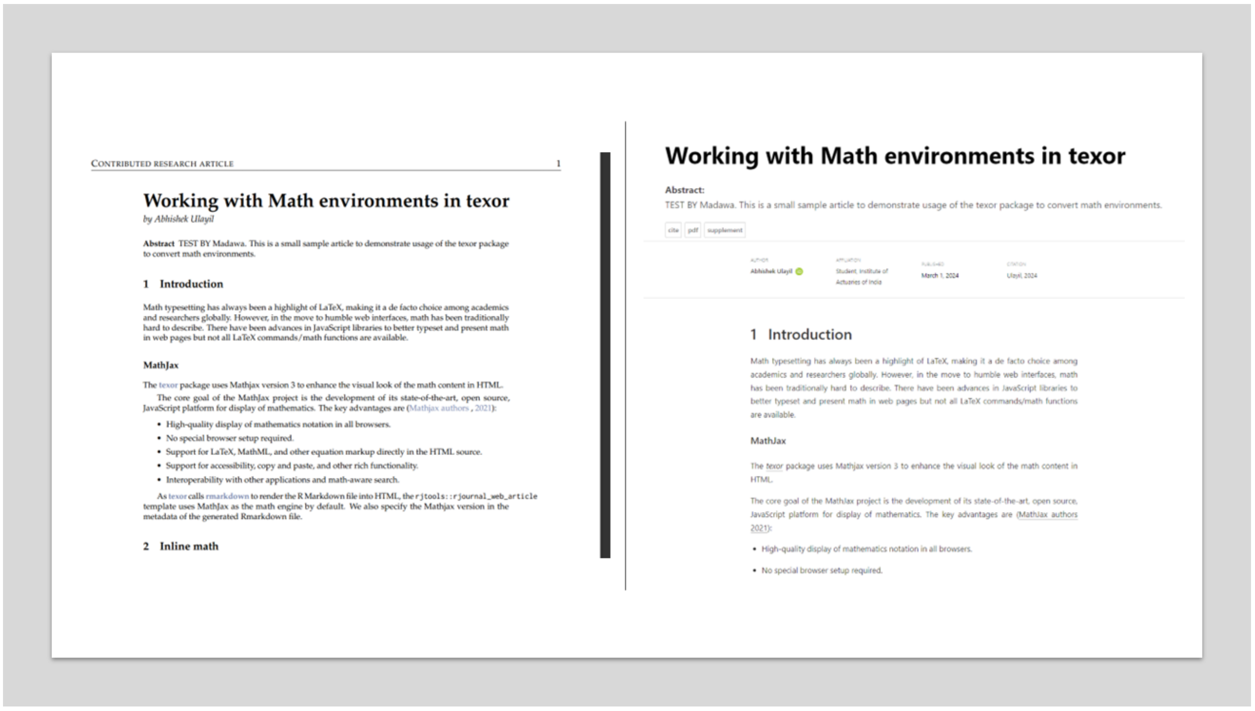
Sweave to Rmarkdown Test Using texor Package

Madawa

2024-03-30

Contents

1 Easy



CONTRIBUTED RESEARCH ARTICLE 1

Working with Math environments in texor

by Abhishek Ulayil

Abstract TEST BY Madawa. This is a small sample article to demonstrate usage of the texor package to convert math environments.

1 Introduction

Math typesetting has always been a highlight of LaTeX, making it a de facto choice among academics and researchers globally. However, in the move to humble web interfaces, math has been traditionally hard to describe. There have been advances in JavaScript libraries to better typeset and present math in web pages but not all LaTeX commands/math functions are available.

MathJax

The texor package uses Mathjax version 3 to enhance the visual look of the math content in HTML.

The core goal of the MathJax project is the development of its state-of-the-art, open source, JavaScript platform for display of mathematics. The key advantages are (Mathjax authors , 2021):

- High-quality display of mathematics notation in all browsers.
- No special browser setup required.
- Support for LaTeX, MathML, and other equation markup directly in the HTML source.
- Support for accessibility, copy and paste, and other rich functionality.
- Interoperability with other applications and math-aware search.

As texor calls rmarkdown to render the R Markdown file into HTML, the `rjtools::rjournal_web_article` template uses MathJax as the math engine by default. We also specify the Mathjax version in the metadata of the generated Rmarkdown file.

2 Inline math

Working with Math environments in texor

Abstract:
TEST BY Madawa. This is a small sample article to demonstrate usage of the texor package to convert math environments.

