R Markdown as an Authoring Tool in Linguistics

Yongfu Liao (廖永賦)[*]

September 28, 2018

Abstract

R Markdown, an alternative document preparation system to Microsoft Word and LaTeX, is introduced for academic and scientific writing in Linguistics. Several strengths of *R Markdown*, including integration of document writing and data analysis, extensibility, customizability, and graphical user interface support, and their potentials for enhancing an integrated, reproducibility-facilitating, and user-friendly document writing workflow for scientific authoring in Linguistics are discussed. Some limitations of *R Markdown* are also pointed out to provide future directions for making *R Markdown* a better authoring tool in Linguistics.

[*]National Taiwan University, Taipei, Taiwan

An online version of this article with animated figures can be found at **bit.ly/LingRmd**.
The source of this article can be found at https://github.com/liao961120/ling-rmd.

1 Introduction

Writing is a common task for all scholars and students in academia, which 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 paper submission.

Currently, two document preparation systems, *Microsoft Word* and *LaTeX*, predominate in academia. There are advantages and disadvantages related to both softwares. *Microsoft Word* has an intuitive user interface and hence is more error-proof regarding typos compared to *LaTeX*. *LaTeX*, on the other hand, excels at writing complex mathematical equations and has extreme flexibilty regarding typesetting. The flexibility comes at a price, however. To achieve this flexibility, *LaTeX* has an extremly complex syntax and has been shown to hinder document writing (Knauff & Nejasmic, 2014).

One feature that *Microsoft Word* and *LaTeX* both lack is the ability to integrate with tasks other than document writing itself. For example, in the context of scientific authoring, computed results from analyses of data need to be copy-and-pasted from a statistical software into the document. This could take the author a substantial of time if the anlysis need to be redone, such as when data are updated or some errors where found in the analysis. This also makes the written document unreliable, since errors may be incurred by manual copy-and-pasting the results into the document. A workflow of scientific authoring using *Microsoft Word* may resemble the one shown in figure 1.

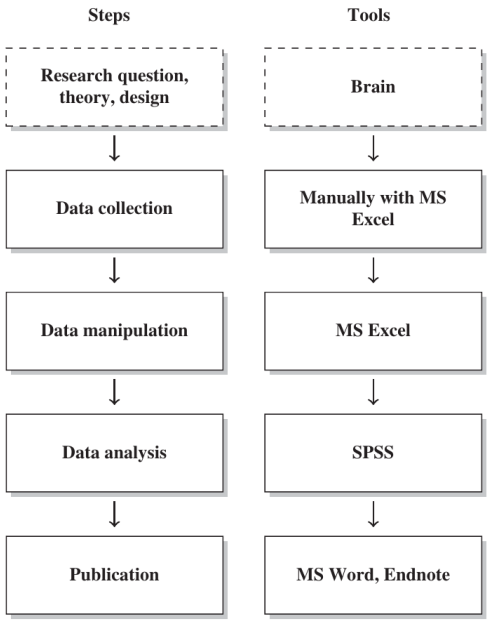


Figure 1: A common but not integrated workflow (Munzert, Rubba, Meißner, & Nyhuis, 2014, pp xviii).

