R Markdown Cookbook

Michael Harper, Yihui Xie

2018-09-24

Contents

Preface

> This book is in a **very** early stage of development. If you have any suggestions on what should be included within this book, please get in touch via GitHub

R Markdown is a powerful tool for combining analysis and reporting into the same document. Since the development of the **rmarkdown** package (Allaire et al., 2018), it has grown to become a diverse ecosystem of code, and reports, books and websites can all easily be generated directly from R code.

There is a wealth of guidance which has grown over the past few years, and the book *R Markdown: The Definitive Guide* (Yihui Xie, 2018) provides an overview of all that can be done with R Markdown. However, as noted by Yihui, it was commented by the publisher that it would be beneficial to provide more practical examples of the use of R Markdown. And so, the idea of this book was born.

To fill the gap for official documentation, users often seek for help on StackOverflow. At the time of writing, there were almost 4,000 different questions with the `r-markdown` tag. However, the use of the website is relatively difficult if you do not have a specific problem you are trying to search for, and therefore it is hard to be able to tell what is possible with R Markdown unless you have the insight to search for a specific question. This book aims to draw together much existing literature from StackOverflow, and provide up-to-date solutions for everyday queries that users commonly face.

This book is designed to provide a range of examples of how to extend the functionality of your R Markdown documents. As a cookbook, this guide is recommended to new or intermediate R Markdown users who are looking for practical examples of how R Markdown documents can be effectively updated.

How to read this book

The book is designed to highlight the many ways in which your R Markdown documents can be customized. Users are encouraged to try using the concepts within their own documents. The code detailed in the book should be sufficient, but the full source code and examples are provided on GitHub.

It is recommended that readers have a basic understanding of R Markdown. Chapter 2 of *R Markdown: The Definitive Guide* (Yihui Xie, 2018) provides a great overview of the basics of R Markdown and is recommended background reading for any new users of R Markdown.

It should be noted that this guide does not intend to provide a full technical reference for R Markdown, and there is already extensive literature available on these topics. This book aims to supplement, not replace, this existing literature, and it is therefore recommended that readers explore the following books if they seek further information:

- *R Markdown: The Definitive guide* (Yihui Xie, 2018): provides an overview of the R Markdown package and the wide range of ways it can be used.
- *Dynamic documents and knitr*: provides more detailed technical guidance on the inner workings of **rmarkdown** and **knitr** (Xie, 2018b).

- *Authoring books with bookdown* (Xie, 2016): a short book which provides details on the **bookdown** package (Xie, 2018a), which is designed to simplify the creation of long-format documents in R Markdown.
- *blogdown: Creating Websites with R Markdown* (Xie et al., 2017): provides details of the **blogdown** package (**?**) package.

Where relevant, this book provides references to these existing resources.

Structure of the book

The book is broken down into small "recipes" which aim to demonstrate a single concept at a time.

Chapter content to be finalised once recipes created.

Software information and conventions

The R session information used when compiling this book is as follows:

```
xfun::session_info(c(
  'bookdown', 'knitr', 'rmarkdown'), dependencies = FALSE)
```

```
## R version 3.5.0 (2017-01-27)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.5 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   bookdown_0.7.19 knitr_1.20      rmarkdown_1.10
```

We do not add prompts (`>` and `+`) to R source code in this book, and we comment out the text output with two hashes `##` by default, as you can see from the R session information above. This is for your convenience when you want to copy and run the code (the text output will be ignored since it is commented out). Package names are in bold text (e.g., **rmarkdown**), and inline code and filenames are formatted in a typewriter font (e.g., `knitr::knit('foo.Rmd')`). Function names are followed by parentheses (e.g., `blogdown::serve_site()`). The double-colon operator `::` means accessing an object from a package.

"Rmd" is the filename extension of R Markdown files, and also an abbreviation of R Markdown in this book.

Acknowledgements

About the Authors

Part I

Get Started

Chapter 1

Installation

We assume you have already installed R (https://www.r-project.org) (R Core Team, 2017) and the RStudio IDE (https://www.rstudio.com). RStudio is not required but recommended, because it makes it easier for an average user to work with R Markdown. If you do not have RStudio IDE installed, you will have to install Pandoc (http://pandoc.org), otherwise there is no need to install Pandoc separately because RStudio has bundled it. Next you can install the **rmarkdown** package in R:

```
# Install from CRAN
install.packages('rmarkdown')

# Or if you want to test the development version,
# install from GitHub
if (!requireNamespace("devtools"))
  install.packages('devtools')
devtools::install_github('rstudio/rmarkdown')
```

Chapter 2

Basics

This book by no means intends to be a comprehesive guide to R Markdown, however before we start, it is important to provide an overview of the R Markdown ecosystem before we explain how to make changes to it. This chapter therefore aims to provide the basic concepts required for the book.

