**An Introduction to R Markdown**

Raphael Merz

2021-06-19

This document was created by someone who is also just getting started with R Markdown. Nevertheless, I tried to give a short overview of the practical implications of R Markdown and where I think the software makes one’s life much easier! If you have any questions or comments, be sure to send me an email ([raphael.merz@rub.de](mailto:raphael.merz@rub.de))!

1. What is R Markdown?

R Markdown is an ecosystem of packages for R (while also compatible with other coding languages) created to increase reproducibility in all things statistics. It can be used to combine text and code in one document and offers a huge amount of output (e.g., html, pdf, or word) and formatting options (e.g., essays, manuscripts, presentations).

Additionally, the combination of text and code can save a lot of time whenever your data changes (e.g., due to further exclusion of participants), because ideally the code will give you the “new” values with all changes included, hence you will not have to change much in your .Rmd file.

1. Installing R Markdown and essential Packages

The easiest way to install R Markdown is to do it within R:

install.packages("rmarkdown")

To create pdf files, you will also need to install LaTeX. This can be done by installing TinyTeX.

install.packages("tinytex")

tinytex::install\_tinytex()

After that you can install different packages for R Markdown that help you to organize your .Rmd files. The package papaja for example offers several templates for different kinds of documents. To date, I have always used the “APA article (6th edition)” option (work on APA 7 will start “in the foreseeable future”; Aust & Barth, 2021), but there will definitely be many more to choose from online.

To install papaja you will need pandoc, which is required to export the different output styles. pandoc should already be installed if you work with R Studio. Alternatively, it can be downloaded [here](https://pandoc.org/installing.html).

Because papaja is not yet available via the install.packages()-function, you will have to install it from GitHub:

# Install devtools package if necessary

if(!"devtools" %in% rownames(installed.packages())) install.packages("devtools")

# Install the stable development versions from GitHub

devtools::install\_github("crsh/papaja")

# Install the latest development snapshot from GitHub

devtools::install\_github("crsh/papaja@devel")

(Aust & Barth, 2021)

The first installation of papaja can take a few minutes. You should also have a look at the console because some functions must be run manually. After that you can open your first APA formatted R Markdown File via “File 🡪 New File 🡪 R Markdown 🡪 From Templates 🡪 APA article (6th edition)”.

1. Writing with R Markdown

When working with R Markdown, your main goal is to create research that can be reproduced by literally anyone with a functioning computer and access to your .Rmd file (and R script). To do this you should change as many values, figures etc. as possible to code so that anyone can understand where these values come from / which data you used exactly.

* 1. Coding within R Markdown

There are two main coding options within R Markdown. Firstly, you can code within the text to specify values right out of the data. Secondly, you can insert larger code chunks, for example to create tables or figures.

* + 1. Inline Coding

Coding within the text works pretty straight forward: you start with an apostrophe and an “r” to state that the following segment will be R code. Then you insert your code and finish off with another apostrophe:

## R Markdown:

We found a mean age of `r round(mean(data$age), digits = 2)` years.

# R Markdown - Output:

We found a mean age of 31.34 years.

To simplify (and shorten) the text in your .Rmd file, you can always create objects for specific values that you saved in a standard R script before. You would then have to run said R script before creating an output document with R Markdown, but the code within the text will be much shorter:

## R Script:

mean\_age <- round(mean(data$age), digits = 2)

## R Markdown:

We found a mean age of `r mean\_age` years.

# R Markdown - Output:

We found a mean age of 31.34 years.

* + 1. R Code Chunks

An R code chunk consists of many lines of code rather than being in a specific line within the text. You create a code chunk with three apostrophes and an “r” inside curly braces on the top of the chunk and three apostrophes at the end of the chunk. In between these apostrophes, you can write your code: e.g., functions to create tables or figures. You can also create the objects that are linked within the text (see above paragraph) here. Just make sure the objects are created before they come up in the text.

## R Markdown:

```{r}

mean\_age <- round(mean(data$age), digits = 2)

```

We found a mean age of `r mean\_age` years.

# R Markdown - Output:

We found a mean age of 31.34 years.

* + 1. Summary-Outputs

When coding specific values into a text, you will potentially need to extract said values from a summary output (at least that`s easier than finding different functions that give you the different value individually). To do that, you can save the summary output from which you want to extract single values from as an object and then save these single values as objects themselves by specifying where they are in the output (in squared brackets).

Note that you once more have to decide whether you want to have the code for creating the objects within your .Rmd file (in a code chunk) or in a different R script.

## R Script:

summary\_age <- summary(data$age)

# Summary-Output:

Min. 1st Qu. Median Mean 3rd Qu. Max.

18.00 21.00 27.00 31.34 39.00 65.00

mean\_age <- round(summary\_age[4], digits = 2) # = 31.34

## R Markdown:

We found a mean age of `r mean\_age` years.

