Programming for Data Science

Introduction to R language

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

September 2021

The R Project

- Environment for statistical computing and graphics:
- Free software and Open-source;
- A simple programming language:
  - it is an open-source implementation of S language;
  - it is among the Top 10 Programming Languages in 2019 for *IEEE Spectrum Journal*.
- software and packages can be downloaded from:

  www.cran.r-project.org

- Versions of R exist of Windows, MacOS, Linux and various other Unix-like OS.

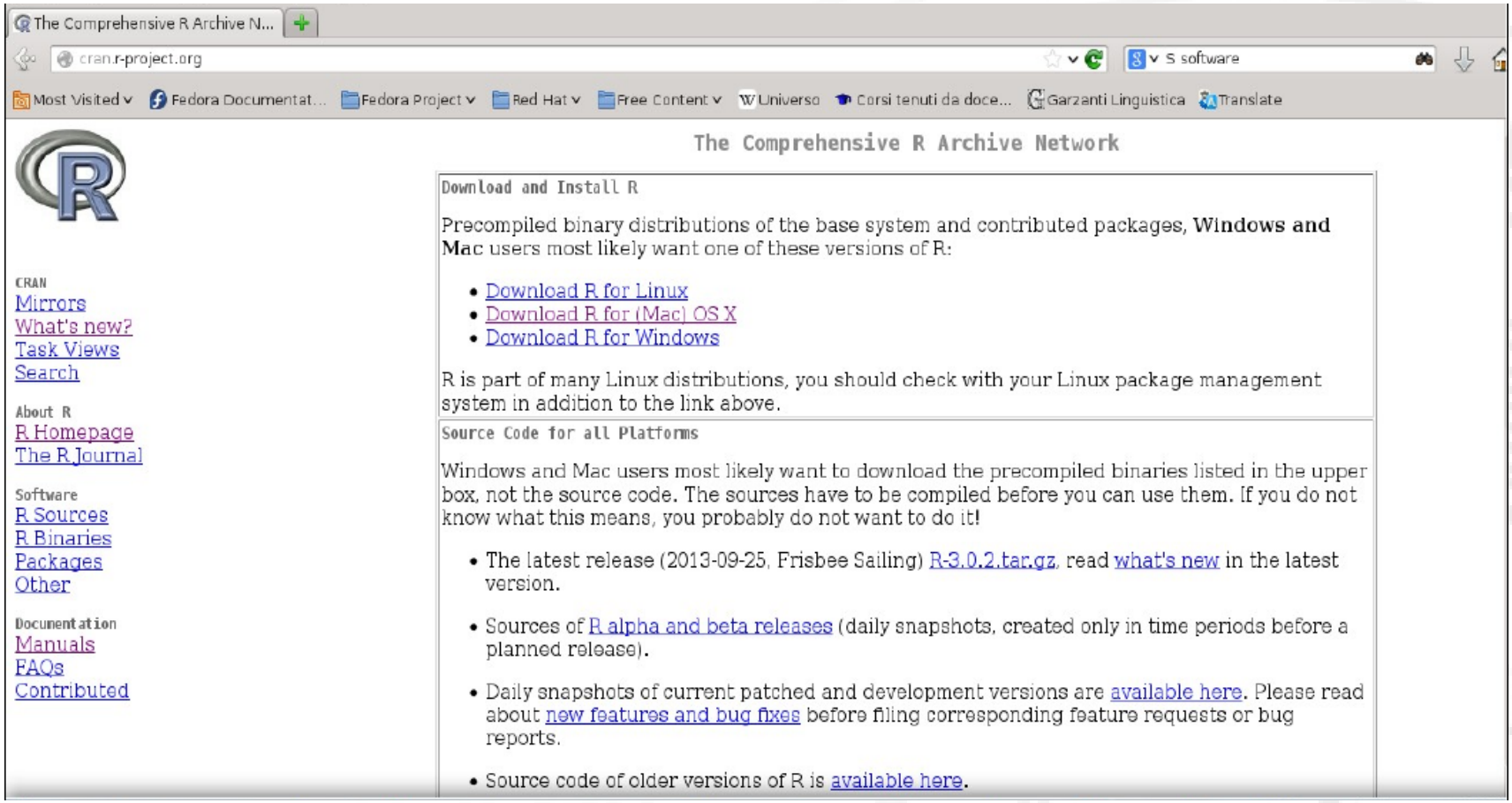
Why to use R language

- Implement many common statistical procedures;
- Provide excellent graphics functionality;
- A convenient starting point for many data analysis projects
- Libraries (namely packages) can be automatically downloaded from:

  www.cran.r-project.org

- It is standard for data mining and statistical analysis;
- Efficient data structures make programming easier.

