Quantitative Understanding in Biology
R Markdown

Luce Skrabanek, Jason Banfelder

September 19th, 2023

1 Keeping your code

Up to this point, we have been using the interactive console panel of RStudio. We have seen that the history panel keeps a record of what we have done, but what would happen if we quit RStudio and did not save the .Rdata? The objects in our environment would disappear, and we would have to trawl through the command history, remember those commands that worked, and re-run them to recreate our environment.

There are two ways in which we can keep track of which commands we want to run to re-generate our analyses, both of which use the Source panel.

1.1 R scripts

A best practice is to store R commands in a script file. Choose *File ⇒ New File ⇒ R Script*, or from the green cross icon, choose R Script. An empty tab should appear, which you can populate with R commands, exactly as you would type them in the Console panel. You can use the Console panel to engineer the command to do exactly what you want, then copy it to the Source window. An easy way to do this is to highlight the command in the History panel and click the 'To Source' button.

From the Source panel, you can run a single line (that the cursor is on), or a selection of highlighted lines (Mac users press cmd-return). Clicking the Source button will run all the commands in the script, from start to finish. Note that nothing will be printed to the Console window unless explicitly declared, or you use the 'Source With Echo' option, which will print both the command and any output.

A good practice is to periodically empty your environment (but not your history - you might need it!), and try to re-create it using just the script. This will let you know if you have inadvertently missed adding a crucial command!

1.2 R Markdown

This is a way of combining not just the code, but also notes and comments about the code and project. It can be thought of as akin to a lab notebook, which you can use to capture scientific ideas and the associated analysis results and communicate these with colleagues.

To start a new R Markdown document, choose *File ⇒ New File ⇒ R Markdown*, or select R Markdown from the green cross icon. You will be asked to fill in some basic metadata (name of document and author) and what you want the default output format to be.

There are three parts to an R Markdown document:

1. a YAML header, surrounded by triple dashes, which is automatically generated for you when you open a new document

   ```
   ---
   title: "Some title"
   author: "Luce"
   output: html_document
   ---
   ```

2. the chunks of R code that will be run, surrounded by triple backticks (```)

   ```
   ``` {r optional_name, optional_options}
   R code here
   ```
   ```

3. text, with Markdown formatting.

To produce a complete report including code and text, click the Knit button. If you haven't already saved the file, it will ask you to do so now. Try this now, with the default content given to you on creation.

By default, an HTML file will be produced, with a preview either in the Preview panel, or externally (which preference can be set from the Options menu (the cog icon)). The HTML file can be sent to colleagues or your PI, and viewed in a browser. Other output options include PDF and Word, although you will need to install separate packages for that functionality. We will talk more about external packages in a later section.

Each R code chunk can be run separately. The output from each chunk can include both the code and the results (use `echo=TRUE` as a chunk option) or just the results (use `echo=FALSE`). The output, including figures, may be shown inline (in the .Rmd file), but many people find this distracting, and prefer to see the output in the Console and Plot panels, which can be set from the Options menu (select Chunk Output in Console).

It is also possible to include mini-code chunks inline, surrounded by single backticks, and prefixed by the letter `r`. These will get evaluated, and get filled in, whenever the document is knitted.

The formatting of the text is accomplished with standard Markdown formatting. Examples include:

- Different level headings are prefixed by varying numbers of `#`.

- Italic or bold text is surrounded by one or two `*`, respectively,

- Bulleted list items are prefixed by `*`,

- Numbered list items are prefixed by numbers.

There are a number of guides to help you construct your R Markdown documents directly from within RStudio:

- an external R Markdown cheatsheet PDF, which lists all the most commonly used commands and syntax, accessed from *Help ⇒ Cheatsheets ⇒ R Markdown Cheat Sheet*,

- an external markdown reference guide PDF, accessed from *Help ⇒ Cheatsheets ⇒ R Markdown Reference Guide*, also found inline at *Help ⇒ Markdown Quick Reference*.