

Introduction to R Software

Swayam Prabha

Lecture 4

Command line, Libraries, Packages and Data Editor

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from
<http://home.iitk.ac.in/~shalab/sp>



Command Line versus Scripts

What is command line?

```
R Console

R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 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]

> |  This is command line
```

Libraries in R

R provides many functions and one can also write own.

Functions and datasets are organised into libraries

To use a library, simply type the `library` function with the name of the library in brackets.

```
library(. )
```

For example, to load the `spatial` library type:

```
library(spatial)
```

Libraries in R

Examples of libraries that come as a part of base package in R.

MASS : package associated with Venables and Ripley's book entitled *Modern Applied Statistics using S-Plus*.

mgcv : generalized additive models.

Contents of Libraries

It is easy to use the `help` function to discover the contents of library packages.

Here is how we find out about the contents of the `spatial` library:

```
library(help=spatial) returns  
      Information on package 'spatial'
```

```
Description:
```

```
Package:    spatial
```

```
Priority:    recommended
```

```
Version:    7.3-8
```

followed by a list of all the functions and data sets.

Then we get....₅

Contents of Libraries

```
R Console
> library(help=spatial)
> |

Documentation for package 'spatial'

Information on package 'spatial'

Description:

Package:          spatial
Priority:         recommended
Version:         7.3-11
Date:            2015-08-29
Depends:         R (>= 3.0.0), graphics, stats, utils
Suggests:        MASS
Authors@R:       c(person("Brian", "Ripley", role = c("aut", "cre",
"          "cph"), email = "ripley@stats.ox.ac.uk"),
"          person("Roger", "Bivand", role = "ctb"),
"          person("William", "Venables", role = "cph"))

Description:      Functions for kriging and point pattern analysis.
Title:           Functions for Kriging and Point Pattern Analysis
LazyLoad:        yes
ByteCompile:     yes
License:         GPL-2 | GPL-3
URL:             http://www.stats.ox.ac.uk/pub/MASS4/
NeedsCompilation: yes
Packaged:        2015-08-28 15:25:37 UTC; ripley
Author:          Brian Ripley [aut, cre, cph], Roger Bivand [ctb],
                William Venables [cph]
```

Installing Packages and Libraries

The base R package contains programs for basic operations.

It does not contain some of the libraries necessary for advanced statistical work.

Specific requirements are met by special packages.

They are downloaded and their downloading is very simple.

Installing Packages and Libraries

To install any package,

- run the R program,
- then on the command line, use the `install.packages` function to download the libraries we want.

Installing Packages and Libraries

The base R package contains some necessary libraries only.

Other libraries are required for advanced statistical work which are downloaded and installed as and when required.

Run the R program, then use the `install.packages` command to download the libraries.