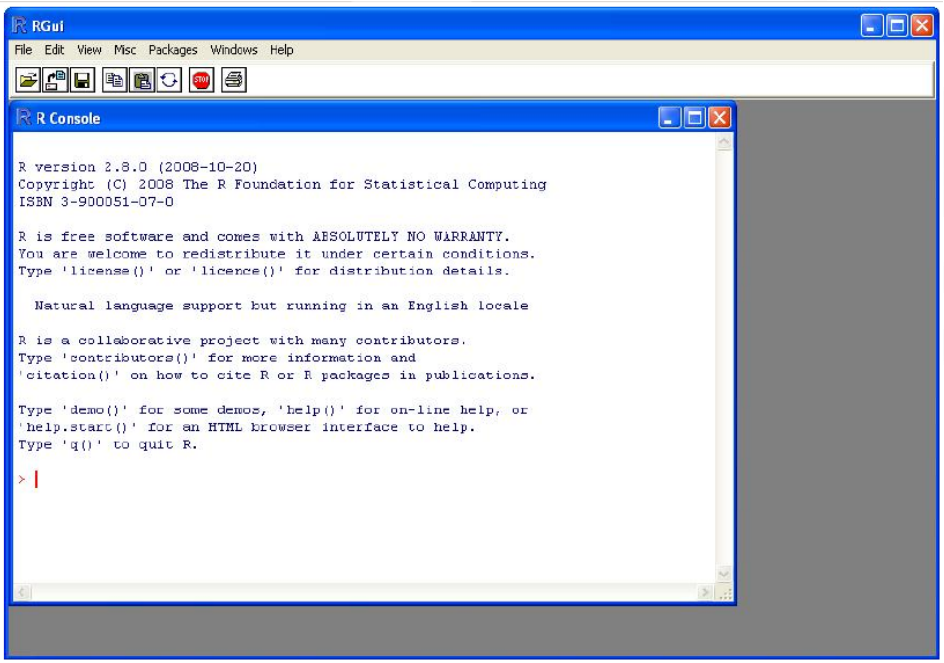
Download and Install R language



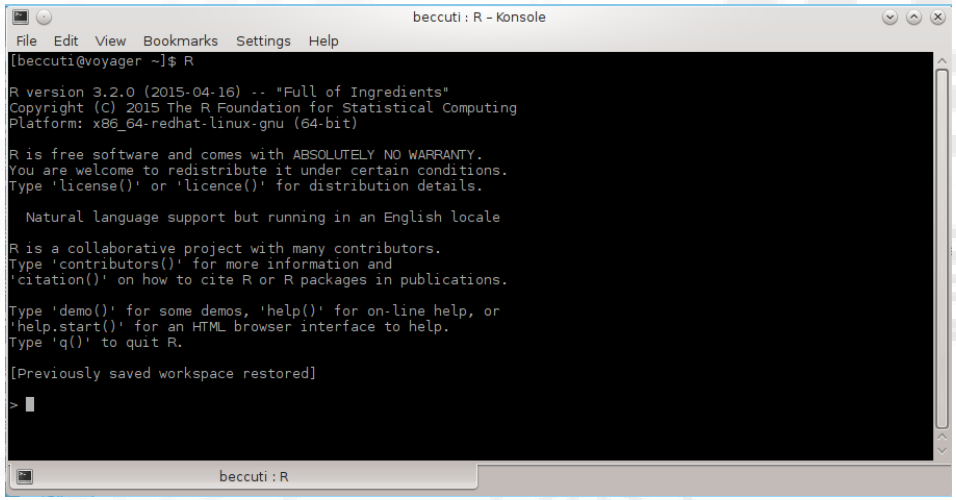
http://cran.mirror.garr.it/mirrors/CRAN/

Download the appropriate version (w.r.t. your OS) and follow the instructions to install the programme.

