

Introduction to R Software

Introduction

:::

Command line, Data Editor and R Studio

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Command Line versus Scripts

What is command line?

R Console

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"  
Copyright (C) 2015 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
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.
```

```
[Previously saved workspace restored]
```

```
> |
```

Type the commands here

This is command line

Command Line versus Scripts

**Execution of commands in R is not menu driven.
(Not like Clicking over buttons to get outcome)**

We need to type the commands.

Single line and multi line commands are possible to write.

When writing multi-line programs, it is useful to use a text editor rather than execute everything directly at the command line.

Command Line versus Scripts

Option 1:

One may use R's own built-in editor.

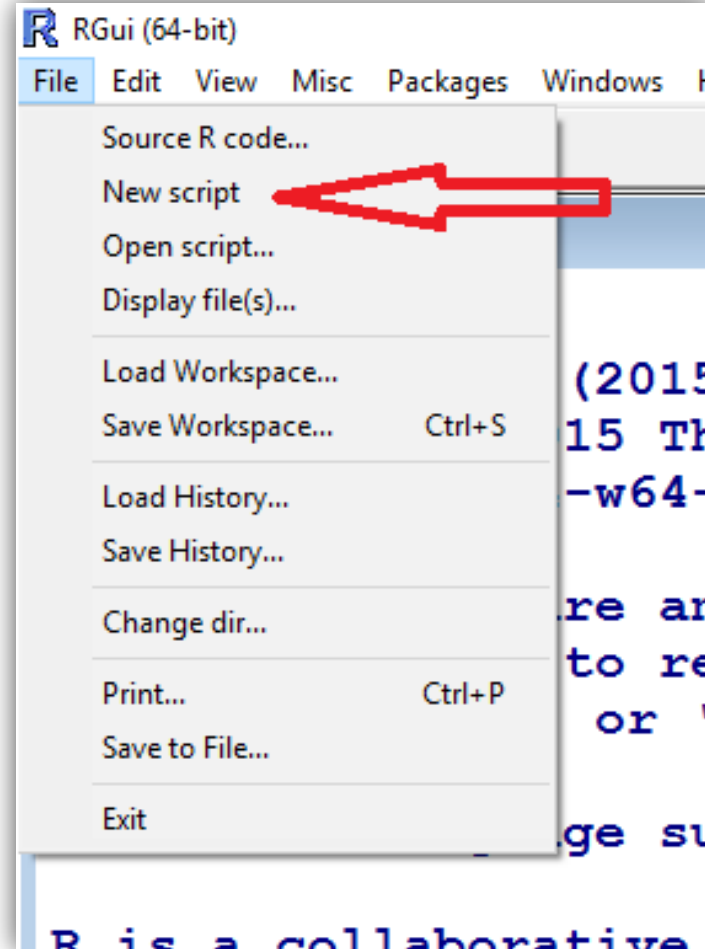
It is accessible from the **RGui** menu bar.

Click **File** and then click on **New script**.

Command Line versus Scripts

At this point R will
open a window entitled
Untitled-R Editor.

We may type and edit in this.

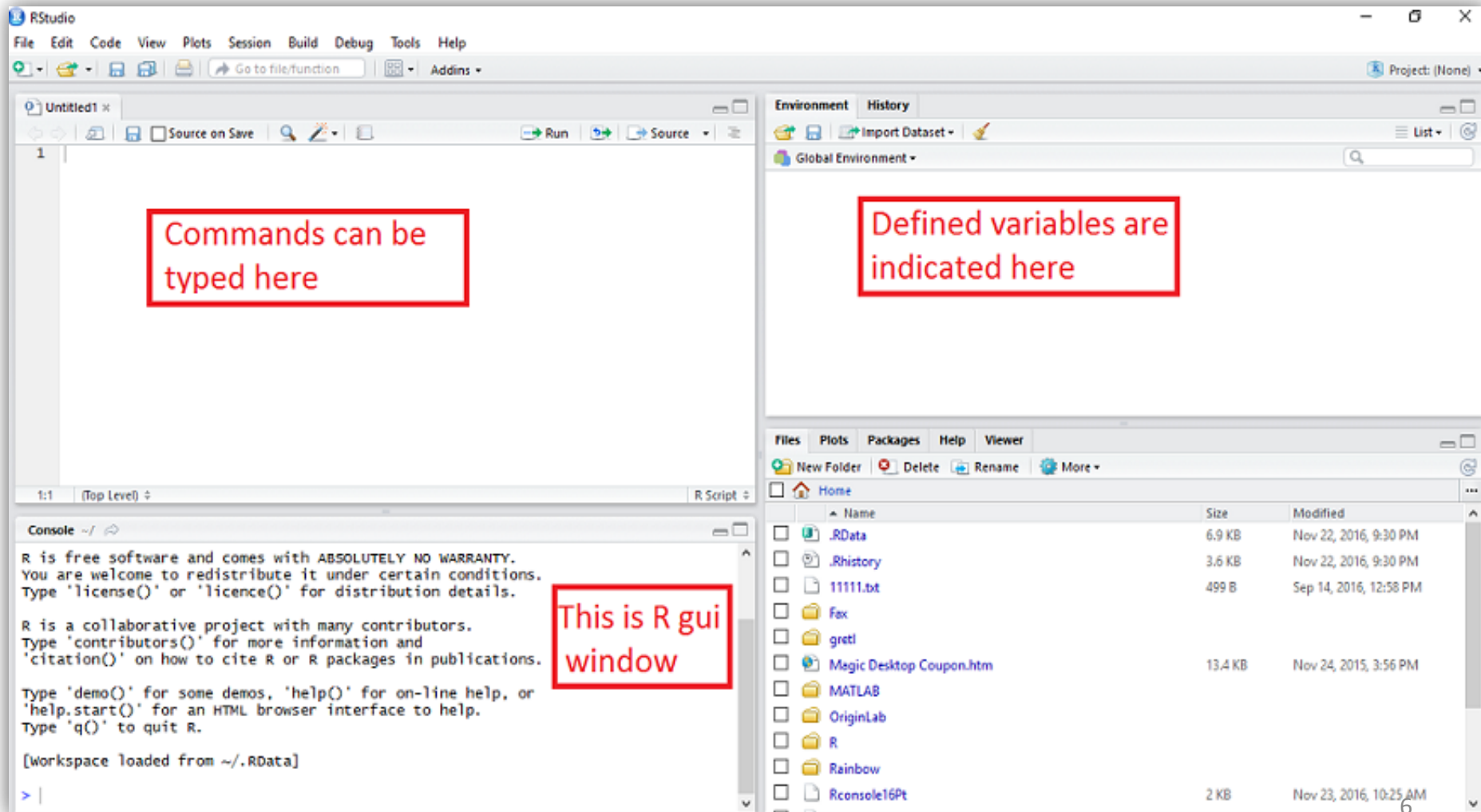


If we want to execute a line or a group of lines, just highlight them and press **Ctrl+R**.

Command Line versus Scripts

Option 2:

Use R studio software.



Command Line versus Scripts

Suppose we want to use following three functions:

Type them.

