Converting LaTeX R Journal Articles into Rmarkdown using texor and rebib

by Abhishek Ulayil, Mitch O'Hara-Wild, Christophe Dervieux, Heather Turner, and Dianne Cook

Abstract An abstract of less than 150 words.

1 Introduction

The R Journal is the primary open-access outlet for publications produced by the R community. It was born in 2008, evolving from a newsletter, that ran from 2001, into a more formal article publication to encourage documenting statistical computing research.

The format is constantly evolving. Early articles were typeset using LateX (The LaTeX Project 2023), from a specific, but changing, template. This requires that code is separated from the documentation, and there is a chance that code chunks in the paper don't reproduce the results reported. With the emergence of dynamic document systems like R Markdown (Xie, Allaire, and Grolemund 2018) a tight-coupling of code and documentation is possible. Code chunks are dynamically executed when the document is typeset using a system like **knitr** (Xie 2015), making reporting of computing research more reproducible.

In 2019, with the help of funding from the R Consortium it was decided that it was time to update operations. One aspect of this was to change from LaTeX paper submissions to a more reproducible format, where code was embedded in the document, and the output could be both HTML and pdf. There are numerous benefits of HTML format:

1. Articles can include interactive graphics and tables.
2. The format is more accessible to screen readers making the work more accessible to vision-impaired researchers.

This latter point is a reason to consider converting all of the legacy articles into HTML.

A key decision for creating conversion software was to decide to directly convert LaTeX to HTML, or PDF to HTML, or LaTeX to R Markdown, and then use the current journal tools to create the HTML. The latter approach was decided to be the most versatile and useful. If an article can be converted from LaTeX to R Markdown, it would help authors make the transition to reproducible publishing, beyond what the R Journal needed. Once an article is in R Markdown format it can be adapted to include the code for dynamic execution.

In addition to article format, changes to the web site structure were important for delivering the publication. Web site architectures are also constantly evolving, and the emergence of **distill** (Dervieux et al. 2022) allows for the journal web site to optimally deliver R Markdown articles.

The **rjtools** was developed to create articles using R Markdown for the R Journal, and to embed them into the journal web site. The packages described here, **texor** and **rebib** describes software to convert legacy LaTeX format articles into Rmarkdown, so that they can be rendered in HTML in the new web site.

The paper is organised as follows. Section XXX gives an overview of the conversion process. Section XXX describes examples of pre-processing using regular expressions, and section XXX provides examples of post-processing using lua filters. Section XXX describes tools to do special handling of bibliography files. Section XXX runs through an example conversion process.

2 Converting from LaTeX to R Markdown

The decision to convert to R Markdown format means that the final output to pdf and HTML will depend on Pandoc (MacFarlane 2023). Pandoc is a versatile document conversion program written in Haskell that is core to numerous documentation systems, including R Markdown and Quarto. Pandoc first converts a document into an abstract syntax tree. From this, it can convert to a different format, including custom ones.

Pandoc can be used to do the conversion from LaTeX to R Markdown also. However, additional pre-processing needs to be done to handle special R Journal LaTeX styling. And further post-processing needs to be done to handle specific R Journal R Markdown styling. The **texor** package contains functionality to handle this pre- and postprocessing of the document.

DC: Add a diagram here showing flow of work, just pre-process .tex -> pandoc -> lua ; rebib operations???

Pre-processing using regular expressions

LaTeX is very descriptive language, that allows authors substantial freedom for customization. Markdown (Gruber 2002), on which R Markdown is based, is more restrictive and was born to make it easier to create web pages without the distraction of a gazillion HTML tags. The purpose of Markdown is that it allows the author to focus on writing, without format cluttering the text.

While pandoc can do most of the heavy-lifting, it cannot cope with all the freedom with which LaTeX documents are written. An example of this is with formatting of code. Pandoc only handles the `verbatim` environment, but there are many ways to format code in LaTeX, and the R Journal template has a special `\code{}` command. If the code environment is not verbatim, then pandoc will also try to process the actual code content as LaTeX commands and will likely lose details. It is better to convert these synonyms into `verbatim` environments this prior to passing the document to pandoc.