R under GUI



from Windows

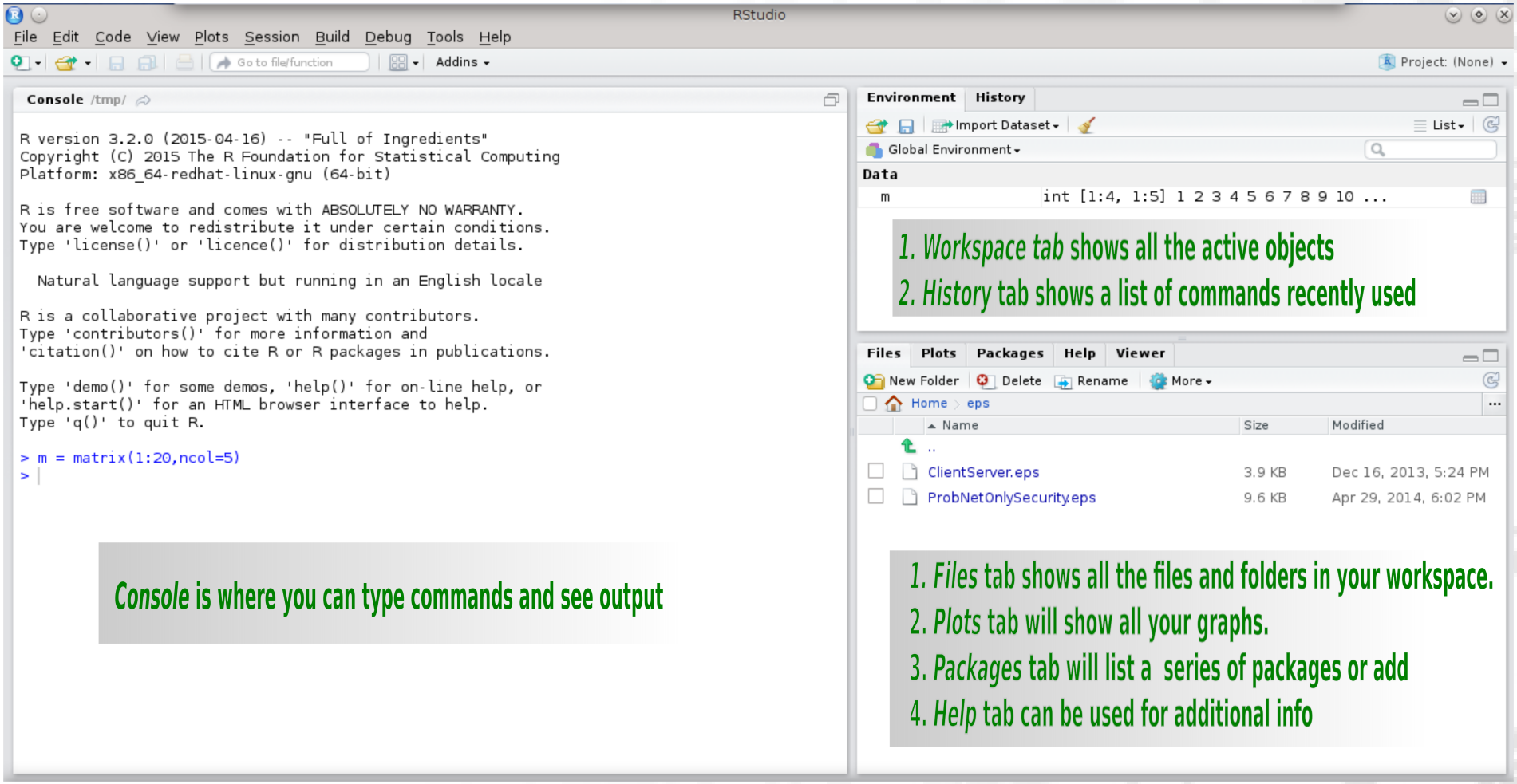


from Linux

R under GUI using Rstudio

RStudio allows the user to run R in a more user friendly environment.

It is open-source and available at `http://www.rstudio.com/`



Starting R

R can be started:

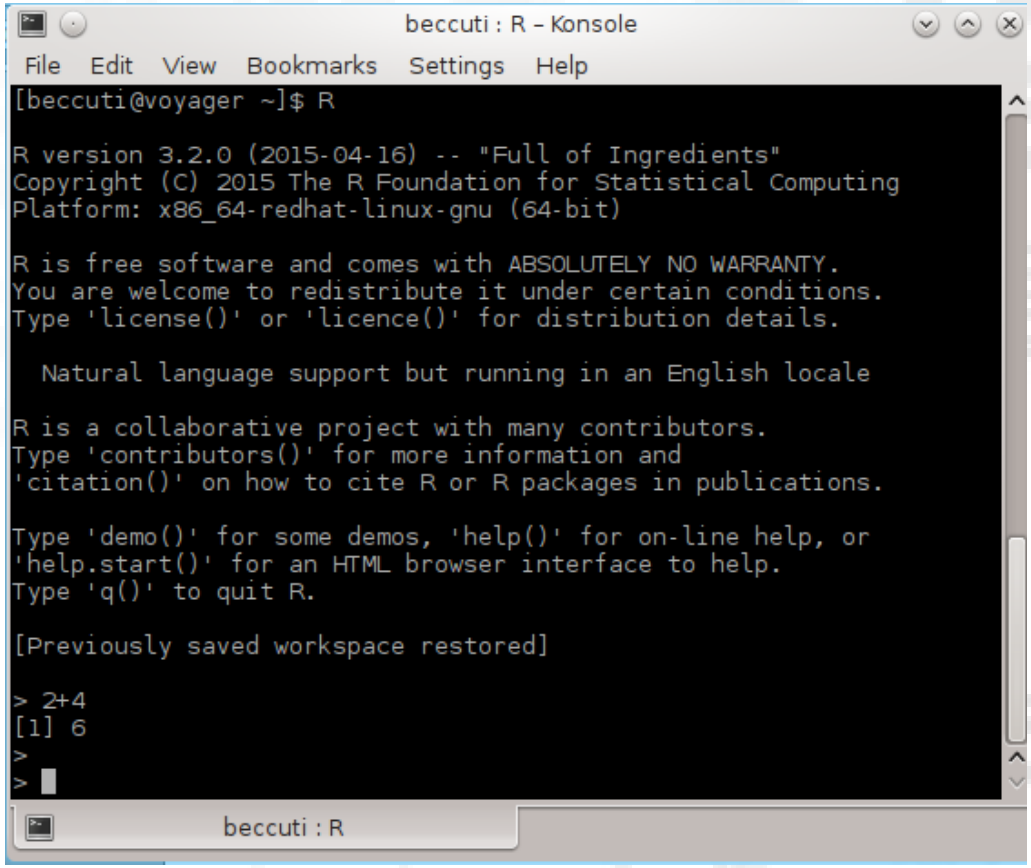
- by double-clicking on the R icon (e.g. Window);
- by double-clicking on the Rstudio icon (e.g. Window + Rstudio);
- by typing R in a shell (e.g. Linux).
- by typing rstudio in a shell (e.g. Linux + Rstudio).

How R works:

