu"	[1,]	"1"	"giallo"		
[2,]	"2"	"giallo"	[2,]	"20"	"verde"		
[3,]	"3"	"blu"	[3,]	"8"	"verde"		
[4,]	"6"	"verde"	[4,]	"5"	"blu"		
[5,]	"6"	"verde"	[5,]	"6"	"verde"		
[6,]	"4"	"blu"	[6,]	"1"	"giallo"		
[7,]	"3"	"blu"	[7,]	"5"	"verde"		

First Steps

Gli oggetti - Dataframes

I Dataframes sono delle matrici in cui i tipi delle colonne possono essere diversi. È il formato dati più comodo da utilizzare in quanto le colonne (e anche le righe) possono essere identificate da un nome.

```
> dataframe = data.frame(age = vec.1, color=vec.2)
```

```
>dataframe
```

	age	color
1	18	blu
2	2	giallo
3	3	blu
4	6	verde
5	6	verde
6	4	blu
7	3	blu

First Steps

Gli oggetti - Dataframes

Per accedere ai singoli vettori (colonne) del dataframe si utilizza il metodo \$:

```
> dataframe$color
```

```
[1] blu   giallo blu   verde verde blu   blu
```

Per accedere ai singoli elementi del dataframe si utilizzano gli indici di riga e colonna:

```
> dataframe[2,1]
```


```
[1] 2
```

Oltre all'indice posizionale è possibile utilizzare il nome della colonna (riga):

```
> dataframe[3,"color"]
```

```
[1] blu
```

	age	color
1	18	blu
2	2	giallo
3	3	blu
4	6	verde
5	6	verde
6	4	blu
7	3	blu



N.B. Possono essere utilizzati tutti i metodi di accesso anche per le singole colonne (righe) del dataframe

First Steps

Importare dati esterni

L'elaborazione di dati esterni è la principale funzione di R. Proprio per questo motivo è possibile importare file di diversa natura, dai file di testo separati da tabulazioni a quelli prodotti da altri software (es: SAS).

La funzione più utilizzata per caricare dati esterni è *read.table()* della libreria *utils*:

```
> df = read.table(file.choose())
```

L'unico argomento obbligatorio della funzione *read.table* è il path del file da caricare. Per comodità si usa al suo posto la funzione *file.choose()* che fa apparire il classico pop-up di navigazione dei file.

Una volta aperto il file *esempio.csv* lo si analizza chiamandolo nella console:

```
> df
```

	V1
1	id; sesso; anni; peso; altezza
2	MT ; M ; 69 ; 76 ; 1.78
3	GF ; F ; 56 ; 63 ;
4	MC ; F ; 53 ; 71 ; 1.60
5	SB ; M ; 28 ; 73 ; 1.78
6	FE ; F ; 61 ; 54 ; 1.54
7	AB ; M ; 46 ; 92 ; 1.84
8	RF ; F ; 31 ; 81 ; 1.56

Il file risulta essere evidentemente corrotto.

L'errato caricamento del file è dovuta alla non specificazione degli altri argomenti che risultano essere necessari per questo tipo di file.

First Steps

Importare dati esterni

Per importare correttamente il file è necessario specificare la presenza dei nomi delle colonne nella prima riga del file (*header*) ed il separatore (;):