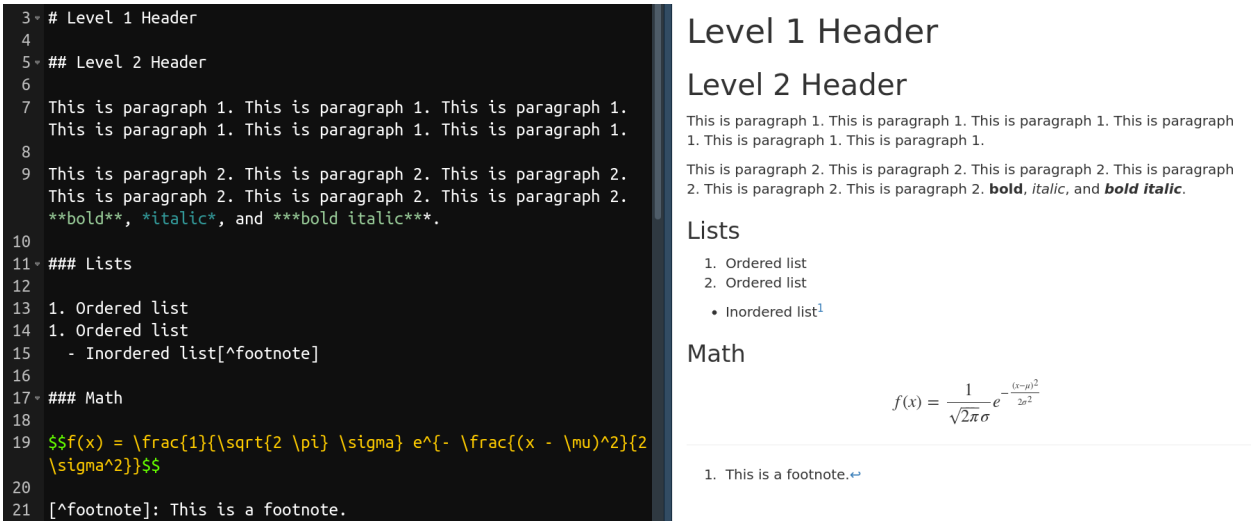
A reproducible and integrated workflow for scientific authoring, from data cleaning and analysis to document writing and generation, can be established using R and *R Markdown*. *R Markdown* has the capability of embedding executable R code, which automatically generates computed results, into the document. Hence, a document written with *R Markdown* is *dynamic*. It gets updated every time when regenerated. R's extensibility also makes other tasks, such as typesetting, easier, and it also enables *R Markdown* to be customized to fit specific needs in a particular field.

This article surveys the *R Markdown ecosystem* and try to illustrate the potential of using *R Markdown* to construct an integrated, reproducibility-facilitating, and user-friendly document writing environment for students and researchers in Linguitics.

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 simple plain text markers, such as #, −, and ∗. The syntax of Markdown is shown in figure 2.



```
# Level 1 Header

## Level 2 Header

This is paragraph 1. This is paragraph 1. This is paragraph 1.
This is paragraph 1. This is paragraph 1. This is paragraph 1.

This is paragraph 2. This is paragraph 2. This is paragraph 2.
This is paragraph 2. This is paragraph 2. This is paragraph 2.
**bold**, *italic*, and ***bold italic***.

### Lists

1. Ordered list
1. Ordered list
  - Inordered list[^footnote]

### Math

$$f(x) = \frac{1}{\sqrt{2 \pi} \sigma} e^{- \frac{(x - \mu)^2}{2 \sigma^2}}$$

[^footnote]: This is a footnote.
```


Figure 2: A demonstration of Markdown synatax. The left is the text content with markdown syntax. The right is the formatted document based on Markdown syntax.

"R Markdown", as it's name implies, combine R and Markdown, which enables users to embed computed results in a document written with Markdown syntax. In addition, *R Markdown* uses *Pandoc Markdown*[2], an enhanced version of Markdown that includes more versatile syntax, such as syntax for citations, footnotes, and tables. The R package *bookdown* (Xie, 2016) further expands R Markdown's ability, allowing authoring documents with publication-ready qualities.

[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] https://pandoc.org/MANUAL.html#pandocs-markdown

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

R is a programming language developed not in a traditional CS[3]-context but for the purpose of statistical computing (R Core Team, 2018). This makes R a 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, there is a huge and friendly community support for R, which means solutions to many problems new users often face can easily be found on the web.

Many fields other than Statistics either start to or already use R substantially, such as Biostatistics and 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-pasting 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.

[3] Abbreviation of "Computer science".

