Getting started with Rmarkdown

Kath Sherratt

2024-02-20

R Markdown can be used for combining code with text to author complete HTML, PDF, or MS Word documents. This is a quick tutorial to get started with R Markdown, with a Resources section at the end to carry on learning more. For practise, we'll use the built-in `cars` dataset (type `?cars` to find out more about it).

Creating a document

We can create a new R Markdown document within RStudio from File > New File > R Markdown.

To create output from an R Markdown document, click the **Knit** button at the top of the RStudio pane. This will create and save a document that includes both the content as well as the output of any executable R code within the document. This process of formatting text and executing code is called "rendering" the document.

> **Try it out:** Render this document by pressing the "Knit" button.

Writing and formatting text

Like a notepad, we can write text directly into the Rmarkdown document. There are two ways to format text (e.g., create headers, lists, emphasis):

1. Use the Markdown formatting syntax. Markdown is a simple formatting language for formatting text. For example, create headers with "#", emphasise text in bold or italics by surrounding text with "**". For advanced formatting (e.g., equations, citations), we can use LaTeX syntax.
2. Use the RStudio Visual editor. This provides point-and-click formatting similar to MS Word. At the top left of the pane, we see a button to switch from "Source" (the underlying markdown-formatted text) to "Visual".

> **Try it out:** Switch between "Source" and "Visual" views to compare how the markdown text in this document is visually formatted.

Adding code

You can write R code within an Rmarkdown document using "chunks". A chunk is the equivalent of a script that's embedded within the Rmarkdown document. Everything within a chunk is assumed to be code, and will be executed when the document is rendered. Chunks of code can do anything a script can do: process data, contain statistical analysis, create tables or plots.

Creating a chunk

1. Create a new chunk for R code using the green Insert button in RStudio, or manually by typing in triple backticks + {r } at the start followed by triple backticks at the end of the chunk.
2. We also need to give each chunk a unique name, just as you would save a script with a unique filename. This goes in the chunk header ("{r name}").
3. We can then execute chunks of code as needed by using the green play button at the top right of each chunk.

```r
# Just as in a script, this is a comment within a code chunk
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

```r
# Code in chunks operates as usual within the RStudio. For example, outputs are saved to the global env
cor_speed_dist <- cor(cars$speed, cars$dist)
```

> **Try it out:** Edit this document to create a new chunk in the space below. Then insert a line of code to print the first rows of the `cars` dataset (try `head(cars)`). Try executing the chunk within your RStudio session by using the "play" button.

Chunk settings

By default, when rendering (publishing) an Rmarkdown, everything you see in the source document will appear in the output. However, depending on what you're writing, you may or may not want to present the code itself alongside the text in the published output of your Rmarkdown. We can control how chunks of code and their outputs are presented by changing settings in the heading of each code chunk (i.e. {r chunk-name, settings...}).

- `echo = FALSE` prevents printing the R code in the published document
- `include = FALSE` prevents printing any output of the chunk; the code will still be executed, but its output won't visibly appear
- `eval = FALSE` prevents the chunk from being evaluated (executed) at all

> **Try it out:** Add one of these settings in the header of chunk you created above (tip: include a comma after the chunk name). Then Knit the document again to see how the output changed.

Embedding code within text

Code doesn't have to be kept within a chunk; we can also include short pieces of code directly within the text as it's written. Include code in-line with:

- In the Visual editor view, press the "</>" icon in the formatting options bar

- In the Source editor view, manually writing: ‘ᵒ.. ‘.

This is great for creating dynamic reports with paragraphs of results that automatically update. For example, in-line code can print a summary number or the output of a calculation performed in a previous code chunk. (Example: in the first chunk above, we saved a correlation between cars' speed and stopping distance, and our result was a 0.81 correlation coefficient.)

> **Try it out:** Complete the example sentence below by adding code to print the number of records in the `cars` data. Knit the document to check the output.

The average car stopping distance was `r` ft, tested among `r` cars.

Including plots and tables

Just as with R scripts, we can create plots and tables within an R chunk. The output will then be printed directly in the document. There are also plenty of packages and formatting options to create good-looking output.

For example, here's a standard plot in base R:

```
# Plot the "cars" dataset. Note I've specified the figure dimensions and caption in the chunk header.
plot(cars)
```

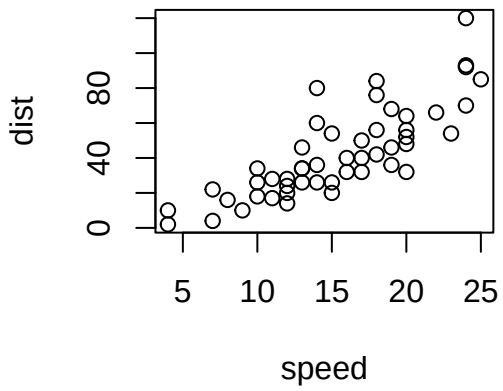

Figure 1: Cars' speed and stopping distance

We can quickly create formatted tables using `knitr::kable()`:

```
# Create a table summarising the "cars" dataset
knitr::kable(summary(cars), caption = "Summary of speeds and stopping distances")
```

Table 1: Summary of speeds and stopping distances

speed	dist
Min. : 4.0	Min. : 2.00
1st Qu.:12.0	1st Qu.: 26.00
Median :15.0	Median : 36.00
Mean :15.4	Mean : 42.98
3rd Qu.:19.0	3rd Qu.: 56.00
Max. :25.0	Max. :120.00

Rendering output

Choose the desired output format (e.g., PDF, HTML, Word) by specifying the output format in the YAML header at the very start of the Rmarkdown document.

> **Try it out:** Change how this document is rendered to create a webpage. Currently the 'output' option is set to 'pdf_document'. Replace this with 'html_document'. Knit the document and check the webpage.

Resources

Getting started

- An overview of R Markdown: http://rmarkdown.rstudio.com
- The complete guide to using R Markdown: https://bookdown.org/yihui/rmarkdown-cookbook/
- Get started with writing in Markdown syntax: https://www.markdownguide.org/cheat-sheet/

Taking R Markdown further

- Include citations within text and an automated bibliography: https://rstudio.github.io/visual-markdown-editing/citations.html
- Use the `trackdown` package to integrate R Markdown with Google Drive for easier authorship collaboration: https://claudiozandonella.github.io/trackdown/
- Render reports multiple times with different options, using `parameters`: https://bookdown.org/yihui/rmarkdown-cookbook/parameterized-reports.html