R – a statistical programming language

Installation and Usage

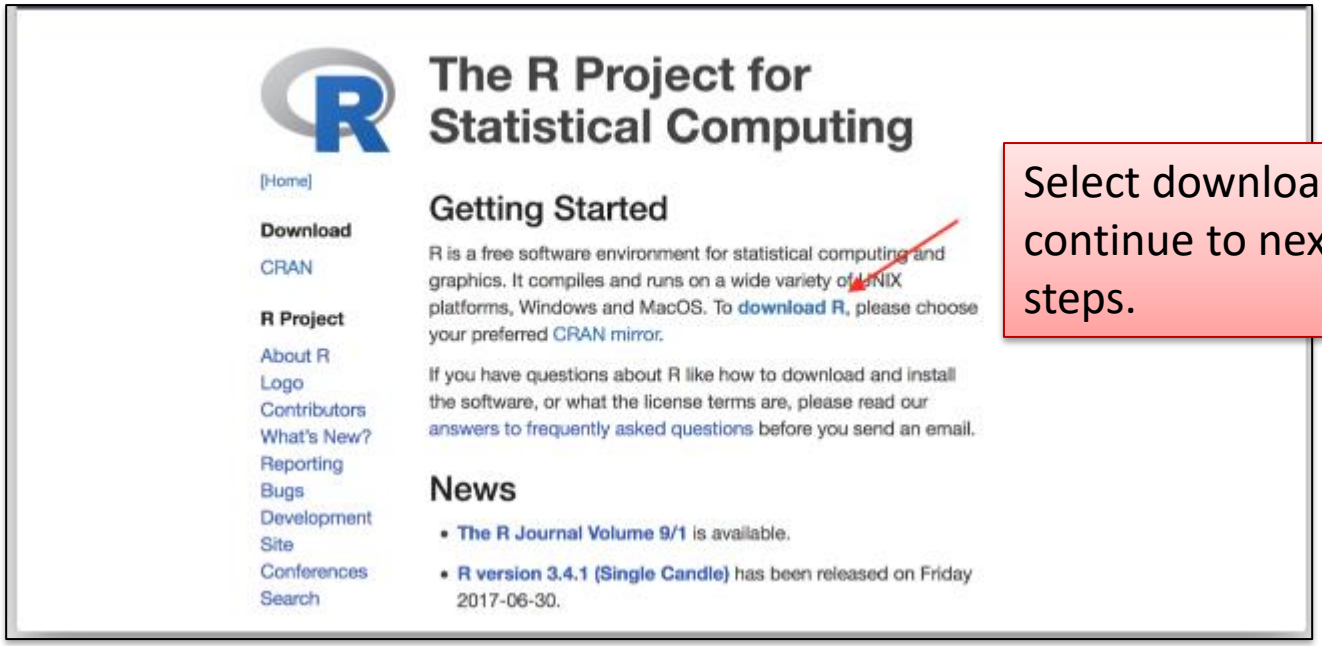
Download and Install

R and RStudio

- To work with R install 2 software packages:
  - **R environment**: provides the environment to execute R commands and scripts.
  - **RStudio**: An IDE (interface) to facilitate users working with R.

Download and Install R(1)

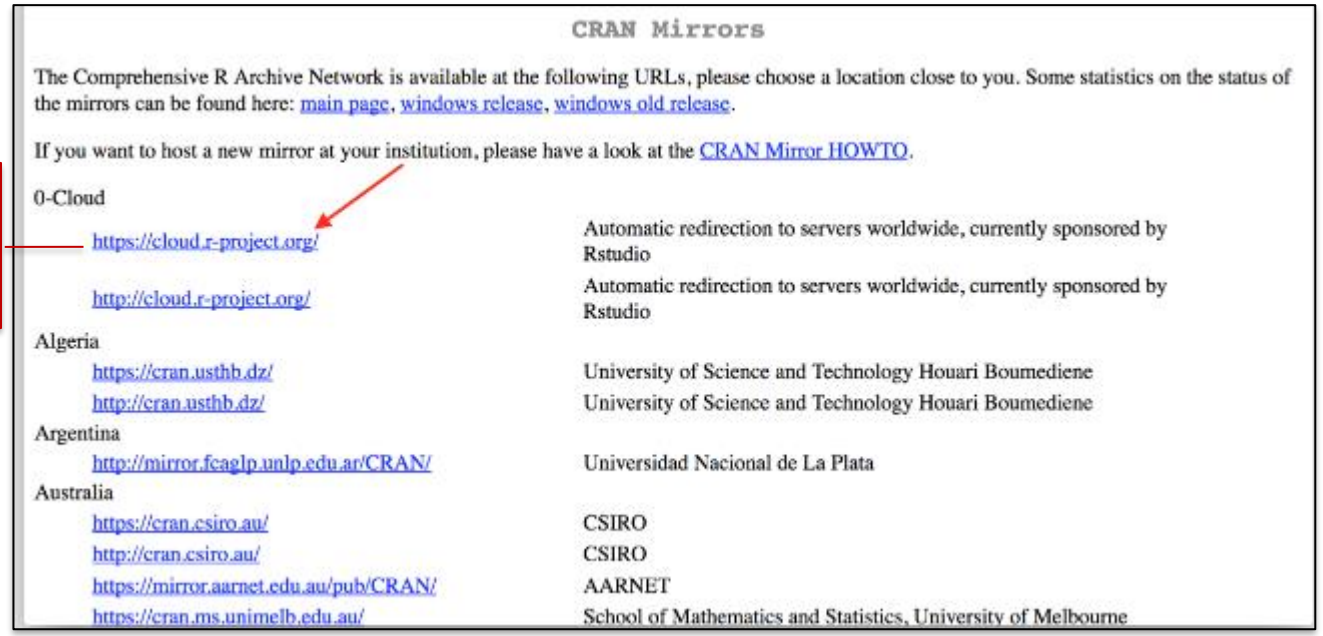
- https://www.r-project.org/



Download and Install R(2)

- Available mirrors

Select first mirror



CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page, windows release, windows old release.