```
library(MASS)
```

```
attach(bacteria)
```

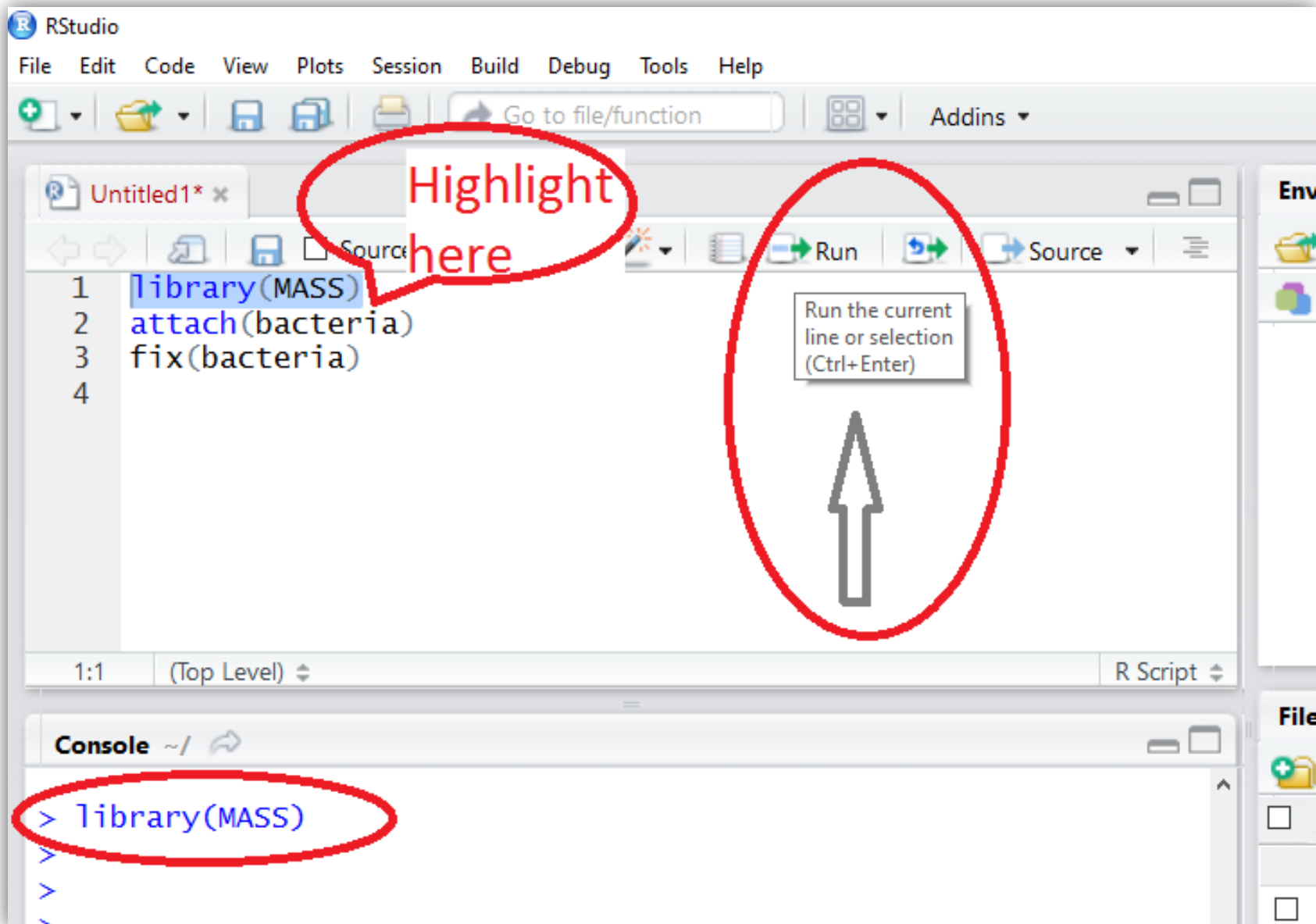
```
fix(bacteria)
```

Suppose we want to run only function: `library(MASS)`

Highlight it and click on **Run**

Then we get....

Command Line versus Scripts



Data Editor

There is a data editor within R that can be accessed from the menu bar by selecting Edit/Data editor.

Provide the name of the matrix or data frame that we want to edit and a **Data Editor** window appears.

Alternatively we can do this from the command line using the **fix** function.

Example:

```
library(MASS)  
attach(bacteria)  
fix(bacteria)
```

Data Editor

We can do it in R Studio as follows :

The screenshot shows the RStudio interface. The source code window contains the following R code:

```
1 library(MASS)
2 attach(bacteria)
3 fix(bacteria)
```

A red circle highlights the `library(MASS)` line with the text "Highlight here".

The Data Editor window is open, displaying a table with 12 rows and 7 columns. The columns are labeled: y, ap, hilo, week, ID, and trt. The data is as follows:

	y	ap	hilo	week	ID	trt
1	y	p	hi	0	X01	placebo
2	y	p	hi	2	X01	placebo
3	y	p	hi	4	X01	placebo
4	y	p	hi	11	X01	placebo
5	y	a	hi	0	X02	drug+
6	y	a	hi	2	X02	drug+
7	n	a	hi	6	X02	drug+
8	y	a	hi	11	X02	drug+
9	y	a	lo	0	X03	drug
10	y	a	lo	2	X03	drug
11	y	a	lo	4	X03	drug
12	y	a	lo	6	X03	drug

A green arrow points to the Data Editor window with the text "We get this window".

A grey arrow points to the Data Editor window with the text "This is the data in MASS".

A green arrow points to the "Source" button in the toolbar with the text "Click here".

Cleaning up the Windows

We assign names to variables when analyzing any data.

It is good practice to remove the variable names given to any data frame at the end each session in R.

This way, variables with same names but different properties will not get in each others way in subsequent work.

`rm()` command removes variable names

For example,

`rm(x,y,z)` removes the variables x, y and z.

Cleaning up the Windows

`detach()` command detaches objects from the Search Path

It removes it from the `search()` path of available R objects.

Usually this is either a `data.frame` which has been attached or a package which was attached by `library`.

To get rid of everything, including data frames, type
`rm(list=ls())`

Then we get....

Cleaning up the Windows

R Console

```
> library(splines)
```

Loads the package 'splines'

