

Reproducible Writing

Dynamic APA-formatted manuscripts with papaja

Daniela Palleschi

2024-06-18

Table of contents

Disclaimer	1
Requirements	2
tinytex	2
papaja	2
Writing	3
Rmarkdown	3
APA-formatting with papaja	3
Task	3
Cross-referencing	4
Figures	4
Images	4
Example sentences	5
Tables	6
Table labels	7
Data tables	7
Placing tables and figures	7
Citations	8
BibTex format	8
In-text citations	8
Zotero	9
Output	9
Collaboration	9

Learning objectives

Today we will...

- learn about R markdown for writing
- integrate citations with Bib(La)Tex
- learn how to cross-reference
- create linguistic example sentences

Resources

- to read more on today's topic, check out the **papaja** manual (Aust & Barth, 2023)
 - https://frederikaust.com/papaja_man/

Disclaimer

- this is also a *very* quick-and-dirty introduction on getting started with APA-formatted manuscripts in R markdown
 - there are a *lot* of resources (e.g., E-books, blog posts, forum threads, manuals) that will address specific formatting problems or wishes you may have
 - Google is your friend!
- also, these slides were written in Quarto, and are published as HTML
 - much of the syntax I'm presenting doesn't actually work in Quarto/HTML
 - but all the raw code that I show will work in R markdown/PDF

Requirements

- packages:
 - `papaja`
 - `tinytex`
- software (optional)
 - Zotero + a Zotero account

- download from the Moodle or GitHub:
 - `references.bib`

tinytex

- includes helper functions for installing LaTeX distribution
 - i.e., helps create PDF outputs

```
# to install tinytex run these two lines
install.packages("tinytex")
tinytex::install_tinytex()
```

papaja

I want to add a citation

- for APA-formatted scientific manuscripts
 - currently uses APA 6, but we can update it to APA 7

```
# to install tinytex run these two lines
install.packages("papaja")
```

Writing

- writing an article or thesis
 - *not* a report
- should be kept separate from the actual analyses
 - e.g., in its own folder or even own project
 - if in its own project: make sure you transfer over files needed (e.g., figures, data, saved models)

Rmarkdown

- we can also write PDFs in Quarto
 - but its relatively new, and there's more support for scientific articles in R markdown
- most everything in R markdown is identical to Quarto
 - some important differences: code chunk options (we'll see these later)

APA-formatting with papaja

- a package specifically for writing APA-formatted manuscripts
- File > New File > R markdown > From template > APA-formatted article (papaja)
 - will open a file with a long YAML
 - render it and see how it looks

Task

- in a new papaja script, do the following:
 1. change the YAML to include your name

Cross-referencing

- e.g., referring to another section
 - in which case, we need `number_sections: TRUE` in our YAML
- simply provide a label in the same line as a heading, either with `{#section_label}` or `\label{section_label}`
 - then provide the label within `\ref{}`, and the section number will be produced in the output
- the example text below would then be written as *Here is some text in Section 1* (assuming the Introduction is numbered as 1)

```
# Introduction {#section_label}
```

```
Here is some text in Section \ref{section_label}.
```

Figures

- or figure, table, example sentence or equation

```
```{r fig-iris, eval = TRUE}
library(ggplot2)
iris |> ggplot() + aes(x = Sepal.Length, y = Sepal.Width) + geom_point()
```
```