2.1 R Markdown components

- Explain the difference between all the parts in r markdown (pandoc, knitr, LaTeX).
- Use this section to signpost readers to the correct chapter.

Chapter 3

Multi-format projects

One of the main benefits of R Markdown can create multiple output formats from a single source document. This book example is available in all three formats, with PDF, ebook and online versions all being available.

Project should therefore ideally be designed to be flexible for the multiple outputs. However, users will often find themselves wanting to fine-tune the output, and doing so will often require

This chapter aims to provide several examples on how a document can be customised with output specific features, whilst also retaining the ability to export documents to multiple formats.

3.1 Control Output

Dsigning custom behavior which can respond to change in outputs using `is_latex_output` etc. For example, you may want to have interactive tables using `DT:datatable` in the HTML output but print static versions in the PDF.

3.2 Output-specific functions

HTML documents have the capacity to contain. The webshot package is available to take screenshots of HTML elements for non-interactive output formats (ebooks, PDF).

```
library(ggplot2)
library(plotly)
p <- ggplot(data = diamonds, aes(x = cut, fill = clarity)) +
            geom_bar(position = "dodge")
ggplotly(p)
```

3.3 Designing output

If we wish to design our own format-specific function we can use the functions `knitr::is_latex_output()` andknitr::is_html_output‘. These function will return a TRUE/FALSE action

```
knitr::is_html_output()
```

This can be used to control the output of statements.

Chapter 4

Knitr Hooks

- Knitr hooks: these are explained within previous R Markdown book
- Include practical examples could be very useful for readers to see the power of it.
- Use some of these: https://gist.github.com/yihui/2629886

Chapter 5

Tables

- provide more detailed documentation on kable
- Using kableExtra to style tables

Chapter 6

Managing Big Projects

- Practical tips on how a big project should be managed
- `source` is particularly useful for loading external scripts so that the R Markdown project isn't too bloated with code.

Ideas: - Use `(ref:tag)` to store page formatting options which might need to be reused. For example a page break

6.1 Sourcing Files

A benefit of using R Markdown is that it is easy

```
source("yourScript.R")
```

6.2 Caching

- Caching
- Ways it can be tailored to suit analysis. This cache invalidation is a great example: https://stackoverflow.com/questions/18376008/invalidate-a-chunks-cache-when-uncached-chunk-changes

6.3 Notifications

- Can link R Markdown with notifications if have long analysis
- Working with non R users
- Use HTML outputs and interactive graphics

Part II

Additional Packages

Chapter 7

Diagrams

It is often useful to express analysis using flowcharts and diagrams within reports. Although there are many separate programs which can be used to produce these, it can be beneficial to create these within our analysis directly. This makes it easier to update and edit the graph if we need to edit it in the future, and allows us to use the results from our analysis directly within the diagrams, making them more information.

While there are several different packages available for R, we will focus on the package **DiagrammeR** (Iannone, 2018). We also recommend that you read at http://rich-iannone.github.io/DiagrammeR/index.html . In this chapter we will explain some of the basic usages of the package.

7.1 Basic Diagrams

DiagrammeR provides methods to build graphs for a number of different graphing languages.

R Studio provides native support for Graphviz (`.gv`) and mermaid (`.mmd`) files. Using files of these types in RStudio provides the advantage of syntax coloring and allowing a quick preview of the diagram. For example, we can make a simple flowchart with the following code:

```
DiagrammeR::grViz("digraph {

        graph [layout = dot, rankdir = LR]

        node [shape = rectangle]
        rec1 [label = 'Step 1']
        rec2 [label = 'Step 2']
        rec3 [label =  'Step 3']
        rec4 [label = 'Step 4']

        # edge definitions with the node IDs
        rec1 -> rec2 -> rec3 -> rec4
        }",
        height = 200)
```

There are extensive controls which can be used to control the shape of nodes, colours, line types and add additional parameters.