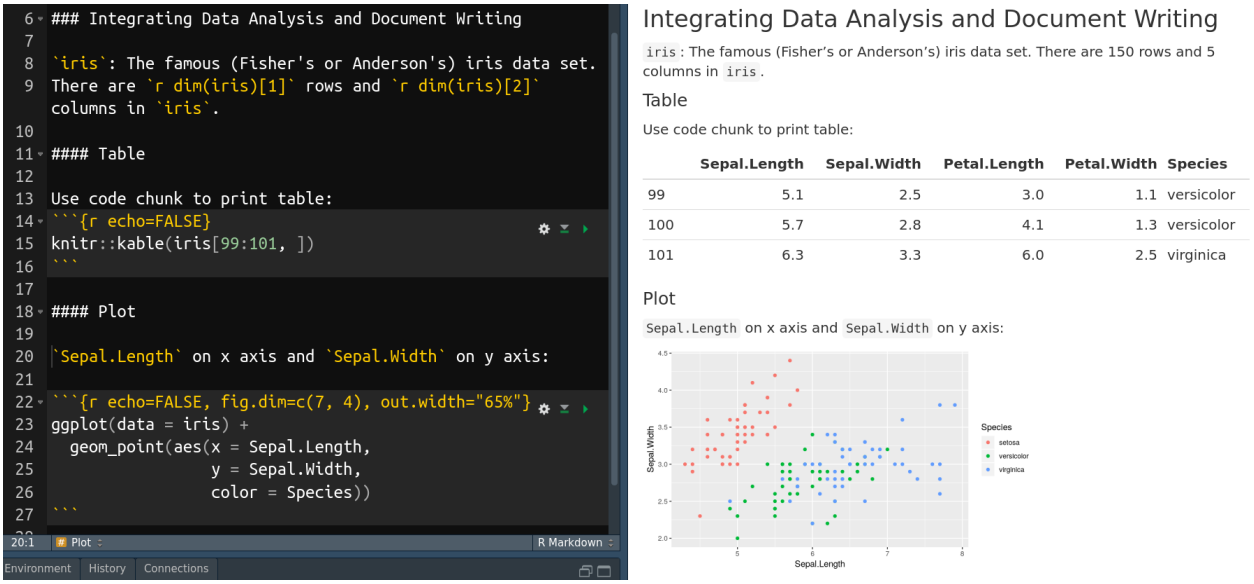


Figure 3: A demonstration of *R Markdown*. The left is the source of *R Markdown* file. The right is the document generated from *R Markdown*.

2.2.2 Supporting Reproducibility

A reproducibile analysis is an analysis with results that can be reproduced from raw data anywhere by anybody (Berez-Kroeker et al., 2018). The nature of R, a programming language, lends itself to facilitating reproducibility, since writing down the analysis as *code* is essentially *recording every step of the analysis*. Besides integration with R, two other features make *R Markdown* a good tool facilitating a reproducible workflow:

1. **R Markdown is plain text**

   Plain text format, contrary to binary files such as MS Word, doesn't require specialied and propriety software to read. People with no access to propriety softwares can still open and read the file, hence facilitates the openness of science.

   Plain text format also makes the file easiler to "version control", i.e, keeping track of the history of modifications, or versions, of the file by version control softwares. Although not directly

related to reproducibility, recording the history of modification protects the file against unanticipated errors that can break down the workflow, since one can always revert to previous versions of the version controled file.

2. **Python support**

   The new R package *reticulate* (Allaire, Ushey, & Tang, 2018) enhances the interoperability between Python and R in an R environment. It is now possible to run Python in R console. The new Python engine enabled by reticulate also solved a major drawback in previous versions of *R Markdown* – Python variables couldn't be shared across different code chunks. Now, Python code chunks share states in *R Markdown* and variables constructed in previous code chunks can be accessed, using either Python or R code. This gives *R Markdown* similar, if not better, power compared to Juyper notebook.

   Integrating R and Python is especially important for Linguistics, as many actively developed packages and libraries for Linguistics are written in Python. For example, the famous software for analying speech sound, Praat (Boersma, 2002), has a third-party support in Python, which allows accessing low-level functions in Praat using Python syntax (Jadoul, Thompson, & de Boer, 2018) instead of using the less popular Praat scripting.

   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. An analysis can be done in R, Python, or both. *R Markdown* can then act as a "glue", combining different parts of analysis together into an integrated and reproducible whole.

