UiO : PROMENTA
Det samfunnsvitenskapelige fakultet

Introduction to the R Language

E. F. Haghish

25/8/2022

Getting to know R and RStudio

- How does it look like
- What should you know before getting your hands on R
- R syntax (basics)
- RStudio IDE
- Getting help online
- Writing script files

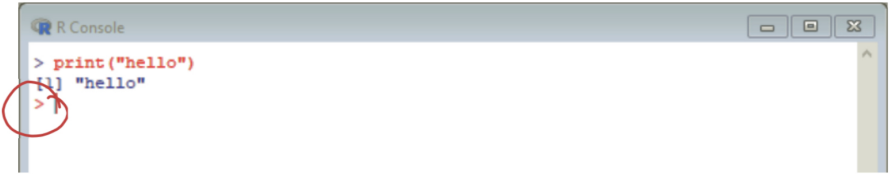
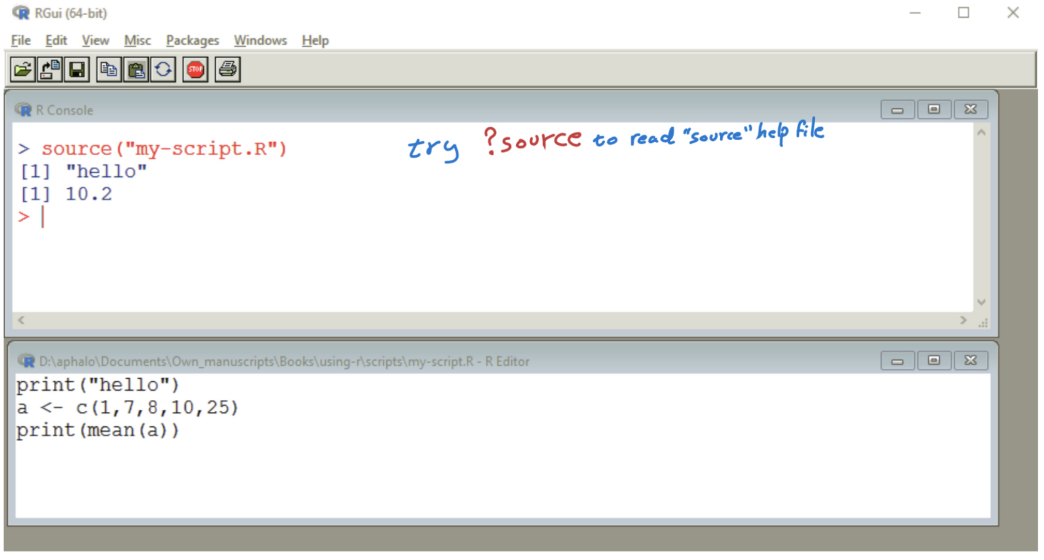


FIGURE 1.1
The R console where the user can type textual commands one by one. Here the user has typed `print("Hello")` and *entered* it by ending the line of text by pressing the "enter" key. The result of running the command is displayed below the command. The character at the head of the input line, a ">" in this case, is called the command prompt, signaling where a command can be typed in. Commands entered by the user are displayed in red, while results returned by R are displayed in blue.


RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> source("my-script.R")
[1] "hello"
[1] 10.2
>
try ?source to read "source" help file
D:\aphalo\Documents\Own_manuscripts\Books\using-r\scripts\my-script.R - R Editor
print("hello")
a <- c(1,7,8,10,25)
print(mean(a))

1.2.3.2 Using R in a "batch job"

To run a script we need first to prepare a script in a text editor. Figure 1.4 shows the console immediately after running the script file shown in the text editor. As before, red text, the command `source("my-script.R")`, was typed by the user, and the blue text in the console is what was displayed by R as a result of this action. The title bar of the console, shows "R-console," while the title bar of the editor shows the *path* to the script file that is open and ready to be edited followed by "R-editor."

When working at the command prompt, most results are printed by default. However, within scripts one needs to use function `print()` explicitly when a result is to be displayed.

