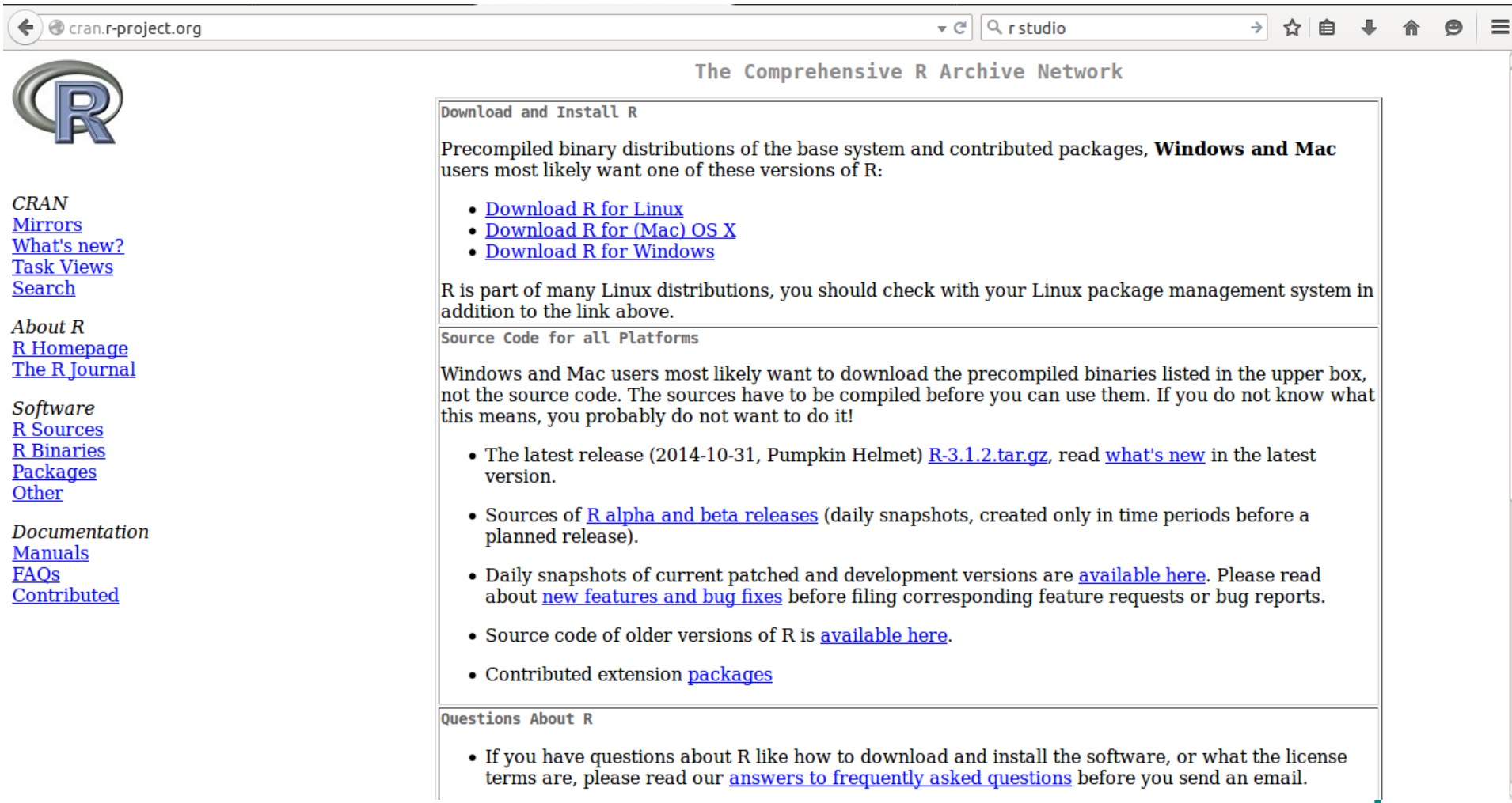
Emanuele Baratti

DICAM – Costruzioni idrauliche

Mail: emanuele.baratti@unibo.it

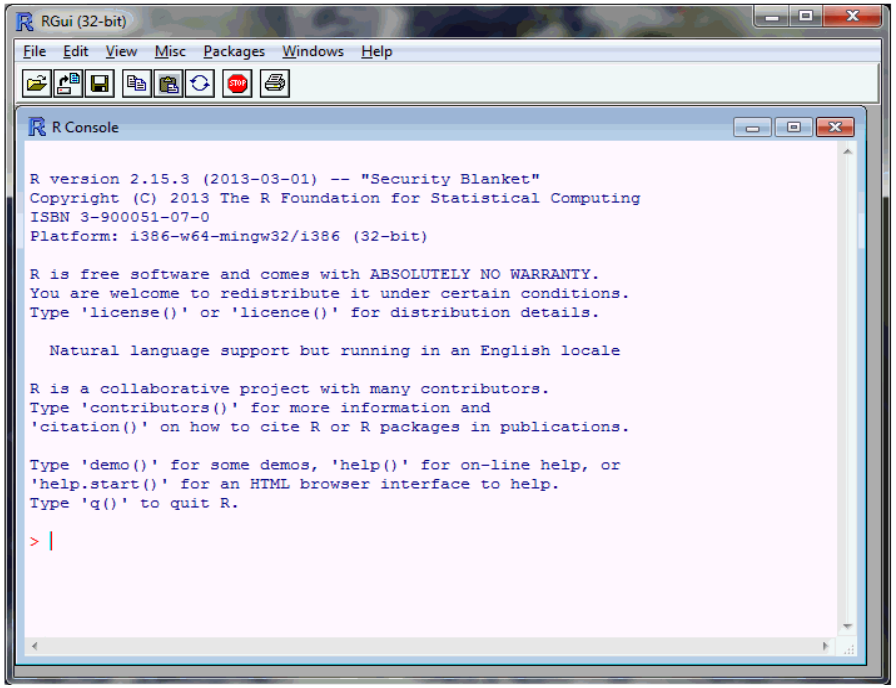
R


cran.r-project.org
r studio
The Comprehensive R Archive Network
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
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
