

# R Markdown as an Authoring Tool in Linguistics

*Yongfu Liao (廖永賦)\**

*September 17, 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. 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, such as report preparation, journal submission, and book authoring. Academic writing takes an author a great deal of time. However, much of the time spent *isn't related to the content or the idea the author wishes to convey* but the chores regarding repetitive works such as manually combining results from different analysis tools or formatting the papers to suit the requirements of journals.

Currently, two document preparation systems, Microsoft Word and LaTeX, predominate in academia. There are advantages and disadvantages in using both softwares. MS 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 flexibility regarding typesetting. The flexibility comes at a price, however. To achieve this flexibility, LaTeX has an extremely complex syntax and has been shown to hinder document writing (Knauff & Nejasmic, 2014).

One feature that MS 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 analyses need to be redone, such as when data are updated or some errors were 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 MS Word may resemble the one shown in figure 1.

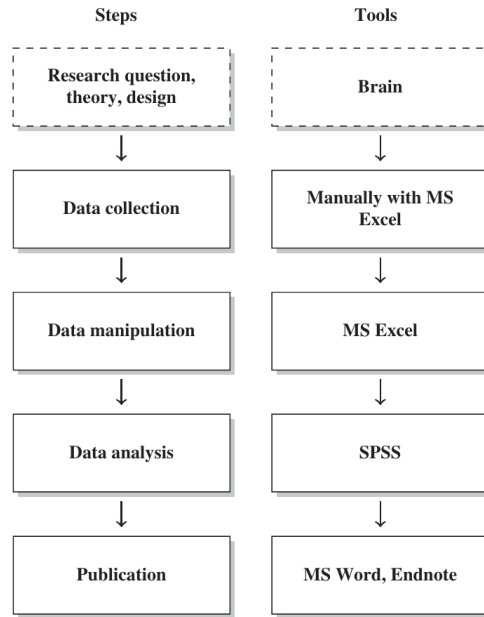


Figure 1: A common but not integrated workflow (Munzert, Rubba, Meißner, & Nyhuis, 2014, pp xviii).

A reproducible and integrated workflow for scientific authoring, from data cleaning and analysis to document generation and publication, can be established using R and R Markdown. R Markdown has the capability of embedding executable R code, which can generate and insert computed results into the document, hence creating a *dynamic document* that gets updated every time it is regenerated. R’s extensibility also makes other tasks, such as typesetting, easier. This extensibility also enables R Markdown to be customized to fit specific needs in a particular field. This article surveys the R Markdown “ecosystem” to provide an integrated workflow for authoring in Linguistics.

## 2 Overview of R Markdown

### 2.1 A Brief Introduction

“Markdown” is a minimalist and easy-to-learn markup language<sup>1</sup>, which formats the text by using plain text markers. Part of the syntax is illustrated in figure 2.

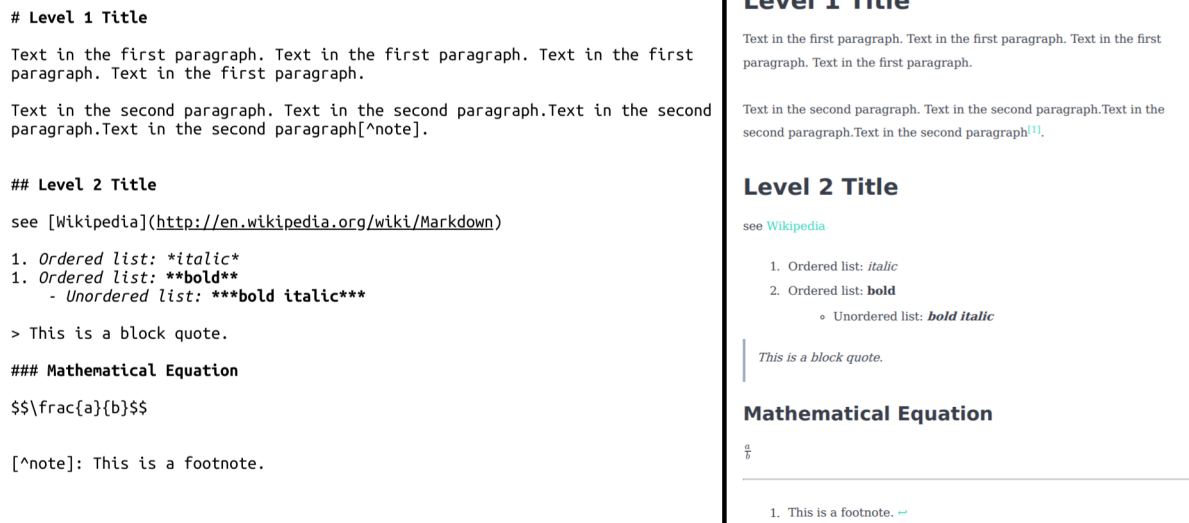


Figure 2: A simple demonstration of Markdown syntax.

