R Markdown Presentation Script

Giulia Grisot

22 February 2021

Part I

Hi and welcome to the first part of today's workshop. My name is Giulia and I will introduce you to R Markdown. I will then show you how to create reports and documents with it.

R Markdown is a tool we use to create *efficient reports*, to *summarize analyses* and *communicate results* to an audience.

R Markdown is especially indicated if you are dealing with datasets and plots, but you can also think of it as a "zero fuss" alternative to word processors. While these generally have a "friendlier" interface, and in certain ways more flexibility, they also often oblige you to waste quite a lot of time on the format adjustments. With R Markdown, once you define the "rules" you just have to decide how you want to organise the hierarchies within your paper, and simply write plain text. The output will come perfectly formatted.

With R Markdown, using only R code, you can create several kinds of documents, including HTML and PDFs, and even presentations. Most importantly, R Markdown ensures that the *results are reproducible.*

Consider the following scenario:

- You've conducted your analyses and created a paper draft where you demonstrate your findings.
- Then you send the paper to a journal and, after a few months, the reviewers request some modifications
- However, let's imagine that, by then, you forgot the exact steps you went through to get to those early results, or that in the meantime you updated your data files. How are you going to do?
- By generating the results and report from code with R Markdown, you *can always reproduce your results, and easily update new data in the workflow.*

R Markdown elements

An R Markdown document is made of three components: *metadata*, *text* and *code*.

YAML

At the top of the file, you can see the *YAML header*, containing the file metadata. It is enclosed between two strings of made of three hashes each. The YAML determines the properties of your markdown output, and depending on your desired format, you can add numerous features to your document, such as weather you want a table of content, whether you want to link the paper to a bib file to add references automatically, and so on.

In this session, we will not focus on the YAML in depth, and we will use a basic syntax that accommodates simple reports. If you want to personalize your YAML more, several tutorials can be found online. I will provide a few links at the end of the session.

One thing that you need to know, however, is that the YAML is sensitive to spaces, tabs and double quotes. Whenever you decide to add some features, pay attention to how these are structured within the YAML.

Text body

The YAML is followed by the content that make up your report, the actual body of your document. This can includes both text and code chunks.

Within a report, we can include a discussion of the data that is presented by adding text.

Headers

Including headers in the report will help clearly label each section. We add headings hashes, followed by and a single space before the text. The more hashes we use, the smaller the header will be.

We can also include sentences to add additional detail to the report. Here, we add a sentence to describe a dataset. Remember that the added sentences have to be included after the heading but before the code chunk, so that when we knit the report, the reader will see the heading, the dataset description, and the code in the intended order.

Text formatting There are many options for formatting text in a markdown file, including making the text bold, italicized, or strikethrough, as long as bulleted and numerated lists. To bold the text, we surround it with two sets of asterisks or underscores. To italicize it, we use one set of asterisks or a single set of underscores, and to strikethrough text, we surround the text with two sets of tildes.

We can also add inline code to the text to clarify that we are referring to object names from the code. To do this, we surround the text with a single set of backticks.

Images and links We can also add images by placing ethe image caption in square brackets preceded by an exclamation mark, and followed by the link to local files or images from the web in parentheses.

Similarly, we can add links to website by placing the link text in square brackets followed by the link in parentheses.

Note that there are no spaces between the square brackets and the parentheses.

Knit

After we have finished modifying a file, we might want to see it in a more legible form, which can be shared with others. Knitting a file is how we generate an output file from the R markdown file.

When a file is knit, R Markdown transforms the document into the desired output file, reading the text and running the chucks of code.

For example, when the R Markdown file shown on the left is knit, it will create the output shown on the right

Practice time

Following the step by step instructions on the Rmarkdown practice file prepared for you, you will build a report using an open source dataset of Student Exams Performance from Keggle. This contains the exam results of a 1000 students, along with their gender, ethnical group and parents level of education.

We want to create a report to show to an audience whether gender has an effect on how the students perform in several subjects.

Part II

Tables

Markdown tables

You can dd a table manually very easily using dashes to separate the heading row from the content ones, and pipes to separate columns. With colons you can indicate the alignment of the content, and the width of the column can be modified “visually” by adding dashes to the separating line. Writing a table manually allows you to write tables independently of the content of your data. As you might notice from the picture, the editing can get confusing, especially if you have longer cell content. It is important to remember that the structure is defined by the separating line, and that the content does not alter it.

Kable tables

Alternatively you can add tables to the report using the kable() function from the knitr package. The knitr package is what runs each code chunk and knits the document. There are a number of options to customize the table, but it isn’t possible to format the data within the table to perform tasks like combining cells. These data wrangling tasks should be done beforehand. For example, we create here a summary table of the average scores in math and reading for female and male students, and then we render it as a table with kable. Kable makes it easy to change column names and add captions.

Don’t be skeptical about what you see in the preview, it will be rendered nicely when you knit.

Adding references

As an academic, one of the most important feature for a report might the managing of references. If you want to add references to a rmarkdown document, the easiest way is to connect the document to a bib file containing your references, which can then be “called” in your document. You can easily produce bib files from almost any reference manager software, such as Mendeley and Zotero.

Once you have you “library.bib” file in the working directory, you have to connect the bibliography to the rmarkdown by adding a line to the YAML. Notice that you can decide to have or not have links for the citations.

The knitting process with deal with the formatting, as well as with the creation of a lovely bibliography at the end. If you need a specific referencing system, there are ways to achieve it. They are not too complicated and there are plenty of instructions online. For this tutorial, we will stick to the basic one provided automatically by rmarkdown.

To add references to the text is really easy, just add the reference ID preceded by the sign @ You can find the IDs in the bib file.