If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

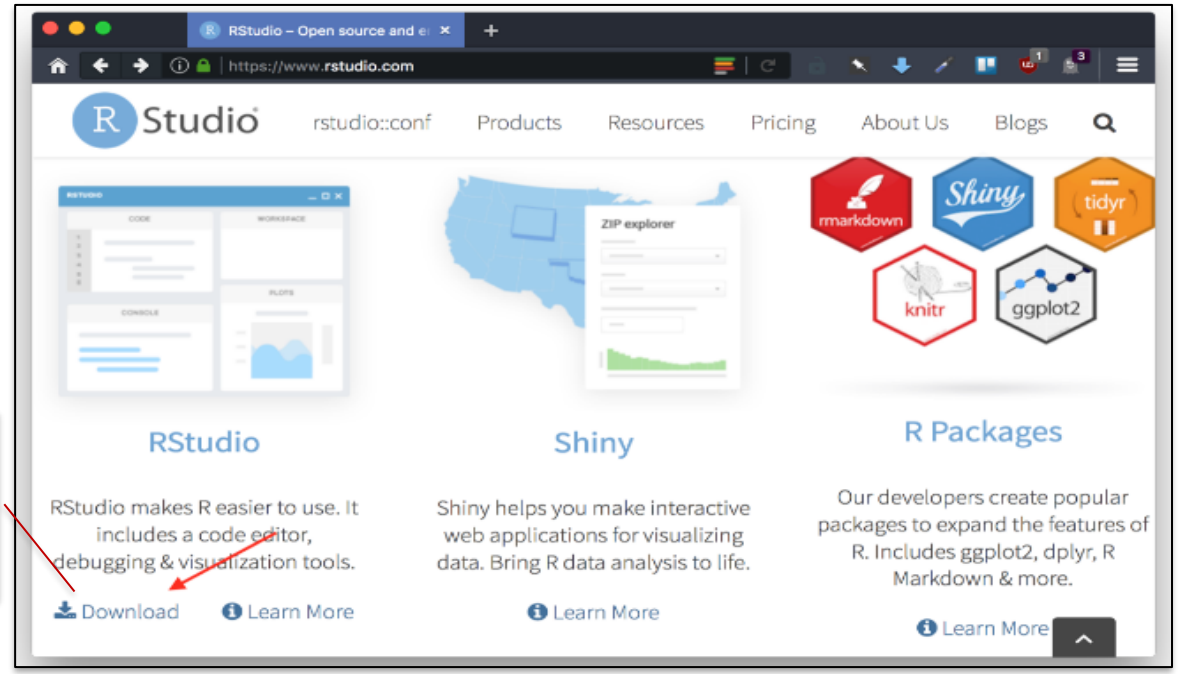
0-Cloud	
https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
http://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	
https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
http://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	
http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	
https://cran.csiro.au/	CSIRO
http://cran.csiro.au/	CSIRO
https://mirror.aarnet.edu.au/pub/CRAN/	AARNET
https://cran.ms.unimelb.edu.au/	School of Mathematics and Statistics, University of Melbourne

Download and Install R(3)

- Next, select Operating System (e.g. Windows).
- Select *Install R for the first time.*
- Select *Download R for Windows.*

Download and Install RStudio(1)

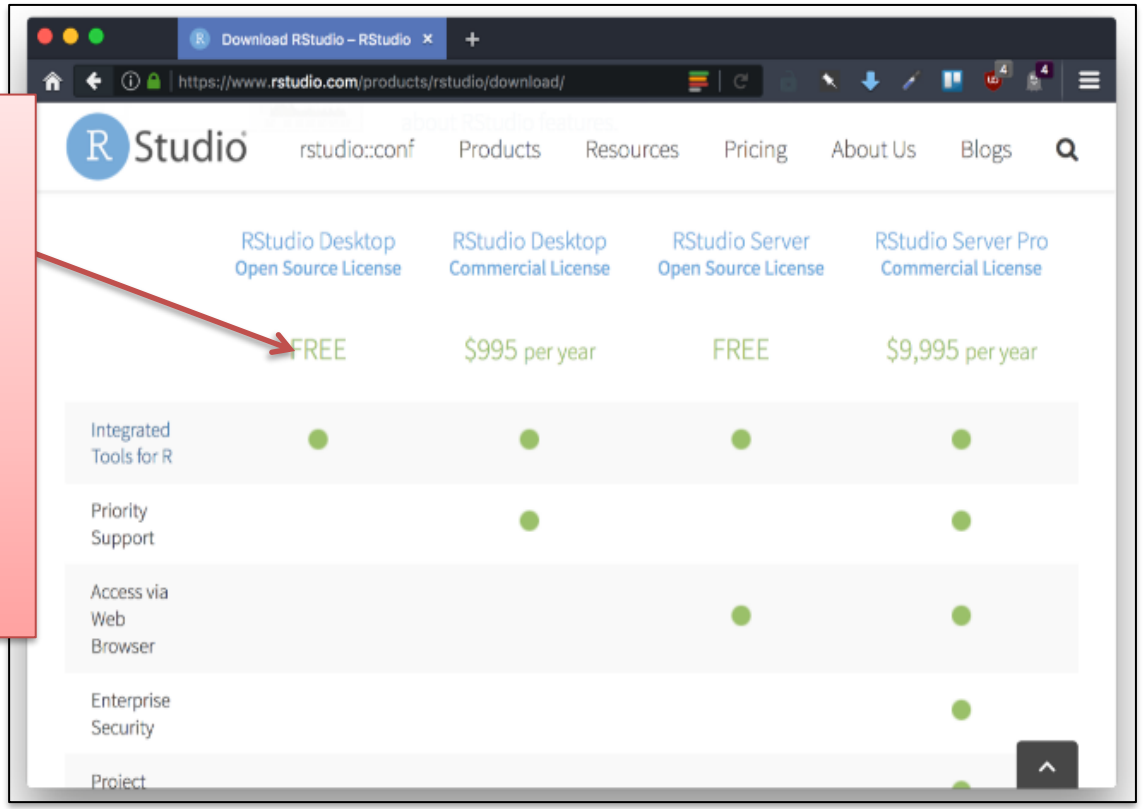
- https://www.rstudio.com/



Follow the 'Download' button

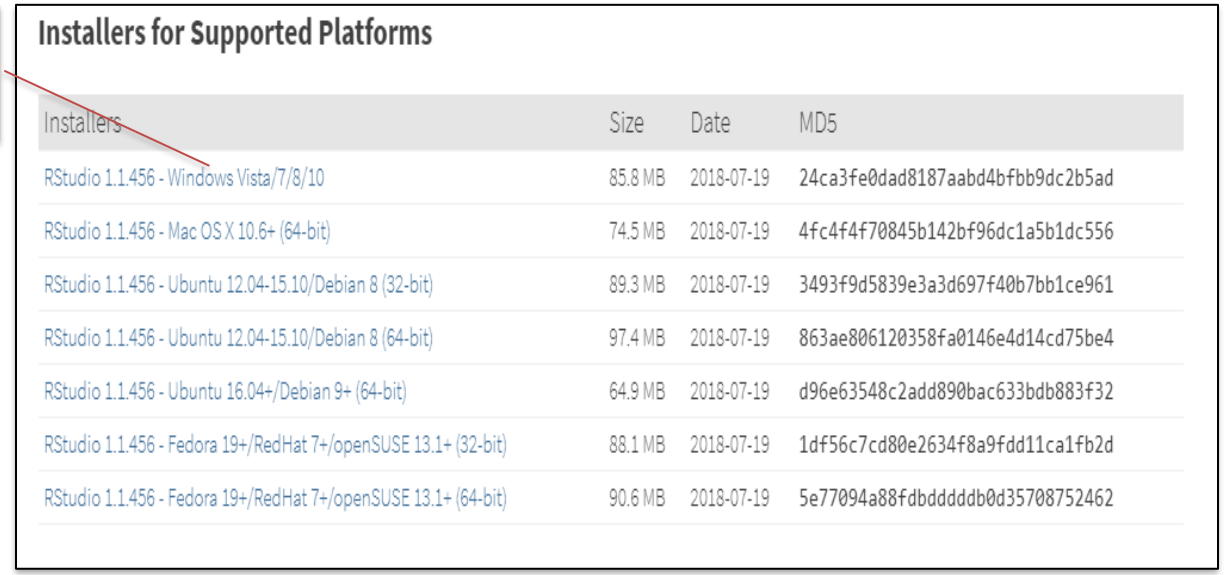
Download and Install RStudio(2)

For ‘RStudio Desktop - Open source license’ version, scroll down and click the ‘Download’ button.



