IV. R Markdown (exercises)

Data Science Lab, University of Copenhagen

17 March, 2021

R Markdown is a file format for authoring dynamic reports that integrate code, plots, results, and text in fully reproducible manner. Using R Markdown with Rstudio enables you to

- Save and execute code
- Generate high quality reports to share (either to future you, or collaborators)
- Tightly combine analysis with interpretation

Using the **knitr** package, an R Markdown file can be compiled into pdf, html, and even word documents. All the html and pdf documents you have seen in this workshop (even this document) have been generated using the R Markdown format.
In this exercise, we will get you started with R Markdown and introduce some of its many features.

Need some help? There are tons of resources online for learning R Markdown, but in particular we can recommend:

- Rstudios own **rmarkdown** cheatsheet is a very handy reference guide!
- The chapter on R Markdown in *R for Data Science*.
- *R Markdown: The Definitive Guide* by Yihui Xie (author of the **knitr** package). Is available for free here. It is a bit technical, but it is really a treasure trove of information.

Opening your first R Markdown file

1. In order to start playing around with R Markdown files, you need to make sure that you have both the **rmarkdown** and **knitr** package installed. This can be done through the *Packages* menu in Rstudio, or by running the command `install.packages("rmarkdown")` and `install.packages("knitr")`.

2. Open a new R Markdown document (*File* → *New File* → *R Markdown*). If you are asked to install some extra packages, then answer "yes". You are asked for a title; just write "test document" or whatever you please. Choose html as output format. Save the document; use file extension .Rmd, if you don't let the computer choose it for you. As you might realize from the R Markdown template you have open, an R Markdown file is just plain text file (like a `.txt` file), but uses the `.Rmd` extension and has some special syntax.

3. Verify that you can knit the document with the example content: Click on the *Knit* button in the console toolbar (the one with the blue yarn and knitting needles), and an example output file should be generated.

4. Edit the title of the markdown document to *My first Markdown document*, and click *Knit* again. Notice how the title of the output document changes. Also, note that, as soon as the output html file is generated, you can view it with any browser (without starting RStudio). Your work is therefore easily shared with collaborators that don't use R themselves.

   Looking at the template R Markdown, there are three important types of content:

   1. An (optional) **YAML header** surrounded by `---`s.
   2. **Chunks** of R code surrounded by ```` ``` ````.

3. Text mixed with simple text formatting like `# heading`, `_italics_`, and `__bold__`.

Try to play around with the formatting. What happens if you write `## heading`? And how do you make a piece of text both bold *and* italicized?

Putting R code in code chunks

5. Code chunks are the main reason that R Markdown is smart! Delete the example content in the markdown document from the first header and down. Insert a new R chunk by clicking on the *Insert* button in the console toolbar (the button with a white *C* in a green square), and choosing *R*. Then, type for example

```
3+2
log10(0.001)
z <- 17
```

inside the chunk. Don't knit yet! Instead: What happens when you click the little green "play" button to the right of the chunk?

1. You can also run each line one at a time with *Ctrl/Cmd + Enter*. Try it.

   As you may already have noticed, it takes a few seconds to knit your document, even if it is very small. It is therefore recommended that you **run your R commands/chunks without knitting until the code works as expected**. You just saw how to do that.

2. Knit the document, and notice what the R chunk has now generated.

3. Click the small cogwheel next to the R chunk, and select *Show output only* in the dropdown menu denoted *Output*. Knit the document, and notice what happens in the output. Obviously, the different options can be used to control the output from an R chunk. You may switch back to *Use document defaults* again.

4. Add the text *Some simple computations in R* above the chunk, and knit the markdown document again.

5. Insert a chunk with the following code, knit, and see what happens:

```
x <- c(1,2,3,4)
y <- c(2,6,3,1)
plot(y ~ x)
```

Basic text formatting

11. Insert a line of text, starting with two pound signs, e.g. `## My Header`. Knit, and see what happens. What happens if you change it to `# My Header` or `### My Header`?