Examples :

```
install.packages("ggplot2") : installs package ggplot2
```

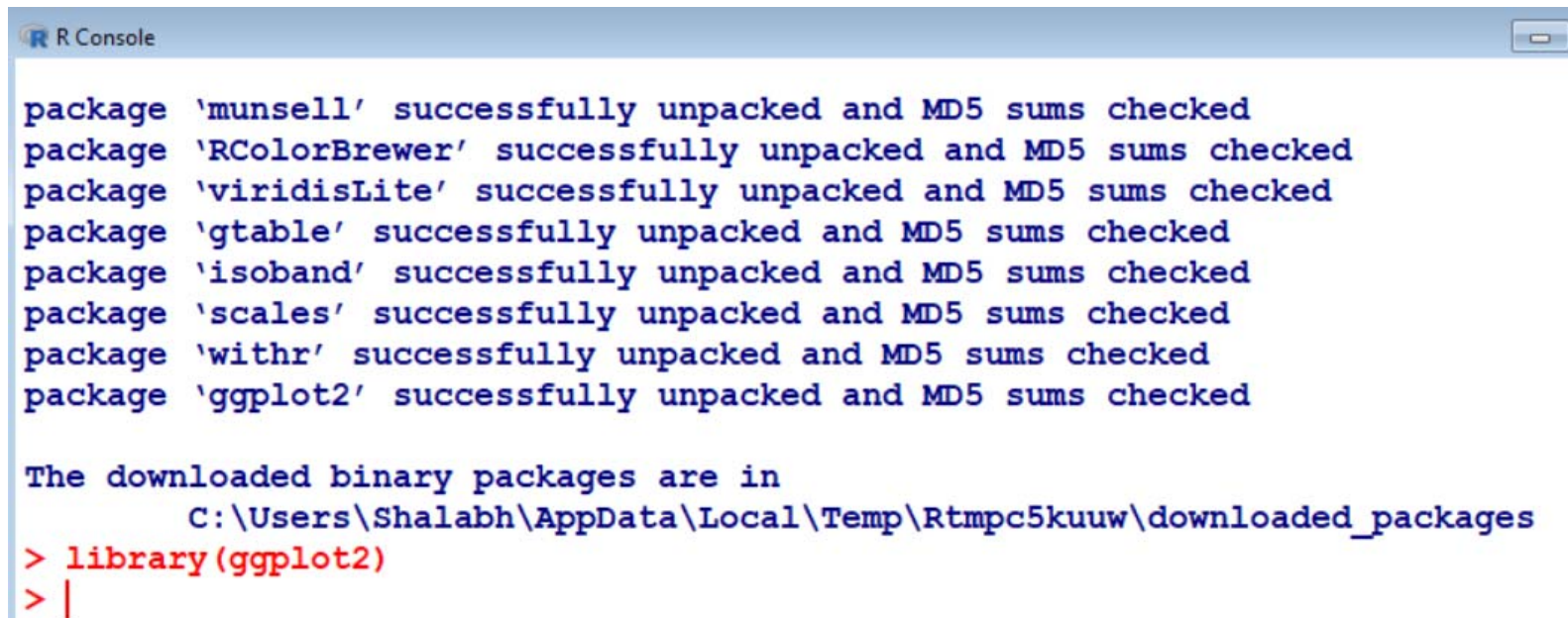
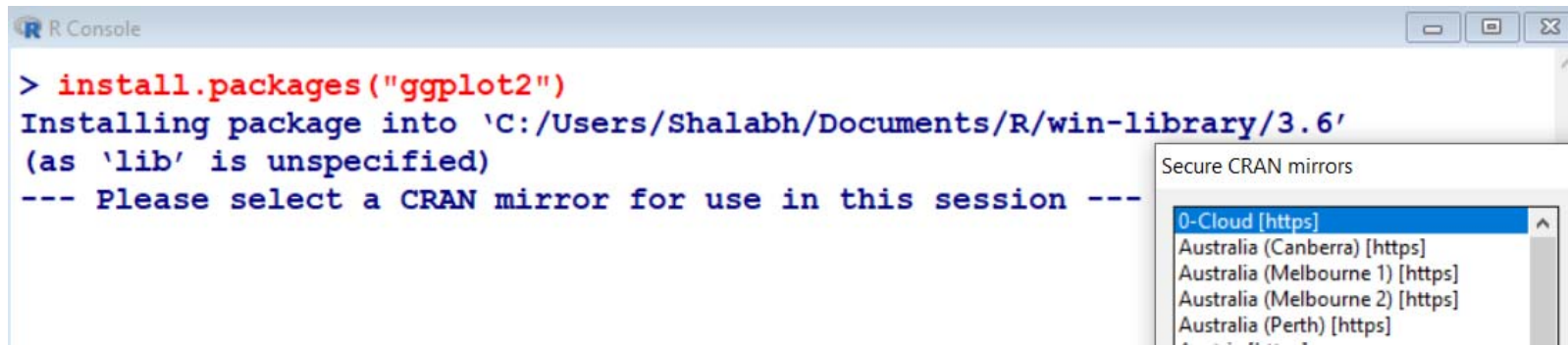
```
install.packages("agricolae") : installs package agricolae
```

```
install.packages("DoE.base") : installs package DoE.base
```

Installing Packages and Libraries

Example

```
install.packages("ggplot2")
```



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.

`rm()` command removes variable names

For example,

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

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.

