Introduction to R and RStudio

matthias.kuhn@tu-dresden.de
Institute for Medical Informatics and Biometry (IMB),
Faculty of Medicine Carl Gustav Carus, TU Dresden

October 9, 2023

The R software is a programming language and software environment that is tailored for statistical analyses and graphics. R is a functional programming language but allows some basic object-orientation, too. It is an interpreted language which makes it not the fastest programming language. R belongs to the GNU project ("free software") and the source code is mainly written in C, Fortran or R itself. The article on Wikipedia gives a good overview of R with examples.

R is command-line oriented and it ships with a very basic user-interface. To add a little comfort when working with R we recommand the software `RStudio`, an integrated development environment (IDE) made for R. Like R, RStudio is also available as open source software. And RStudio also handles Python scripts quite well. The pretence is to be able to have the best of the R-world and the Python-world available in one IDE (more information).

This document gives you a primer in using R. It covers installation and shows you the first steps in R and RStudio.

1 Installation

We give a short set of instructions how to install R and RStudio on your computer at home.

Installation of R. R is open source software and as such available at no cost at the r-project.org. To effectively download the files you click on `CRAN` that stands for "Comprehensive R Archive Network" and choose a server in your vicinity. On Windows and on Mac the R program comes with a basic graphical user interface. On Linux R is merely a console program. Verify that your R installation was successful by launching the R program. You can quit it by typing `q()` (for quit) and hitting ↲.

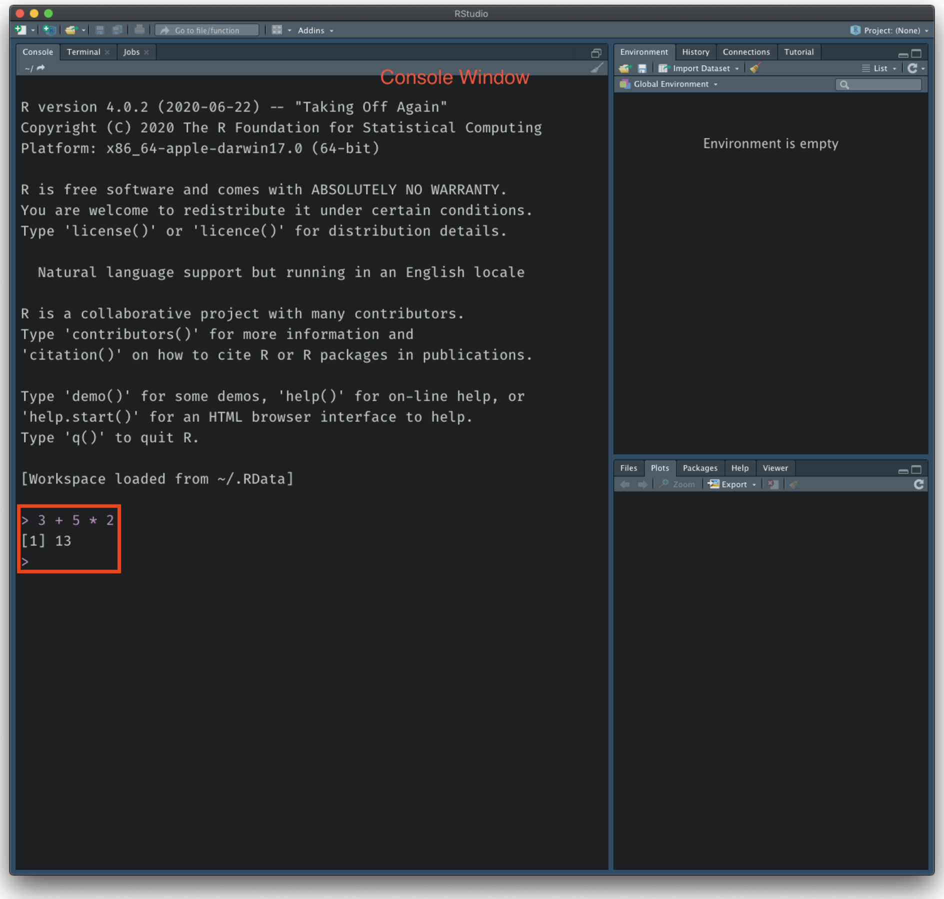
Installation of RStudio. RStudio is developed by the company Posit and you can download the RStudio IDE at no cost at `https://posit.co/download/rstudio-desktop/`. RStudio auto-detects your local R installation and provides you with a more convenient graphical user interface. So you can work with R via RStudio, i.e., it suffices to start only the RStudio application.

