How to set up R and Rstudio

Alexander Gamerdinger

2022-02-15

Contents

About R and Rstudio **2**

Install R and RStudio **3**
Uninstall R and Rstudio . . . 3
Install R . . . 3
Install R-Studio . . . 3

Adapt global options in R Studio **5**
Turn off "Save workspace" . . . 5
Rstudio Appearance . . . 5
Pane layout . . . 7
Line softwrapping . . . 8
Line highlighting, margins and rainbow parenthesis . . . 10

Important

To minimize errors in class, I urge everyone who already has R and Rstudio installed to uninstall them and start from scratch. This is important, because some R packages require you to be on the latest version. If you run them with older R or Rstudio versions, it might cause errors.

We will also setup Rstudio different from the default, so even if you have the newest R and Rstudio version, please follow the all the steps.

Dear all

Please make sure to follow the steps below to set up your Rstudio environment for this course. This document was originally written by Emil Begtrup-Bright and slightly refined by me to make things even more clear before we get started.

About R and Rstudio

This section briefly explains some basic concepts about computers that are usually not introduced, but just tacitly assumed. It is my experience that explaining them – briefly – can be a remedy for some basic confusion. **Feel free to skip this section, it is not essential.**

R isn't a program that you can open and start using, like Microsoft Word or Internet Explorer. Instead, R is a computer language, like C, C++, or UNIX. You use R by writing commands in the R language and asking your computer to interpret them. In the old days, people ran R code in a UNIX terminal window - as if they were hackers in a movie from the 1980s, like this:

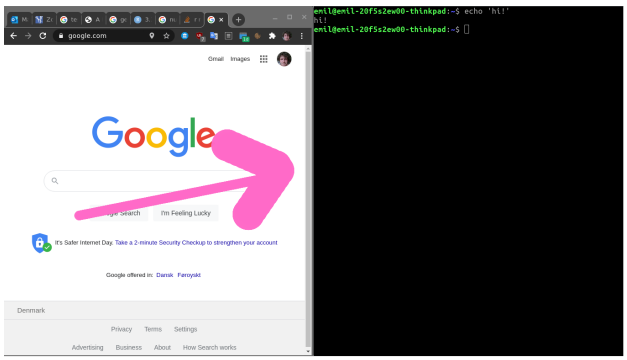

Figure 1: The view of a terminal (here a Linux terminal)

Windows and Mac users usually do not program from a terminal window, so the Windows and Mac downloads for R come with a simple program that opens a terminal-like window for you to run R code in. This is what opens when you click the R icon on your Windows or Mac computer. These programs do a little more than the basic terminal window, but not much. You may hear people refer to it as the R GUI (*Graphical User Interface*).

We will instead be using a GUI for R called Rstudio. Rstudio *is* an application like Microsoft Word — except that instead of helping you write in English or Danish, RStudio helps you write in R.

RStudio is a like an advanced text editor, like Notepad in Windows. It basically helps us write and send commands to the R programming language. Rstudio has a sort-of terminal built in, called the console. this is where R lives in Rstudio, you might say.

RStudio, being a piece of software that provide facilities for working with code, is also what we call an IDE (*Integrated Development Enviroment*). So for example, Microsoft Word is a GUI for working with text, but it is not an IDE, like Rstudio, because Microsoft Word isn't made for working with code.

For the very interested: Here's a very well-written in-depth introduction to the basic metaphors of computers: In the Beginning was the Command Line, by Neal Stephenson.

Install R and RStudio

Uninstall R and Rstudio

If you already have R or Rstudio from a previous course, please uninstall it first.

Uninstalling R

R for macOS consists of two parts: the GUI (R.APP) and the R framework. Un-installation is as simple as removing those folders (e.g. by dragging them onto the Trash aka Bin). The typical installation will install the GUI into the /Applications/R.app folder and the R framework into the /Library/Frameworks/R.framework folder. The links to R and Rscript in /usr/local/bin should also be removed.