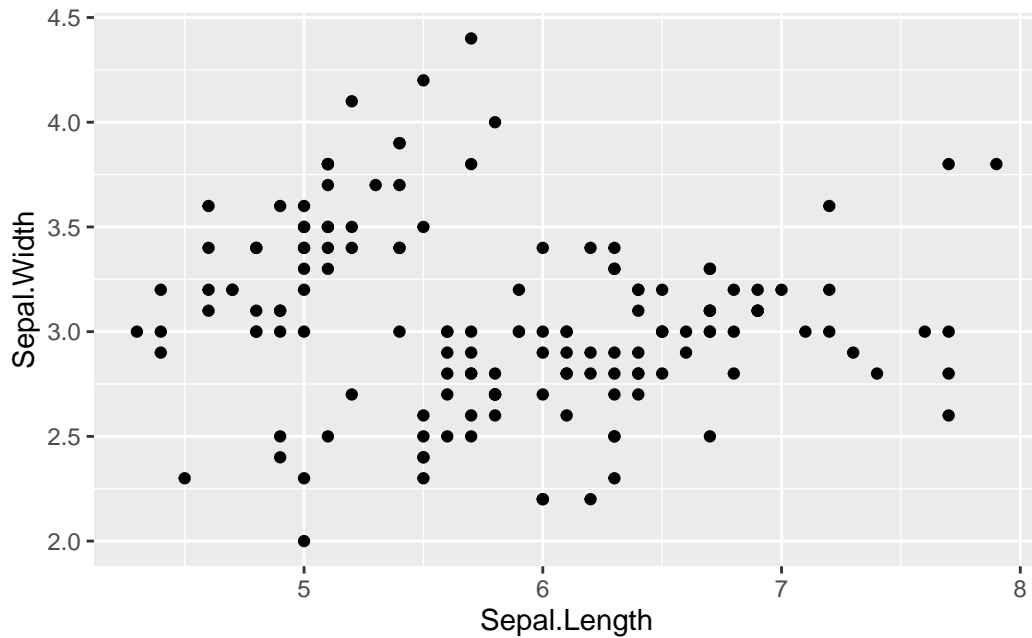


Figure 1

- now if we were to write *As seen in Figure \ref{fig-iris}, we would get: As seen in Figure 1*
- be careful not to use underscores (`_`) in your figure labels, this causes problems

Images

- You might also include a figure of the trial procedure, or some other visual description of your data
- For example, in Figure Figure 2 we see an overview of the types of iris (flowers) that make up the data from the built-in `iris` dataset (figure from Mijwil & Abttan, 2021)
- you can then cross-reference to images the same was, by putting the label inside `\ref{}`

```

```{r fig-summary, out.width="100%", fig.pos="t", fig.cap="\label{fig-summary}Visual depiction of the iris dataset"}
knitr::include_graphics(here::here("figures", "iris_photo.png"))
```

```

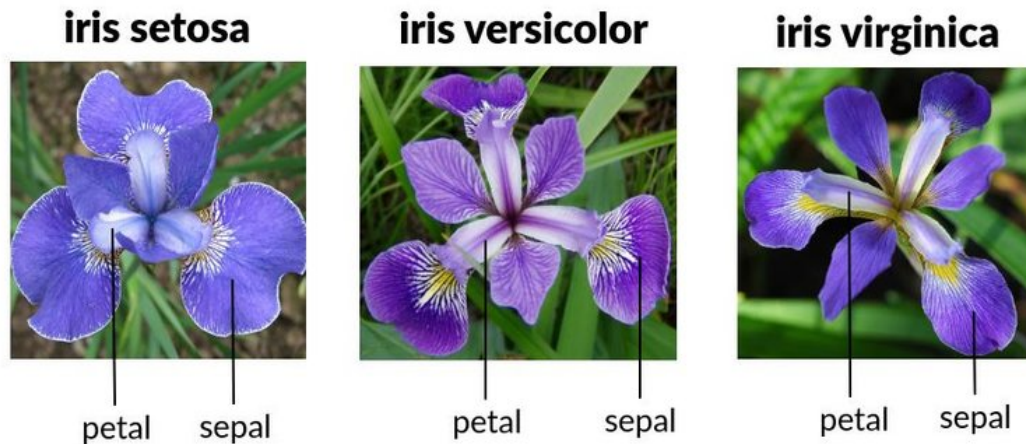


Figure 2: Visual depiction of dependent variables from the `iris` dataset

Example sentences

- we can write example sentences with latex syntax
- first, add this to your YAML

```

header-includes:
  \usepackage{float} \usepackage{gb4e} \noauthomath

```

- then, you can write an example as follows:

```

\begin{exe}
\ex \label{ex:example} This is an item with just one example.
\end{exe}

```

(1) This is an item with just one example.

- and reference it in your text with `See example \ref{ex:example}`, which will be written as: *See example 1*

Tables

- e.g., you can give an overview of your stimuli (you could also do this with example sentences)
 - if producing your table with R code, remember to feed it into a function that formats tables
 - e.g., `knitr::kable()` or `papaja::apa_table()`

```
```{r apa-table, eval=F, echo = "fenced"}
library(tidyverse)
tribble(
 ~"Item", ~"Condition", ~"Sentence",
 "1", "a", "Example sentence of condition A",
 "1", "b", "Example sentence of condition B",
 "1", "c", "Example sentence of condition C",
 "1", "d", "Example sentence of condition D",
) |>
 papaja::apa_table(caption = "Example stimuli")
```
```

Table 1: Example stimuli

| Item | Condition | Sentence |
|------|-----------|---------------------------------|
| 1 | a | Example sentence of condition A |
| 1 | b | Example sentence of condition B |
| 1 | c | Example sentence of condition C |
| 1 | d | Example sentence of condition D |

Table labels

- writing “See Table `apa-table` for example stimuli” will print:
 - *See Table 1 for example stimuli.*
- For this to work, you need to provide a label in the code chunk settings: `{r apa-table, echo=F, eval=T}`. + remember to use `\ref{tab:label}` and replace label with yours (i.e., don’t forget the `tab:` prefix).