2.2.3 Easy Collaboration

When multiple authors need to work on the same document, it can become hard to manage the document. Things become easy when *R Markdown* is used. In cases where authors work on their own resposible sections of the document, each author can write her sections in a standalone *R Markdown* file. By using R Markdown's "child document" functionality[4], one can then combine multiple *R Markdown* files into a single file for document generation.

In cases where multiple authors need to work on the same section or when one author often overwrites another author's text, the aforementioned version control system and a web-based hosting service for version control, such as GitHub, can be used. Version control systems, such as Git, can automatically merge changes contributed by different authors into the same file. If there are conflicts in the changes, such as when two authors modified the same line in the file, the conflicts are recorded by Git and are can be solved by manual effort.

2.2.4 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 *Microsoft Word*, *Microsoft Powerpoint*, *LaTeX*, *PDF*, and *HTML*. There are also multiple styles of document, such as slides, books (Xie, 2016), journal papers, and even websites or blogs (Xie, Hill, & Thomas, 2017). This large variety of output formats enables authors to publish their works through different formats with the same underlying *R Markdown* file.

2.2.4.1 HTML Support

It can be argued that the most prominent output format of *R Markdown* is HTML. In the Digital Age, the

[4] https://yihui.name/knitr/demo/child/

web becomes a popular, if not dominant, way to distribute publications. Web pages enable displaying more varieties of contents, such as GIFs (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 visualization of complex 3D graphs by using animation to rotate 3D objects in the graphs (Wilke, 2018). Using HTML output thus enhances communication with the readers – the explanatory text is written next to dynamic visual elements, rather than links to external files or web pages.

Using HTML outputs also enables authors to self-publish the content through the web, making resources available to a wider audience. This is especially useful for educational purposes.

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 several examples that reduce the burden of writing Linguistics-related documents in *R Markdown*.

2.3.1 Bibliographies and Citations

It is necessary to include citations for academic writing, and many use EndNote, a reference management software, together with *Microsoft Word* to accomplish this task. *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 much more comfortable. For example, the R package *citr* (Aust, 2017) lets authors insert citations through a graphical interface,

where authors can search information (author, year, title, etc.) about the articles 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. *citr* integrates well with *Zotero*[5], 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[6].

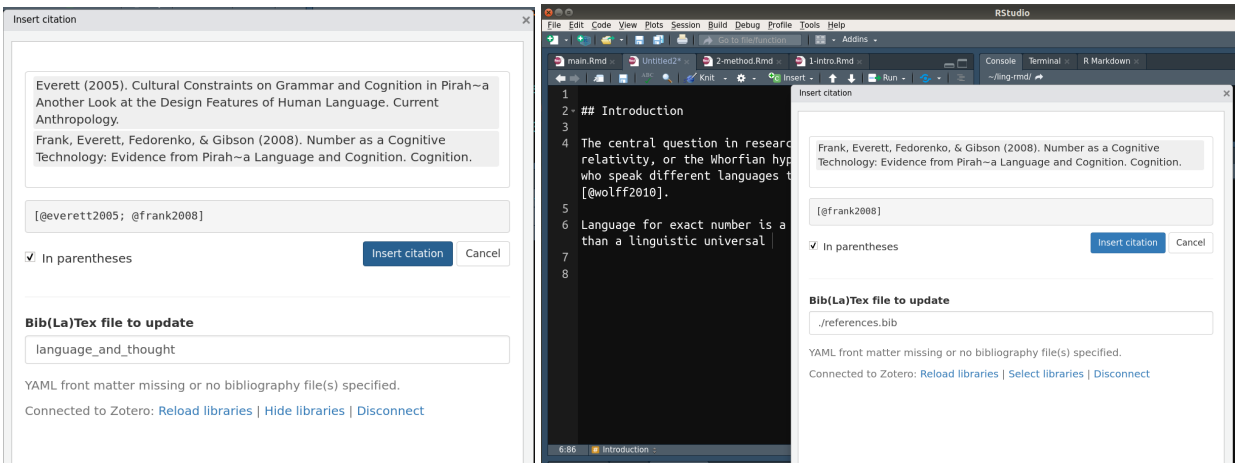


Figure 4: Using citr with *R Markdown*.