2 First steps with RStudio

1. Start **RStudio** (Look for the icon).
2. RStudio will start R for you. In the **Console Window** (*Console*) on the left side of RStudio runs an R instance where you can directly type in and execute R commands. The „>“-character at the beginning of the line means that R is ready to take input from you. An R-command can stretch over multiple lines. If this is the case the first character of the line is a „+“.

   An entered command is executed as soon as you press ↵. You can abort the input of a command with the Esc key.

   In this way you can use R as a simple pocket calculator:

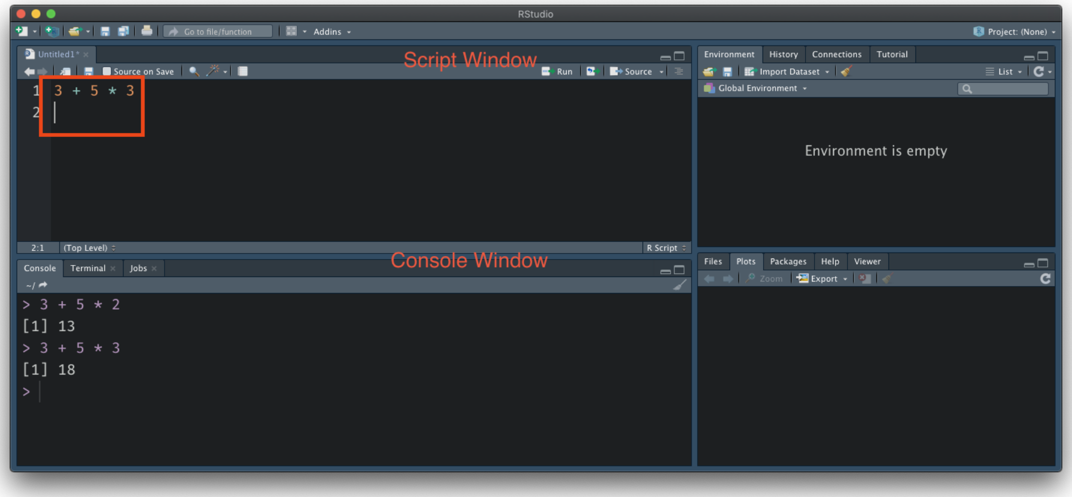


```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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.

[Workspace loaded from ~/.RData]

> 3 + 5 * 2
[1] 13
>
```

3. You should make it a habbit to put your R commands in a separate text file (a so-called **R-script**) as soon as you start issuing more than just a few commands. This way you can re-run your commands easily at a later point in time.

   To start a new R-script in RStudio you can use the menu File ⟩ New ⟩ R Script or the shortcut Ctrl + ⇧ + N. A **script window** will open on the left side above the Console window. Here you can type your R commands and save them to file per menu via File ⟩ Save or per shortcut Ctrl + S. To open an existing R script in RStudio's script window use File ⟩ Open File... or simply Ctrl + O.

   But how can you actually execute R commands from your R script? — The shortcut Ctrl + ↲ executes the current line of your R script in the R console. If you have text selected then only this selected area is executed in R. Ctrl + ↲ is arguably the **most important shortcut** in RStudio.



```
3 + 5 * 3
```

```
> 3 + 5 * 2
[1] 13
> 3 + 5 * 3
[1] 18
>
```

4. Anything in a line after a # is treated as comment. **Comments** are text in your R script that is not meant to be executed by R but serves as documentation.

5. A nice property of RStudio is that you have **auto-completion** in the console and the script window. As soon as you have typed the beginning of an R-command (like `med`) you can press ↹ to get a list of possible completions RStudio knows of, for instance `median`. This can save you a lot of typing!

   The console and the script windows are the two most important windows in RStudio. But there are more and in what follows we introduce some of them.

6. On the lower right side of RStudio you can spot the **graphics window** under the name *Plots*. Any graphical output produced by an R command will be shown there.