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 (2014-10-31, Pumpkin Helmet) R-3.1.2.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
Questions About R
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.

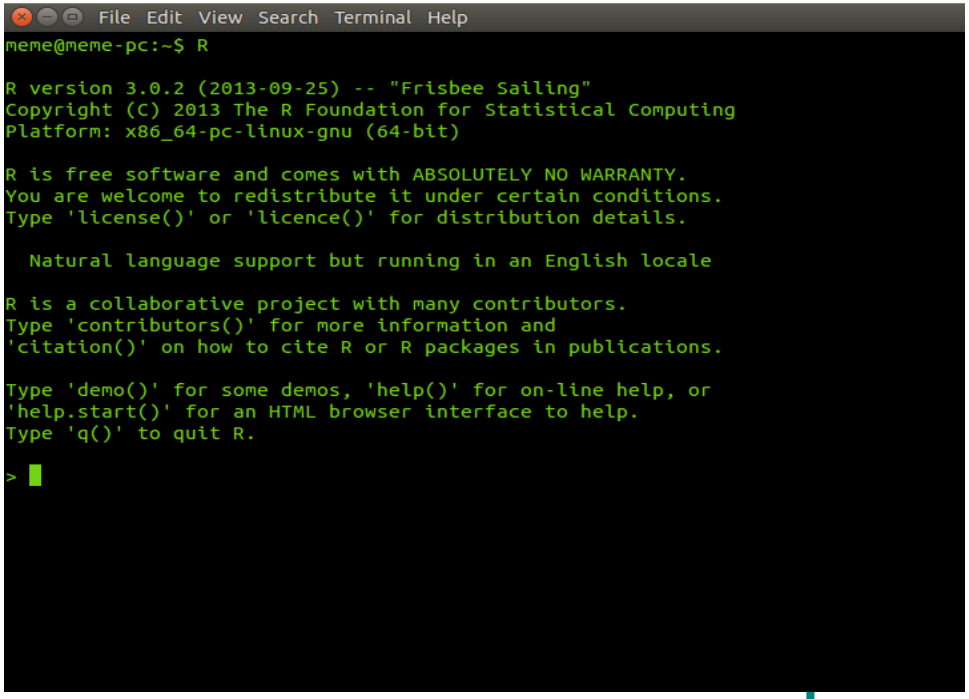
The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Is a powerful statistics system.