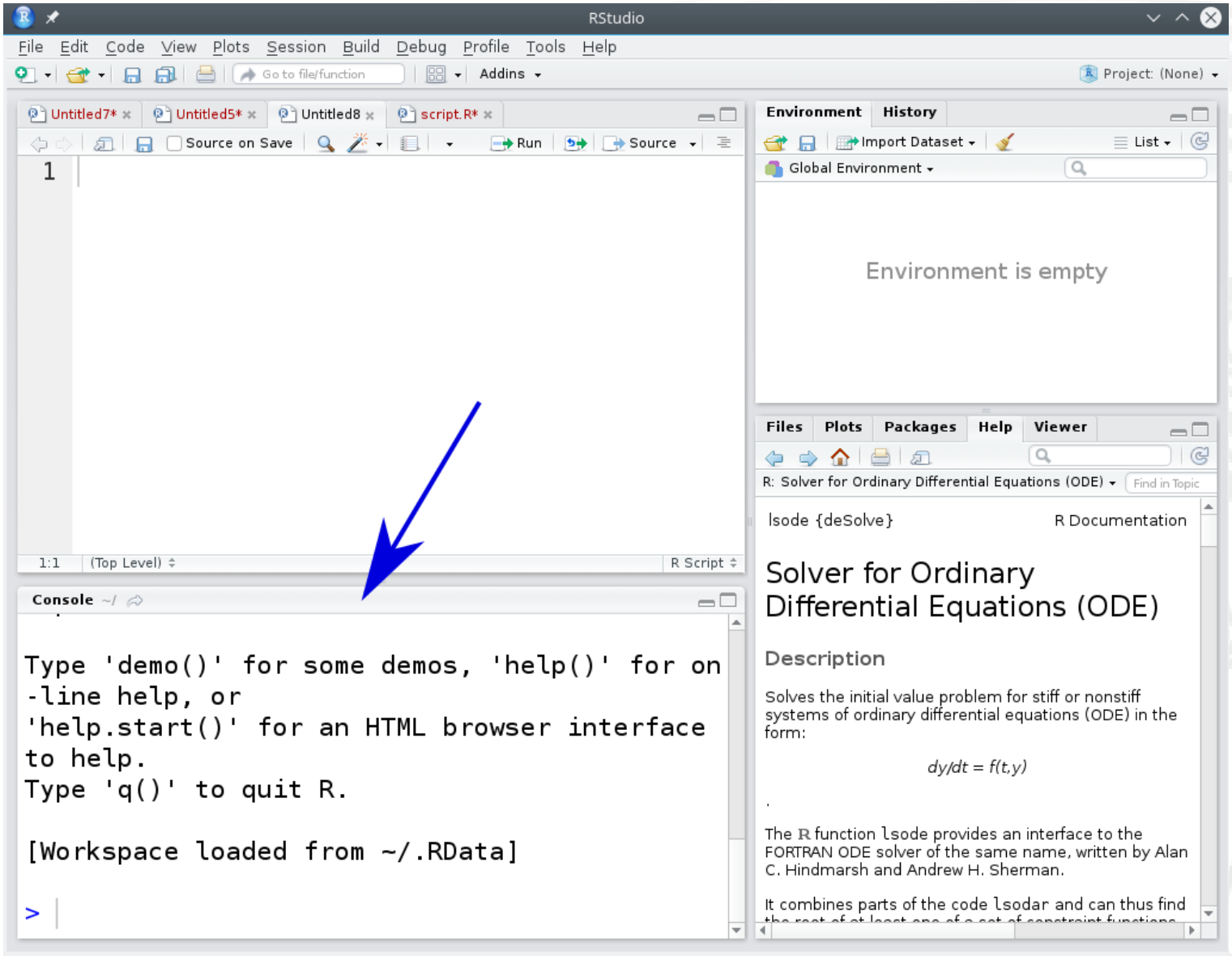
- R creates its objects in memory and saves them in a file called .RData (by default);
- Commands are recorded in an .Rhistory file, Command can be recalled using up- and down-arrow;
- Recalled commands may be edited;
- Commands may be abandoned by pressing <Esc>;
- To end your session type q() or just kill the window.
- A concept of *working directory* is introduced: each project is associated with a working folder containing each data.

Interactive R

- R defaults to an interactive mode;
- A prompt ">" is presented to users;
- Each input command is evaluated and a result returned;
- Commands
  - consist of expressions or assignments;
  - are separated by a semi-colon (;) or by a newline
  - can be grouped together using curly brackets({ and })