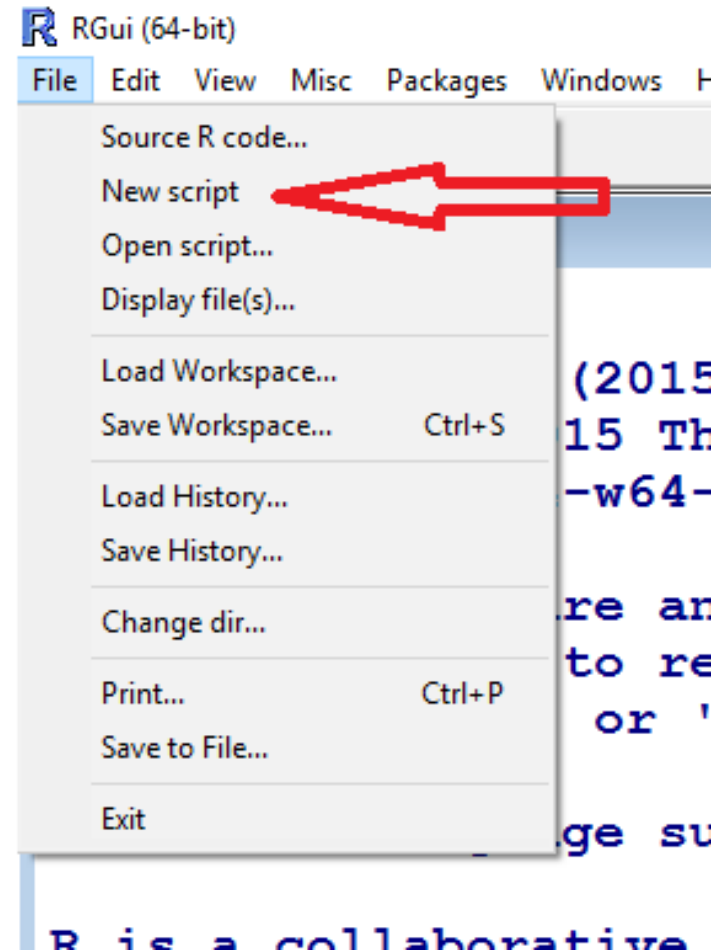
It is accessible from the **RGui** menu bar.

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

Command Line versus Scripts

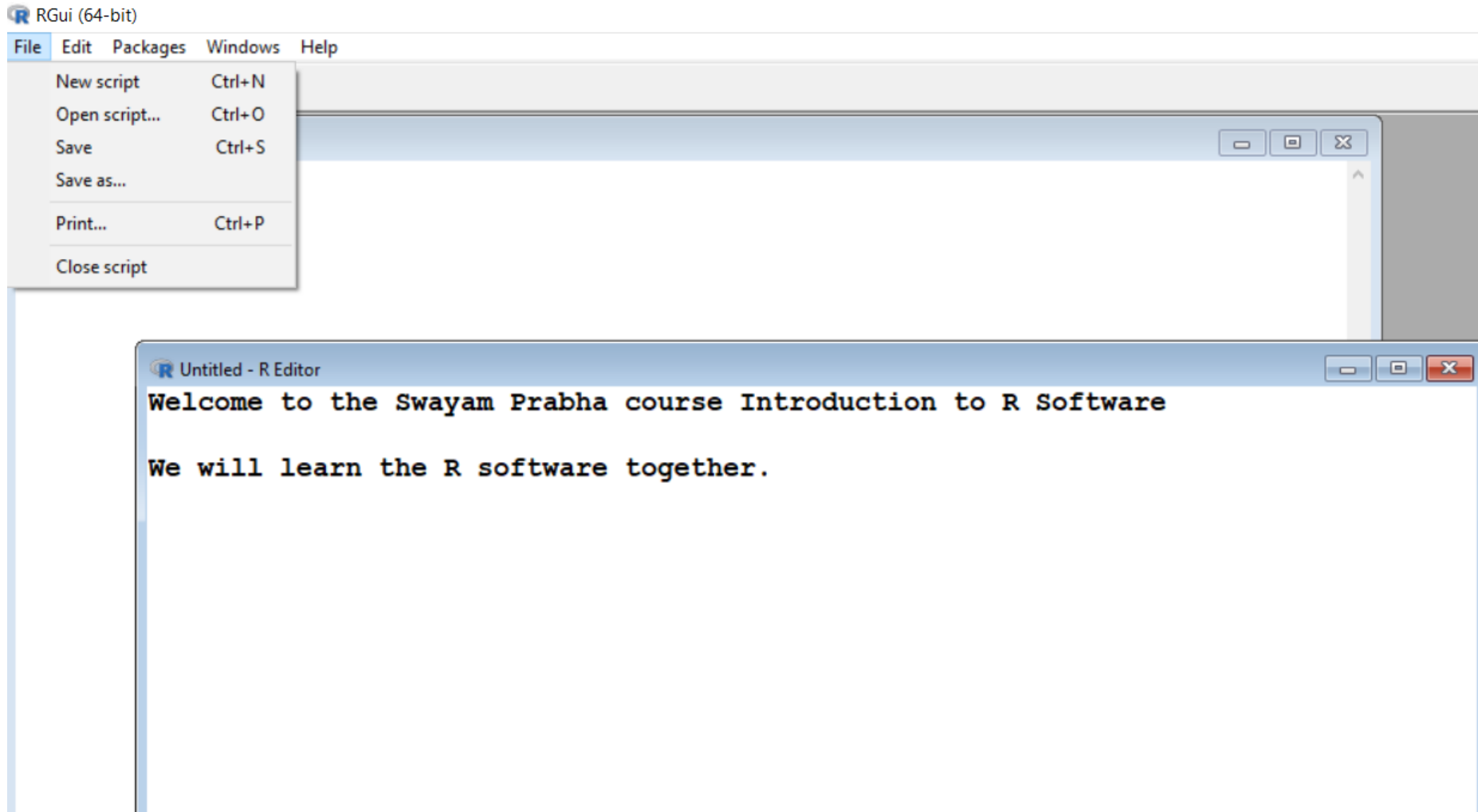
At this point R will open a window entitled `Untitled-R Editor`.