Windows system



UNIX system



R operates on "Objects":

- vectors

- matrices

- data frames

- lists

- factors

- “user define” functions / R functions

- Import/export object
  - >read.table(...)
  - > scan(...)
  - > write.table (...)

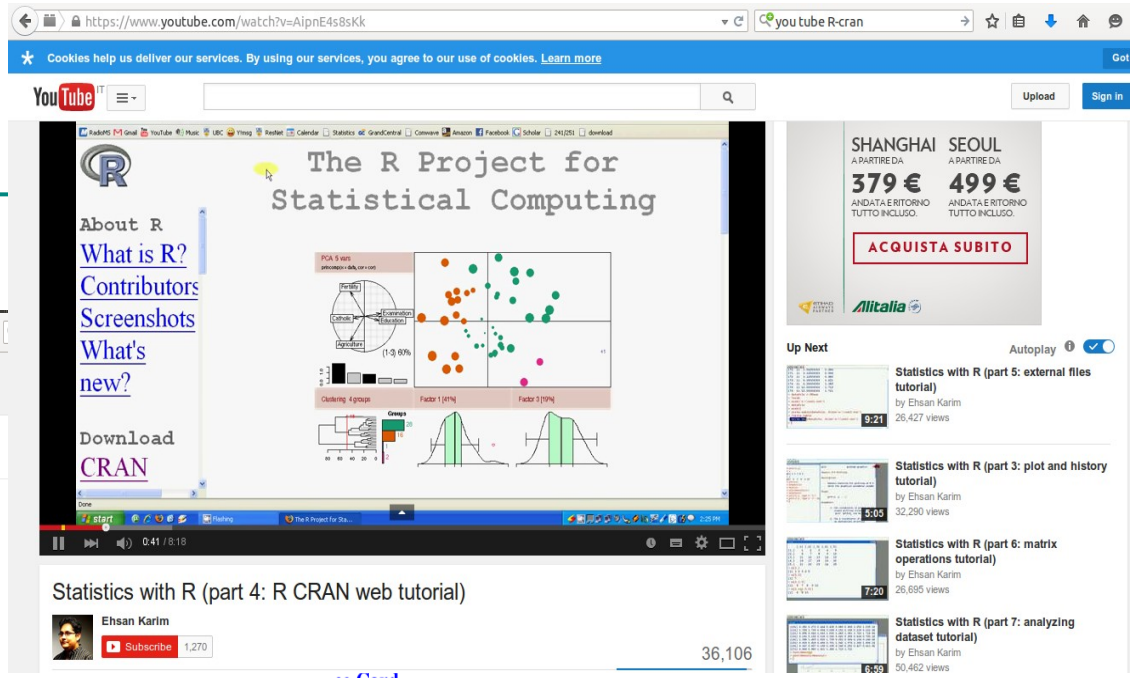
- Conditional structures
  - If,then,else
  - For, while loop