A true "batch job" is not run at the R console but at the operating system command prompt, or shell. The shell is the console of the operating system—Linux, Unix, OS X, or MS-Windows. Figure 1.5 shows how running a script at the Windows

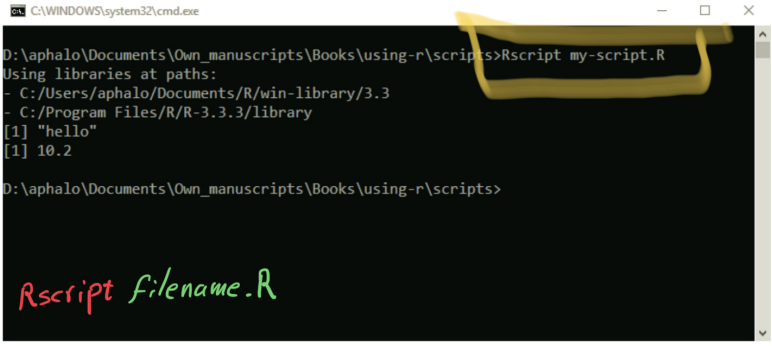


FIGURE 1.5
Screen capture of the MS-Windows command console just after running the same script. Here we use `Rscript` to run the script; the exact syntax will depend on the operating system in use. In this case, R prints the results at the operating system console or shell, rather than in its own R console.

command prompt looks. A script can be run at the operating system prompt to do time-consuming calculations with the output saved to a file. One may use this approach on a server, say, to leave a large data analysis job running overnight or

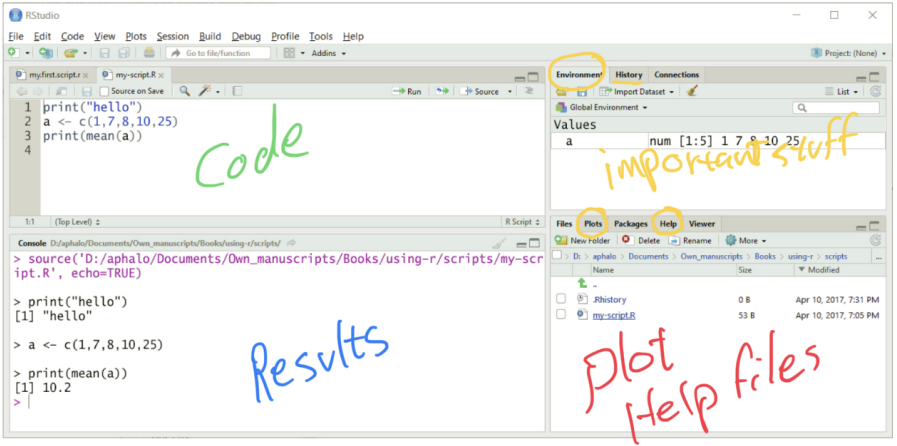


FIGURE 1.6
The RStudio interface just after running the same script. Here we used the "Source" button to run the script. In this case, R prints the results to the R console in the lower left pane.

1.3 Reproducible data analysis

Reproducible data analysis is much more than a fashionable buzzword. Under any situation where accountability is important, from scientific research to decision making in commercial enterprises, industrial quality control and safety and environmental impact assessments, being able to reproduce a data analysis reaching the same conclusions from the same data is crucial. Most approaches to reproducible data analysis are based on automating report generation and including, as part of the report, all the computer commands used to generate the results presented.

A fundamental requirement for reproducibility is a reliable record of what commands have been run on which data. Such a record is especially difficult to keep when issuing commands through menus and dialogue boxes in a graphical user interface or interactively at a console. Even working interactively at the R console using copy and paste to include commands and results in a report is error prone, and laborious.