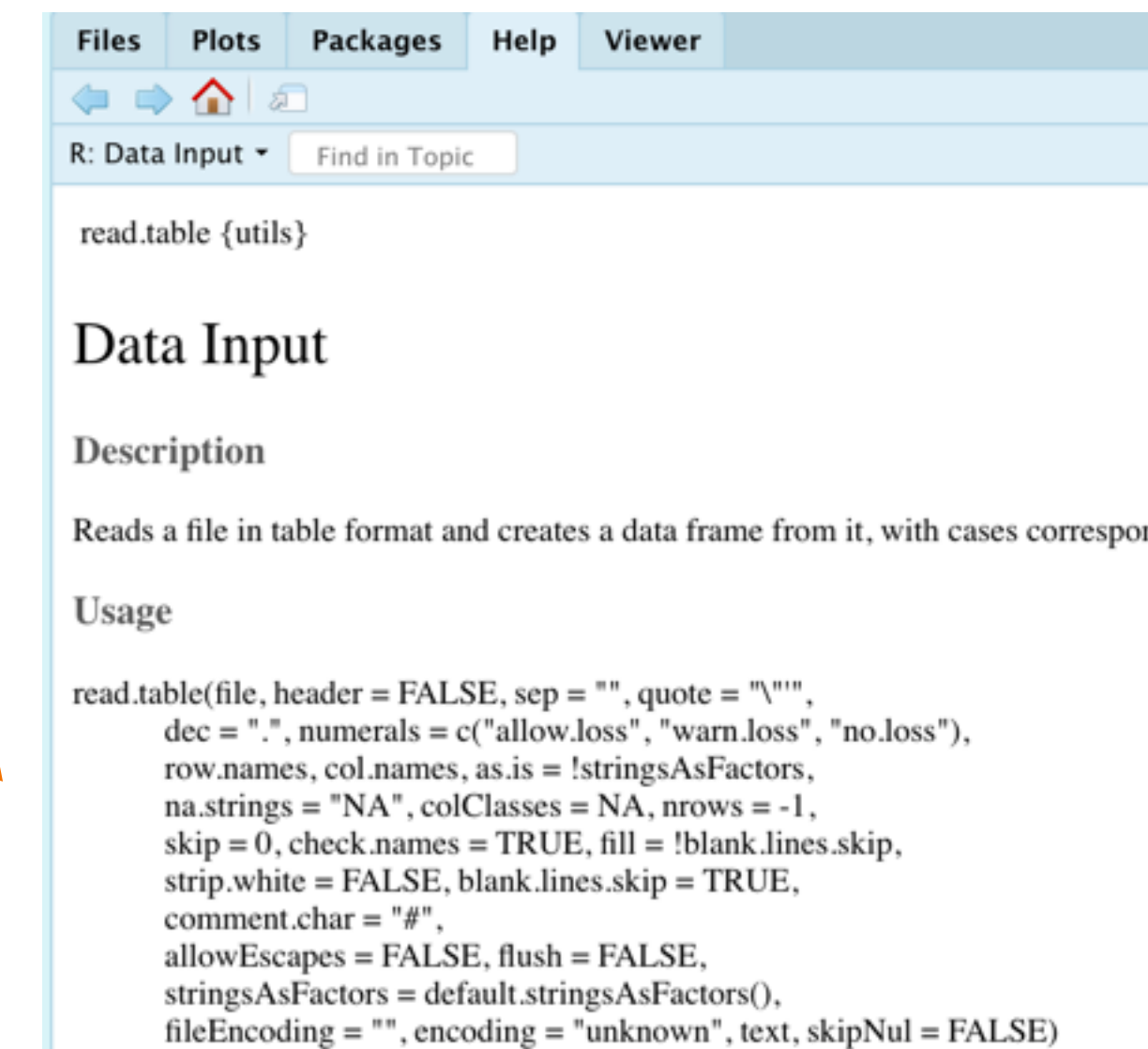
```
> df = read.table(file.choose(), sep=";", header=TRUE )
```

```
> df
```

	id	sex	age	weight	height
1	MT	M	69	76	1.78
2	GF	F	56	63	NA
3	MC	F	53	71	1.60
4	SB	M	28	73	1.78
5	FE	F	61	54	1.54
6	AB	M	46	92	1.84
7	RF	F	31	81	1.56

Si noti che oltre al *sep* e *header* sono presenti molti altri argomenti nella funzione *read.table*, proprio per la moltitudine di file che possono essere importati.

```
> ?read.table
```



The screenshot shows the R help page for the `read.table` function. The window has tabs for Files, Plots, Packages, Help, and Viewer. The 'Help' tab is active, showing the 'R: Data Input' section. The 'Data Input' section includes a 'Description' and a 'Usage' section. The 'Usage' section contains the full signature of the `read.table` function with all its arguments and their default values.

```
read.table {utils}
```