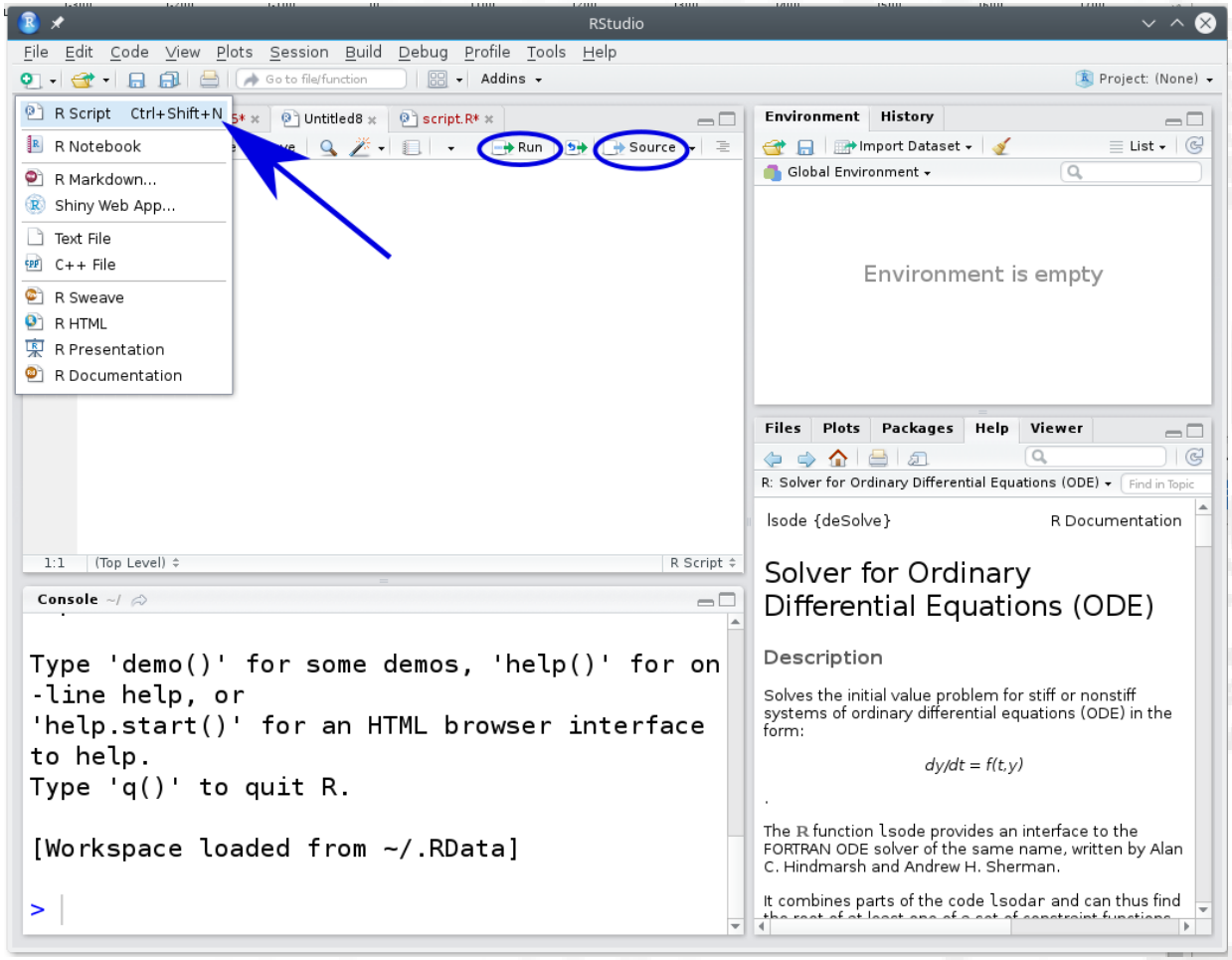


RStudio prompt and script



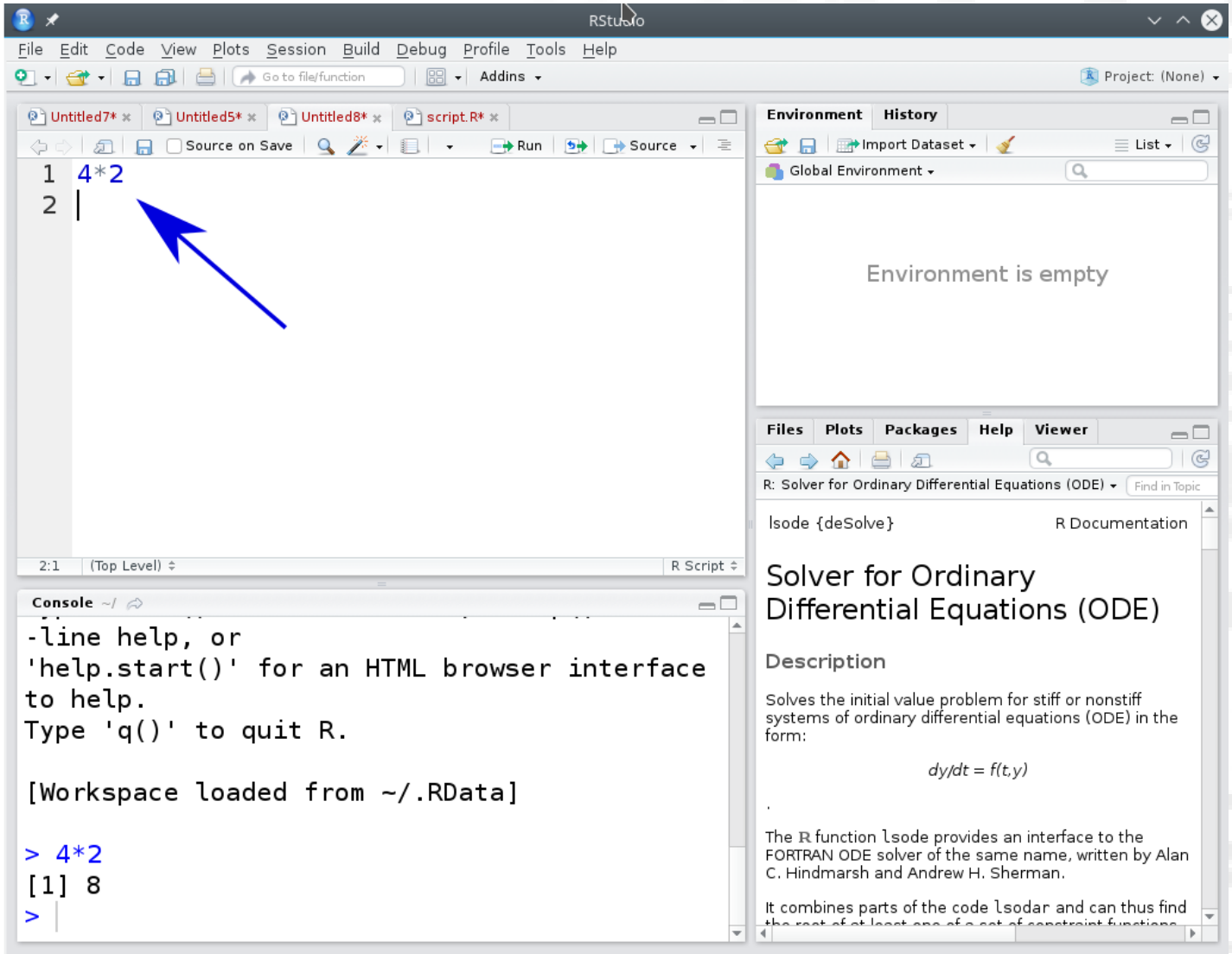
RStudio prompt and script

- R script can be used to save R commands into a file;
- Commands into R script can be executed line by line (clicking on Run) or globally (clicking on Source).