Download and Install RStudio(3)

e.g. Installer for Windows



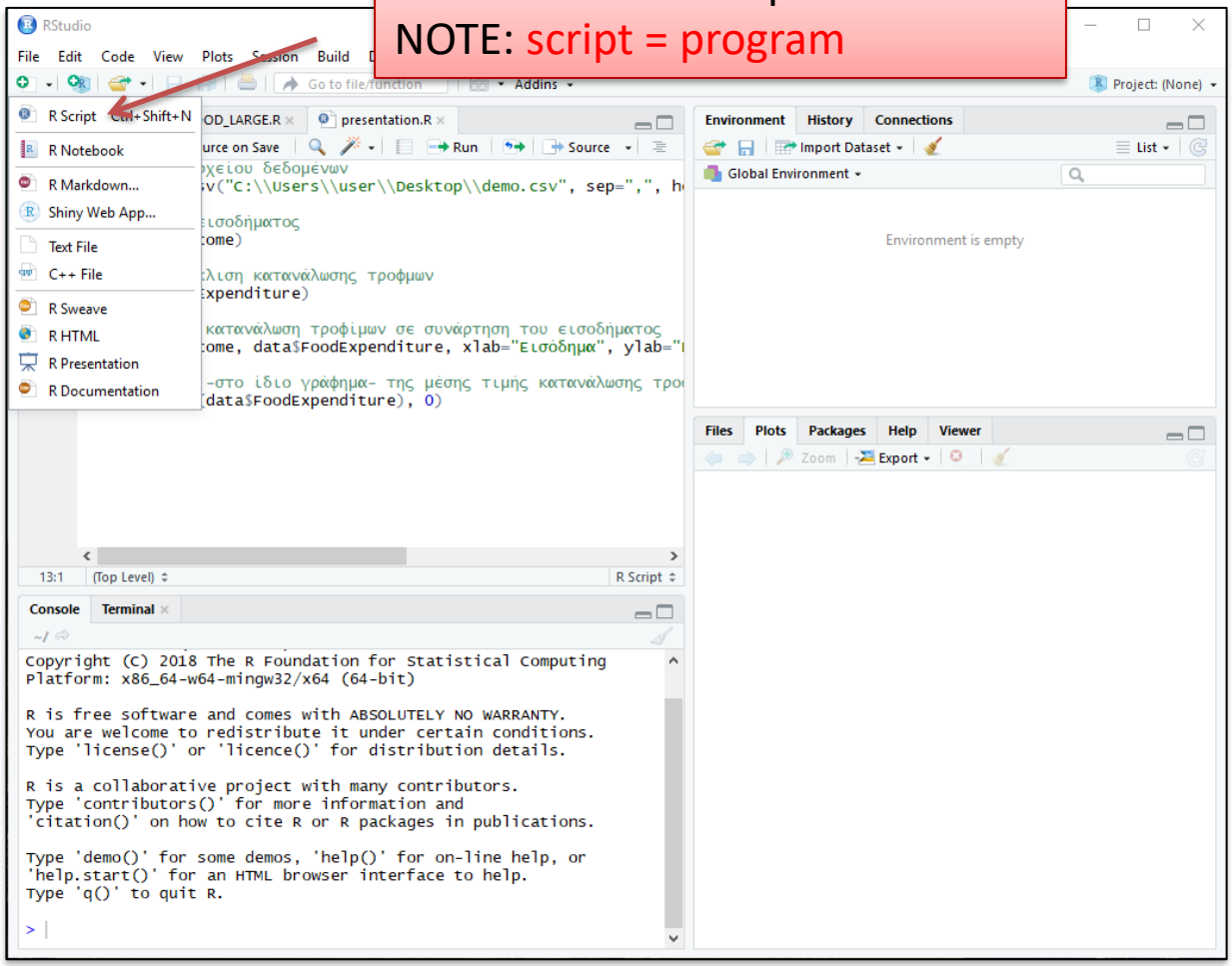
Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.456 - Windows Vista/7/8/10	85.8 MB	2018-07-19	24ca3fe0dad8187aabd4bfbb9dc2b5ad
RStudio 1.1.456 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-07-19	4fc4f4f70845b142bf96dc1a5b1dc556
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-07-19	3493f9d5839e3a3d697f40b7bb1ce961
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-07-19	863ae806120358fa0146e4d14cd75be4
RStudio 1.1.456 - Ubuntu 16.04+/Debian 9+ (64-bit)	64.9 MB	2018-07-19	d96e63548c2add890bac633bdb883f32
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-07-19	1df56c7cd80e2634f8a9fdd11ca1fb2d
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-07-19	5e77094a88fdbdddddb0d35708752462