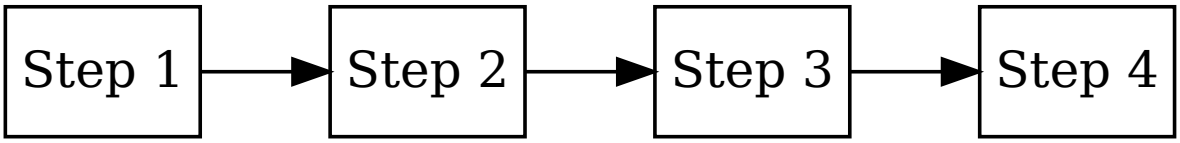
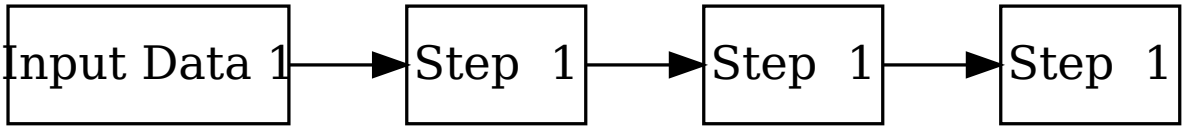


Figure 7.1: A basic graphic using DiagrammeR

7.2 Adding parameters to plots

Graphviz substitution allows for mixing in R expressions into a Graphviz graph specification without sacrificing readability. If you specify a subsitution with `@@`, you must ensure there is a valid expression for that substitution. The expressions are placed as footnotes and their evaluations must result in an R vector object (i.e., not a data frame, list, or matrix). Because there is the possibility to have multiple substitutions, numbering is required. Thus, the @@ notation is immediately followed by a number and that number should correspond to the number of the footnoted R expression. For example, suppose we have a dataset which is being analysed, and we would like to identify how many values are removed at each stage of a process.

```
DiagrammeR::grViz("
digraph graph2 {

graph [layout = dot, rankdir = LR]

# node definitions with substituted label text
node [shape = rectangle]
a [label = '@@1']
b [label = '@@2']
c [label = '@@3']
d [label = '@@4']

a -> b -> c -> d

}

[1]: paste('Input Data', 1)
[2]: paste('Step ', 1)
[3]: paste('Step ', 1)
[4]: paste('Step ', 1)
")
```


Input Data 1
Step 1
Step 1
Step 1

Chapter 8

References

Part III

Recipes

Chapter 9

Document Elements

There are lots of small tips and tricks that can be used to customize the content of R Markdown documents. This chapter provides short recipes which are useful for tweaking the behaviour of your own R Markdown documents.

9.1 Set Date to Current Date

It may be useful for the date of the knitted R Markdown document to automatically update each time we rerun the file. To do this, we can add R code directly to the `date` field in the YAML, and use the `Sys.time()` function to extract the current date. As this function will by default provide the date and time, we must specify the desired date time format as shown below:

```
---
title: Your title
date: !r format(Sys.time(), '%d %B, %Y')
---
```

This will automatically update each time you Knit your document i.e. 24 September, 2018. If we wish to customize the format of the dates, we can alter the time format by providing our own text string. Some examples are shown as follows:

- `%B %Y`: September 2018
- `%d/%m/%y`: "24/09/18
- `%d/%m/%y`: Mon 24 Sep

A full table of POSIXct formats is shown in Table 9.1, which allows users to construct their required date format.

9.2 Adding Multiple Authors to Document

We can add multiple authors to a R Markdown document within the YAML in a number of ways. If we simply want to list them on the same line, we can provide a text string to the document:

```
---
title: "Untitled"
author: "John Doe, John Smith"
---
```

Table 9.1: Date Time Formats within R

Code	Meaning	Code	Meaning
%a	Abbreviated weekday	%A	Full weekday
%b	Abbreviated month	%B	Full month
%c	Locale-specific date and time	%d	Decimal date
%H	Decimal hours (24 hour)	%I	Decimal hours (12 hour)
%j	Decimal day of the year	%m	Decimal month
%M	Decimal minute	%p	Locale-specific AM/PM
%S	Decimal second	%U	Decimal week of the year (starting on Sunday)
%w	Decimal Weekday (0=Sunday)	%W	Decimal week of the year (starting on Monday)
%x	Locale-specific Date	%X	Locale-specific Time
%y	2-digit year	%Y	4-digit year
%z	Offset from GMT	%Z	Time zone (character)

Alternatively, if we wish for each entry to be on its own line we can provide a list of entries to the YAML field. This can be useful if you wish to include further information about each author such as an email address or instituion:

```
---
author:
- John Doe, Institution One
- John Smith, Institution Two
---
```

