Using RMarkdown for reproducible research

BIO8068 Data visualisation in Ecology

# 1. Reproducible research

Ideally you want to be able to create reproducible research, so that it can be undertaken by someone else, following your instructions, in the same way as you would a report of a laboratory experiment or field survey. In the latter two examples, your “Methods” section will usually have a description of how you collected the data, and if someone else were to follow the identical set of methods they should obtain roughly the same results

In ecological data analysis, if you are writing your own documents, you want to ensure that the text in it accords exactly with the data. This is why it is best to clean up the raw data in R, rather than editing it in Excel. RStudio projects greatly facilitate good practice, for example in an RStudio project folder I often have separate sub-folders for

* data (raw data)
* figs (usually ggplot figs or maps generated by R)
* R ( if it is a very complex set of analyses, I might put R functions into their own sub-folder, and use the source function at the start of a main.R script in the project folder itself)

Depending on the complexity of the project, you might want output folders for Word documents or processed data. Until a few years ago, my typical working pattern was to:

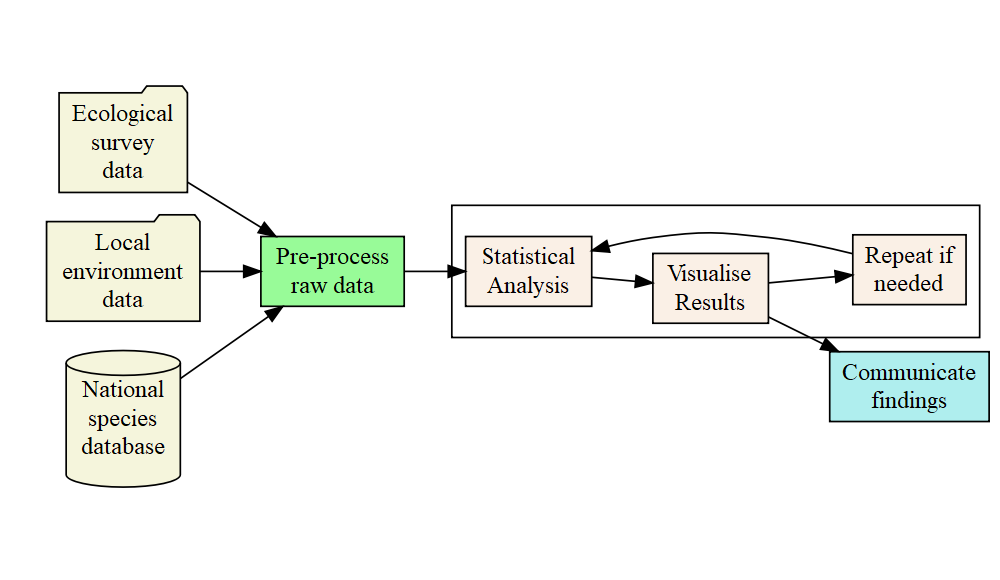
* Collect the data
* Import and analyse the data in R
* Copy / paste / export graphs, tables, numbers from R into the Microsoft Word document.

The main problem with this approach was that regularly I would accidentally miss an update from the first to last steps. For example, if analysing species distribution records, I might receive an updated spreadsheet with a few extra records. I’d then re-analyse the data in R, and copy the new graphs into Microsoft Word. The problem was that, especially when changes to the original raw data were fairly subtle, that it was easy to forget to make the corresponding edits to figures, tables or numbers lurking in a paragraph. Gradually the Word document was wrong, but would take ages to correct.

You also want a system whereby anyone can take your raw data, analyse it according to your instructions, and obtain the **identical** results, including statistical analyses, graphs, maps, tables. This is true reproducible research. Many ecological journals are now demanding this level of quality. You can do all this in R.

# 2. Typical workflow

In many ecological surveys you may collect data in the field (or be sent it if it is raining!), analyse it iteratively until it makes sense, and report it. There are also “intermediate” steps:



The step in green **Pre-process raw data** also known as “data munging” takes a long time, as you will already have discovered from other modules. Indeed, some ecologists have estimated this takes up to 50 to 90% of the whole lifecycle of a piece of work. The “interesting” bit is in the box: trying to understand what is going on the natural world based on the fun data you have collected. The final step can be hard work, writing reports etc. to **Communicate findings** to policy-makers, the press, other scientists and the wider world. These communications can be in the form of PowerPoint presentations, pdf files, Word documents, interactive websites etc.