We may type and edit in this.

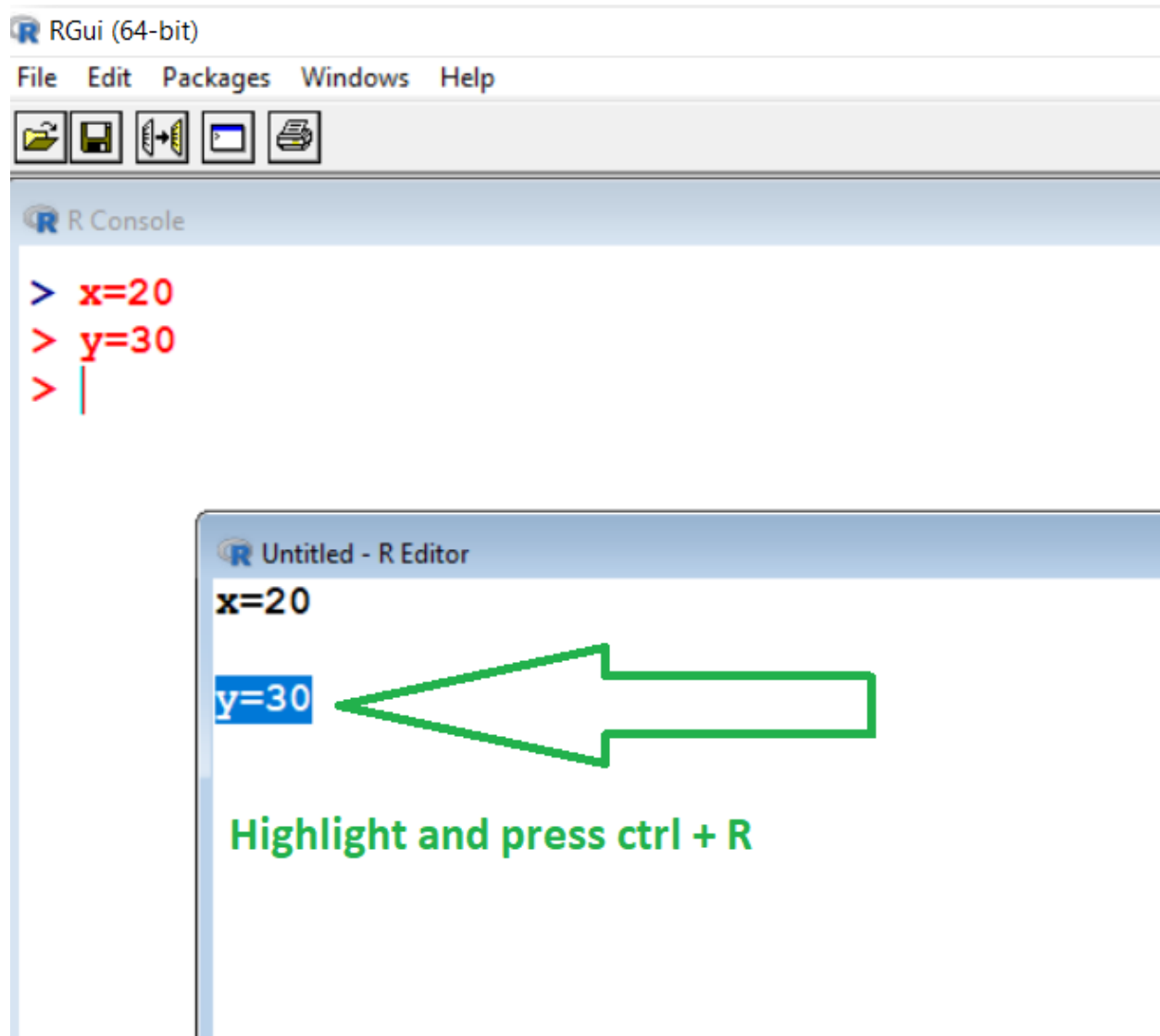


If we want to execute a line or a group of lines, just highlight them and press `Ctrl+R`.

