Examples

Macros

Set of real number

LaTeX: `\newcommand \RR {\mathbb{R}}`

MathJax: `RR: "{\\mathbb{R}}"`

Call: `$\RR$`

Rendering: $\RR$

Normal distribution

LaTeX: `\newcommand \normal [2] {\mathcal{N}\left({#1},{#2}\right)}`

MathJax: `normal: ["{\\mathcal{N}\\left({#1},{#2}\\right)}", 2]`

Call: `$\normal{0}{\sigma^2}$`

Rendering: $\normal{0}{\sigma^2}$

Differential operator

LaTeX: `\newcommand \dx [2] [] {\mathrm{d}^{#1}\mspace{-1mu}\mathord{#2}}`

MathJax: `dx: ["{\\mathrm{d}^{#1}\\mspace{-1mu}\\mathord{#2}}", 2, ""]`

Call: `$\dx{x} + \dx[2]{f}$`

Rendering: $\dx{x} + \dx[2]{f}$

Colors in maths mode

LaTeX: `\usepackage{color}` + `\newcommand \warning [1] {\textcolor{red}{#1}}`

MathJax: `warning: ["\\color{red}{{#1}}", 1]`

Call: `$(a+b)^2 = a^2 + \warning{2ab} + b^2$`

Rendering: $(a+b)^2 = a^2 + \warning{2ab} + b^2$

Colors in text mode

As MathJax doesn't work with text mode, to have a consistent method between different outputs, you could use an R function to write raw HTML or LaTeX code depending on the output.

```
colorize <- function(x, color) {
  if (knitr::is_latex_output()) {
    sprintf("\\textcolor{%s}{%s}", color, x)
  } else if (knitr::is_html_output()) {
    sprintf("<span style='color: %s;'>%s</span>", color, x)
  } else {
    x
```

```
  }
}
```

Call: `r colorize("Hello world!", "blue")` (enclosed with backsticks)

Rendering: Hello world!

Writing and including macros files into `.Rmd`

File `macros.tex`

```
\usepackage{color}
\newcommand \RR {\mathbb{R}}
\newcommand \normal [2] {\mathcal{N}\left({#1},{#2}\right)}
\newcommand \dx [2] [] {\mathrm{d}^{#1}\mspace{-1mu}\mathord{#2}}
\newcommand \warning [1] {\textcolor{red}{#1}}
```

File `macros.html`

```
<script type="text/x-mathjax-config">
MathJax.Hub.Config({
  TeX: {
    Macros: {
      RR: "{\\mathbb{R}}",
      normal: ["{\\mathcal{N}\\left({#1},{#2}\\right)}", 2],
      dx: ["{\\mathrm{d}^{#1}\\mspace{-1mu}\\mathord{#2}}", 2, ""],
      warning: ["\\color{red}{{#1}}", 1]
    }
  }
});
</script>
```

`.Rmd` YAML

```
---
title: "Examples"
output:
  pdf_document:
    includes:
      in_header: macros.tex
  html_document:
    includes:
      before_body: macros.html
---
```