```
>
```

```
>
```

```
> detach(package:splines)
```

```
>
```

```
>
```

Detaches the
package 'splines'

Introduction to R Studio

It is an interface between R and us.

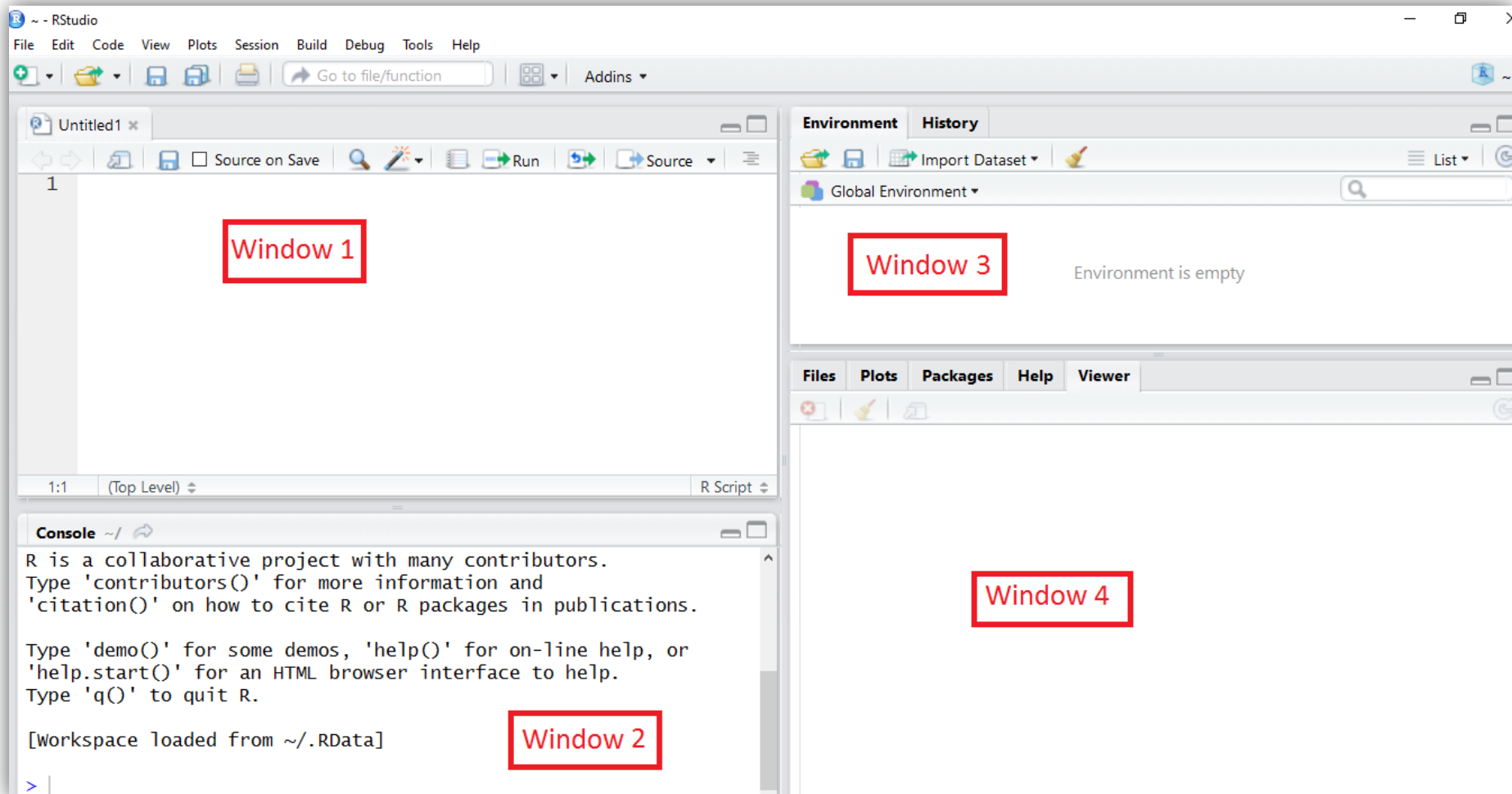
More useful for beginners.

It makes coding easier.

When we start R studio, we see 4 windows