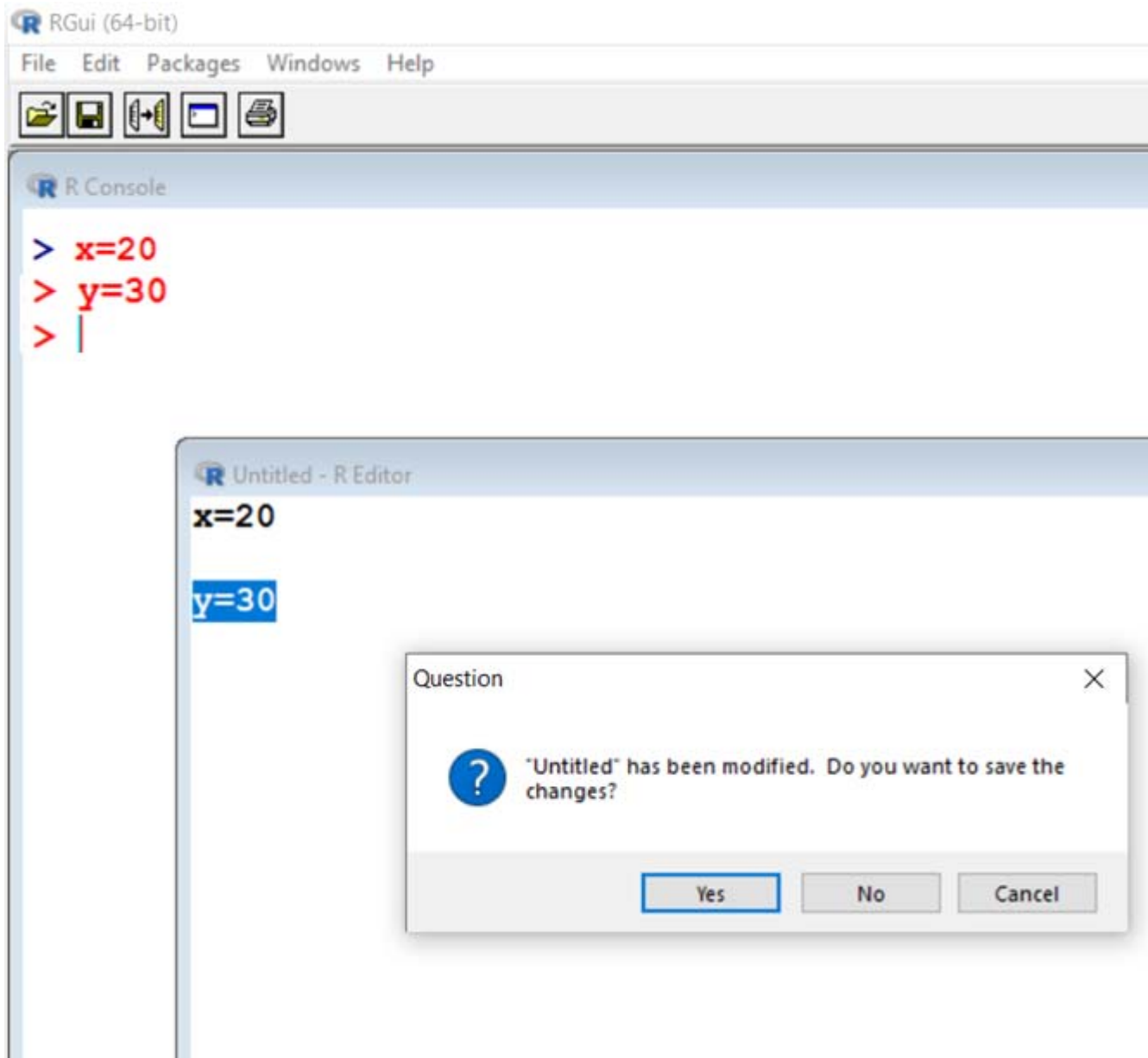
Command Line versus Scripts



Command Line versus Scripts