We can make use of the markdown syntax `^[]` to add additional information as a footnote to the document. This may be more useful if you have extended information you wish to include for each author, such as providing a contact email, address etc. The exact behaviour will depend on the output format (HTML/PDF/Word):

```
---
author:
- John Doe^[Institution One, email@domain.com]
- John Doe 2^[Institution Two, email2@domain.com]
---
```

Certain R Markdown templates will allow you to specify additional parameters directly within the YAML. For example, the Radix output format allows `url`, `affiliation` and `affiliation_url` to be specified. We must first install the Radix package from GitHub:

```
devtools::install_github("rstudio/radix")
```

We can use the new format by changing the `output` option as specified below:

```
---
title: "Radix for R Markdown"
author:
  - name: "JJ Allaire"
    url: https://github.com/jjallaire
    affiliation: RStudio
    affiliation_url: https://www.rstudio.com
output: radix::radix_article
---
```

Chapter 10

LaTeX

For many authors, the main of long reports or books, the primary output will be LaTeX. In this chapter, we discuss approaches which can be used to customise the output of PDF reports.

Users should approach with a note of caution. One of the major benefits of R Markdown is the fact that a single source document can create documents with multiple formats. By tailoring your work to a single output format (PDF/Word/HTML), you may improve the appearance and performance of a single output but at the expense of this tranferability.

10.1 Inserting Commands

10.2 LaTeX preamble

10.3 Multi-figure plots

If we need to print multiple output graphs or figures, there are several ways this can be achieved.

10.4 Cross-output

As explained in https://bookdown.org/yihui/rmarkdown/r-code.html#figures of Yihui Xie (2018), we can place multiple figures side-by-side using the `fig.hold='hold'` along with the `out.width` option. As an example below, we have set the `out.width="50%"`:

```
plot(1:10)
plot(rnorm(10), pch=19)
```

The main benefits of this approach is that it is easily achieved, and also works for both PDF and HTML outputs.

10.5 LaTeX subfigures

When writing a document you may want to include some slightly more complicated figures with multiple images. Subfigures are a useful LaTeX feature which allows us to achieve this by plotting multiple figures within a single plot and providing each with their own subcaption.

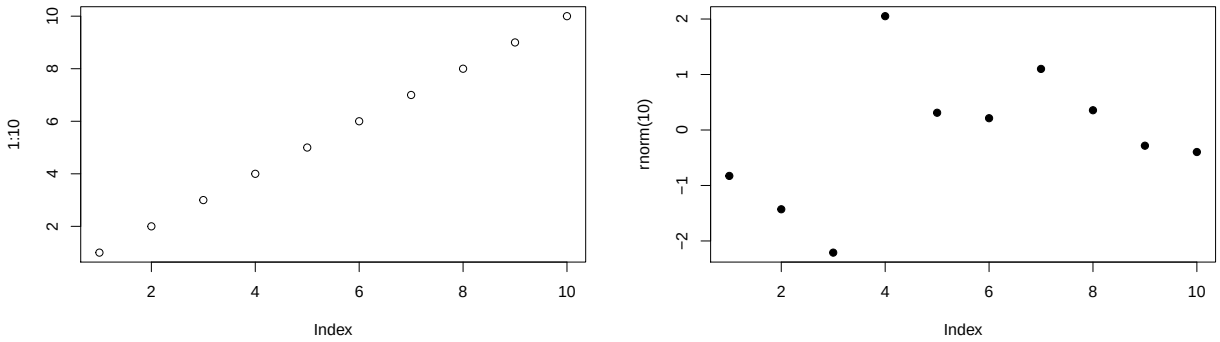


Figure 10.1: Side-by-side figures

Subfigures require the LaTeX package `subfig`. The line `\usepackage{subfig}` must therefore be included within the YAML, or if you are using an external tex template you can add this to that file. For example:

As listed within the knitr chunk options, subfigures require a few additional settings to be set in the chunk header:

- `fig.subcap` is a list of the captions for subfigures
- `fig.ncol`: the number of columns of subfigures
- `out.width`: the output width of the figures. You will normally set this 100% divided by the number of sub columns.