“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 extensibility of R, the language R Markdown bases on.

<sup>1</sup>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.

### 2.2.1 R

R is a programming language developed not in a traditional CS<sup>2</sup>-context but for the purpose of statistical computing (R Core Team, 2018). This makes R a special language – although 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 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 Bioinformatics, Ecology and Evolution, Finance, Psychometrics, Geospatial analysis, and even Linguistics (CRAN, 2018b). This is due to R’s great extendibility, with more than 13,000 packages hosted on CRAN (2018a). As noted later, this extendibility 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 incurred by manual copy-and-pasting are eliminated. This also enhance the *reproducibility* 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 Reproducible 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

---

<sup>2</sup>Abbreviation of “Computer science”.

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 binary files such as MS<sup>3</sup> Word, doesn't require specialised (and often propriety) software to open, hence facilitates the openness of science. Plain text format also makes the file easier 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 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 similar power to Jupyter 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 phonetic software, 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).

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.

---

<sup>3</sup>Abbreviation of “Microsoft”.

### **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 (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.

#### **2.2.3.1 HTML Support**

It can be argued that the most prominent output format of R Markdown is HTML. In the Digital Age, the 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 enough 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. 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 extendibility. 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 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 formatting 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 (author, year, titles, 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<sup>4</sup>, 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<sup>5</sup>.

---

<sup>4</sup><https://www.zotero.org/>

<sup>5</sup>Citation Style Language. See <https://citationstyles.org/> for details. For supported citation formats, see <https://github.com/citation-style-language/styles>.



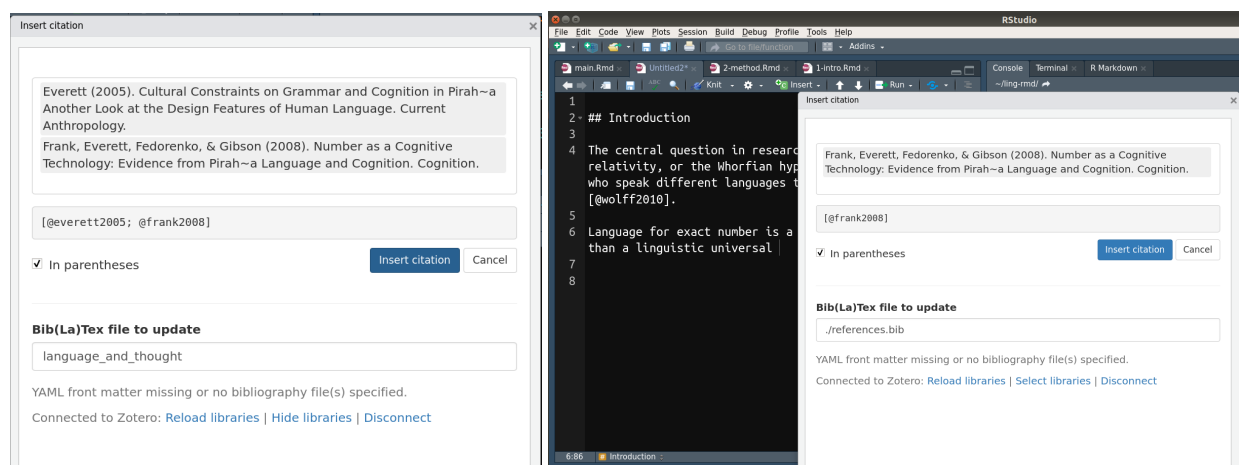


Figure 3: Using citr with R Markdown.