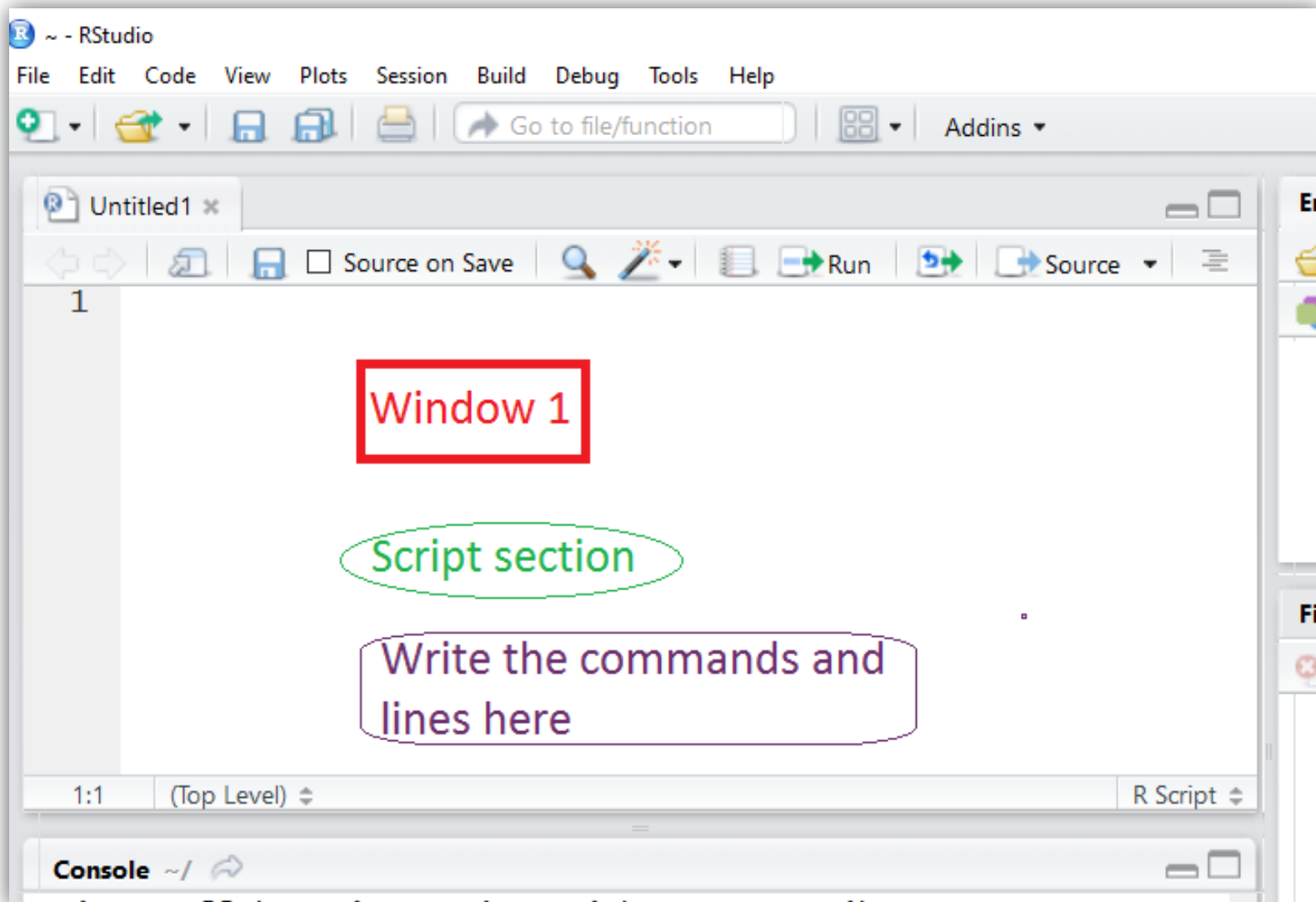
Introduction to R Studio

First opening window of Rstudio is as follows having four windows.



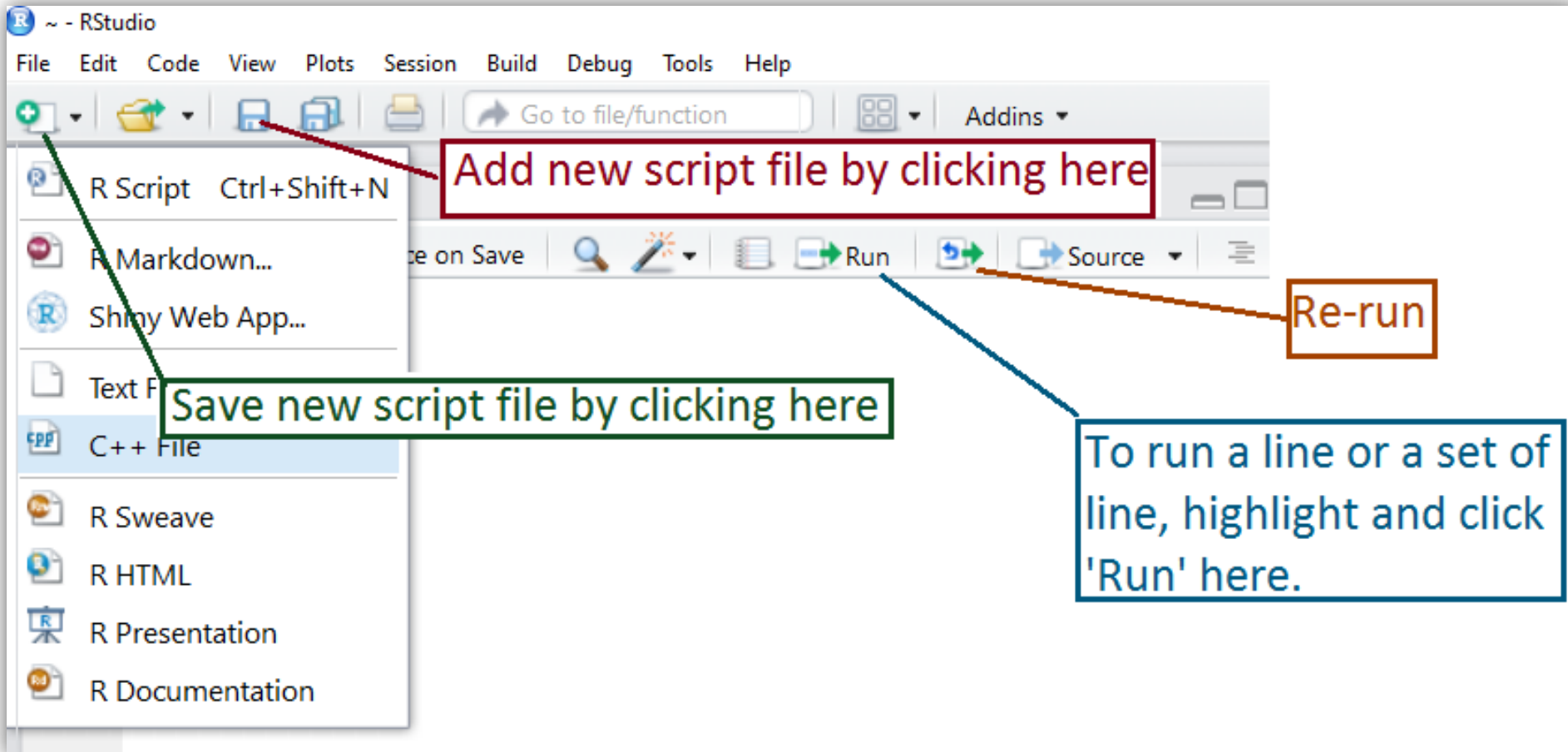
Introduction to R Studio

Description of Window 1



Introduction to R Studio

Description of Window 1