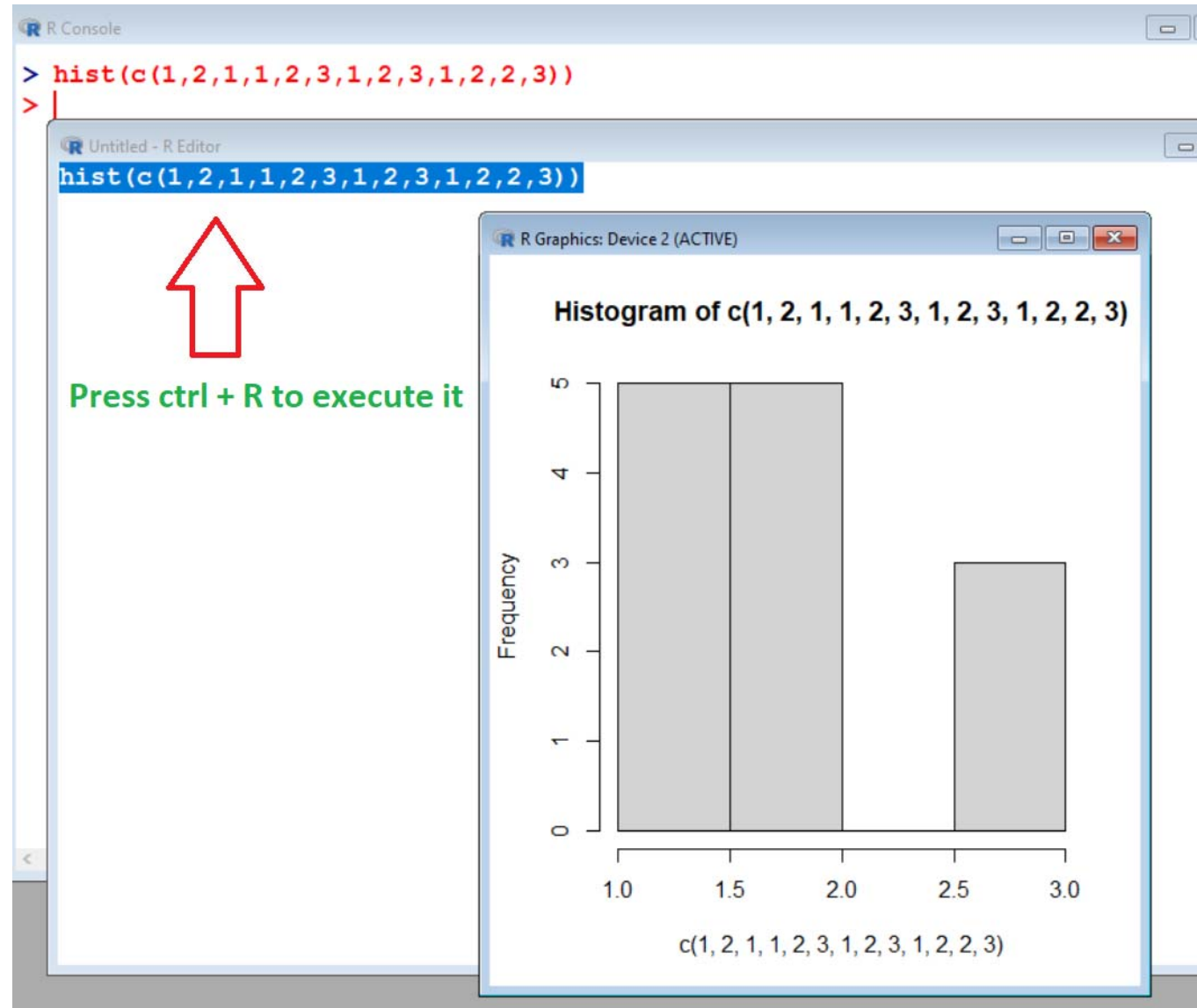


Command Line versus Scripts

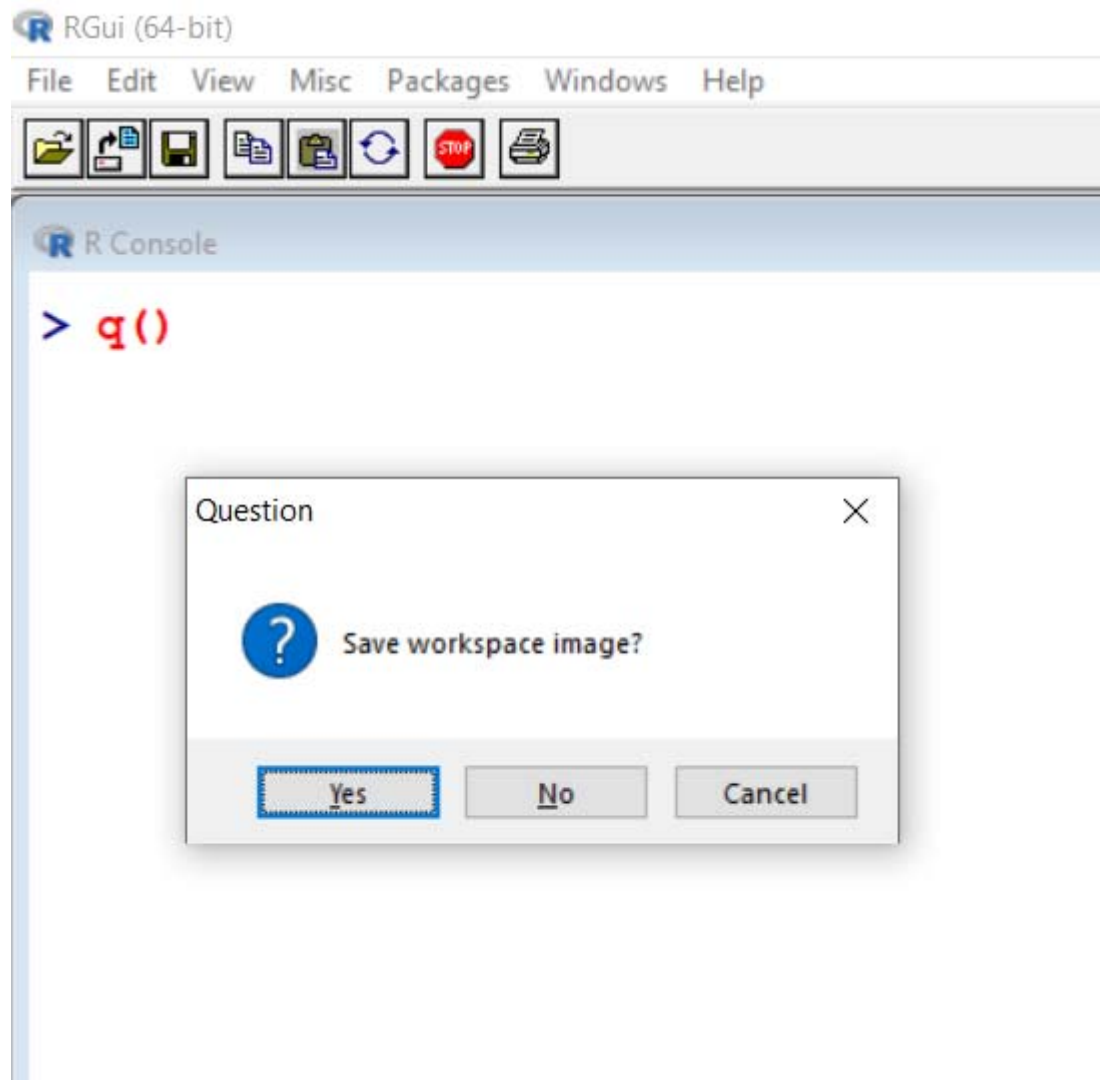