AUTHOR	AFFILIATION	PUBLISHED	CITATION
Abhishek Ulayil	Student, Institute of Actuaries of India	March 1, 2024	Ulayil, 2024

1 Introduction

Math typesetting has always been a highlight of LaTeX, making it a de facto choice among academics and researchers globally. However, in the move to humble web interfaces, math has been traditionally hard to describe. There have been advances in JavaScript libraries to better typeset and present math in web pages but not all LaTeX commands/math functions are available.

MathJax

The *texor* package uses Mathjax version 3 to enhance the visual look of the math content in HTML.

The core goal of the MathJax project is the development of its state-of-the-art, open source, JavaScript platform for display of mathematics. The key advantages are (MathJax authors 2021):

- High-quality display of mathematics notation in all browsers.
- No special browser setup required.

Figure 1: A sample image

Q:Convert LaTex article article to HTML

Below code snippet show how that works and specifically converting them for web use and managing bibliographies

```r
article_dir <- ("D:\\GSoC Research\\Mentor test - Converting Sweave to Rmarkdown
using texor package\\Test\\git_repo\\supplement\\math-env")
#texor::latex_to_web(article_dir)
```

2 Medium

Q: Write an R function that takes an example Markdown file with multiple figures and converts it to HTML

2.1 Sample Images That Used

The function is to provide a perfect way to convert Markdown documents to HTML format using R, with the additional ability to apply custom transformations via Lua filters. This can be useful for batch processing of documents, applying consistent formatting rules, or integrate custom processing steps that are not directly supported by Markdown or HTML. After completing conversion it confirm with text message.

```r
#' @param markdown_file Path to the Markdown file to be converted.
#' @param lua_filter_path Path to the Lua filter file for image numbering (optional).
#' @param output_file Path where the HTML output should be saved.
#' @return Invisible NULL. The function prints a message indicating whether
#' the conversion was successful or if it failed.
#' @examples
#' @export

convert_md_to_html_with_lua_filter <-
  function(markdown_file, lua_filter_path = NULL, output_file = NULL) {
  if (is.null(output_file)) {
    output_file <- gsub("\\.md$", ".html", markdown_file)
  }

  pandoc_args <- c("--from","markdown", "--to","html5", "--output",output_file)

  if (!is.null(lua_filter_path)) {
    pandoc_args <- c(pandoc_args, "--lua-filter", lua_filter_path)
  }

  # === Attempt the conversion and catch any errors ===
  tryCatch({
    rmarkdown::pandoc_convert(markdown_file, options = pandoc_args)
    cat("Conversion completed successfully: ", output_file, "\n")
  }, error = function(e) {
    cat("Conversion failed with error: ", e$message, "\n")
  })


  invisible(NULL)
}

convert_md_to_html_with_lua_filter("example.md", "image_numbering_filter.lua",
                                   "filtered-example.html")
```

Conversion completed successfully: filtered-example.html

3 Hard

Q: Write a Custom Pandoc Reader in Lua to just extract code chunks from Sweave files

This allows for custom processing of documents where code blocks and text follow a specific notation that's not natively recognized by Pandoc.

3.1 Example 1

```
library(tools)

sweave_file <- "Hard/030-landscape.Rnw"

lua_reader <- "sweave_reader.lua"

output_format <- "native"

# === Pandoc command for results ===
pandoc_command <- paste("pandoc", sweave_file, "-f", lua_reader, "-t", output_format)

output <- system(pandoc_command, intern = TRUE)

cat(output, sep = "\n")
```

```
## [ CodeBlock
##     ( "" , [] , [ ( "attributes" , "results='asis'" ) ] )
##     "library(knitr)\nkable(head(iris))"
## , CodeBlock
##     ( ""
##     , []
##     , [ ( "attributes"
##         , "results='asis', message=FALSE, warning=FALSE"
##         )
##       ]
##     )
##     "library(Hmisc)\nlatex(head(iris), file='', landscape=TRUE)"
## ]
```

3.2 Example 2

```
## [ CodeBlock
##     ( ""
##     , []
##     , [ ( "attributes"
##         , "setup, tidy.opts=list(width.cutoff=60)"
##         )
##       ]
##     )
##     "library(knitr)\nknit_hooks$set(document = function(x) {\n  gsub('(\\\\\\\\end\\\\{knitrout\\\\}
## , CodeBlock
##     ( "" , [] , [ ( "attributes" , "" ) ] ) "# hello world\n1+1"
## ]
```

3.3 Example 3

```
## [ CodeBlock
##     ( ""
##     , []
##     , [ ( "attributes" , "persp-plot, cache=TRUE" ) ]
##     )
##     "x <- seq(-10, 10, length= 30)\ny <- x\nf <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
## ]
```