RStudio prompt and script

- Commands can be directly typed into the R script console.



R as a calculator

Simple Arithmetic

```
> 3 + 4
[1]7
```

Operator precedence

```
> 2 + 3 * 5
[1]17
```

Exponentiation

```
> 3^5
[1]243
```

Basic mathematical functions

```
> exp(4)
[1]54.59815
> sqrt(4)
[1]2
```

Predefined constant

```
> pi
[1]3.141593
> Inf
[1]Inf
```

Assignments in R

It is often required to store intermediate results so that they do not need to be re-typed over and over again. To assign a value of 324 to the variable X type:

$> X < -324$

or

$> X = 324$

Variable X can be used in next expressions:

Example

$> X$
[1]324

$> X + X$
[1]648

$> sqrt(X)$
[1]18

$> X = X + X; X$
[1]648

$> X/4$
[1]162

$> X \hat{} sqrt(X)$
[1]$1.54814e+45$

Variable name in R

R is a case-sensitive language, hence x and X do not refer to the same variable.

Variable name:

- can be created using letters, digits and the . (dot) symbol;

  > *data*1.*address*
  > *d*1_4.*f*

- must not start with **a digit** or a . followed by a digit.

- some names are reserved by the system: *if, while, NULL, TRUE ...*

Variable type in R

Basic variable types are:

Numeric: integer, floating point values;

Boolean: values corresponding to **True** or **False**;

Strings: sequences of characters.

Type is determined automatically when variable is created with $<-$ or $=$ operator.

Data structures/Objects are:

R provides types of different object.

Vector: a collection of elements (numbers, logical values and character strings) with same type;

Array: a generalization of a vector;

List: collections of objects of any type;
e.g. list of vectors, list of matrices, etc.

Data Frame: an array in which the type of each element can be different;

Factor: takes on a limited number of values;

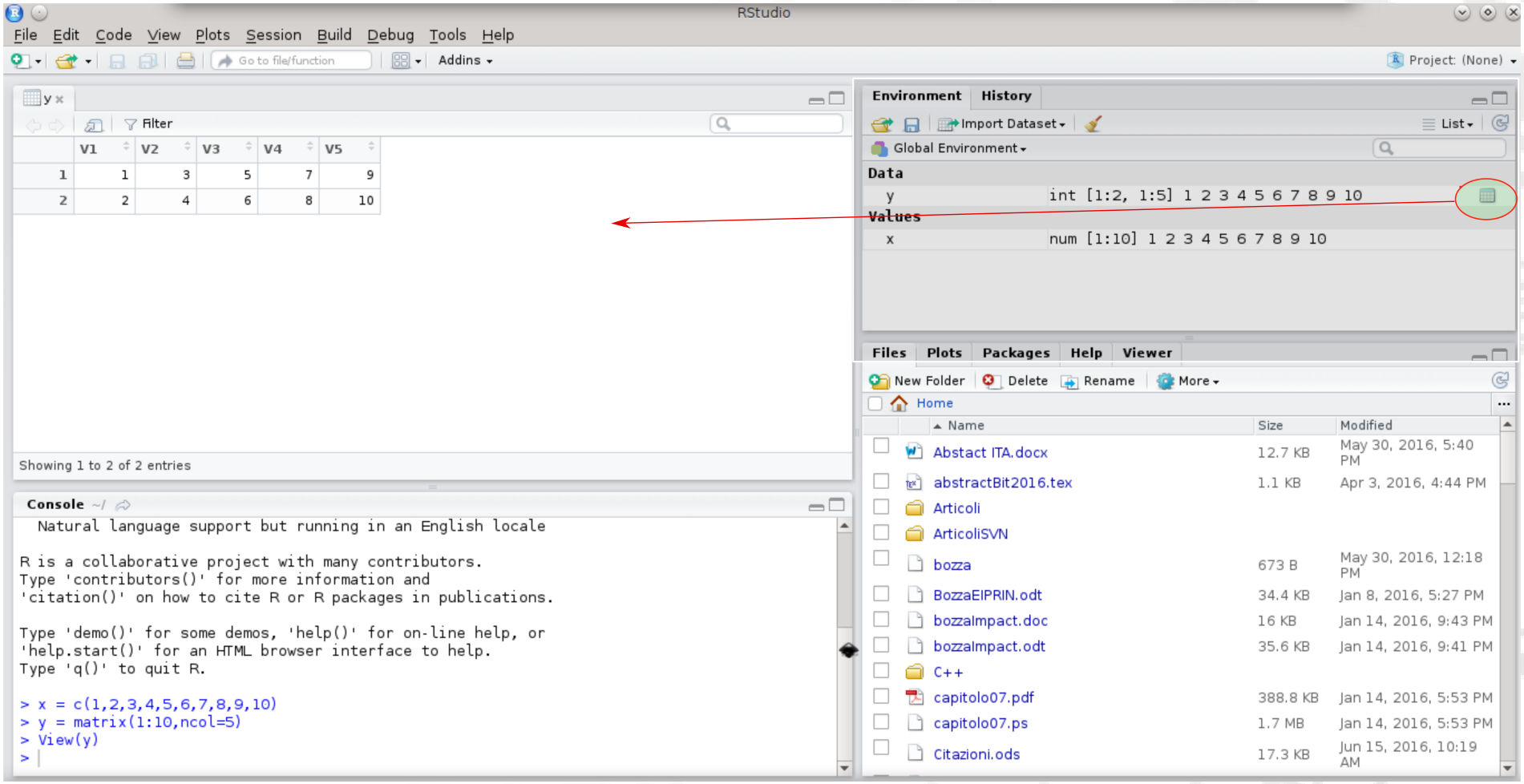
Variable in R

- During an R session, objects are created and stored by name;
- The command *ls()* displays all currently-stored objects (workspace);
- Objects can be removed using *rm(variable_name)*;
- All the objects in the workspace are removed using *rm(list=ls())*.