12. Go to a new line and write something, then add underscores around the text like so `_some text_`. What do the underscores do?

13. Let's make a itemized list in markdown. You do this with `*` followed by a space, one star and space per item in a new line. Also try to use tab (indent) followed by `*` space and text.

14. Stars and underscores are essential in R Markdown, you already know that one star followed by a space makes a list item, but try out some other uses of the star as shown below:

    - `*Here is what one star does to text*`
    - `**Here is what two stars do to text**`
    - `***Here is what three stars do to text***`

Adding external images

15. In one of the previous example, we saw that if include R code that produces a plot, then that plot is automatically included in the output document. Sometimes we also want to add figures or images that are not generated by R. Such external figures can be included either with the Markdown command `![caption](file_path)` or the R command `knitr::include_graphics("file_path")`. The latter provides more control in terms of specifying e.g. figure width and height. The file path can be a path to a file on your computer or a URL to a picture found online. Try including the image `ExceltoR.jpeg` that can be found in the "Exercises" folder. Remember to put it in a chunk!

16. If you use method two, you can by adding the two arguments; `fig.cap="A caption", out.width = '60%'` to the chunk (inside the curly brackets), give your figure a title and control the size of it. Try adding these arguments and see what changes in the document when you knit it.

Other output formats

17. Notice the line `output: html_document` in the beginning of the Rmd file. It implies that output is written to a html-file. If you have MS Word installed, then it is easy to have the output generated as a docx-file instead: Simply write `word_document` instead of `html_document`. Try it!

18. Optional: if you prefer pdf-output, you can write `pdf_document` which uses **LaTeX** to generate a pdf-file.

    - **NOTE**, if you work locally on your laptop, then you need to install a *LaTeX* distribution on your computer: commonly *MikTex* for Windows and *MacTeX* for MacOS. Alternatively, there is a light-weight distribution called `tinytex` that works on cross-platform and can be installed from R (`install.packages("tinytex")`). If you knit without having *LaTeX* installed, then you probably get an error message which includes links for installation of LaTeX. We will **not** knit to pdf now, but you can try it at home when you have *LaTex* installed.

Tables

19. As with images, there are several ways to present data in tables in R Markdown. The most common approach is to use the `kable()` function from **knitr**. Insert at chunk with the following code and knit!

```
knitr::kable(head(iris, 5))
```

1. Markdown also has specific syntax for manually constructing tables. Try copying the following table and insert into your R Markdown document (**NB**, *not* as a code chunk, just as "text")

```
|                  |  Sample 1| Sample 2| Sample 3|
|:-----------------|---------:|--------:|--------:|
|Gene A            | 100.0    |     16.0|     0.12|
|Gene B            | 107.0    |     15.0|     0.45|
|Gene C            | 122.8    |     10.0|     0.08|
```

Various tips and tricks

In this session you have had a whirlwind tour of R Markdown. Even though we've only scratched the surface, we still hope that you've by now realized that R Markdown is a great tool. We encourage you to keep learning, and luckily there are many resources available online for further self-study. To end this exercise, we have here collected a series of "tips and tricks" that we have found helpful.

- Don't knit all the time. Instead, run code lines and R chunks without knitting while you are figuring out how they should be.
- Do not put all your R commands into one big R chunk. Instead, split it into well-defined smaller chunks, which you may even give names.

- The code in Rmd-file must be self-contained in the sense that you cannot use datasets (or other objects) that you have imported "outside" the Rmd-file. You therefore have to include the commands for data import in the file.
- If no output is generated, then read the error message. It is not always easy to read for a beginner, but at least you are informed where the problem occurs.
- It is possible to "cache" R chunks such that the commands are not rerun every time you knit. This is a nice feature if you have time consuming computations.
- As always: Make sure to organize your file in an appropriate way, save the file often, and make sure to save only relevant commands (not all the stuff you played around with at preliminary stages).
- You can open the files in other programs than RStudio, e.g. your favourite browser for html-files or your favourite pdf viewer for pdf-files.