1.4.1 R's built-in help

To access help pages through the command prompt we use function `help()` or a question mark. Every object exported by an R package (functions, methods, classes, data) is documented. Sometimes a single help page documents several R objects. Usually at the end of the help pages, some examples are given, which tend to help very much in learning how to use the functions described. For example, one can search for a help page at the R console.

```
help("sum")
?sum
```

?help ← try this

Look at help for some other functions like `mean()`, `var()`, `plot()` and, why not, `help()` itself!

```
help(help)
```

itself. R function `citation()` when called with the name of a package as its argument provides the reference that should be cited for the package, and without an explicit argument, the reference to cite for the version of R in use.

```
citation()
##
## To cite R in publications use:
##
##   R Core Team (2020). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2020},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

Look at the help page for function `citation()` for a discussion of why it is important for users to cite R and packages when using them.

1.4.2.1 Netiquette

In most internet forums, a certain behavior is expected from those asking and answering questions. Some types of misbehavior, like use of offensive or inappropriate language, will usually result in the user losing writing rights in a forum. Occasional minor misbehavior, will usually result in the original question not being answered and instead the problem highlighted in the reply. In general following the steps listed below will greatly increase your chances of getting a detailed and useful answer.

- Do your homework: first search for existing answers to your question, both

Say that you are new to R

online and in the documentation. (Do mention that you attempted this without success when you post your question.)

- Provide a clear explanation of the problem, and all the relevant information. Say if it concerns R, the version, operating system, and any packages loaded and their versions.
- If at all possible, provide a simplified and short, but self-contained, code example that reproduces the problem (sometimes called *reprex*).

1.4.2.2 StackOverflow

Nowadays, StackOverflow (`http://stackoverflow.com/`) is the best question-and-answer (Q&A) support site for R. In most cases, searching for existing questions and their answers, will be all that you need to do. If asking a question, make sure that it is really a new question. If there is some question that looks similar, make clear how your question is different.

StackOverflow has a user-rights system based on reputation, and questions and answers can be up- and down-voted. Those with the most up-votes are listed at the top of searches. If the questions or answers you write are up-voted, after you accumulate enough reputation, you acquire badges and rights, such as editing other users' questions and answers or later on, even deleting wrong answers or off-topic questions from the system. This sounds complicated, but works extremely well at ensuring that the base of questions and answers is relevant and correct, without relying on nominated *moderators*. When using StackOverflow, do contribute by accepting correct answers, up-voting questions and answers that you find useful, down-voting those you consider poor, and flagging or correcting errors you may discover.

Numeric values and arithmatic

- You can use R as a calculator
- type `2 + 2` in R console and press enter
- type `2 * 5` in R console and press enter

Numeric values and arithmatic

```
1+2*5+1/2 = ?
```

2.3 Numeric values and arithmetic

When working in R with arithmetic expressions, the normal mathematical precedence rules are respected, but parentheses can be used to alter this order. Parentheses can be nested, but in contrast to the usual practice in mathematics, the same parenthesis symbol is used at all nesting levels.

Both in mathematics and programming languages *operator precedence rules* determine which subexpressions are evaluated first and which later. Contrary to primitive electronic calculators, R evaluates numeric expressions containing operators according to the rules of mathematics. In the expression $3 + 2 \times 3$, the product 2×3 has precedence over the addition, and is evaluated first, yielding as the result of the whole expression, 9. In programming languages, similar rules apply to all operators, even those taking as operands non-numeric values.

Assigning values

There are some syntactically legal statements that are not very frequently used, but you should be aware that they are valid, as they will not trigger error messages, and may surprise you. The most important thing is to write code consistently. The "backwards" assignment operator `->` and resulting code like `1 -> a` are valid but less frequently used. The use of the equals sign (=) for assignment in place of `<-` although valid is discouraged. Chaining assignments as in the first statement below can be used to signal to the human reader that `a`, `b` and `c` are being assigned the same value.

A <- 2
B <- 5
5 -> N

Reading R help files ==========

- They are written in plane text
  - in most recent R and RStudio version, **finally**, the code receives syntax highlighter!

Operators

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2 is 2

```
x <- c(1:10)
x[(x>8) | (x<5)]
```

```
## [1]  1  2  3  4  9 10
```

Logical Operators

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x | y	x OR y
x & y	x AND y

Vocabulary check

*What words did I present so far that are known to **R interpreter**?*

TASK

Read one or more of the following help files and get ready to answer some questions.

Read the help files of the following vocabularies

- `help`
- `source`
- `print`
- `citation`
- `c`
- `sum`
- `mean`
- `var`
- `plot`