Package ‘mathjaxr’

December 1, 2025

Version 2.0-0

Date 2025-12-01

Title Using 'Mathjax' in Rd Files

Description Provides 'MathJax' and macros to enable its use within Rd files for rendering equations in the HTML help files.

License GPL-3

Encoding UTF-8

BuildManual TRUE

URL `https://github.com/wviechtb/mathjaxr`

BugReports `https://github.com/wviechtb/mathjaxr/issues`

Contents

`mathjaxr-package` *Using MathJax in Rd Files*

Description

The **mathjaxr** package allows for easy inclusion of MathJax equations in Rd files. Package authors wanting to make use of the package and its functionality need to:

1. install the **mathjaxr** package,
2. add `mathjaxr` to `Imports` in the ‘`DESCRIPTION`’ file of their package,
3. add `mathjaxr` to `RdMacros` in the ‘`DESCRIPTION`’ file of their package (or add `RdMacros: mathjaxr` if the ‘`DESCRIPTION`’ file does not yet contain a `RdMacros` entry),
4. add `BuildManual: TRUE` to the ‘`DESCRIPTION`’ file,

5. add `import(mathjaxr)` in the ‘`NAMESPACE`’ file of their package.

One can then enable the use of MathJax by calling the `\loadmathjax` macro (that is provided by the **mathjaxr** package) within the `\description{}` section of an Rd file (or within the `@description` section when using **roxygen2**).

An inline equation can then be added with the `\mjeqn{latex}{ascii}` macro, with the LaTeX commands for the equation given between the first set of curly brackets (which will be rendered in the HTML and PDF help pages) and the plain-text version of the equation given between the second set of curly brackets (which will be shown in the plain-text help). With the `\mjdeqn{latex}{ascii}` macro, one can add ‘displayed equations’ (as in LaTeX’s `displaymath` environment).

Single argument versions of these macros, `\mjseqn{latexascii}` and `\mjsdeqn{latexascii}`, are also available. In case that one must specify different LaTeX commands for the PDF and HTML pages, there are also triple argument versions of these macros, namely `\mjteqn{pdflatex}{htmllatex}{ascii}` and `\mjtdeqn{pdflatex}{htmllatex}{ascii}`.

Details

The JavaScript code for MathJax is contained in this package. If a user viewing a help page has **mathjaxr** installed, it will be retrieved from there, otherwise it will be retrieved from the CDN site `https://cdn.jsdelivr.net/npm/mathjax@4/tex-mml-chtml.js`. To force use of the CDN site, the user can set the environment variable `MATHJAXR_USECDN` to any non-blank value (e.g., `Sys.setenv(MATHJAXR_USECDN=TRUE)`). The URL for a diferent CDN can be specified via the environment variable `MATHJAXR_CDN`.

Notes/Issues

- Care must be taken when using the less-than and greater-than symbols in equations as these might get interpreted by the browser as HTML tags. See here for further details. The best solution is to avoid these symbols completely for the HTML pages and instead use the `\lt` and `\gt` macros provided by MathJax. However, these macros cause problems when rendering the PDF help pages. Hence, one should then use `\mjteqn{}` and `\mjtdeqn{}` to specify different LaTeX commands for the PDF and HTML pages. For example, `\mjteqn{i < j}{i \lt j}{i < j}` will work across all output formats and does not cause any problems for the HTML help pages. For the less- and greater-than-or-equal-to symbols, one can use the `\le` and `\ge` macros that work across all output formats.
- Curly braces/brackets in equations also cause problems. Using `\lbrace` and `\rbrace` (possibly in combination with `\left` and `\right` to make them sufficiently large) is a solution (e.g., `\mjeqn{\left\lbrace ... \right\rbrace}{\{...\}}` should render nicely in the PDF/HTML help pages and the plain-text version).
- Using the percent symbol (i.e., %) inside of equations is also problematic. The percent symbol needs to be ‘escaped’ by using a backslash, but backslashes need to be escaped as well. For this to work, we need to use the correct number of backslashes, which works slightly differently for producing the PDF, HTML, and plain-text help pages. The equation `\mjteqn{100\\\%}{100\\\\\\\%}{100\%}` should be rendered correctly in all three help pages.
- While MathJax supports a large number of LaTeX commands, only the math-mode commands are implemented. See here for a list of the supported commands.

- When using R via an RStudio Server, MathJax equations in the help files are not automatically rendered (this is not a problem on RStudio Desktop). A solution to get the rendering to work in RStudio Server is to add ‘‘`server-add-header=X-Content-Type-Options:`” to ‘`/etc/rstudio/rserver.conf`’ (and then restarting the server). This of course requires admin rights on the machine running the server.

Example

The probability density function of a normal distribution is given by

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

where μ denotes the mean of the distribution and σ its standard deviation.

Author(s)

Wolfgang Viechtbauer <`wvb@wvbauer.com`> `https://www.wvbauer.com/`

See Also

`preview_rd`

`preview_rd` *Preview rendered version of an Rd file*

Description

Function to preview the rendered version of an Rd file.

Usage

```
preview_rd(Rdfile, view = TRUE, type = "html", verbose = FALSE, dark = FALSE, ...)
```

Arguments

`Rdfile`	character string with the name of the Rd file to preview (either with or without the `.Rd` or `.rd` extension).
`view`	logical indicating whether the rendered version of the help file should be displayed.
`type`	character string indicating which version should be rendered (either `"html"`, `"txt"`/`"text"`, or `"pdf"`).
`verbose`	logical indicating whether diagnostic output will be shown.
`dark`	logical indicating whether the rendererd HTML page should use a dark mode.
`...`	other arguments.

Details

The function is useful when writing a help file that contains MathJax equations. Instead of having to reinstall the package under development to check if the equations are being rendered correctly, one can just set the current working directory to the root of the package (or its `man` directory) and then use `preview_rd()` to preview the HTML, plain-text, or PDF version of an Rd file on the fly.

For `type="html"`, the HTML page will be opened in the browser with the `browseURL` function. When making further changes to the Rd file, reopening the page each time with `preview_rd()` is inconvenient as this will usually open up a new tab in the browser. Setting `view=FALSE` prevents this. Reloading the page in the open tab should then reflect the updates. In RStudio, the generated HTML version will be displayed in the 'Viewer' pane and the `view` argument is then irrelevant.

Setting `dark=TRUE` (in combination with `type="html"`) renders the HTML page in dark mode. Unfortunately, this does not work in RStudio.

For `type="txt"` (or `type="text"`), the plain-text version of the help file will be shown (using the `file.show` function and directly on the console in RStudio).

For `type="pdf"`, the PDF is generated using `R CMD Rd2pdf` and should open up in the default PDF viewer.

Note

Any links to local help pages used in the Rd file to preview will be shown in the rendered HTML or PDF help file, but are non-functional.

Due to some limitations as to how MathJax can be loaded via the **mathjaxr** package, MathJax must be loaded via the CDN when using the `preview_rd()` function. Hence, rendering of equations in HTML will only work in the preview with an active internet connection.

Author(s)

Wolfgang Viechtbauer <`wvb@wvbauer.com`> `https://www.wvbauer.com/`

Examples

```
## Not run:
setwd("/path/to/root/of/package")
preview_rd("someRdfile")

## End(Not run)
```

Index