RStudio Interface

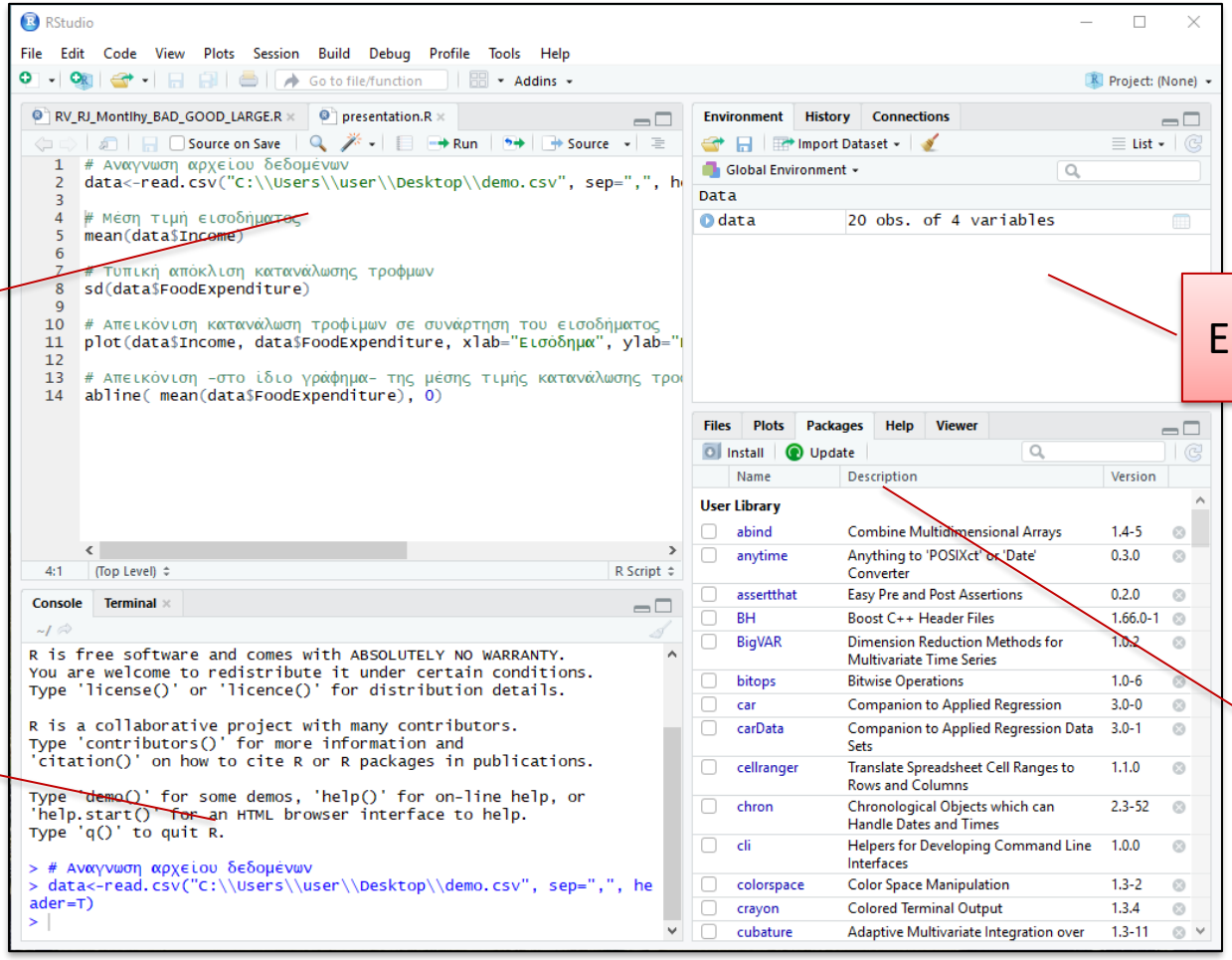
Rstudio Interface

Create a new R Script.
NOTE: script = program



Rstudio Panes

Editor.
Write your R program here



Environment

Console

Plots, Packages, Help

R sessions in RStudio (Basic Examples)

- An R session starts R and allows us to type command lines.
  - R Session? A set of variables that define the context of execution
  - R allows definition of variables of various data types. E.g. built – in datatypes **data frame, numeric, matrix, vector character, Lists, Factors** but also user-defined data types.
- Execute R code directly from the console or save commands as script files (plain text files that contain R code).
- A Package in R is a collection of functions, compiled code, data.
- Many packages are already installed.
- Install additional packages into Rstudio with two different methods.
- Packages allow us to perform specific functions.

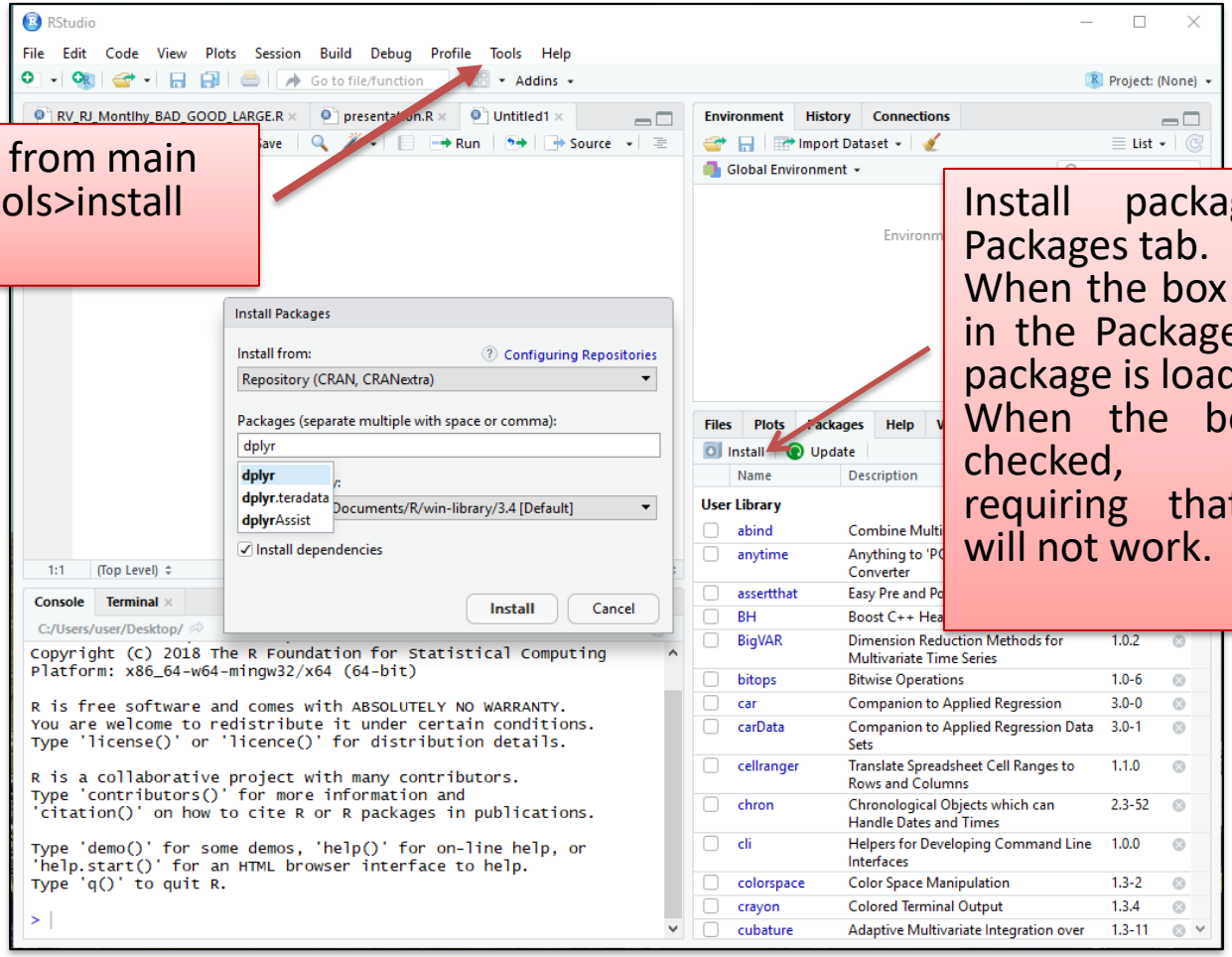
R libraries

R libraries

- R offers a great number of libraries (created by others) that enables the use of the appropriate statistical (or other) method
  - Such libraries is the strong aspect of R! You don't need to create them yourself – someone else has probably created a similar one.
- Make a package's contents available to use in the current R session.
- To use the package for a specific functionality we use the function library().
- library() function loads the package into memory.
- We can load as many libraries as we need.
- library(e1071)→Naïve Bayes classifier.
- library(stats)→k-means clustering.
- library(randomForest)→Random Forest classification and regression.