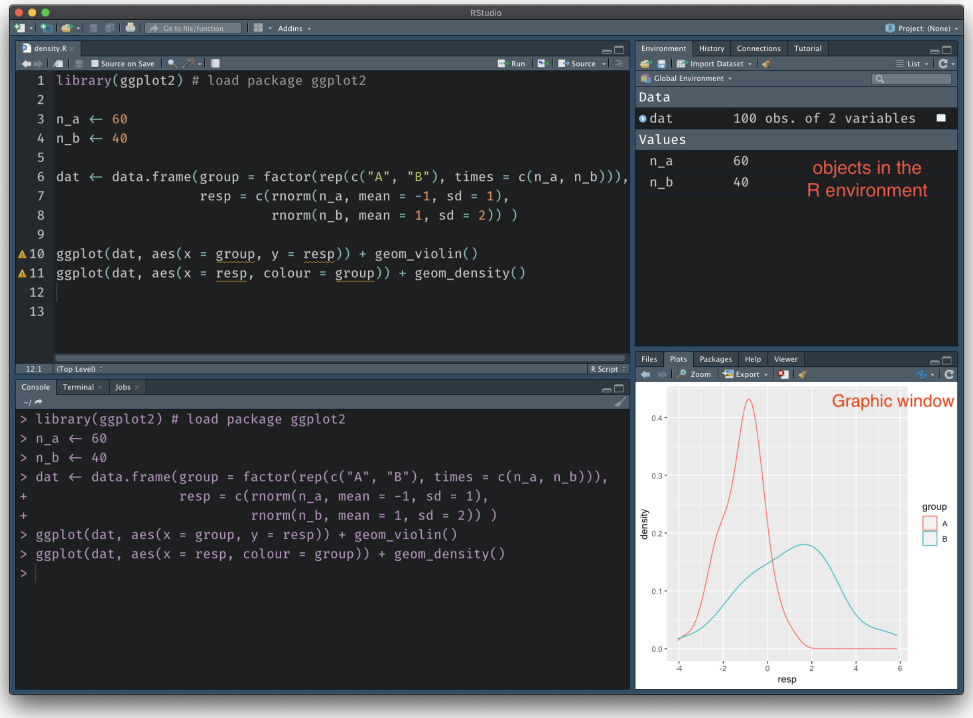
7. There is an extensive help system integrated in R and it can be accessible in RStudio via the **help window** (*Help*). To get to the start page click on the house icon 🏠 in the

headline of the help window.

If you look for the help of a particular R-command (say, `mean`) you can type `?mean` and press Enter to see the help page for this command. Alternatively, you can also use the general search pane 🔍 on the right side of the headline within the help window. The example section at the end of a help page is often most instructive.

In case you do not know the exact command (and hence can not look up its help page) the R command `RSiteSearch()` might help you further: just give a keyword as argument and it will look up related R-commands in an online R-specific search engine and shows the results in a browser.

8. At the upper right side there is the **environment window** where all R objects are listed that are available in your current session. If you click on *Import Dataset* in the headline an assistant will guide you to read in data from different sources (text files or on the web).
9. Via the **packages window** (*Packages*) you can install and load R extensions aka packages. The most important host for open-source R-packages is CRAN, the Comprehensive R Archive Network. Packages provide new functions and often also data sets.



```
library(ggplot2) # load package ggplot2

n_a ← 60
n_b ← 40

dat ← data.frame(group = factor(rep(c("A", "B"), times = c(n_a, n_b))),
                  resp = c(rnorm(n_a, mean = -1, sd = 1),
                           rnorm(n_b, mean = 1, sd = 2)) )

ggplot(dat, aes(x = group, y = resp)) + geom_violin()
ggplot(dat, aes(x = resp, colour = group)) + geom_density()
```

```
> library(ggplot2) # load package ggplot2
> n_a ← 60
> n_b ← 40
> dat ← data.frame(group = factor(rep(c("A", "B"), times = c(n_a, n_b))),
+                  resp = c(rnorm(n_a, mean = -1, sd = 1),
+                           rnorm(n_b, mean = 1, sd = 2)) )
> ggplot(dat, aes(x = group, y = resp)) + geom_violin()
> ggplot(dat, aes(x = resp, colour = group)) + geom_density()
>
```

Data	
dat	100 obs. of 2 variables
Values	
n_a	60
n_b	40

Congratulations! You have just made your first steps in RStudio! Be curious and experiment what you already can do with R & RStudio.