R Markdown as an Authoring Tool in Linguistics

Yongfu Liao (廖永賦)[*]

September 16, 2018

Abstract

bold

[*]National Taiwan University, Taipei, Taiwan

There is an online version of this article with animated figures. To have a better reading experience, visit **https://bit.ly/2D0JtIT** for the article.

1 Introduction

Authoring is a common task for all scholars and students in academia, from report preparation, teaching, journal submission to book writing. Yet many people in academia aren't familiar with using specialized tools and an integrated workflow for academic writing, especially in the fields of Social science.

Academic writing takes a great deal of time. However, much of the time spent *isn't realated to the content or the idea the author wishes to convey* but the chores regarding repetive works such as manually combining results from differnt analysis tools or formatting the papers to suit the requirements of journals. A great deal of time can be saved if there exists a recommended and integrated workflow for academic writing, either as a norm or a culture that encourages this. Currently, however, workflows for academic writing are a matter of tastes for different authors and are considered as personal skills. It should be argued that a workflow be proposed in a specific field of science, especially in the fields of Social science, where students often have no formal programming training in building custom programs to facilitate an integrated workflow.

1.1 A Non-integrated Workflow

An integrated workflow for academic writing is crucial in science. An example of a common, but not integrated, workflow is illustrated in figure 1:

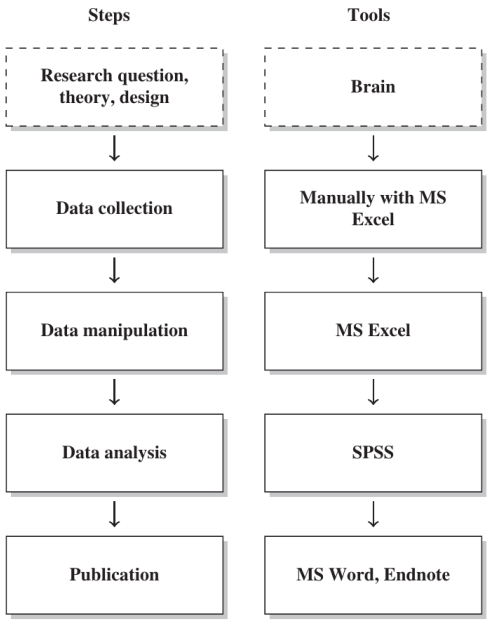

Figure 1: A common but not integrated workflow (Munzert, Rubba, Meißner, & Nyhuis, 2014, pp xviii).

One of the problem of this workflow is that authors have to manually copy-and-paste results from different sources, which are error-prone, and when errors are found, authors have to go through the process again, wasting substantial time. An even more serious problem related to this workflow is that it can hinder the progress of science, since when errors occured but are not found, erroneous results enter a published paper, in which subsequent researches might base on.

Facilitating an integrated workflow for acedemic writing is related to specific tasks a field often does. Hence, no preexisting tools exist. However, general frameworks exist and are extendable and customizable to fit the specific needs of a field. This article surveys the R Markdown ecosystem and focuses perticularly on features related to Linguistics in order to propose a integrated workflow for authoring.

2 Overview of R Markdown

2.1 A Brief Introduction

"Markdown" is a minimalist and easy-to-learn markup language[1], which formats the text by using plain text markers, e.g., lines beginning with "#" are first level titles and "##" are second level titles, wrapping text with "*" results in italics, etc. "R Markdown" extends the syntax of Markdown to allow more versatile styling of the text and therefore allows authoring documents with publication-ready qualities.

2.2 Benefits of Using R Markdown

There are several benefits for using R Markdown as an authoring tool. Most of them results directly from the extensibilty of R, the language R Markdown bases on.

2.2.1 R

The core feature of R Markdown is its integration with R, a programming language developed not in a traditional CS[2]-context but for the purpose of statistical computing (R Core Team, 2018). This makes R an special language – althought R has a steep learning curve compared to other GUI-based statistical softwares, it has a gentle learning curve compared to other "hardcore" programming languages. R is designed for scientists, not programmers. In addition, R has a huge and friendly community support, which means solutions to many problems new users often confront can easily be found on the web.

Many fields other than statistics either start to or already use R substantially, such as Biostatistics and

[1] Markup languages are used to style text appearance. For example, HTML (Hyper Text Markup Language) is one of the most popular markup language, which is used to format web pages.

[2] Abbreviation of "Computer science".