Additional information can be found here: https://cran.r-project.org/doc/manuals/r-release/R-admin.html#Uninstalling-under-macOS

For Windows users, follow this guide:

https://stackoverflow.com/questions/55204017/how-to-uninstall-r-and-rstudio-with-all-packages-settings-and-everything-else

Upgrade RStudio

Rstudio -> Help -> Check for Updates

Install R

Download the newest R version from here:

https://cran.r-project.org/

Pick your operation system.

- If on Windows -> click on 'base' and then click the big letters 'Download R X.X.X for Windows' (instead of X.X.X there will numbers, eg the newest version of R currently available)
- If on Mac -> click the .pkg-file under 'latest release', named something like R-X.X.X.pkg, like the Windows-version
- If on Linux -> follow the guide here

Then install the program(ming language). You don't need to adapt the default install options during installation.

Double check

In this course, we will be using **R version 4.1.2**, so please make sure to download this version. You can check which version you are on by typing `R.Version()` in the console.

Install R-Studio

Go to the following link and choose the version for your operating system, to download RStudio onto your computer:

https://rstudio.com

Install the program. It is not necessary to check/uncheck special options during installation.

Double check

In this course, we will be using **Rstudio Desktop version 2021.09.2**, so please make sure to download this version. You can find out the version of RStudio IDE by simply selecting "About RStudio" from the Help menu.

Adapt global options in R Studio

Most of these options I think improve the readability, navigation and overall coding experience. But if you have a setup you like more, feel free to use it. The only step here that is absolutely required is the first, to turn of "save workspace".

Turn off "Save workspace"

Set the following option from the Rstudio global options menu.

In RStudio, go to the Menu `Tools -> Global Options`. Set the options for 'Workspace' as highlighted in glorious pink in the screenshot below and then press "OK".

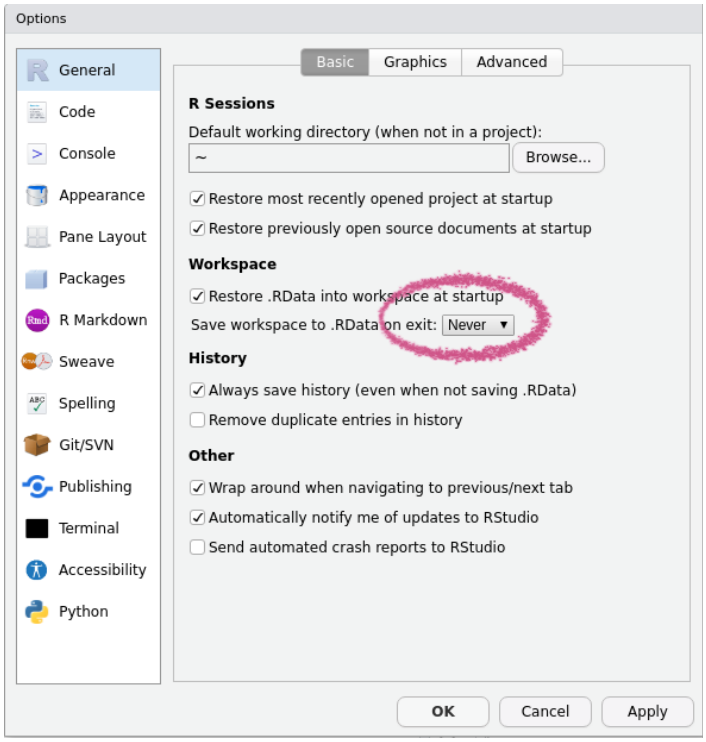

Figure 2: Turning off "Save workspace"

This means that Rstudio will always erase objects in memory when you close the program, and will load an empty enviroment every time you start it up. this leads to less confusion and encourages better coding practice.

Rstudio Appearance

The default color scheme in RStudio is ill-chosen for beginners, because the normal text color of the ouput shown in the console is in red, which is typically associated with an error. Instead, go to `Tools -> Global Options -> Appearance`. Under `Editor theme` select a different color scheme. Pretty much every color scheme is better than the default, but I recommend `Tomorrow Night`, which is a dark theme, where errors will be written in dark orange (almost red, with a little good will).