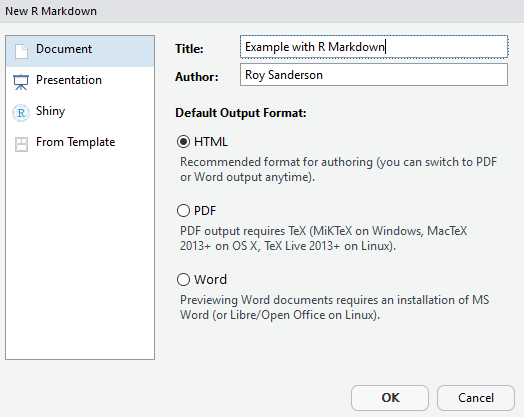
At this point you might be wondering how to create Word documents. The Rmarkdown and knitr packages provide a powerful method of writing text in R, and creating high-quality documents. Indeed, this practical schedule is written in Rmarkdown, irrespective of whether you are reading it in a Microsoft Word document, or an embedded HTML page in Canvas.

# 3. Installing and getting started with RMarkdown

Begin by installing the rmarkdown package:

# Install rmarkdown from CRAN  
install.packages("rmarkdown")

Now, instead of a standard R script, you are going to create an RMarkdown script. Click on File -> New File and select the R Markdown option. A setup screen will appear for you to enter details about your document:



A new Untitled1 file will be created, that is already pre-populated with some text. Save the file in your project, giving it a name ending in .Rmd to indicate that it is markdown format. Click on the knit button at the top of the RMarkdown file and it will be instantly converted into HTML format for viewing. If you want a Word document, that is also available via the down-arrow next to the Knit button. You may have to open Word first, but usually it should display automatically. When preparing documents I tend to work mainly in HTML for previewing the output, as it is slightly quicker to Knit together.

The example script already shows you a lot of features of RMarkdown. You can embed R code, decide whether or not to display the R code, or just the output, decide whether to execute the code, etc. You also have control on text size, can create equations, and so on. For example \_italics if prefix or suffix with an underscore or one asterisk\_ will display *italics if prefix or suffix with an underscore or one asterisk* whilst \*\*bold if prefix or suffix with two asterisks\*\* will give **bold if prefix or suffix with asterisks**. Lines that begin with a # symbol control the headings, sub-headings etc., with the more # symbols the lower the subheading. Therefore

## This is a level 2 heading gives

## This is a level 2 heading

whilst ### This is a level 3 heading gives

### This is a level 3 heading

Look at the code “chunks” in the example markdown document. Notice how sets of R commands begin and end with three backslash symbols. The buttons on the right allow you to execute the code in all preceding chunks, or just the one you are working on.

*Exercise* Copy and paste **small** sections of R code from an R analysis script that you used in e.g. BIO8075 for some of the meta-analysis examples into your new RMarkdown document. Also remember to copy over any CSV data files you used. Cross refer back to my practical schedules in BIO8075 and try and write a simple narrative about what you are doing. You will then have a complete document that is reproducible, and shows the results of your analyses.

There is a lot of useful online information on RMarkdown, including a ‘cheat sheet’ (available from RStudio Help menu), the main website <https://rmarkdown.rstudio.com> and an online book <https://bookdown.org/yihui/rmarkdown/>. The latter was only published in 2019 but constantly updated, so this is a fast-moving field. You can even include references and bibliographies in the system.

# 4. Editing existing RMarkdown files that you have already seen

Finally, this practical schedule was not written in Microsoft Word, it was written in RMarkdown! As it is hosted on Github, you can download this document and edit and play with it yourself. This will also improve your skills. If you want to clone the RMarkdown file, and all the associated documents:

* Save all your existing files, and if using git version control, commit and push all changes
* Close your project
* Start a New Project, selecting the choice of **Version Control** as to where you are getting the document
* When asked for the URL, use: git@github.com:rasanderson/BIO8068\_reproducible.git

You can view the files at <https://github.com/rasanderson/BIO8068_reproducible.git> As this is a public repository, and you are pulling rather than pushing changes, it should not prompt you for a username or password. If you look in the top level, you will see an R\_and\_markdown.Rmd file which is used to create this document, as well as the reproducible.Rmd for the git and GitHub practicals.

Begin by seeing if you can recreate (in HTML) this document. Before you can do so, install the DiagrammeR package which was used to create the flowchart. You will also need to install and initialise the webshot package:

install.packages("DiagrammeR")  
install.packages("webshot")  
webshot::install\_phantomjs()

In case you’re wondering, the webshot package was needed as by default DiagrammeR flowcharts are only available in HTML, not Microsoft Word, and the webshot package does an auto-convert if needed into a simple image for Microsoft Word.

Now that you have the repository locally, start to play around with editing settings, headings, lines in either this practical, or the git practical, to get a good grasp of the capabilities.