Bioinformatics, Ecology and Evolution, Finance, Psychometrics, Geospatial analysis, and even Linguitics (CRAN, 2018b). This is due to R's great extendibility, with more than 13,000 packages hosted on CRAN (2018a). As noted later, this extandibility also enables turning R Markdown into a specialized tool for authoring in Linguistics.

By integration with R language, R Markdown allows computed results be directly embedded into the document. To put it another way, the analysis of data (through R) is directly integrated into document writing, and hence, errors incured by manual copy-and-paste are eliminated. This also enhance the *reproduciblity* of the workflow (Baumer, Cetinkaya-Rundel, Bray, Loi, & Horton, 2014), since every time the document is generated from R Markdown, the underlying code for data analysis is rerun to generate the embedded output. Generating the document is essentially reproducing the analysis of data.

2.2.2 Supporting Reproducibility

A Reproducibile analysis is an analysis with results that can be guaranteed to be reproduced from raw data anywhere by anybody (Berez-Kroeker et al., 2018). Besides rerunning analysis to generate results every time, two other features make R Markdown a good tool for facilitating a reproducible workflow:

1. **R Markdown is plain text**

   Plain text format, contrary to binaries such as MS[3] Word, doesn't require specialied (and often propriety) software to open, hence facilitates the openness of science. Plain text also makes the file easiler to "version control", i.e, keeping the history of modifications, or versions, of the file by version control softwares.

2. **Python support**

   The new R package reticulate (Allaire, Ushey, & Tang, 2018) enhances the ability of Python integration in R. It is now possible to run Python in R console. The new Python engine enabled by

[3] Abbreviation of "Microsoft".

reticulate also solved a major drawback in previous versions of R Markdown – Python variables are shared across different code chunks, i.e., Python code chunks share states in R Markdown. This gives R Markdown the same power as Juyper notebook.

Integrating R and Python is especially important for Linguistics, as many actively developed packages and libraries for Linguistics are written in Python. Even softwares like Praat (Boersma, 2002) has a third-party support in Python, which allows accessing low-level functions in Praat with Python (Jadoul, Thompson, & de Boer, 2018). Some tasks that R certainly does better than Python are data manipulation, statistical analysis, visualization, and report generation, hence, integrating R and Python combines the strengths of both languages. Here, R Markdown acts as a "glue", combining different parts of analysis together into an integrated whole.

2.2.3 Wide Range of Output Formats and Styles

R Markdown supports a variety of output formats, owing to its foundation, Pandoc (MacFarlane, 2013). Some of the supported formats are MS Word, MS Powerpoint, LaTeX, PDF, and HTML. There are also multiple styles of document support, such as slides, books, journal papers[4], and even websites. This large variety of output formats enables authors to publish their works through different formats with the same underlying R Markdown file.

It can be argued that the most prominent output format is HTML. In this Digital Age, the web becomes a popular, if not dominant, way to distribute publications. Web pages enable displaying more varieties of contents, such as GIF (animated figures) and tables with search bars, thus enhancing the ability to convey ideas. With research becoming more complex, traditional medium might not be enought to present the results. For example, it might be useless to display a complex 3D graph in a static PDF. Using GIF, however, can facilitate the visualizing of complex 3D graphs by using animation to rotate the 3D objects

[4]For a list of supportted journal templates, see https://github.com/rstudio/rticles.

in the graphs. R Markdown's native support of HTML output makes it a valuable tool for communicating – the explanatory text is written directly with the dynamic visual elements, rather than using links to link to external files or web pages.

With the ability to generate HTML outputs easily, authors also gain the ability of self-publishing the content through web, making resources available to a wider audience. This is especially useful for educational purposes in academia.

2.3 Making R Markdown Suitable for Linguistics

The power of R Markdown as an authoring tool that facilitates an integrated workflow in Linguistics comes from R's extendiblity. By using R extensions, or R packages, R Markdown can be turned into a specialized tool for writing Linguistics-related documents. Below introduces two examples that reduce the burden of writing Linguistics-related documents in R Markdown.

2.3.1 Bibliographies and Citations

It is necessary to insert citations when writing articles for journal submission, and many use EndNote, a reference management software, together with MS Word to accomplish this. R Markdown has native support for inserting and formating citations and bibliographies, using a citation syntax provided by Pandoc (RStudio, 2018). With R's extendibility, the experience of inserting citations can become more comfortable. For example, the R package citr (Aust, 2017) lets authors insert citations through a GUI interface, where authors can search information about the articles (author, year, titles etc.) they want to insert. This extension makes the experience of inserting citations and references in R Markdown similar to that of using EndNote with MS Word.