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases correspor

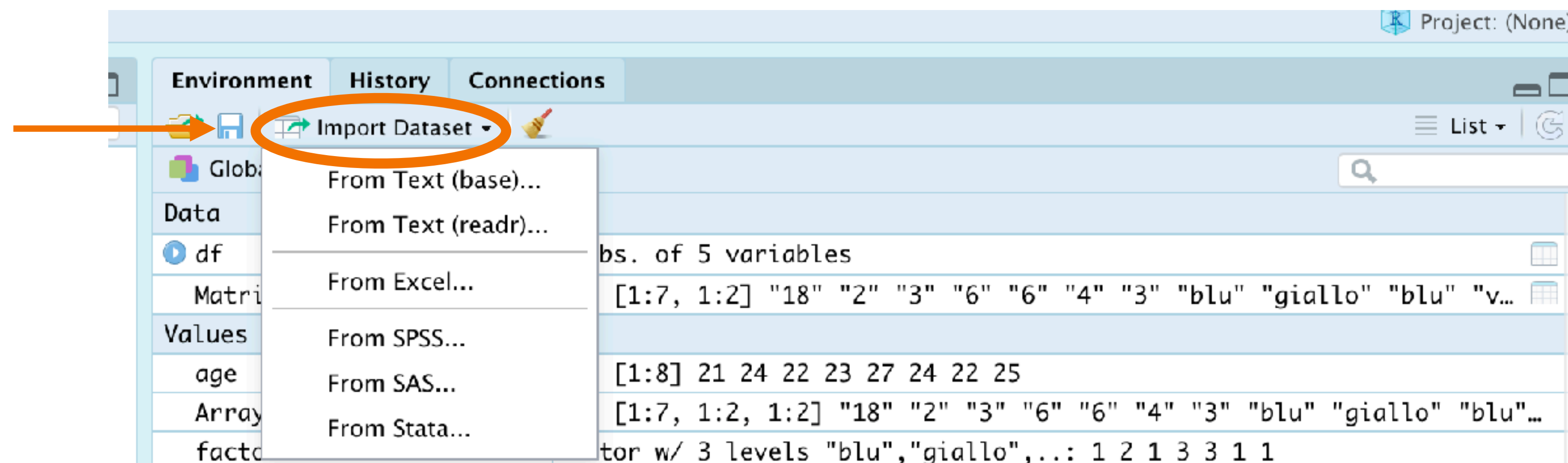
Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"",  
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
  row.names, col.names, as.is = !stringsAsFactors,  
  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 = "#",  
  allowEscapes = FALSE, flush = FALSE,  
  stringsAsFactors = default.stringsAsFactors(),  
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```


First Steps

Importare dati esterni

Come gran parte delle funzioni di bas, l'import dei dati può essere eseguito sia da console che in modo grafico sfruttando l'apposito tab in Rstudio.



First Steps

Importare dati esterni

Row names

Automatic

Automatic

Use first column

Use numbers

Separator

Semicolon

Whitespace

Comma

Semicolon

Tab

Comments

None

None

#

!

%

@

/

~

Decimal

Period

Period

Comma

Import Dataset

Name

esempio

Encoding

Automatic

Heading

Yes

No

Row names

Automatic

Separator

Semicolon

Decimal

Period

Quote

Double quote (")

Comment

None

na.strings

NA

☒ Strings as factors

Input File

id; sesso; anni; peso; altezza
MT ; M ; 69 ; 76 ; 1.78
GF ; F ; 56 ; 63 ;
MC ; F ; 53 ; 71 ; 1.60
SB ; M ; 28 ; 73 ; 1.78
FE ; F ; 61 ; 54 ; 1.54
AB ; M ; 46 ; 92 ; 1.84
RF ; F ; 31 ; 81 ; 1.56

Data Frame

id	sesso	anni	peso	altezza
MT	M	69	76	1.78
GF	F	56	63	NA
MC	F	53	71	1.60
SB	M	28	73	1.78
FE	F	61	54	1.54
AB	M	46	92	1.84
RF	F	31	81	1.56

Import

Cancel

Environment

History

Connections

View(dataframe)

rm(dataframe)

t

First Steps

Esercizi