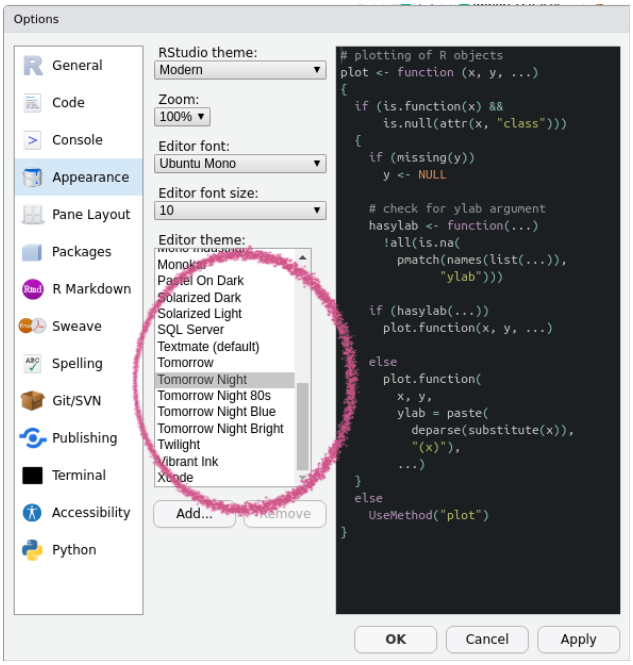

Figure 3: Change color sheme

Optional

For those of you interested in advanced color schemes (such as the one that I use), you can follow this quick guide. Since the package `rsthemes` by Garrick Aden-Buie is not yet on CRAN, you have to download it from GitHub with the following steps:

```
# first, install devtools
install.packages("devtools")

# second, install rsthemes
devtools::install_github("gadenbuie/rsthemes")

# Voila, you should have more themes to choose from under the Appearance panel.
# I use the theme: Material Palenight (rsthemes) in case you are interested!
```

Pane layout

Next up is the pane layout, which is also available through `Global Options`.

To improve readability, I recommend that you place the "Console" to the right and the "Source" to the left of your screen. Place the "Environment" pane beneath the "Source" layout, and the "Files, Plots" pane beneath your "Console" as seen in **Figure 5**.

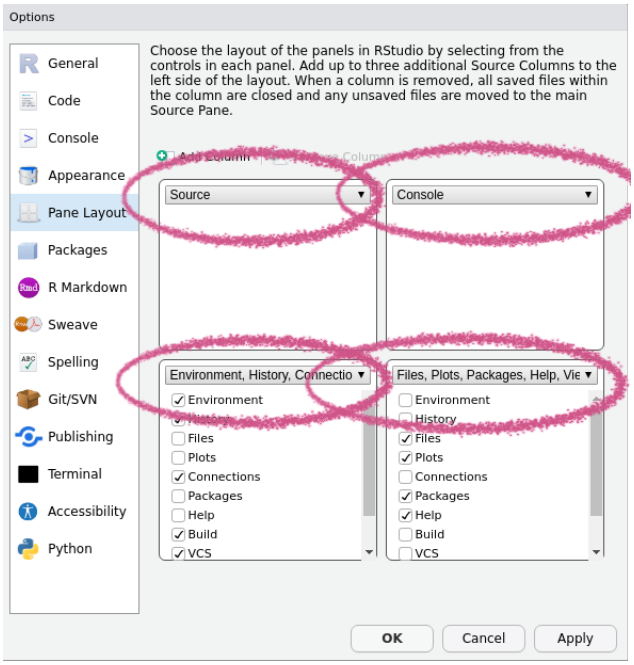

Figure 4: Changing Pane layout

You can use the mouse to resize the area of each pane, by using the mouse to pull in the border between each pane. Your layout should look like this, the lines where you can resize the panes are hightlighted.

