Introduction to RStudio

Introduction

RStudio is an interface to the statistical software R, extending upon the so called "base R" interface. When you open RStudio you will see three windows, to the left the console, to the upper right one with the environment (imported datasets and variables) and to the lower right one with files in your working directory. If you open a file this will extend to four windows as below in Figure 1.

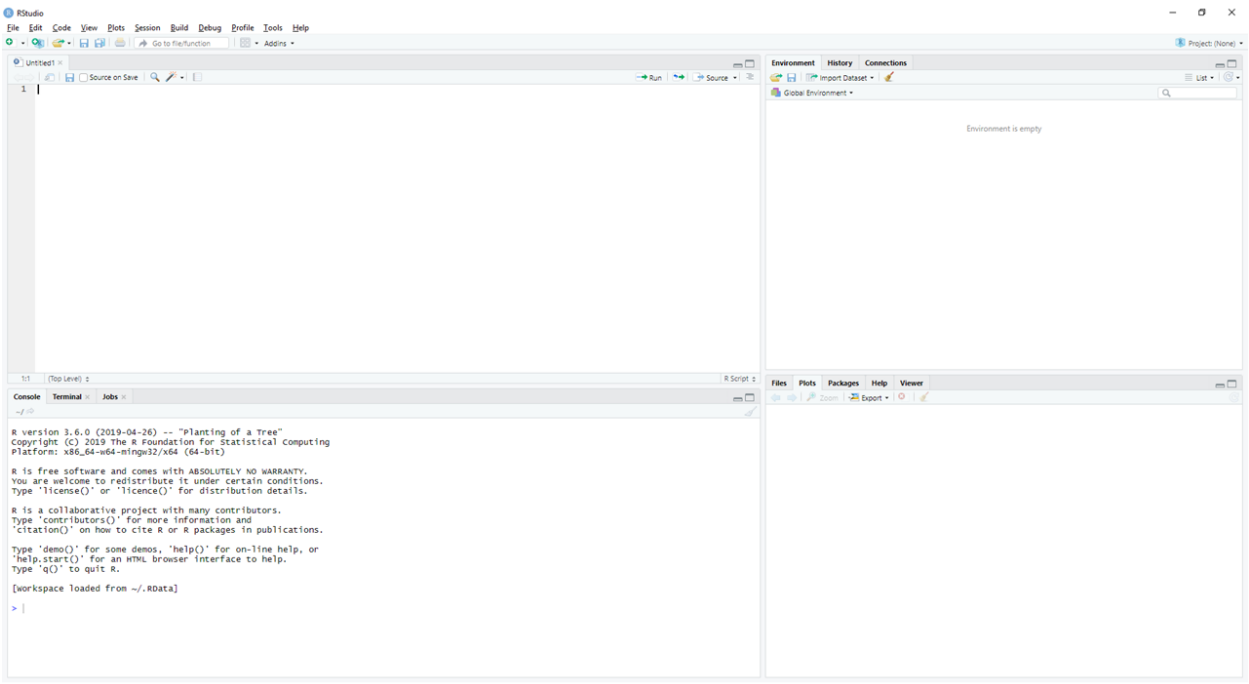

Figure 1: The opening window in RSTudio.

In these boxes there are also others tabs, which is more or less self-explantory, such as *Plots*, *Packages* (where you can install and update packages) in the lower window and *History* (where you can see which commands you have run) in the upper window.

Projects

In contrast to base R, RStudio is based on a concept called projects. The working directory is always tied to the location of the project. This means that files created in the project will be saved in the same location as the project. You do not have to specify the whole path,e.g. when reading a data set. texttt{"mydata.txt"} is enough, `"C:user/R files/data/mydata.txt"` is not needed, as long as the dataset is in the same folder as the project. You can create a new project by clicking on the R-symbol up to the top right, and select new project. If you use RMarkdown files, a project are automatically created.

Getting started

In R you can write the programming code in the console, but ususally you want to save your code while working and then you have several options of working with script files:

Figure 2: The opening window when you choose a R Markdown file.

- Regular R script files: You can open a new R script by clicking the empty sheet icon in the upper left corner (or choose File–> New File –> R Script). The keyboard shortcut for running a selected code chunk is Ctrl/Cmd+Return[1]
- R Markdown. There are two ways to do this, with some minor differences. Either a R Markdown notebook, or a R Markdown document. The R Markdown notebook allows for a quick preview of the document in HTML format with the content that is in the editor, whereas an R Markdown document needs to be compiled, in the same sense as a LaTeX document[2]. You can open a new R Markdown file by choosing the small arrow on the empty sheet icon in the upper left corner (or choose File –> New File –> R Markdown...). Chuncks of code are again run by Ctrl/Cmd+Return.

Exercise

1. Open a regular R Script file and write in the first line: $5 + 7$. With the curser in this line press Ctrl/Cmd+Return and see when results you get in the console window.
2. Open a R Markdown file. You will see that there is already som text there (Figure 1).

   In the file you have opened replace the middle line of the code marked in blue with 5+7. Place your cursor on the line with code and press Ctrl/Cmd+Return. Find the results in the window that opens under the code chunk in the script window. If you instead assign the result to a variable name you will not get any output. Instead the variable/value `res` will appear in the 'environment' window.

   ```
   res <- 5+7
   ```

   Plots will also appear in the window under the code. Test this by making a plot, e.g. for the annual Canadian lynx trappings 1821-1934, which is a dataset always available in R. The code must be written in the same or a new chunk that starts with ```` ```{r} ```` and ends with ```` ``` ````, like this:

[1] Find a list of Windows and MAC shortcuts here: https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts

[2] In fact, when compiling a R Markdown to pdf, it gets converted into a TeX file, which then is compiled into a pdf file. To compile R Markdown into pdf you need to have a TeX distribution installed, e.g. MiKTeX, This is usually available at SLU computers

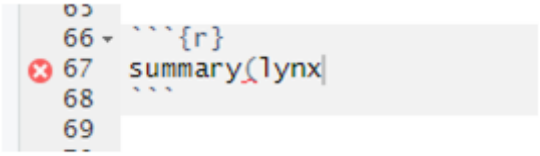

Figure 3: A paranthesis is missing in the code.

```
```{r}
plot(lynx)
```
```

Save your Markdownfile (.Rmd) and press the "Knit" button, which you can find on the top (just below the file names).

Continue with R Script or R Markdown

You can now choose if you want to proceed with R Script files or R Markdown files. Both work well for your analysis and it is a question of preferences. R Markdown files can be compiled into HTML/PDF/DOCX regardless of whether it is a R Markdown notebook or document and gives you better possibilities to include comments and other information (Pictures from files, formulas) in the same file[3]. R Script files are easier to handle if you only want to save your code. When using R Script files make sure to also specify a project to read from and write to your current project folder.

Read file and work with code

Datasets, data frames or tibbles, are created by combining several rows and columns of data. Usually you will read data to R from a file (e.g. by `read_csv` or \texttt{read_delim}). It can also be convenient to use RStudios *Import Dataset* option, which you can find under File –> Import Dataset.

Code is written in the R script or in the code chunks of R Markdown. RStudio will give you hints if some of your code is wrong (Figure 2).

[3]All our R teaching material is written in R Markdown