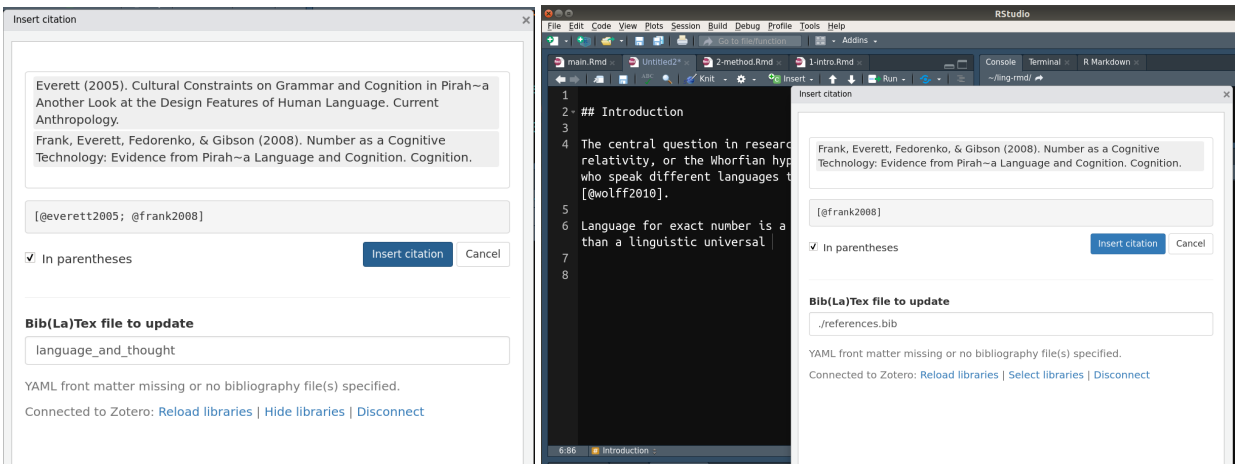

Figure 2: Using citr with R Markdown.

2.3.2 Inserting IPA Symbols

Problem often arise when inserting IPA symbols into documents, since there is no simple way to type IPA symbols with the keyboard. The author of this article created a package to deal with this problem. linguisticsdown (Liao, 2018) makes it possible to type IPA symbols by searching their phonetic descriptions, such as “plosive”, “bilabial”, “aspirated” etc., or by using the X-SAMPA input method (Wells, 1995). With this extension, authors can write documents containing IPA symbols with ease.

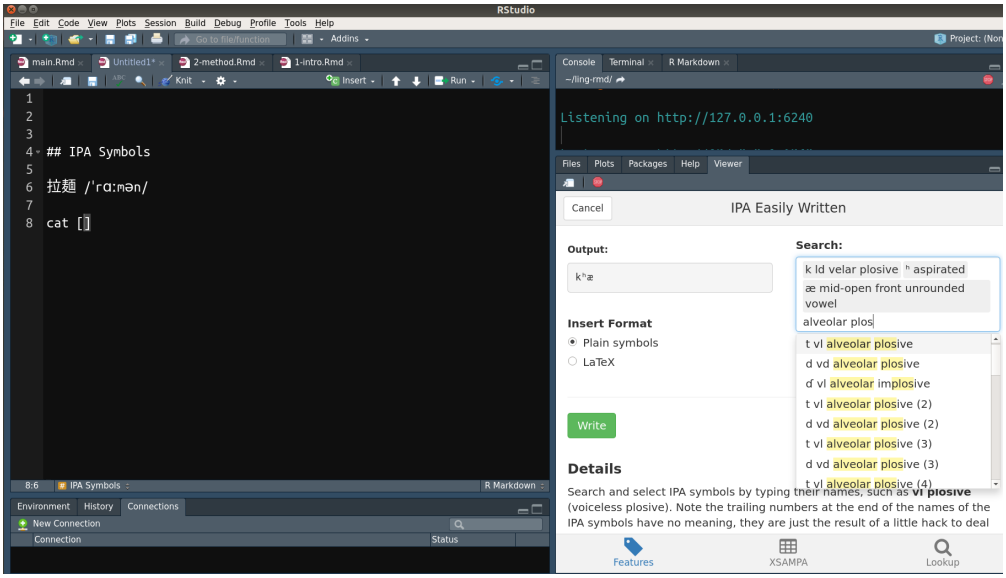

Figure 3: Using linguisticsdown with R Markdown.