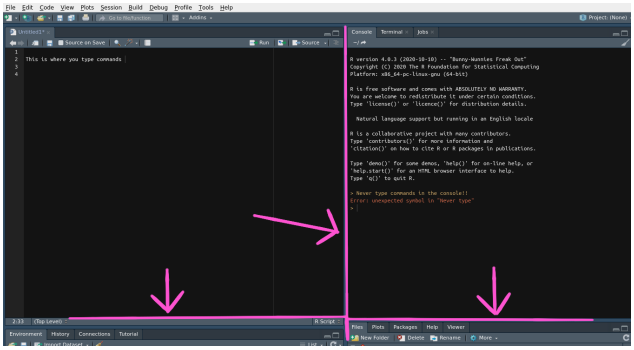

Figure 5: Rezising panes in Rstuido

This pane layout is a suggestion, if you like some other layout then feel free to use it. This is, however, the layout I use, so it might be easier for you to interpret my screen and apply it yourself if your layout is the same.

You should always have Rstudio in fullscreen, eg. maximized, so you use all your screen real estate. This gives you the best overview of your code.

Line softwrapping

This one can be a little hard to wrap your head around at first. Hopefully it will become clear by using it. If it is hard to understand, don't worry about it and just follow the instructions below to enable it, and learn what it means during our exercises instead.

Go to `Tools -> Global Options -> Code` under the `Editing` tab. select 'Soft-wrap R source files'.

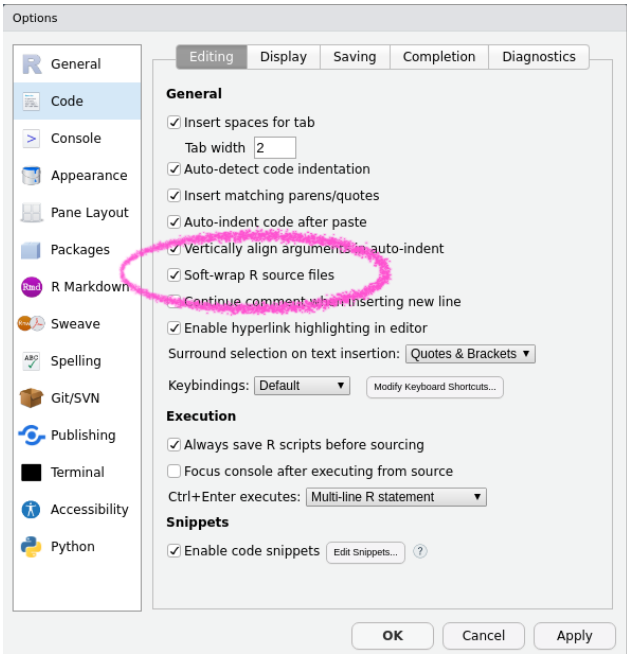

Figure 6: Setting line softwrapping

This means that R scripts - the source pane on the left in the layout we just set up – will 'break' a line into several lines, if it's a very long command, for example. The important thing here is that it is a *soft* line break. This means that the line isn't actually two lines, but is only displayed that way so that the entire content can fit onto your screen.

Figure 7 shows a long line without soft-wrapping, where the text can't be displayed on my screen all at once:

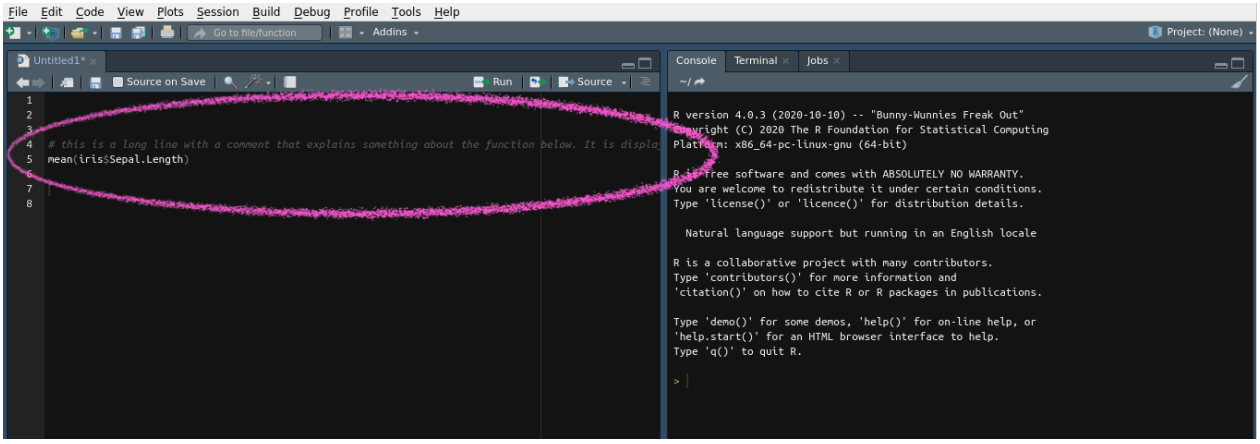

Figure 7: Line without soft-wrapping

Figure 8 on the other hand, shows the same line with soft-wrapping. As you can see, line number 4 on the left of the script now *stretching* over two lines.

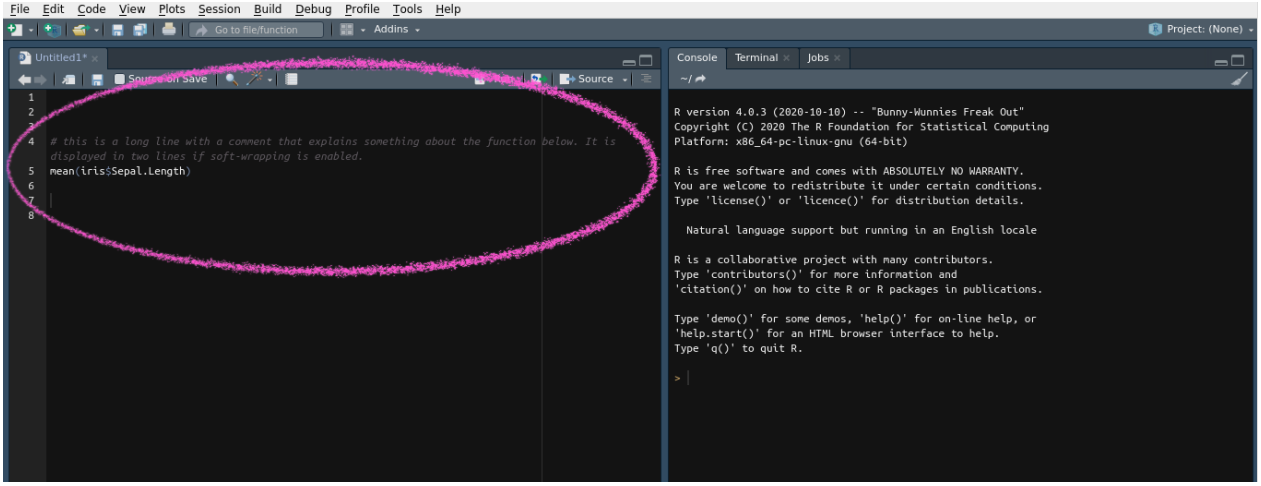

Figure 8: Line with soft-wrapping

What this means is that the comment-sign '#' is still part of the second line, and we do not have to manually write it every time our comments stretch over two lines.