Install Packages - GUI method

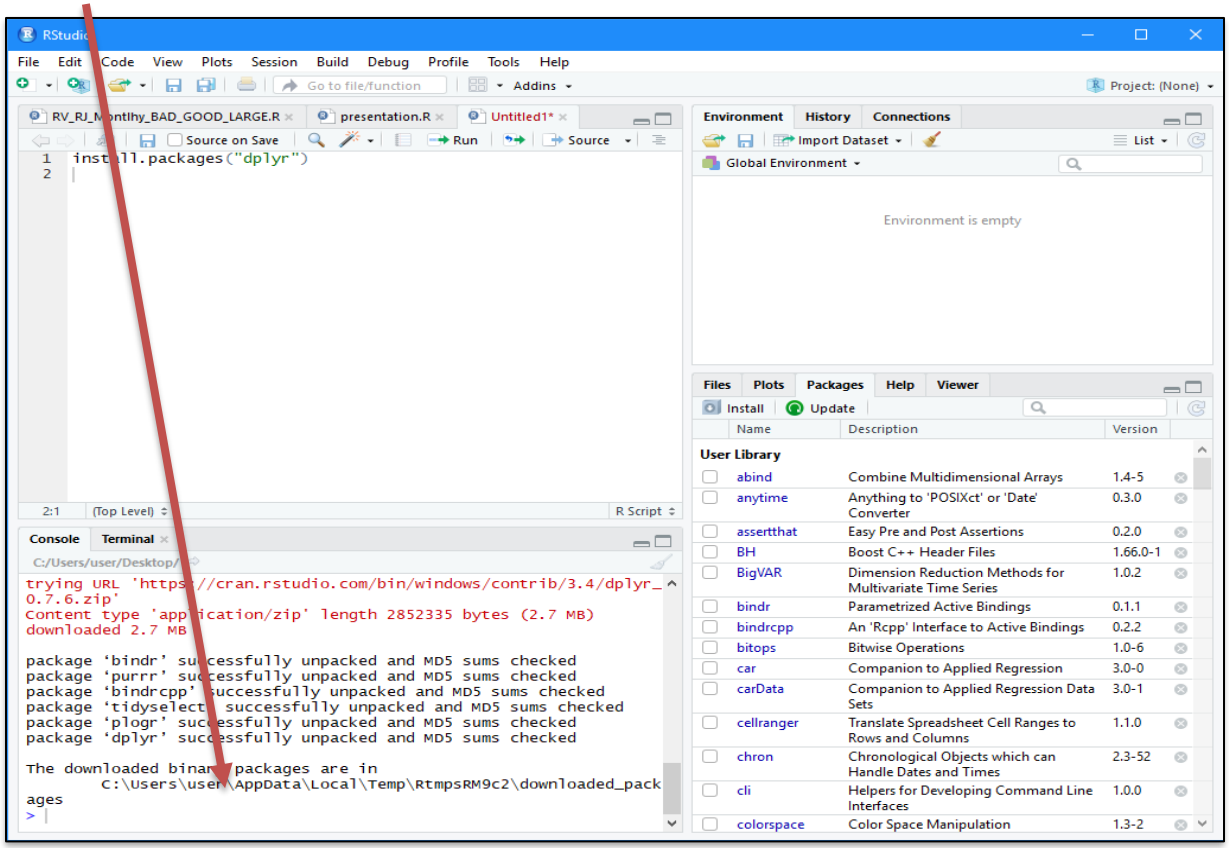
Install packages from main menu. Select Tools>install packages.



Install packages from Packages tab.
When the box is checked in the Packages Tab, the package is loaded.
When the box is not checked, commands requiring that package will not work.

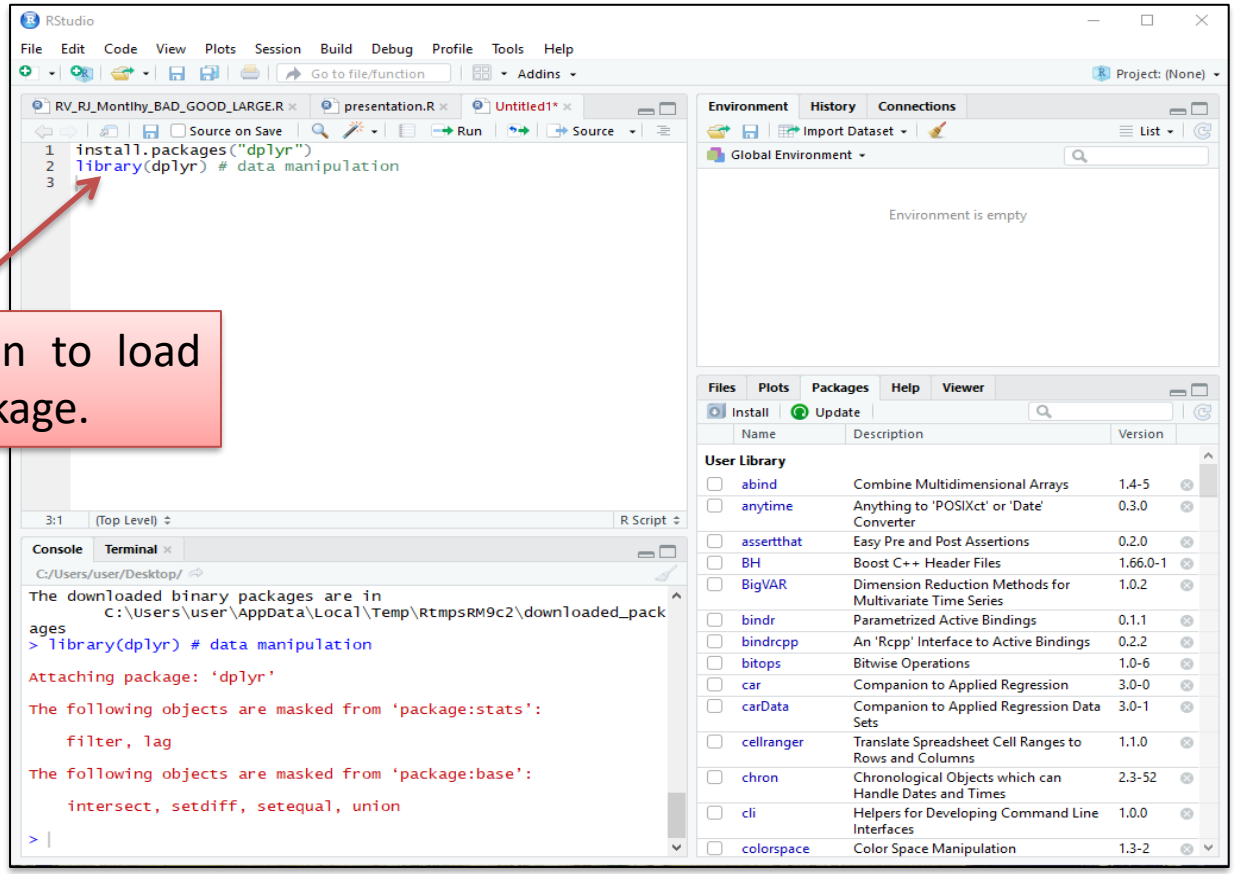
Install Packages - Console method

- The install.packages() function.



