## 2.2 Compile an R Markdown document

The usual way to compile an R Markdown document is to click the `Knit` button as shown in Figure 2.1, and the corresponding keyboard shortcut is `Ctrl + Shift + K` ( `Cmd + Shift + K` on macOS). Under the hood, RStudio calls the function `rmarkdown::render()` to render the document *in a new R session*. Please note the emphasis here, which often confuses R Markdown users. Rendering an Rmd document in a new R session means that *none of the objects in your current R session (e.g., those you created in your R console) are available to that session*.[2] Reproducibility is the main reason that RStudio uses a new R session to render your Rmd documents: in most cases, you may want your documents to continue to work the next time you open R, or in other people's computing environments. See this StackOverflow answer if you want to know more.

If you must render a document in the current R session, you can also call `rmarkdown::render()` by yourself, and pass the path of the Rmd file to this function. The second argument of this function is the output format, which defaults to the first output format you specify in the YAML metadata (if it is missing, the default is `html_document` ). When you have multiple output formats in the metadata, and do not want to use the first one, you can specify the one you want in the second argument, e.g., for an Rmd document `foo.Rmd` with the metadata:

```
output:
  html_document:
    toc: true
  pdf_document:
    keep_tex: true
```

You can render it to PDF via:

```
rmarkdown::render('foo.Rmd', 'pdf_document')
```

The function call gives you much more freedom (e.g., you can generate a series of reports in a loop), but you should bear reproducibility in mind when you render documents this way. Of course, you can start a new and clean R session by yourself, and call `rmarkdown::render()` in that session. As long as you do not manually interact with that session (e.g., manually creating variables in the R console), your reports should be reproducible.

Another main way to work with Rmd documents is the R Markdown Notebooks, which will be introduced in Section 3.2. With notebooks, you can run code chunks individually and see results right inside the RStudio editor. This is a convenient way to interact or experiment with code in an Rmd document, because you do not have to compile the whole document. Without using the notebooks, you can still partially execute code chunks, but the execution only occurs in the R console, and the notebook interface presents results of code chunks right beneath the chunks in the editor, which can be a great advantage. Again, for the sake of reproducibility, you will need to compile the whole document eventually in a clean environment.

Lastly, I want to mention an "unofficial" way to compile Rmd documents: the function `xaringan::inf_mr()`, or equivalently, the RStudio addin "Infinite Moon Reader." Obviously, this requires you to install the **xaringan** package (Xie 2022d), which is available on CRAN. The main advantage of this way is LiveReload: a technology that enables you to live preview the output as soon as you save the source document, and you do not need to hit the `Knit` button. The other advantage is that it compiles the Rmd document *in the current R session*, which may or may not be what you desire. Note that this method only works for Rmd documents that output to HTML, including HTML documents and presentations.

A few R Markdown extension packages, such as **bookdown** and **blogdown**, have their own way of compiling documents, and we will introduce them later.

Note that it is also possible to render a series of reports instead of single one from a single R Markdown source document. You can parameterize an R Markdown document, and generate different reports using different parameters. See Chapter 15 for details.

## References

———. 2022d. *Xaringan: Presentation Ninja*. https://github.com/yihui/xaringan.

[2] This is not strictly true, but mostly true. You may save objects in your current R session to a file, e.g., `.RData`, and load it in a new R session.↩