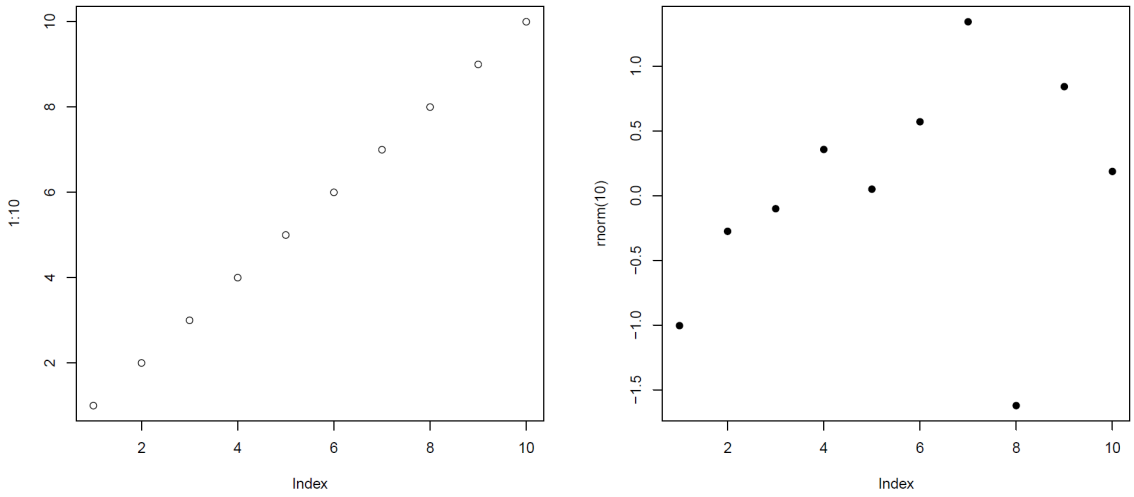
An example is demonstrated below:

````
```yaml
output: pdf_document
header-includes:
   - \usepackage{subfig}
```

```{r fig-sub, fig.cap='two plots', fig.subcap=c('one plot', 'the other one'), out.width='.49\\linewidth
plot(1:10)
plot(rnorm(10), pch=19)
```
````

The output is shown in Figure 10.2.

```
knitr::include_graphics("images/subfigure.png", dpi = NA)
```

10.5.1 Using with list



(a) one plot

(b) the other one

Figure 1: two plots

Figure 10.2: An example subcaption

Chapter 11

HTML Output

How to customise the appearance of HTML documents:

- CSS styles
- Adapt some of the details from here: https://rmarkdown.rstudio.com/html_document_format.html

11.1 Tabbed headings

https://stackoverflow.com/questions/38062706/rmarkdown-tabbed-and-untabbed-headings?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

11.2 Altering Citation Style

- Using short author citations: https://stackoverflow.com/questions/48303890/using-short-author-citations-in-bookdown utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

11.3 Automatically Generate Package Citations

It is important to that an acknowledgements is provided within your work to authors of any literature or software referenced. Within a typical workflow, this can be a frustrating experience if you are required to manually identify the correctly formatted citations, load them into a referencing software and then refer to them within the text. However, R Markdown makes it easy to streamline this process. We can generate a BibTex file for packages used within your analysis using the `write_bib` function within R Markdown.

```
# automatically create a bib database for R packages
knitr::write_bib(c(
  .packages(), 'bookdown', 'knitr', 'rmarkdown', 'DiagrammeR'
), 'packages.bib')
```

To refer to the packages in the text, we must add `bibliography: packages.bib` to the YAML frontmatter, and then references can be made using the format `[@R-package]` where `package` is replaced with the package name i.e. `[@R-rmarkdown]`. This will be replaced with an intext citation and will be added to the bibliography at the end of your document.

Note, that the `write_bib` command is designed to overwrite the existing bibliography. If you want to manually add any other items to the bibligraphy in your document, it is recommended that you create a second `.bib` file which is also referred to within the YAML `bibliography` field, as shown below:

```
---
bibliography: [packages.bib. references.bib]
---
```

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.

Iannone, R. (2018). *DiagrammeR: Graph/Network Visualization*. R package version 1.0.0.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman & Hall/CRC The R Series. CRC Press.

Xie, Y. (2018a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.7.19.

Xie, Y. (2018b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.20.

Xie, Y., Hill, A., and Thomas, A. (2017). *blogdown: Creating Websites with R Markdown*. Chapman & Hall/CRC The R Series. CRC Press.

Yihui Xie, J.J. Allaire, G. G. (2018). *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida, 1st edition. ISBN 978-1138359338.