R Markdown as an Authoring Tool in Linguistics

廖永賦*

September 13, 2018

Abstract

bold

*國立臺灣大學心理學系

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 wished 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 the 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 allowing them to build custom programs that facilitate an integrated workflow.

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

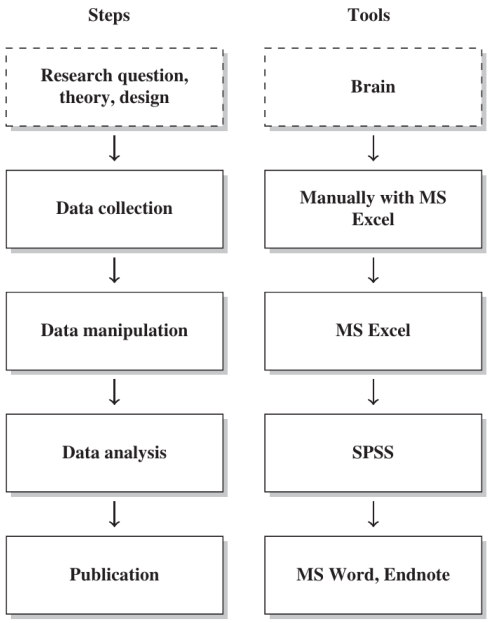

Figure 1: A common workflow (from Munzert, Rubba, Meißner, & Nyhuis, 2014, pp xviii)

One of the problem of this workflow is that authors have to manually copy-and-paste results from different sources, which is error-prone, and when errors are found, authors have to go through the process again, wasting a substantial time. An even more serious problem is that this workflow hinders the progress of science, since when errors occured but are not found, erroneous results enter a published paper, in which subsequent researches would 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 in specific needs in a field. This article surveys the R Markdown ecosystem, and focuses perticularly on features related to Linguistics.

R Markdown as an Authoring Tool

"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 "##" for 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.

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.

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. 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, e.g., Biostatistics and Bioinformatics, Ecology and Evolution, Finance, Psychometrics, Geospatial analysis, and even Linguitics (CRAN, 2018b). This is due to R's great extendible potential, with more than 13,000 packages hosted on CRAN (2018a).

[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 for Computer science.

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.

Supporting Reproducibility

A Reproducibile analysis is an analysis with results that can be generated directly from its underlying raw data anywhere by anybody [cite needed]. Besides the above mentioned properties of R and R Markdown, 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 Word, doesn't require specialied (and often propriety) software to open, and 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, and the new Python engine enabled by reticulate also solved a major drawback in previous versions of R Markdown – Python variables are now shared across different code chunks in R Markdown documents. This makes gives R Markdown the same power as the Juyper notebook.

   Integrating R and Python is especially important for Linguistics, as many actively developed packages and libraries related to Linguistics are written in Python. Some tasks that R certainly does better than Python are graphics and document authoring, hence, integrating R and Python combines

the strengths of both languages. R Markdown acts as a “glue” here, gluing different parts of analysis together into an integrated whole.

A Proposed Workflow

Discussion

References

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

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

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/

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/