Command Line versus Scripts

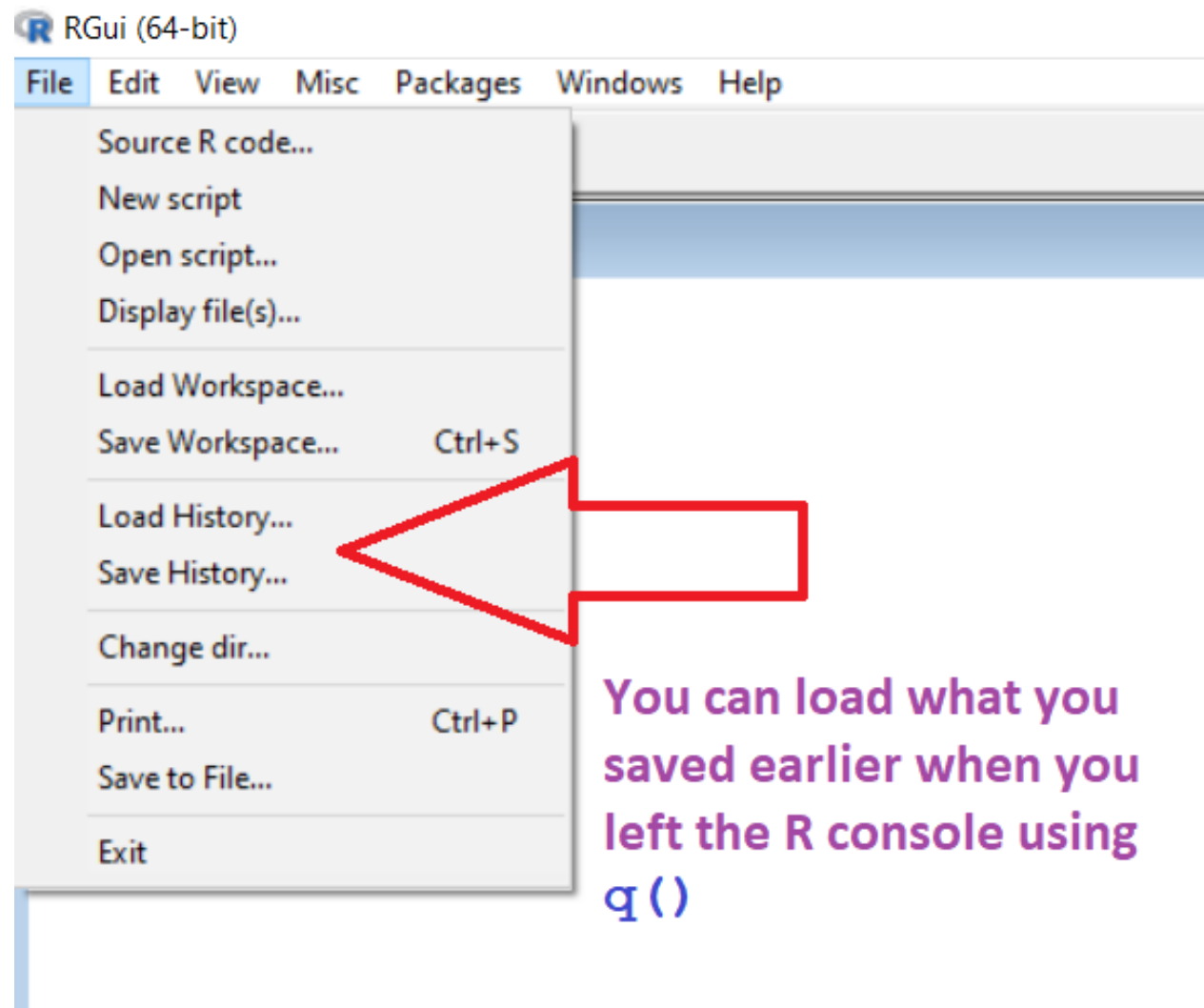
```
hist(c(1,2,1,1,2,3,1,2,3,1,2,2,3))
```



Command Line versus Scripts



Command Line versus Scripts



Command Line versus Scripts

Option 2:

Use R studio software.