2.3.3 Using Templates

As suggested previously, R Markdown can output formats meeting the requirements for journal submission. Authors can thus write documents with the simple R Markdown syntax without worrying the typesetting. Currently, officially supported journal templates[5] are mostly journals under big publishers such as Elseiver, Springer, and SAGE. The support can be extended, however, as long as there are LaTeX template provided. For example, thesisdown (Ismay, 2016/2018) provide an R Markdown template for writing thesis at Reed College, and several other thesis templates based on thesisdown modified it to fit their institutions' needs.

[5]For a list of supportted journal templates, see https://github.com/rstudio/rticles.

3 A Proposed Workflow

Figure 4 illustrates the flow of text (for readers) and data (for analysis) processing in R Markdown. Based on figure 4, authors can do the following to achieve an integrated workflow and don't need to navigate around several softwares during authoring:

1. Put R or Python code that clean, manipulate, and analyze data in the R Markdown source file. If there are too many code, put them in separated R or Python scripts and include them into R Markdown by using "source functions". For even more complex analyses, use the structure of *R Package* as a basis for project management, in which documentation of data sets and functions[6], raw data, and reports can be put together into a well-structured package (Flight, 2014; H. Wickham, 2015).

2. Put R code that generate figures or tables in the R Markdown source file, making it easy to see how they were generated.

3. For values that can be computed from data and needed to be included as inline text in the document, save them as R variables and put them inline with special syntax. For example, use *p*-value = ‘r p_val‘. The variable `p_val` will be converted to the value when the output document is generated. Hence, when data changes (e.g. addition of new data) or the analysis code are modified, the value of the variable `p_val` gets automatically updated as well.

4. Use citr to search and insert citations. citr integrates well with Zotero[7], a free and open-source reference management software similar to EndNote. The citation and bibliography format is automatically styled based on the provided csl file[8].

5. Use linguisticsdown to type IPA symbols.

[6]For complex analyses, there are often long and repetitive code. Wraping these redundant code into functions can make the analyses more manageble.

[7]https://www.zotero.org/

[8]Citation Style Language. See https://citationstyles.org/ for details.

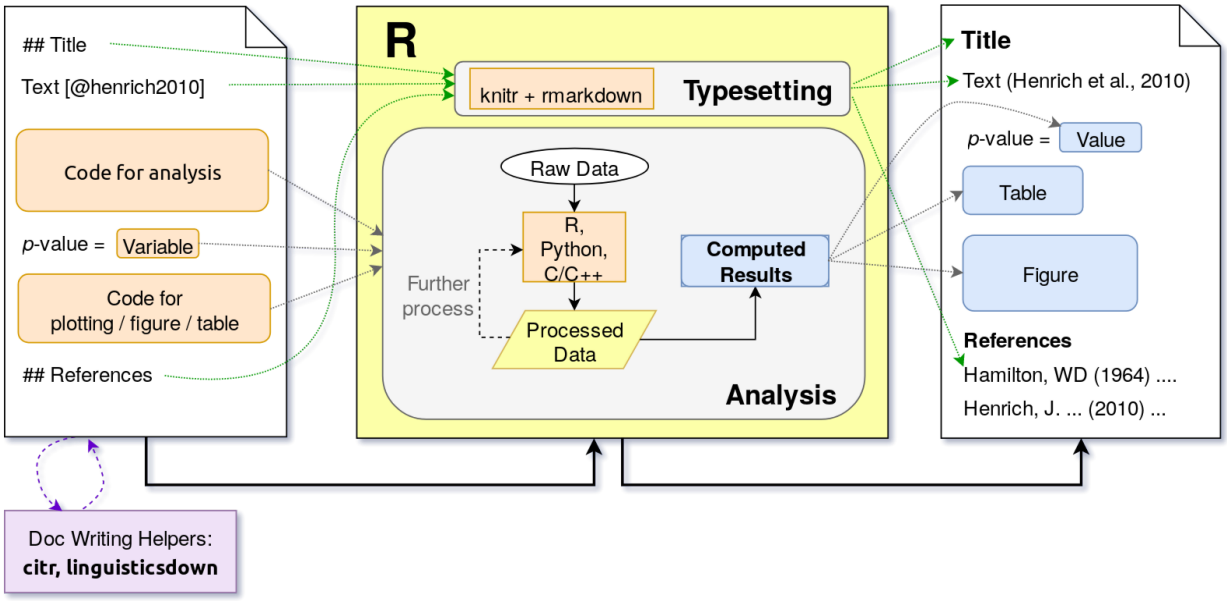

Figure 4: Combining R's data analysis and typesetting abilities to generate a reproducible report. The leftmost is a representation of an R Markdown source file; the rightmost is a generated document. The middle is the process underlying the conversion from source file to output document. For documents with complete template support, the typesetting is automated, so authors only need to focus on analysis and contents of the article.