Data tables

- You can of course also present tables of your data or models

Table 2: Mean values for `iris` measures

| Species | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|------------|--------------|-------------|--------------|-------------|
| setosa | 5.006 | 3.428 | 1.462 | 0.246 |
| versicolor | 5.936 | 2.770 | 4.260 | 1.326 |
| virginica | 6.588 | 2.974 | 5.552 | 2.026 |

- cross-referencing works the same:
 - you write: `Mean values are given in Table \ref{tab:iris-table}`.
 - R markdown prints: *Mean values are given in Table 2*.

Placing tables and figures

- To allow figures and tables to appear in-text (i.e., not at the end of the document), change `floatsintext` in the YAML to `yes` (it will be `no` by default)
 - otherwise `papaja` pushes all tables and figures to the very end of the document

```
floatsintext      : yes # CHANGE TO YES to allow figures and tables to float in text
```

Citations

- the most straightforward way to include citations is by manually adding BibTex citations into your `.bib` file
 - you can define which `.bib` file to use in your YAML (we currently have `bibliography: r-references.bib`)
- you can easily get the BibTex formatted citation via Google Scholar
 - although I suggest using Zotero with the Better BibTex plug in, which stores them locally

BibTex format

- below is an example of a BibTex formatted citation
 - the first info after the opening curly bracked is the reference key (`knuth1984literate`)
- add this reference to your `.bib` file

```
@article{knuth1984literate,  
  title={Literate programming},  
  author={Knuth, Donald Ervin},  
  journal={The computer journal},  
  volume={27},  
  number={2},  
  pages={97--111},  
  year={1984},  
  publisher={Oxford University Press}  
}
```

In-text citations

- to then include a reference in-text, include the BibTeX reference key preceded @
- so if we write `@knuth1984literate` we should get a formatted citation: Knuth (1984)
 - and the full citation should be added to our references section
- if we were to write `[@knuth1984literate]` we would get the reference in brackets (Knuth, 1984)
 - to learn more about how to control the formatting of in-text references check out [Section 3.2 \(Citations\) in the papaja manual](#)

Zotero

- this process can be streamlined by using Zotero + Better BibTex (BBT)
 - there are several walk-throughs of how to do this online, e.g.,
- the benefit: using Zotero keeps a record of your PDFs/readings
 - Zotero Desktop is a nice way to annotate readings and take notes
 - direct integration of BBT with RStudio is possible
- check out this [blogpost](#) to learn more

Output

- PDF: tex file is generated in the process
 - `keep_tex: true`
 - will keep the .tex file produced
 - if you want to move the document to Overleaf or LaTeX, I recommend:
1. Add `keep_tex: true` to your YAML
 2. Render your document
 3. Go find the .tex output in the folder
 4. Upload this tex file to an Overleaf project
 5. Make sure to also copy over any figures created in the output

Collaboration

- unfortunately, there's no elegant method for collaborative writing in R markdown/Rstudio
 - the only real option is to use a remote git repository (e.g., GitHub or GitLab)
 - but this has a steep learning curve and is prone to problems when collaborators aren't familiar with git
 - track changes are also not as elegant as in Overleaf, Google Docs, Word documents, etc. (e.g., with accept/reject buttons or pop-up comments)
- if you have co-authors, consider they may or may not be R (markdown) or LaTeX or R-savvy
- you could send collaborators a PDF that they annotate and then you make the changes back in your R markdown script(s)
 - but this is quite labour intensive on your side
- alternatively, you can also output your first draft as a Word document and then use that as a starting point for collaborative writing
 - keep in mind that any changes to the analyses will then need to be done in Rmarkdown and imported to the edited Word document
- there is also the `trackdown` package which integrates R markdown scripts with Google Docs
 - but there are obvious data protection/ethical concerns with doing so
- currently, I prefer to move the first draft to Overleaf
 - I can always re-run my analyses, re-write up my results section, and just replace the LaTeX code for that section

Thesis writing

- there are also ways to write books in R markdown
 - a lot of web-books are written with **bookdown**, see the website for more: <https://bookdown.org/>
 - I personally prefer Quarto books for web books, for more info: <https://quarto.org/docs/books/>
- to write your thesis, there's the **oxforddown** template
 - <https://ulyngs.github.io/oxforddown/>
- with these options, each chapter is in a self-contained .Rmd script
 - a 'parent' document contains the metadata to knit all chapters into a book

References

- Aust, F., & Barth, M. (2023). *papaja: Prepare reproducible APA journal articles with R Markdown*. <https://github.com/crsh/papaja>
- Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2), 97–111.
- Mijwil, M., & Abttan, R. (2021). Utilizing the Genetic Algorithm to Pruning the C4.5 Decision Tree Algorithm. *Asian Journal of Applied Sciences*, 9, 45–52. <https://doi.org/10.24203/ajas.v9i1.6503>