Observe

At the end of each R session, you are prompted to save your workspace. If you click Yes, all objects are written to the .RData file. When R is re-started, it reloads the workspace from this file and the command history stored in .Rhistory is also reloaded.

Variable in RStudio



Getting help in R

R provides a built-in help facility.

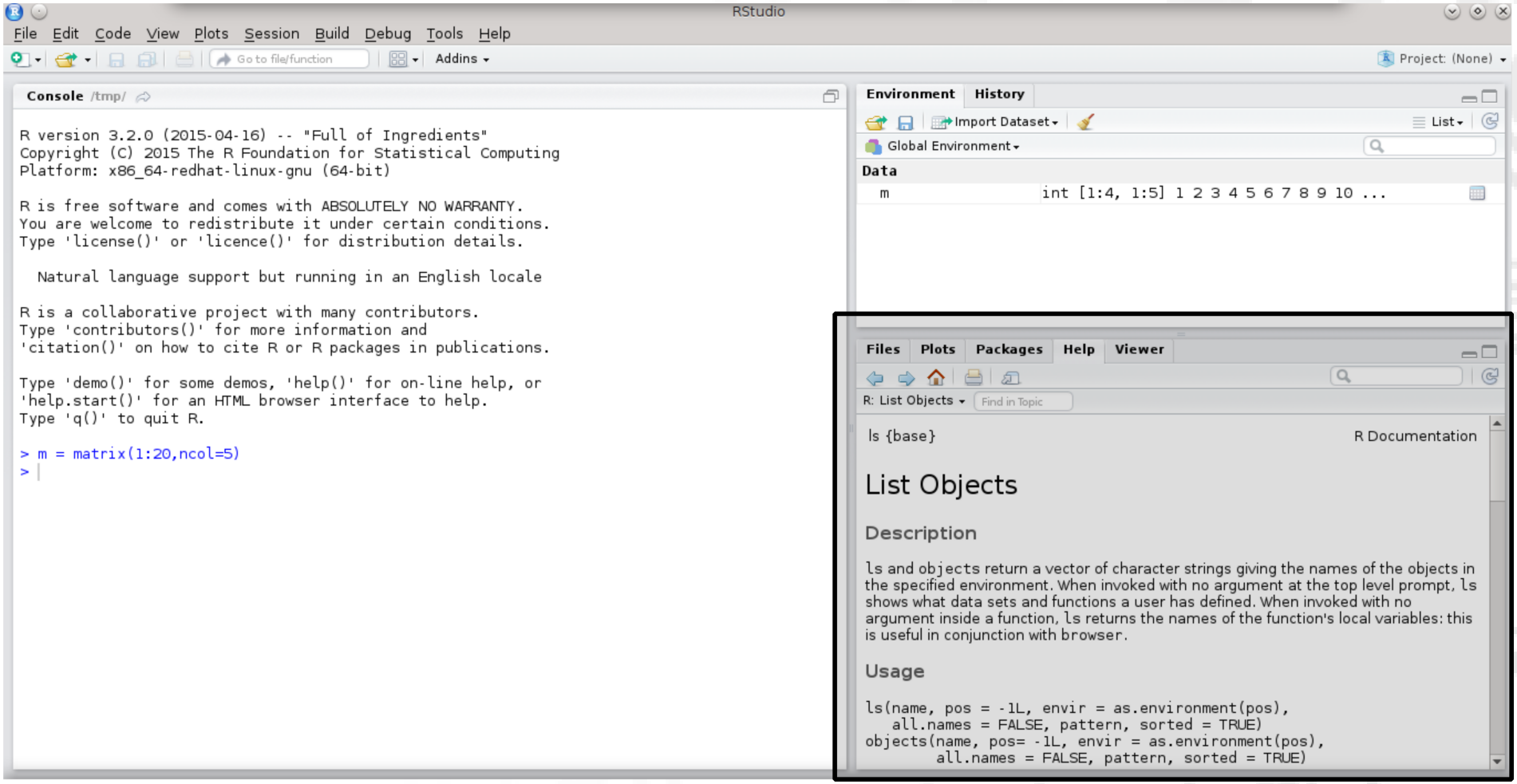
- To get more information on any specific function, e.g. sqrt(), the command is:
  help(sqrt)
  or
  ?sqrt

- help on features specified by special characters must enclose in single or double quotes (e.g. "[[") *help("[[")*