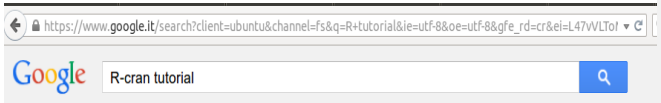
- Graphical procedures
  - Plot, hist, boxplot

cran.r-project.org/doc/manuals/r-release/R-intro.html

An Introduction to R

Table of Contents





ce Card

peac.com 2004-11-07
v.Rpad.org for the source and latest
ginners by Emmanuel Paradis (with

tion.
n

help system
objects in the search list matching

n of help
of an R object
isually a statistical summary but it is
perations for different classes of a

`ls()` show objects in the search path; specify `pat="pat"` to search on a pattern
`ls.str()` str() for each variable in the search path
`dir()` show files in the current directory
`methods(a)` shows S3 methods of a
`methods(class=class(a))` lists all the methods to handle objects of class a

Input and output

`load()` load the datasets written with `save`
`data(x)` loads specified data sets
`library(x)` load add-on packages
`read.table(file)` reads a file in table format and creates a data frame from it; the default separator `sep=""` is any whitespace; use `header=TRUE` to read the first line as a header of column names; use `as.is=TRUE` to prevent character vectors from being converted to factors; use `comment.char=""` to prevent "#" from being interpreted as a comment; use `skip=n` to skip `n` lines before reading data; see the help for options on row naming, NA treatment, and others
`read.csv("filename",header=TRUE)` id. but with defaults set for reading comma-delimited files
`read.delim("filename",header=TRUE)` id. but with defaults set for reading tab-delimited files
`read.fwf(file,widths,header=FALSE,sep="",as.is=FALSE)` read a table of fixed width formatted data into a 'data.frame'; `widths` is an integer vector, giving the widths of the fixed-width fields

character or factor columns are surrounded by quotes ("); `sep` is the field separator; `eol` is the end-of-line separator; `na` is the string for missing values; use `col.names=NA` to add a blank column header to get the column headers aligned correctly for spreadsheet input
`sink(file)` output to `file`, until `sink()`
Most of the I/O functions have a `file` argument. This can often be a character string naming a file or a connection. `file=""` means the standard input or output. Connections can include files, pipes, zipped files, and R variables.
On windows, the file connection can also be used with `description = "clipboard"`. To read a table copied from Excel, use
`x <- read.delim("clipboard")`
To write a table to the clipboard for Excel, use
`write.table(x,"clipboard",sep="\t",col.names=NA)`
For database interaction, see packages `RODBC`, `DBI`, `RMySQL`, `RPgSQL`, and `ROracle`. See packages `XML`, `hdf5`, `netCDF` for reading other file formats.

Data creation

`c(...)` generic function to combine arguments with the default forming a vector; with `recursive=TRUE` descends through lists combining all elements into one vector
`from:to` generates a sequence; ":" has operator priority; 1:4 + 1 is "2,3,4,5"
`seq(from,to)` generates a sequence `by=` specifies increment; `length=` specifies desired length
`seq(along=x)` generates `1, 2, ..., length(along)`; useful for for loops
`rep(x,times)` replicate `x` `times`; use `each=` to repeat "each" element of `x` `each` times; `rep(c(1,2,3),2)` is 1 2 3 1 2 3; `rep(c(1,2,3),each=2)` is 1 1 2 2 3 3
`data.frame(...)` create a data frame of the named or unnamed arguments; `data.frame(v=1:4,ch=c("a","B","c","d"),n=10);` shorter vectors are recycled to the length of the longest
`list(...)` create a list of the named or unnamed arguments; `list(a=c(1,2),b="hi",c=3i);`
`array(x,dim=)` array with data `x`; specify dimensions like `dim=c(3,4,2)`; elements of `x` recycle if `x` is not long enough
`matrix(x,nrow=,ncol=)` matrix; elements of `x` recycle
`factor(x,levels=)` encodes a vector `x` as a factor
`gl(n,k,length=n*k,labels=1:n)` generate levels (factors) by specifying the pattern of their levels; `k` is the number of levels, and `n` is the number of replications
`expand.grid()` a data frame from all combinations of the supplied vectors or factors
`rbind(...)` combine arguments by rows for matrices, data frames, and others
`cbind(...)` id. by columns