4 Discussion

The nature of R, a programming language, lend itself to facilitating reproducibility, since writing down the analysis as *code* is essentially *recording every step of the analysis.*

R Markdown fully harnesses the capabilities of R language, not only R's abilitity in dealing with data but also its ability to typesetting documents. Putting these two features together makes R Markdown a powerful tool for scientific authoring.

4.1 Limitations

4.1.1 Template Support

There are some limitations however. For complete template supports, such as templates provided by the package rticles (Allaire et al., 2018), R Markdown can be used in an integrated fashion, from dealing with data to typesetting, without additional manual setup. For partial template supports, such as native LaTeX templates exist but are not modified to work in harmony with R Markdown, it can be extend to fully support R Markdown without too much efforts.

However, there are instances where no LaTeX templates exist. For example, *Chinese Journal of Psychology* doesn't provide any template and only accepts submission in MS Word format. This problem can be fixed by using the MS Word output (`.docx`) provided by R Markdown. A Word template can also be set up manually (Layton, 2015).

4.1.2 Extension Specific to Linguistics

R Markdown's capability depends on the R community. With more R packages being developed, R Markdown becomes more powerful. The number of users in a field matters as well, with more users comes more demand of functionalities, and hence more volunteers creating new packages to meet the needs.

There are few supports of R and R Markdown related to Linguistics, except for fields like Text Mining and Natural Language Processing. To make R Markdown an authoring tool more suitable for Linguitics, a larger userbase is needed. Linguists familiar with R can also make functions that they used regularly availble to others by bundling the functions into a package.

References

Allaire, J., Ushey, K., & Tang, Y. (2018). *Reticulate: Interface to 'Python'*. Retrieved from https://CRAN.R-project.org/package=reticulate

Allaire, J., Xie, Y., R Foundation, Wickham, H., Journal of Statistical Software, Vaidyanathan, R., … Öğreden, O. (2018). *Rticles: Article Formats for R Markdown*. Retrieved from https://github.com/rstudio/rticles

Aust, F. (2017). *Citr: 'RStudio' Add-in to Insert Markdown Citations*. Retrieved from https://github.com/crsh/citr

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., & Horton, N. J. (2014). R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics. *ArXiv E-Prints*. Retrieved from http://arxiv.org/abs/1402.1894

Berez-Kroeker, A. L., Gawne, L., Kung, S. S., Kelly, B. F., Heston, T., Holton, G., … Woodbury, A. C. (2018). Reproducible research in linguistics: A position statement on data citation and attribution in our field. *Linguistics*, *56*(1), 1. https://doi.org/10.1515/ling-2017-0032

Boersma, P. (2002). Praat, a system for doing phonetics by computer. *Glot International*, *5*(9/10), 341–345.

CRAN. (2018a). Contributed Packages. Retrieved September 13, 2018, from https://cran.r-project.org/web/packages/

CRAN. (2018b). CRAN Task Views. Retrieved September 13, 2018, from https://cran.r-project.org/web/

views/

Flight, R. M. (2014, July). Analyses as Packages. Retrieved September 15, 2018, from https://rmflight.github.io/posts/2014/07/analyses_as_packages.html

Ismay, C. (2018). *Thesisdown: An updated R Markdown thesis template using the bookdown package*. Retrieved from https://github.com/ismayc/thesisdown (Original work published 2016)

Jadoul, Y., Thompson, B., & de Boer, B. (2018). Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, *71*, 1–15. https://doi.org/https://doi.org/10.1016/j.wocn.2018.07.001

Layton, R. (2015, July 21). Happy collaboration with Rmd to docx. Retrieved September 16, 2018, from https://rmarkdown.rstudio.com/articles_docx.html

Liao, Y. (2018). *Linguisticsdown: Easy Linguistics Document Writing with R Markdown*. Retrieved from https://liao961120.github.io/linguisticsdown/

MacFarlane, J. (2013). Pandoc: A universal document converter. *URL: Http://Pandoc.org*.

Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2014). *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons.

R Core Team. (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

RStudio. (2018). Bibliographies and Citations. Retrieved September 15, 2018, from https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html

Wells, J. C. (1995). Computer-coding the IPA: A proposed extension of SAMPA. Retrieved from https:

//www.phon.ucl.ac.uk/home/sampa/ipasam-x.pdf

Wickham, H. (2015). *R packages: Organize, test, document, and share your code*. O'Reilly Media, Inc.