# R Markdown - Output:

We found a mean age of 31.34 years.

Some outputs may consist of different sections or more than one column of values. To extract a specific value from these outputs, you will have to find out in which section the value is located (by looking at the created summary object or by using the levels()-function) and then specify the section, column, and row with a $ sign (similar to different variables of a data frame) and numbers inside the squared brackets:

## R Script:

summary\_age <- summary(model1)

view(summary\_age)

# Output (shortened):

$FIT

par fmin chisq df pvalue

8.00 0.153 160.18 8.00 0.003

$PE

lhs est se pvalue

component1 -0.02965 0.00568 0.002

component2 0.24862 0.16874 0.014

component3 0.01954 0.02684 0.001

model1\_chisq <- round(summary\_age$FIT[3], digits = 2) # = 160.18

model1\_p <- round(summary\_age$FIT[5], digits = 3) # = 0.003

model1\_comp3\_est <- round(summary\_age$PE[3,2], digits = 2) # = 0.02

## R Markdown:

We found a test size of `r model1\_chisq` with a p-value of `r model1\_p`.

[...] The third component had an estimate of `r model1\_comp3\_est`.

# R Markdown - Output:

We found a test size of 160.18 with a p-value of 0.003.

[...] The third component had an estimate of 0.02.

* 1. Citations

When you use the citation function R Markdown offers, you will also not have to create a reference list yourself. To use this function, you will have to create a .bib file of your bibliographic library (+ cite keys), which might require installing a few plugins for both R and your bibliographic manager. The exact procedure for the bibliographic manager Zotero is explained [here](https://ikashnitsky.github.io/2019/zotero/).

In the end, you will only have to link the .bib file from your bibliographic manager to your R Markdown (this can be done inside the .Rmd file, below the abstract 🡪 bibliography; note that the .bib and .Rmd file must be in the same folder) and insert the cite keys starting with an @ into squared brackets, and the reference list will be rendered automatically.

## R Markdown:

R Markdown can be used to create pdf outputs [@Aust\_Barth21;@merz21].

# R Markdown - Output:

R Markdown can be used to create pdf outputs (Aust & Barth, 2021; Merz, 2021).

1. Exporting the Outputs

If you are not working with inline code or code chunks (e.g., when you only use R Markdown for the reference list), you can use the knit button above your script window to export an output. Otherwise, you should use the render-function and put it right at the end of your .Rmd file:

# For a document named R\_Markdown\_Intro:

rmarkdown::render("R\_Markdown\_Intro.Rmd")

You can adjust the format and output options (amongst others), look out for the following:

# For a APA formatted manuscript pdf

documentclass : "apa6"

classoption : "man"

output : papaja::apa6\_pdf

So far, I have only worked with the class options man (manuscript), doc (document) and jou (journal), and output options papaja::apa6\_pdf and papaja::apa6\_word. For a more detailed overview of all options, see Aust and Barth (2021): [2.1.2 Rendering options](http://frederikaust.com/papaja_man/r-markdown-components.html#yaml-front-matter).

1. Final Notes

I hope this short overview helped you understanding the key benefits of working with R Markdown! I strongly recommend trying all these things out hands on – maybe even with your own data from previous (or current) projects, and just see for yourself if working with R Markdown helps you and your research.

1. References and other useful resources

R Markdown

R Markdown. (n.d.) *Introduction*. <https://rmarkdown.rstudio.com/lesson-1.html>

R Markdown. (n.d.) *Reference Guide*. <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf?_ga=2.59950765.146562034.1623589576-1913646398.1619274903>

R Markdown. (n.d.). *Cheat Sheet*. <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

R Markdown. (n.d.). *Quick Tour*. <https://rmarkdown.rstudio.com/authoring_quick_tour.html>

Yihui, X., Allaire, J.J., & Grolemund, G. (2021, April 09). *R Markdown: The Definitive Guide*. <https://bookdown.org/yihui/rmarkdown/>

Feldman, G. (n.d.) RMarkdown / Reproducibility. *JAMOVI / JASP / R / Rmarkdown collaborative guide* (pp. 46-48). <https://docs.google.com/document/d/1A9BbOCsrg1ikLaBltKhXVKj-eetlrBqR-1u-2V99I2c/edit#heading=h.j8sehv>

Papaja for R Markdown

Aust, A, & Barth, M. (2021, May 14). *papaja: Reproducible APA manuscripts with R Markdown*. <http://frederikaust.com/papaja_man/>

Buchanan, E. M. (2017, October 10). *R - Markdown with Papaja* [Video file]. YouTube. <https://youtu.be/I_HW5Rraqg8>

Citation and Bibliographic Management

Kashnitsky, I. (2019, March 14). *Zotero hacks: unlimited synced storage and its smooth use with rmarkdown*. <https://ikashnitsky.github.io/2019/zotero/>