2.3.2 linguisticsdown

linguisticsdown is an R package developed by the author of this article to support Linguistics-related doucument writing in *R Markdown* (Liao, 2018). Currently, the package provides graphical interface as well as functions to facilitate writing IPA symbols in *R Markdown*.

Problem often arise when inserting IPA symbols into documents, since there is no simple way to type IPA symbols with the keyboard. *linguisticsdown* 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

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

[6]Citation Style Language. See https://citationstyles.org/ for details. For supported citation formats, see https://github.com/citation-style-language/styles.

ease.

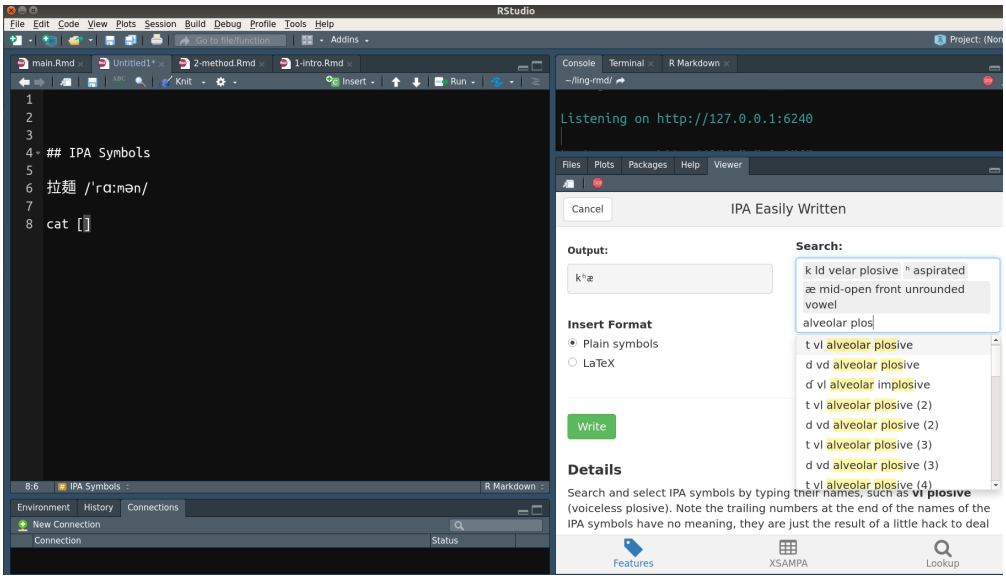


Figure 5: Using linguisticsdown with *R Markdown*.

2.3.3 Using Templates

As suggested previously, *R Markdown* can output formats meeting the requirements of journal submission. Authors can thus write documents with the simple *R Markdown* syntax without worrying about the typesetting. Currently, officially supported journal templates[7] are mostly journals under big publishers such as Elseiver, Springer, and SAGE. The range of support can be extended, however, as long as *LaTeX* templates are 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.

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

3 An Example Workflow

Figure 6 illustrates the flow of *text* and *data processing* in *R Markdown*. Based on figure 6, authors can do the following to facilitate an integrated workflow for document writing:

1. Put R or Python code that clean, manipulate, and analyze data in the *R Markdown* source file. If the code is too long, separate the code into R or Python scripts and include them into *R Markdown* by "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[8], raw data, and reports can be put together into a well-structured package (Flight, 2014; H. Wickham, 2015).