- Help is also available in HTML format by running *help.start()*

- For more information use
  ?help

Getting help in Rstudio

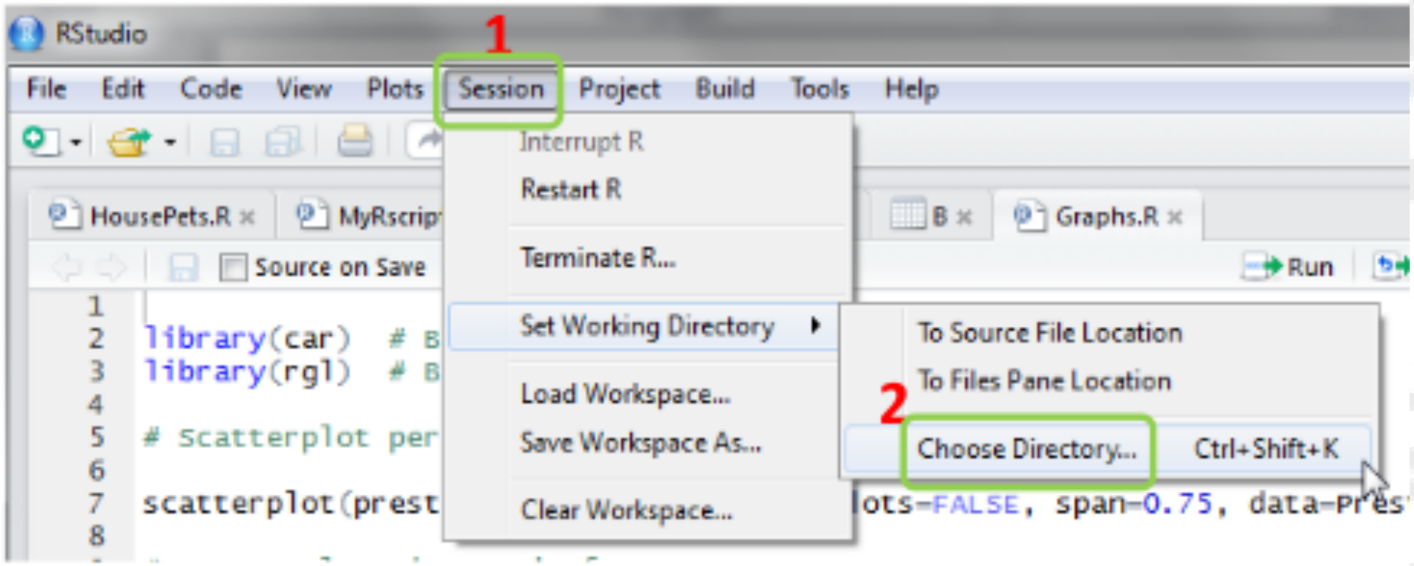


Working directory

Working directory in R:

- Working directory contains data and R scripts. It is a directory of the file-system;
- *getwd()* returns the current Working directory;
- *setwd("new_path")* sets Working directory;

Working directory in RStudio:



Packages in R

- R provides libraries of packages. Packages contain various functions and data sets for numerous purposes;
- Some packages are part of the basic installation. Others can be downloaded from CRAN:
  $>$ *install.packages*("*ggplot2*")
- To use functions and data sets of a package, it must be loaded into the workspace:
  $>$ *library*(*ggplot2*)
- To check what packages are currently loaded into the workspace:
  $>$ *search*()
- A loaded package can be removed:
  $>$ *detach*("*package*:*ggplot2*")

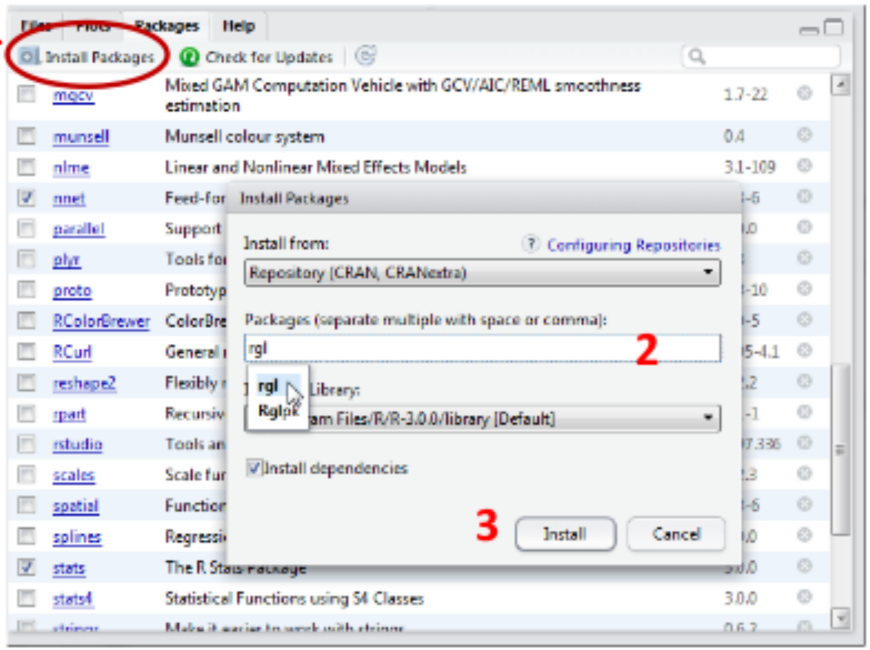
Observe:

if you terminated your session and start a new session with the saved workspace, you must load the packages again.

Packages in Rstudio

RCurl	General network (HTTP/FTP/...) client interface for R	1.95-4.1	
reshape2	Flexibly reshape data: a reboot of the reshape package.	1.2.2	
rpart	Recursive Partitioning	4.1-1	

Before



We focus on Package tab(bottom-right)

After

RCurl	General network (HTTP/FTP/...) client interface for R	1.95-4.1	
reshape2	Flexibly reshape data: a reboot of the reshape package.	1.2.2	
rgl	3D visualization device system (OpenGL)	0.93.952	
rpart	Recursive Partitioning	4.1-1	

DSS/OTR