### 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.

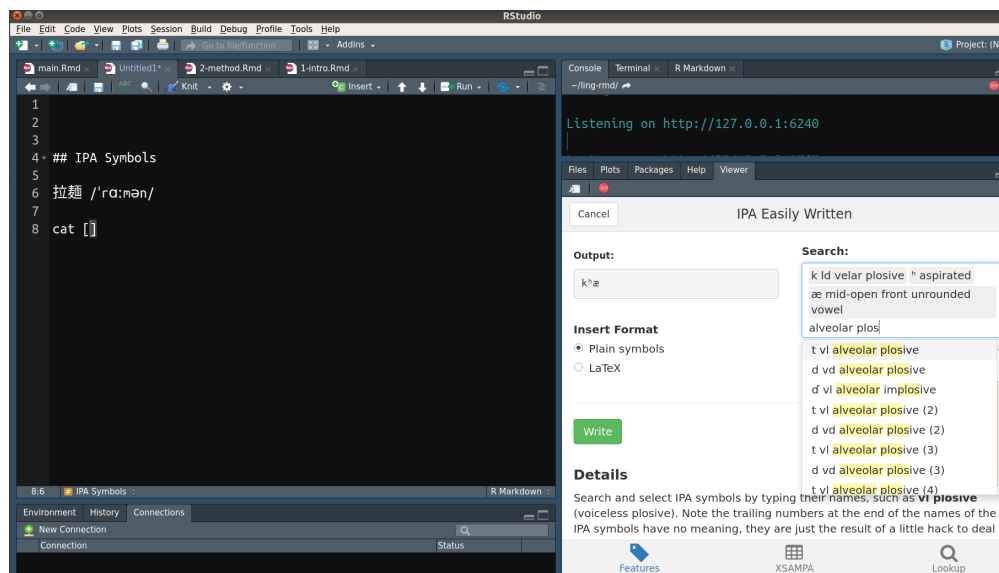


Figure 4: Using linguisticsdown with R Markdown.

### 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<sup>6</sup> are mostly journals under big publishers such as Elsevier, 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.

<sup>6</sup>For a list of supported journal templates, see <https://github.com/rstudio/rarticles>.

### 3 A Proposed Workflow

Figure 5 illustrates the flow of text (for readers) and data (for analysis) processing in R Markdown. Based on figure 5, 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<sup>7</sup>, 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.
5. Use linguisticsdown to type IPA symbols.

---

<sup>7</sup>For complex analyses, there are often long and repetitive code. Wrapping these redundant code into functions can make the analyses more manageable.

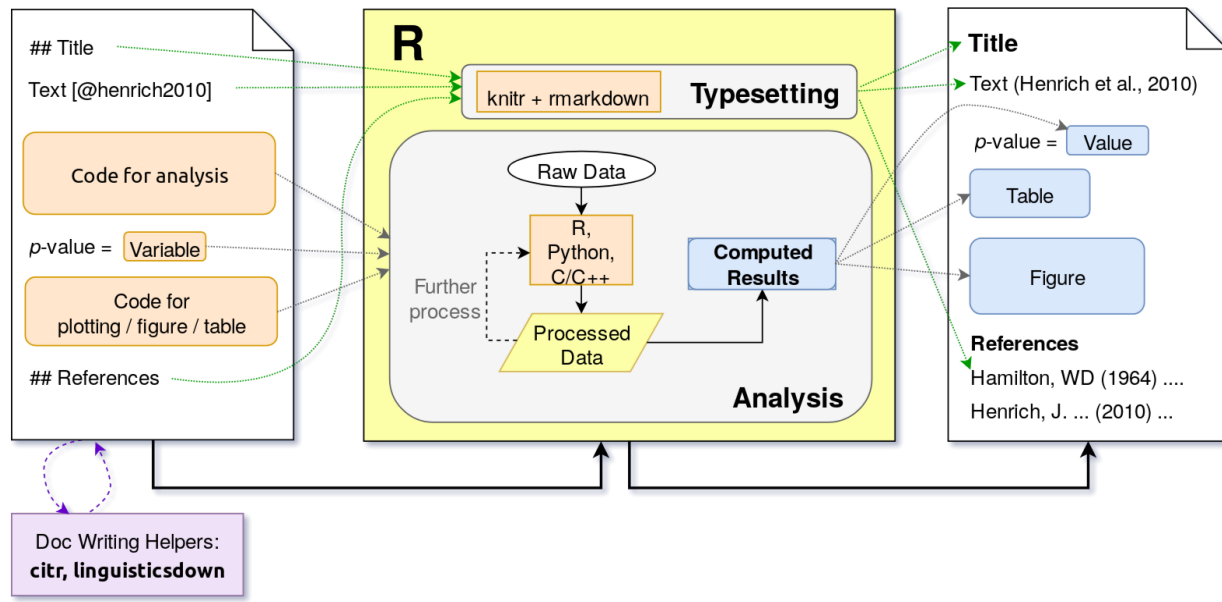


Figure 5: Combining R's data analysis and typesetting abilities to generate a reproducible report. The leftmost is a representation of an R Markdown source file; the rightmost is a generated document. The middle is the process underlying the conversion from source file to output document. For documents 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 ability 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, 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, the *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) 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.

#### **4.1.2 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 an authoring tool more suitable for Linguistics, a larger userbase is needed. Linguists familiar with R can also make functions that they used regularly available 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., ... Ögreden, 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. *Glott 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](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/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](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](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](https://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.

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.