Manage/Use Packages

- library()



library() function to load an *installed* package.

Using R

R language

- As any other language it has the basic commands and operators such as (but slightly different syntax):
  - **Assignment** - create a new variable and assign a value to it using the <- operator. E.g. x← -42 create a variable named x and assign the value -42 to it.
  - **Control flow** with if else
  - **Iteration** with for and while.

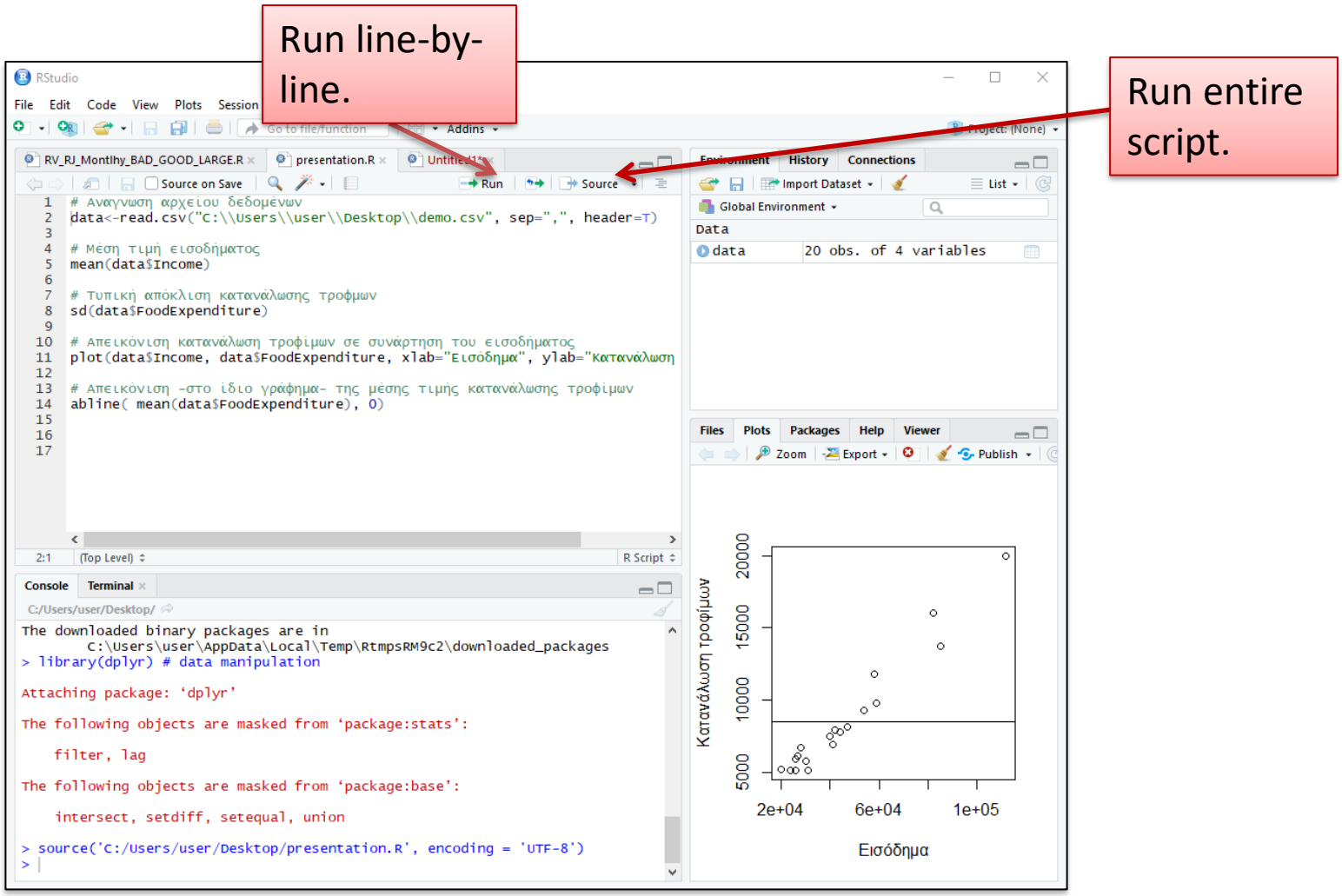
R language

- Data types
  - Like in python, variables defined in R have a data type
  - R supports the following built-in data types:
    - **Data frame**
    - **Numeric**
    - **Character**
    - **Matrix**
    - **Factor (categorical values)**
    - **Arrays**
    - **Vector**
    - **List**
    - **Logical (Boolean/binary)**
    - **Complex**

Example Code

```
# Read data from file
data<-read.csv("demo.csv", sep=",", header=T)

# Compute arithmetic mean of Income
mean(data$Income)

# Compute the standard deviation of Food Expenditure
sd(data$FoodExpenditure)

# Plot showing the relation between Food Expenditure and Income
plot(data$Income, data$FoodExpenditure, xlab="Εισόδημα",
ylab="Κατανάλωση τροφίμων")

# Add line -in the same plot- of mean value of Food Expenditure
abline( mean(data$FoodExpenditure), 0)
```