The functions in **texor** that handle the pre-processing using regular expressions are:

- `stream_editor()`: operates like the `sed` function in unix (Ritchie and Thompson 1978) and allows generic text pattern matching and replacing.
- `patch_code_env()`: replaces the common code environments, `code`, `example`, `Sin`, `Sout`, `Scode`, `Sinput`, `smallverbatim`, `boxedverbatim`, `smallexample` with `verbatim`.
- `patch_equations()`: coordinates various equation environments.
- `patch_figure_env()`: coordinates various figure environments.
- `patch_table_env()`: coordinates table environments.

These functions are verbose and describe all the changes being made. They also create a backup of the original file before making the changes.

Post-processing using Lua filters

This was one of the earliest and most basic Lua filters, and it was part of the **texor** package:

```
function Div(el)
    if el.classes[1] == 'thebibliography' then
        return { }
    end
end
```

Above filter reads the abstract syntax tree and filters out all the Div elements. Then it looks for the class "thebibliography." It turns out that this Div element contains the LaTeX bibliographic records, which usually appear at the very end of papers. We don't need this portion in the article with the "RJ-web-article" layout, given its inclusion as meta-data in the footer.

3 Structure of the texor package

DC: List the main functions in an organised way, could be the pre-processing functions, the post-processing functions, utility functions. This can be also be summarized in a table, but there needs to be text explanations of the main functionality.

4 Using texor

Install instructions

Examples / getting started / usage and so on

5 Managing the bibliography using rebib

Overview

The **rebib** package addresses the issue with LaTeX articles using built-in bibliography options with or without BibTeX files. While this works well with LaTeX, it won't work with Rmarkdown. Initially the

goal was to use external software like Biber to convert the embedded bibliography to BibTeX. However the integration of external software was not viable, hence the experimental idea of a bibliography parser gained momentum.

It was initially a part of the **texor** package, as those functions grew in features and became more involved; at that point, it made sense to move those functions as a separate package. Initially, there were some reservations about the usage of **rebib** and its stability with various formats. However, the package has improved over time and proven to be a good performer.

Using rebib

Install instructions

Examples / getting started / usage and so on

6 Summary

Where this might be useful in the future, other applications.

7 Acknowledgments

pandoc "TODO : Later"

8 Supplementary materials

github repos for the packages, texor, rebib, rjtools

References

Dervieux, Christophe, JJ Allaire, Rich Iannone, Alison Presmanes Hill, and Yihui Xie. 2022. *Distill: 'R Markdown' Format for Scientific and Technical Writing*. https://CRAN.R-project.org/package=distill.

Gruber, John. 2002. "Markdown." https://daringfireball.net/projects/markdown/.

MacFarlane, John. 2023. "Pandoc: A Universal Document Converter." https://pandoc.org.

Ritchie, O. M., and K. Thompson. 1978. "The UNIX Time-Sharing System." *The Bell System Technical Journal* 57 (6): 1905–29. https://doi.org/10.1002/j.1538-7305.1978.tb02136.x.

The LaTeX Project. 2023. "LaTeX – a Document Preparation System." https://www.latex-project.org.

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Abhishek Ulayil
Student, Institute of Actuaries of India
Mumbai, India
ORCiD: 0009-0000-6935-8690
perricoq@outlook.com

Mitch O'Hara-Wild
Monash University
Melbourne, Australia
ORCiD: 0000-0001-6729-7695
mail@mitchelloharawild.com

Christophe Dervieux
Posit PBC
Paris, France
ORCiD: 0000-0003-4474-2498
christophe.dervieux@gmail.com

Heather Turner
University of Warwick
Newport, United Kingdom
ORCiD: 0000-0002-1256-3375
ht@heatherturner.net

Dianne Cook
Monash University
Melbourne, Australia
ORCiD: 0000-0002-3813-7155
dicook@monash.edu