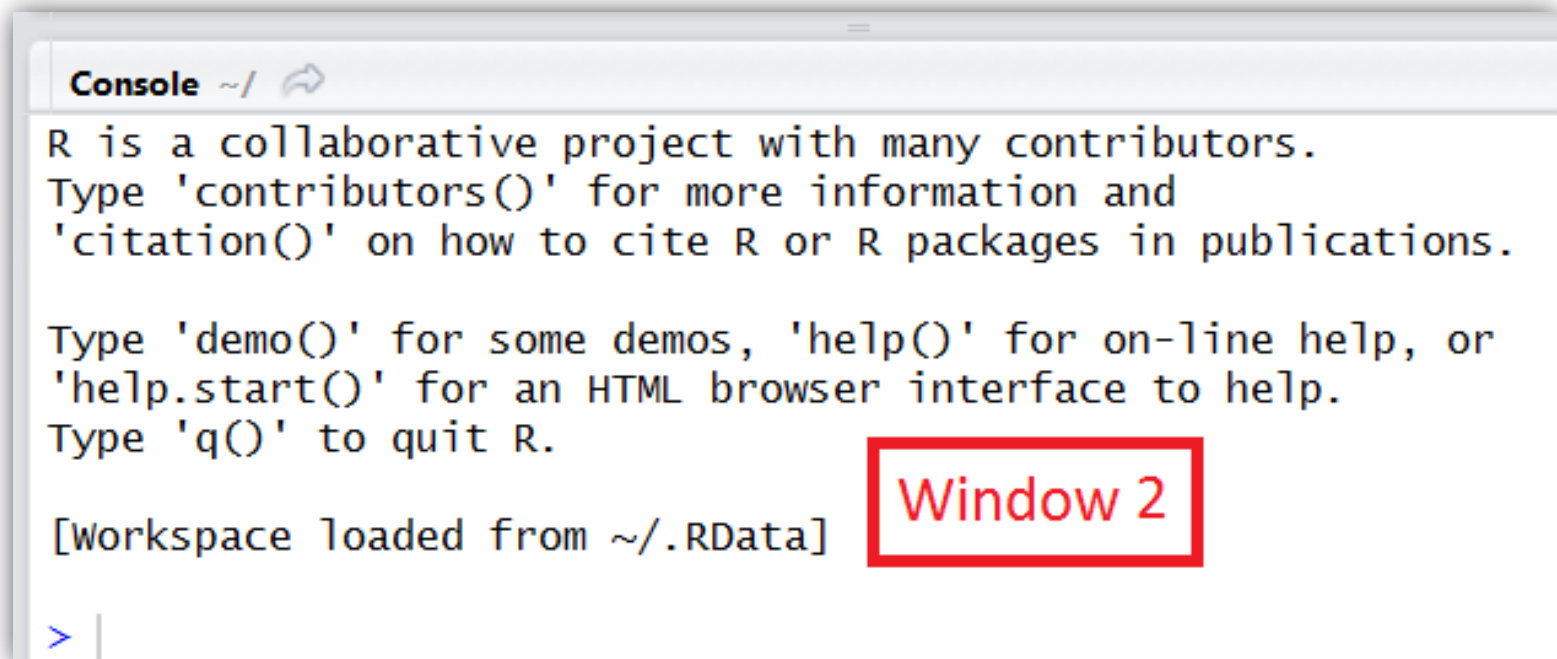
Introduction to R Studio


Description of Window 2 : Console

R program window appears here.

Calculations take place in console window.

One can write programmes in console also but it is hard to make corrections and experiments with the coding.