Slicing and extracting data

Indexing lists
`x[n]` list with elements `n`
`x[[n]]` n^{th} element of the list
`x[["name"]]` element of the list named "name"
`x$name` id.
Indexing matrices
`x[i,j]` element at row `i`, column `j`
`x[i,]` row `i`
`x[,j]` column `j`
`x[,c(1,3)]` columns 1 and 3
`x["name",]` row named "name"
Indexing data frames (matrix indexing plus the following)
`x[["name"]]` column named "name"
`x$name` id.

Variable conversion

`as.array(x), as.data.frame(x), as.numeric(x), as.logical(x), as.complex(x), as.character(x), ...` convert type; for a complete list, use `methods(as)`

Variable information

`is.na(x), is.null(x), is.array(x), is.data.frame(x), is.numeric(x), is.complex(x), is.character(x), ...` test for type; for a complete list, use `methods(is)`
`length(x)` number of elements in `x`
`dim(x)` Retrieve or set the dimension of an object; `dim(x) <- c(3,2)`
`dimnames(x)` Retrieve or set the dimension names of an object
`nrow(x)` number of rows; `NROW(x)` is the same but treats a vector as a one-row matrix
`ncol(x)` and `NCOL(x)` id. for columns
`class(x)` get or set the class of `x`; `class(x) <- "myclass"`
`unclass(x)` remove the class attribute of `x`
`attr(x,which)` get or set the attribute `which` of `x`
`attributes(obj)` get or set the list of attributes of `obj`

Data selection and manipulation

`which.max(x)` returns the index of the greatest element of `x`
`which.min(x)` returns the index of the smallest element of `x`
`rev(x)` reverses the elements of `x`
`sort(x)` sorts the elements of `x` in increasing order; to sort in decreasing order: `rev(sort(x))`
`cut(x,breaks)` divides `x` into intervals (factors); `breaks` is the number of cut intervals or a vector of cut points
`match(x, y)` returns a vector of the same length than `x` with the elements of `x` which are in `y` (NA otherwise)