Run entire script or line by line



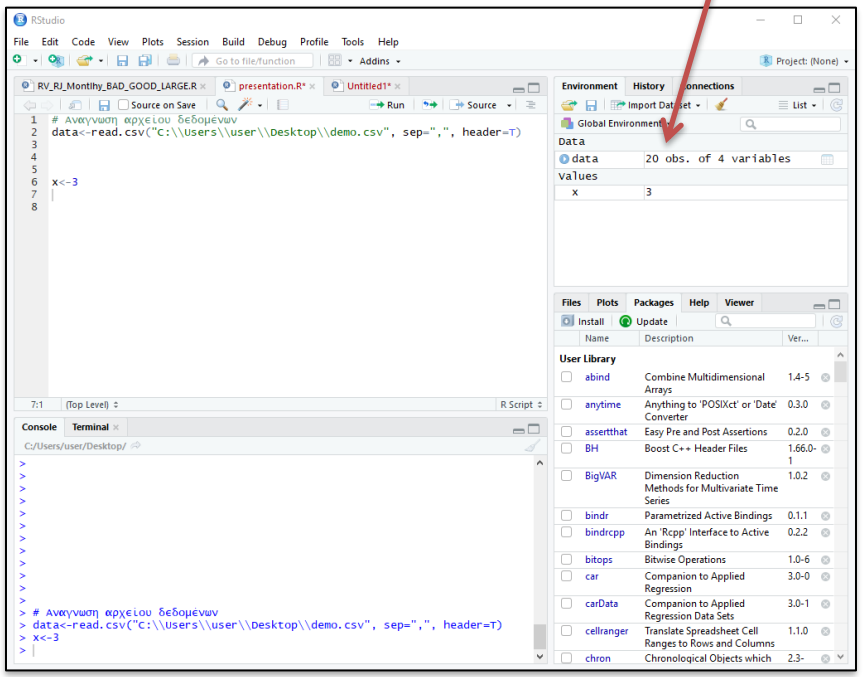
Type commands in the console

- Execute R language commands by typing them line by line.
- Assign values to variables.
- Example → Declare 2 variables ‘x' and ‘y' to have values 3 and 2 respectively.
- >x<-3
- >y<-2
- Type commands directly in the console allows us to see the content of a variable or the structure of a created object.
- >data[1,c(2,3)]
- >head(data, 10)
- >tail(data, 15)

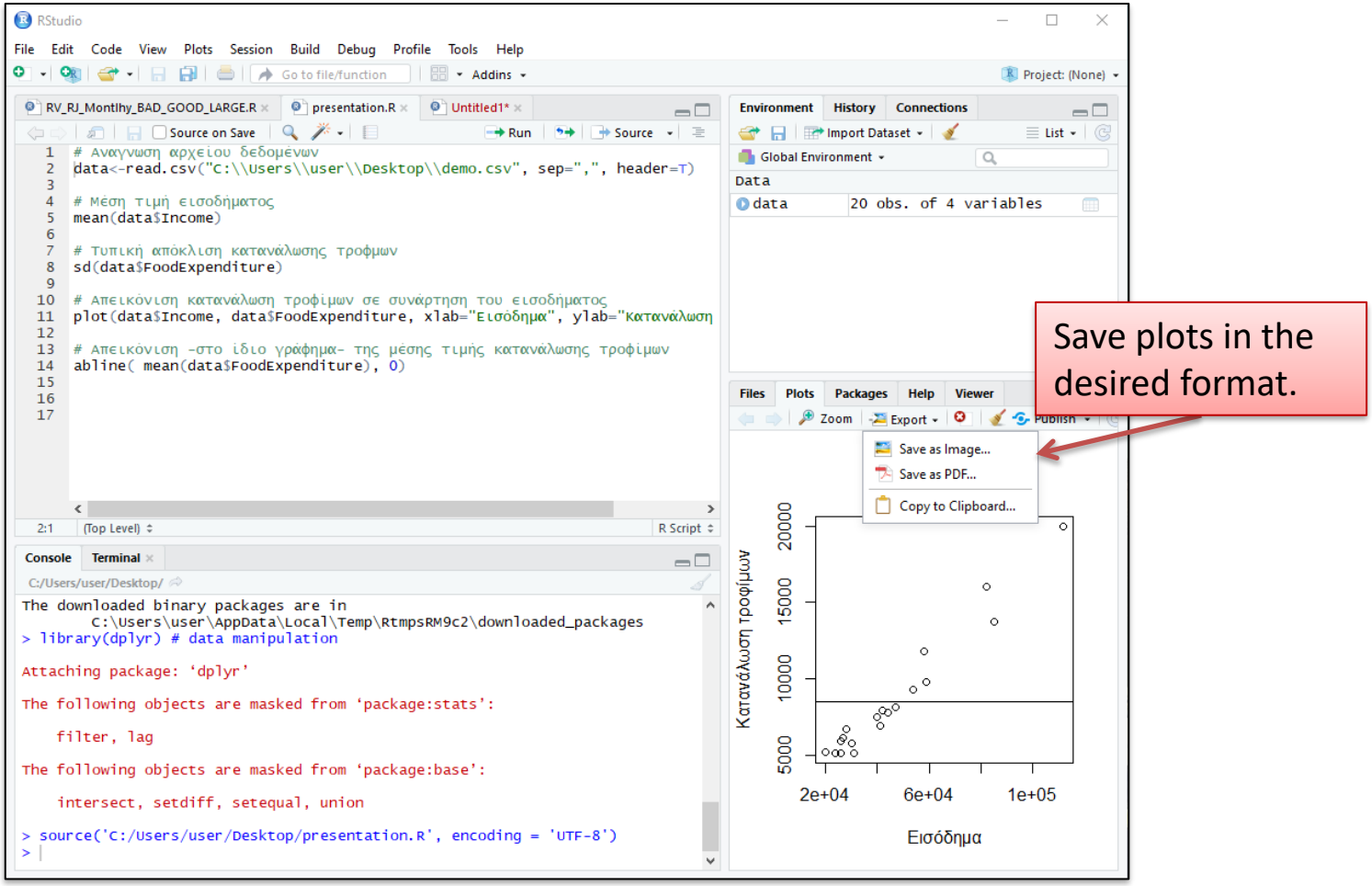
Environment Pane

- List of every function or symbol that is defined in the Console.
- Datasets loaded into the Console.

  e.g. *data<-read.csv("demo.csv",sep=",",header=T)*
- OR directly importing datasets to the Environment. The result is the same as if typing the command into the console.