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

3. For values that are computed from data and need 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.

5. Use *linguisticsdown* to type IPA symbols.

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

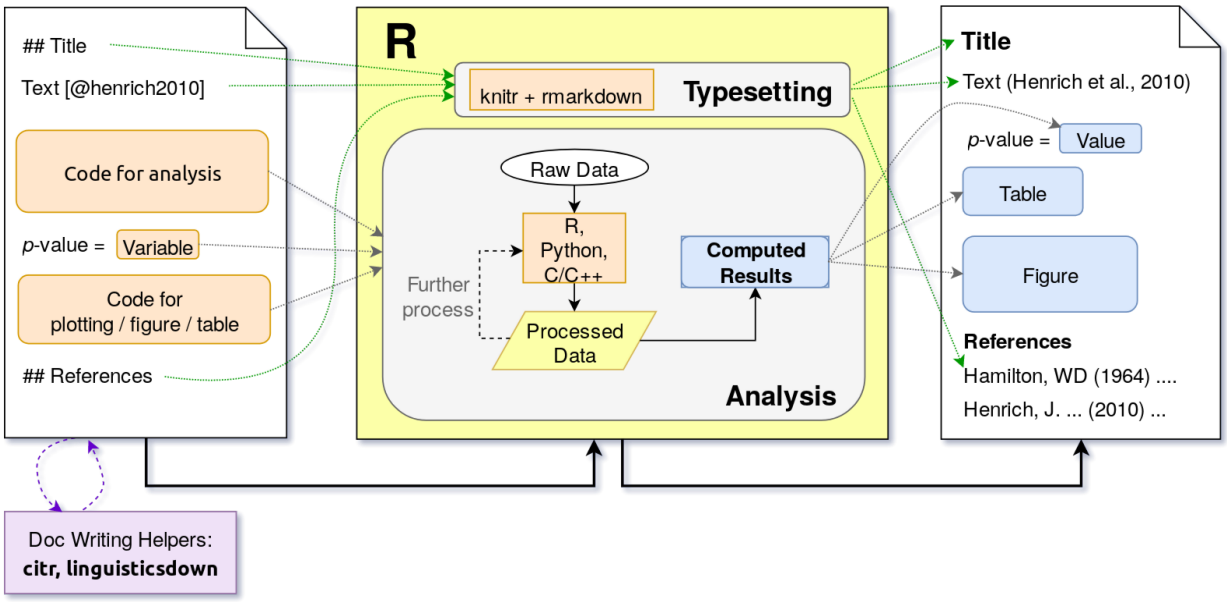


Figure 6: Combining R's data analysis and typesetting abilities to generate a reproducible report. The leftmost is a representation of an *R Markdown* source file; the rightmost is a generated document. The middle is the process underlying the conversion from source file to output document. For documents with complete template support, the typesetting is automated, so authors only need to focus on analysis and contents of the article.

4 Discussion

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 academic and scientific authoring. R Markdown's ability, however, depends much on its support. This article points out many strengths of *R Markdown* but also some current limitations. These limitations provide future directions to improve *R Markdown*, which can make it more suitable for document writing in Linguistics.

4.1 Template Support

For complete template support, 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, i.e, native *LaTeX* templates exist but are not modified to work natively in *R Markdown*, it can be extended to fully support *R Markdown* without too much efforts.

However, there are instances where no *LaTeX* templates exists. For example, many journal doesn't provide any template and only accepts submission in *Microsoft Word* format. This problem can be fixed by using the *Microsoft Word* output (`.docx`) provided by *R Markdown*. A Word template can also be set up manually (Layton, 2015) in advance, so the output format complies the one set in that template. Formatting thus has to be done only once in the Word template.

4.2 Extension Specific to Linguistics

R Markdown's capability depends on the R community. With more 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* a better authoring tool for Linguitics, a larger userbase is needed. Linguists familiar with R can also help to build the community by bundling regularly used custom functions into a package and making it availble to other users.

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

Knauff, M., & Nejasmic, J. (2014). An Efficiency Comparison of Document Preparation Systems Used in Academic Research and Development. *PLOS ONE*, *9*(12), 1–12. https://doi.org/10.1371/journal.pone.0115069

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.

Wilke, C. (2018). *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O'Reilly Media, Incorporated. Retrieved from https://books.google.com.tw/books?id=L3ajtgEACAAJ

Xie, Y. (2016). *Bookdown: Authoring Books and Technical Documents with R Markdown*. CRC Press.

Xie, Y., Hill, A. P., & Thomas, A. (2017). *Blogdown: Creating Websites with R Markdown*. CRC Press.