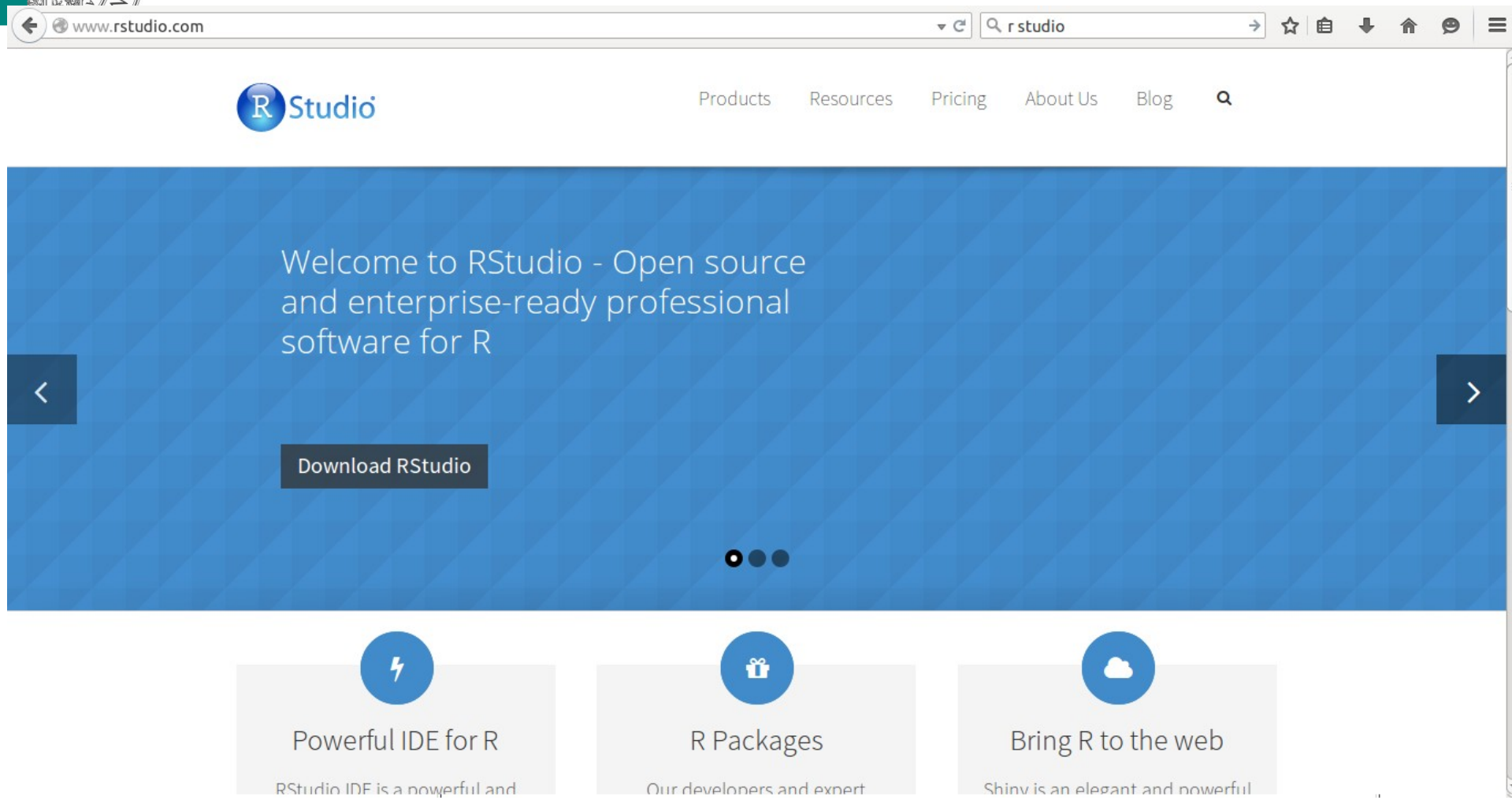
Help function!!

>help(...)