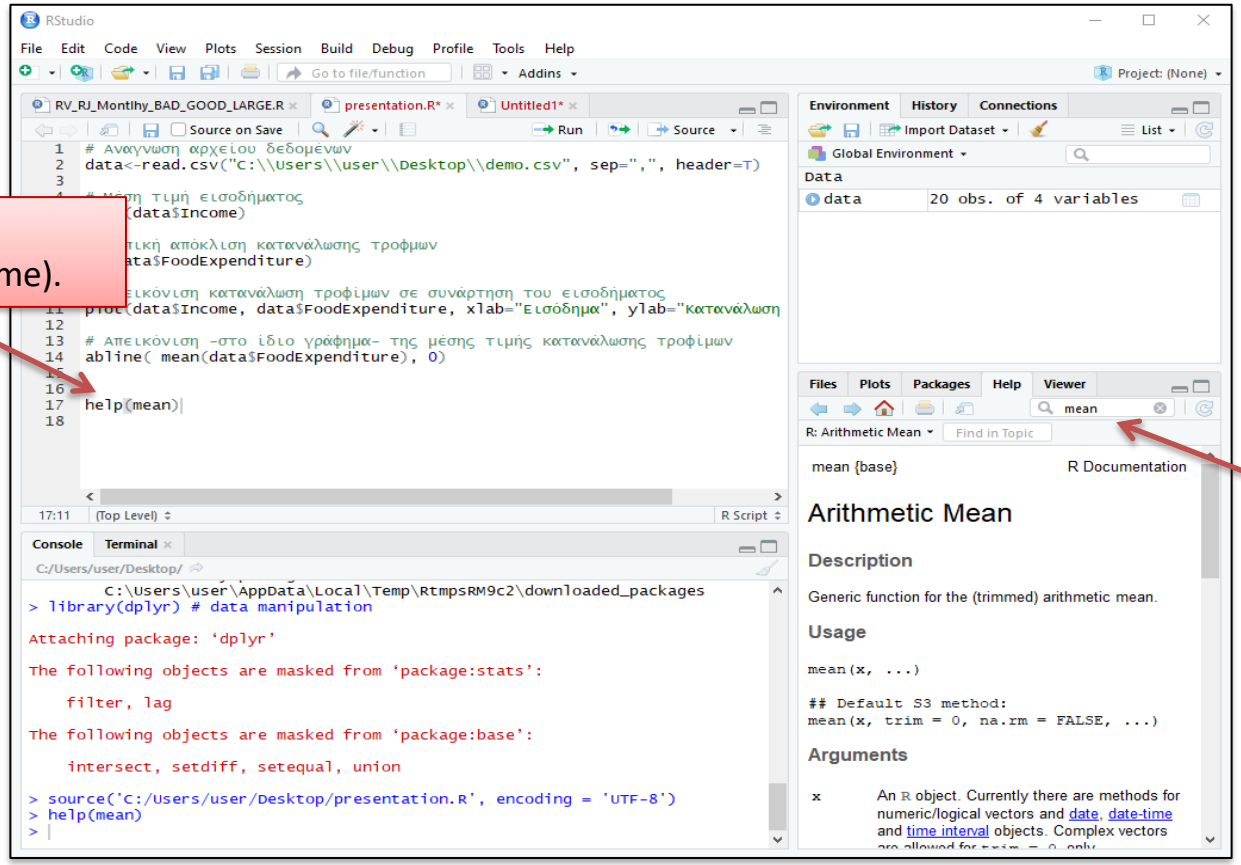


Export the graphics created (Plots Tab)



Help Tab

- Get help on any function of R.



Type help(function-name).

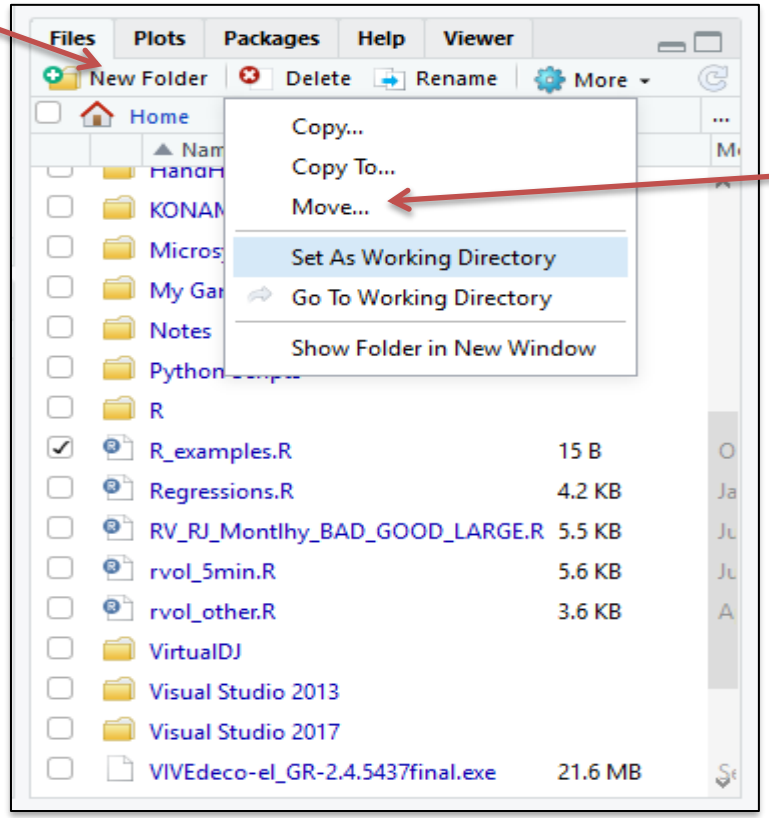
Type function name on Help tab.

Save an R script

- Menu ‘File > Save’
- Choose any directory.
- getwd() (To get the home directory for RStudio).
- Returns a path to the current working directory.
- NOTE: Scripts saved have usually a .R file extension (e.g. myProgram.R) . Such files are simple text files and can be opened with any text editor.

View File in Files Tab

- Create a new directory inside of working directory to save the new file.



Move .R file to new directory.

Quit an R Session

- Menu ‘File > Quit session…’
- Save the workspace or not?
- Save → begin next session with variables and history loaded.
- Don’t save → lose variables and history, files will be preserved.

Useful links

- R for Beginners
  https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
- R
  https://www.r-project.org/
- RStudio
  https://www.rstudio.com/

Working with R

Working with R

- Data frames in R
  - A two dimensional structure/array with rows and columns exactly like a data frame in Python
  - Can hold data of any data type e.g. in a data frame one column may be a number, another a character or logical.
  - Allows slicing and indexing
    - Syntax differs from Python though
    - Also **NUMBERING IN R STARTS FROM 1, NOT 0!**

Working with R

- Data frames in R
  - What does a data frame look like?

Data frame in R

ID	Age	Name
1	23	Maria
2	35	Jim

Working with R

- Data frames in R
  - Slicing/indexing is done using the [] operator specifying row and column in various ways.
  - **Keep in mind: Indexing in R starts at 1 (not 0)!**

```
myData[2, 5]
```

Show the value of the second row and firth column of the data frame myData.

Working with R

- Example: create a variable that is a Data frame that is empty with three columns named “ID”, “Age” and “Name” by specifying the data type of each column:

```
myData <- data.frame(ID=integer(), Age=integer(), Name=character(),
stringsAsFactors=FALSE )
```

Working with R

- Adding a single new row to an empty data frame

```
myData[1,] <- c( 1, 23, "Alice")
```

Add the new row to the first row (notice the 1) of the data frame myData.

Creates a new vector – notice the c() – which will be a row where the first value of the vector is the value for column ID, the second value for column Age and third value for column Name.

Working with R

- Structure of a data frame
  - What columns does it have and what data types these columns are
  - Using function **str()**

```
str(myData)
```

```
'data.frame': 1 obs. of 3 variables:
$ ID : chr "14"
$ Age : chr "33"
$ Name: chr "Alice"
```

- Similar functions: **names(), attributes()**

Working with R

- Indexing and Slicing

```
>myData$Age # Only Column Age from Data frame myData
>myData[,"Age"] # Equivalent to previous expression
>myData[ , 1:2] # Columns 1 (ID) and 2 (Age) from all rows- operator : expressing from to

>myData[1, c("Age", "Name") ] # From row 1, get only Columns Age and Name
```

Working with R

- Reading csv files
  - Using function read.csv() with the proper arguments.
  - Reads a csv file and returns its **content as a Data Frame.**

Working with R

- Reading csv files