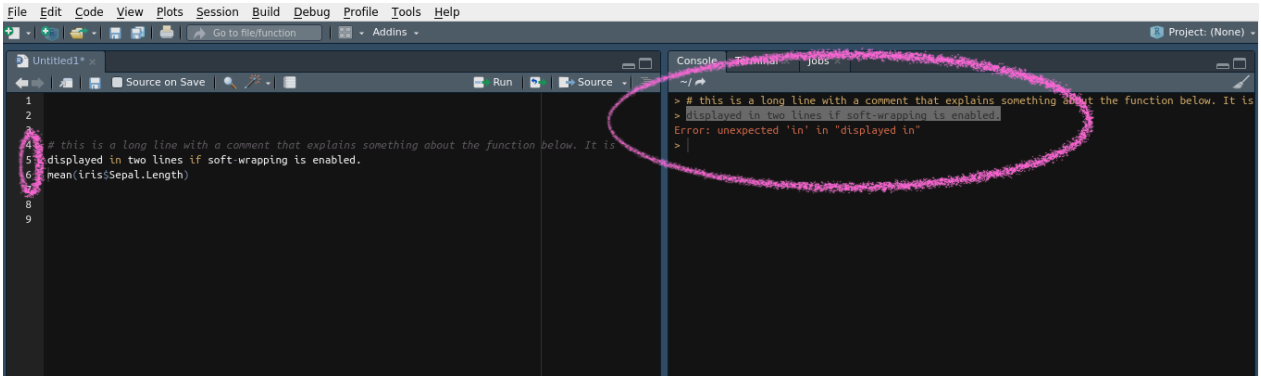

Figure 9: Potential error without soft-wrapping

Figure 9 shows how this setting can help to prevent a common error. '*If soft-wrapping is enabled.*' and you break a comment into two lines for visibility reasons, it might give you an error message, because your comment is now interpreted as code. Every comment demands a comment-sign '#' in the beginning of the line.

Line highlighting, margins and rainbow parenthesis

By highlighting the line that the cursor is on, navigation becomes easier. Two optional options – removing margins and using rainbow parenthesis – are also suggested below, but is less important.

Go to `Tools -> Global Options -> Code` under the `Display` tab. select 'Highlight selected line', marked with pink.

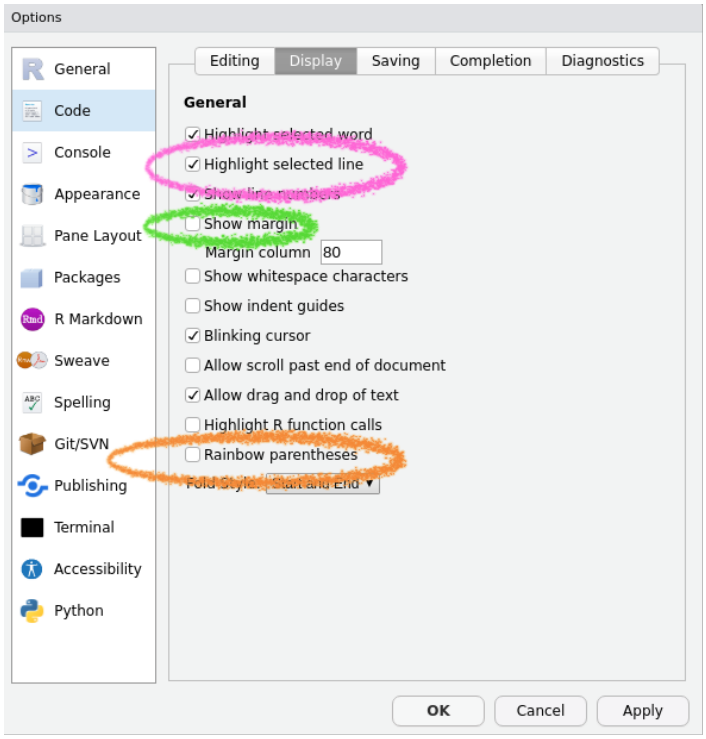

Figure 10: Setting display options

Even though soft-wrapping is useful, it is often considered bad practice to write too much on one line. That is why there is a tiny, dotted vertical line in the source-pane. It's quite distracting, and can be disabled with 'Show margin' (hightlighted in green). Although it is nice to disable, the advice about not writing too much on one line is good advice, try to follow it.

Also, rainbow parenthesis can be enabled (highlighted in orange). This means that parenthesis will be colored in pairs to improve visibility. This can make it easier to navigate in a nested function call (that is, a function where the output is immediately sent to another function, like in `mean(c(1,2,3))`).

Improvements

If you have any suggestions as to how to improve this guide, please reach out to: aga.ioa@cbs.dk. I would be more than happy to adapt the guide so it will be most helpful to you!