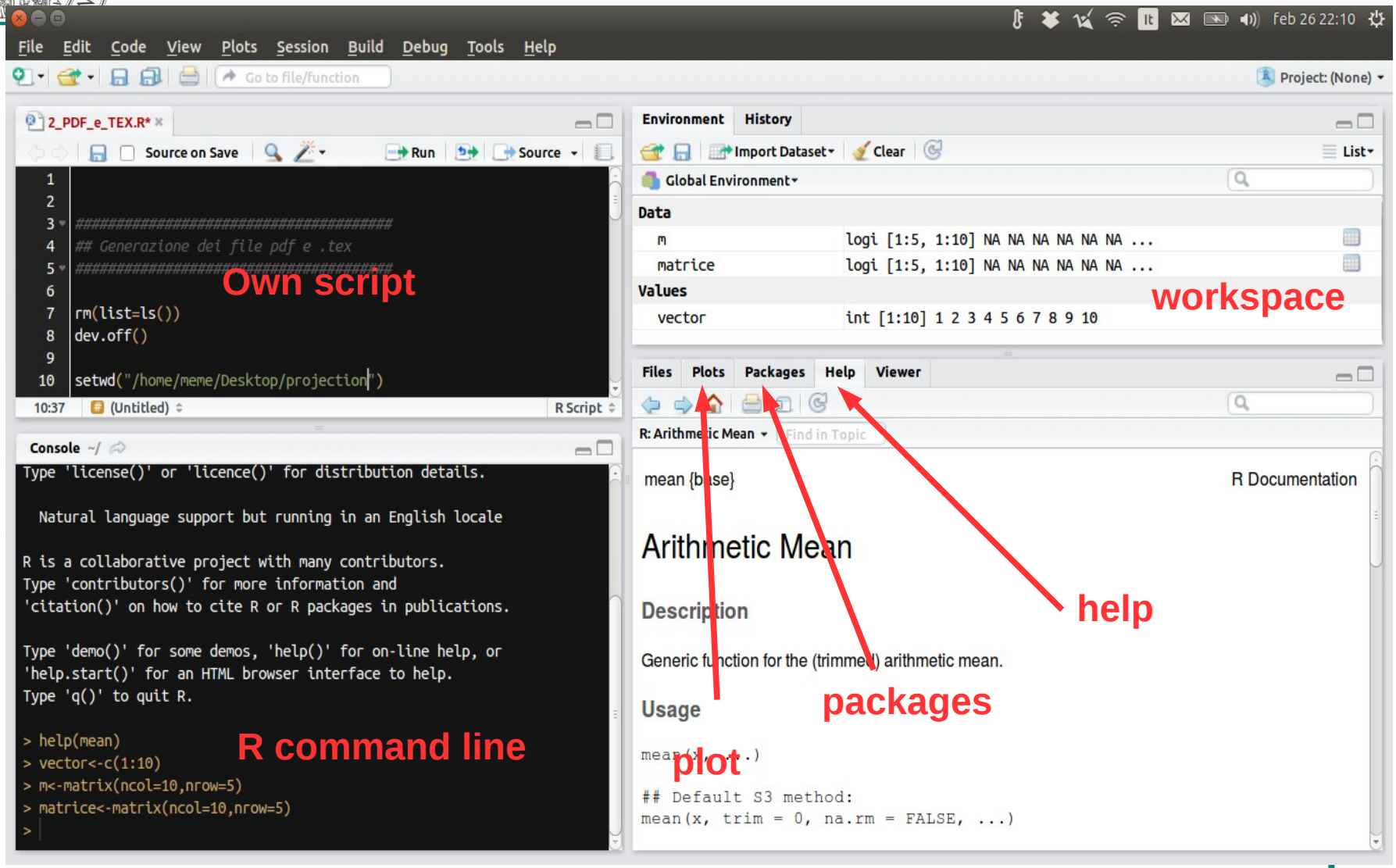

WORK IN PROGRESS


WORK IN PROGRESS

RStudio


www.rstudio.com
r studio
RStudio
Products
Resources
Pricing
About Us
Blog
Welcome to RStudio - Open source and enterprise-ready professional software for R
Download RStudio
Powerful IDE for R
R Packages
Bring R to the web

RStudio


File Edit Code View Plots Session Build Debug Tools Help
Project: (None)
2_PDF_e_TEX.R*
Source on Save
Run
Source
1
2
3 ##################################
4 ## Generazione dei file pdf e .tex
5 ##################################
6
7 rm(list=ls())
8 dev.off()
9
10 setwd("/home/meme/Desktop/projection")
Own script
10:37
(Untitled)
R Script
Console ~/
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> help(mean)
> vector<-c(1:10)
> m<-matrix(ncol=10,nrow=5)
> matrice<-matrix(ncol=10,nrow=5)
>
R command line
Environment
History
Import Dataset
Clear
List
Global Environment
Data
m
logi [1:5, 1:10] NA NA NA NA NA NA ...
matrice
logi [1:5, 1:10] NA NA NA NA NA NA ...
Values
vector
int [1:10] 1 2 3 4 5 6 7 8 9 10
workspace
Files
Plots
Packages
Help
Viewer
R: Arithmetic Mean
Find in Topic
mean {base}
R Documentation
Arithmetic Mean
Description
Generic function for the (trimmed) arithmetic mean.
Usage
mean(x, ...)
Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
help
packages
plot