```
Console ~/   
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.  
  
[Workspace loaded from ~/.RData]  
> |
```

Window 2

Introduction to R Studio

Description of Window 3 : Environment window

All the variables and objects used in the programme appear here.
The nature and values of variables and objects also appear here.

The screenshot shows the R Studio Environment window. At the top, there are two tabs: 'Environment' and 'History'. Below the tabs is a toolbar with icons for file operations and a button labeled 'Import Dataset'. A dropdown menu shows 'Global Environment'. Below this is a table titled 'values' with two columns: the variable name and its value. The first row shows 'x' with the value '1'. Annotations include: an orange box pointing to the 'History' tab stating 'History tells about the codes used earlier.'; a blue box pointing to the 'Import Dataset' button stating 'Data can be imported from other files by clicking here'; a blue box pointing to the 'x' column header stating 'Stored value can be erased from here'; and a green box pointing to the 'x' and '1' row stating 'The stored value x = 1 appears here'. A red box in the bottom left corner is labeled 'Window 3'.

Environment History

Import Dataset

Global Environment

values

x	1
---	---

History tells about the codes used earlier.

Stored value can be erased from here

The stored value x = 1 appears here

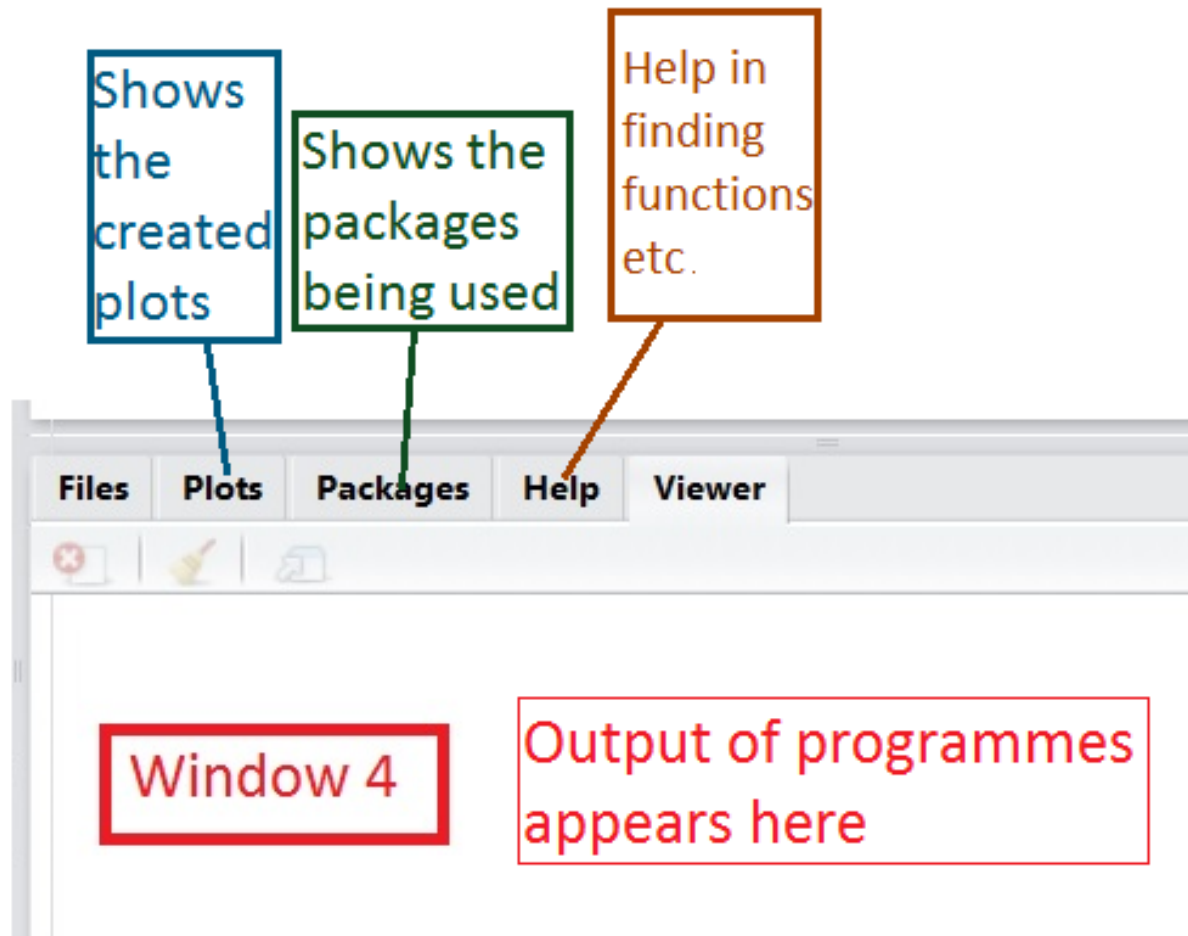
Window 3

Data can be imported from other files by clicking here

Introduction to R Studio

Description of Window 4 : Output window

The output of programmes appears in this window.



Introduction to R Studio

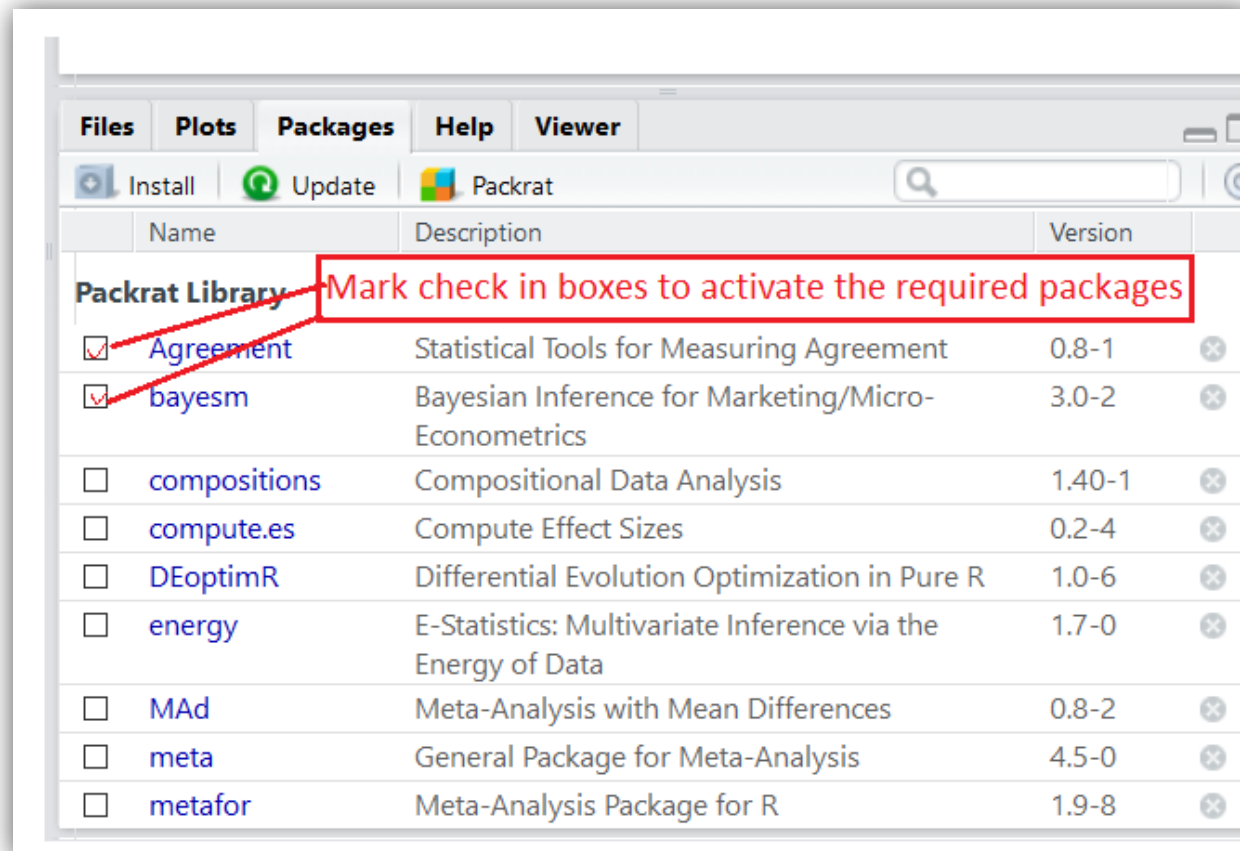
Description of Window 4 : Output window

Packages:

All the packages being installed appear here.

Packages are not active.

Check mark in the boxes to activate them.



Introduction to R Studio

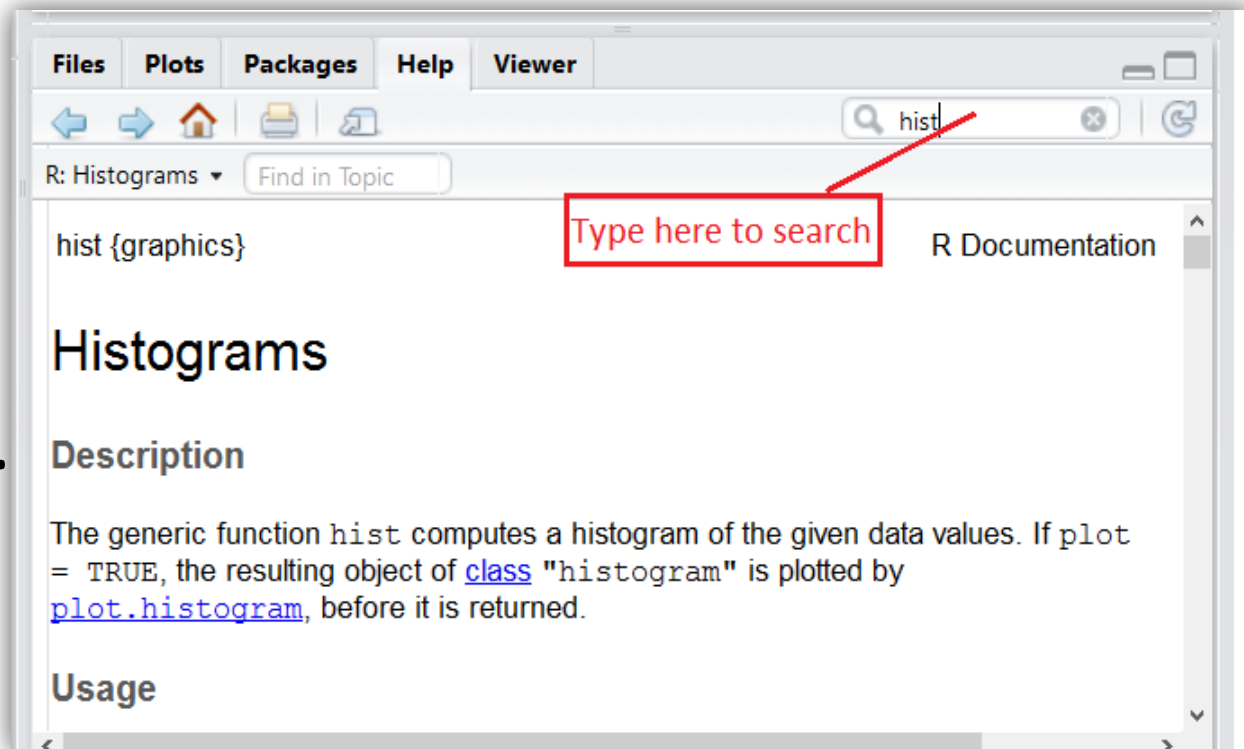
Window 4 : Output window

Help:

Various types of help can be asked.

E.g., to know about histogram,
type **hist**.

Information appears.



Introduction to R Studio

Example:

Histogram of values 1,2,1,1,2,3,1,2,3,1,2,2,3

R studio has following operation and output:

```
1 x=c(1,2,1,1,2,3,1,2,3,1,2,2,3)
2 hist(x)
3
```

Step 1: Type commands here

Step 2 :Click on 'Run'

Console ~/

> hist(x)

Step 3: The executed command appear here

