R Markdown

David Aceituno

Contents

R Markdown

This is an R Markdown document compiled with knitr. Normal Markdown syntax applies. For instance, the title above was written as `# Introduction`.

Math

Regular math can be done by `MathJax` where using double dollar signs will create a latex-style mathematical chunk.

$$y = kx + m$$

Latex

We can write plain latex directly in this document, which will call knitr -> pandoc -> pdflatex for that block of code. For instance, the following block includes a .png image.

```
1  \begin{figure}[h!]
2  \centering
3  \includegraphics[width=0.3\textwidth]{hydrostaticparadox.png}
4  \caption{Hydrostatic paradox}
5  \end{figure}
```

Tikz

There are two ways of using tikz. Either we write it as an R chunk like
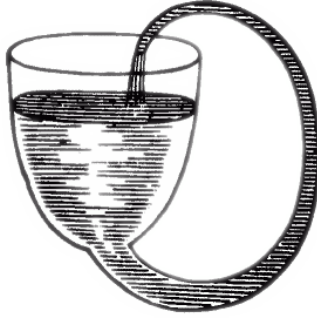
```
```{r,engine='tikz'}\n tikzcode \n```
```

Figure 1: Hydrostatic paradox

where backslash n are newlines, or directly into the file, like

```
1 \usetikzlibrary{arrows}
2 \centering
3 \begin{tikzpicture}
4 tikzcode
5 \end{tikzpicture}
```

Tensors

We can draw tensors after \usepackage{tensorgraphics} which calls commands in the file tensorgraphics.sty.

Rank 2 (matrix)

\tensorII{name}{left index}{right index}

$$\alpha\underset{}{\boxed{A}}\beta$$

Rank 3

\tensorIII{name}{physical index}{left index}{right index}

$$\overset{i}{\underset{j\ \boxed{\Gamma}\ k}{}}$$

Rank 4

\tensorIV{name}{physical index 1}{physical index 2}{left index}{right index}

$$\overset{i\quad j}{k\ \boxed{\theta}\ l}$$

These figures also work in a math environment, as seen below.

$$\alpha \boxed{\;\theta\;}^{\,j\;\;k}\,\beta \;=\; \alpha \boxed{\Gamma^A}^{\,j}\,\gamma \;+\; \gamma \boxed{\Gamma^B}^{\,k}\,\beta \tag{1}$$

Matrix Product States

An MPS can be drawn using

`\mps{name}{length}{physical index}{left index}{right index}`

$$\alpha \boxed{A^1}^{\,\sigma_1}\boxed{A^2}^{\,\sigma_2}\boxed{A^3}^{\,\sigma_3}\boxed{A^4}^{\,\sigma_4}\beta$$

Similarly, the Vidal canonical form of an MPS can be drawn with `\mpsVidal{name rank 3}{name singular values}{length}{physical index}{left index}{right index}`

$$\alpha \boxed{\Gamma^1}^{\,\sigma_1}\boxed{\Lambda^1}\boxed{\Gamma^2}^{\,\sigma_2}\boxed{\Lambda^2}\boxed{\Gamma^3}^{\,\sigma_3}\boxed{\Lambda^3}\boxed{\Gamma^4}^{\,\sigma_\Lambda}\beta$$

Rcpp

We can write and execute C++ code directly in an `Rcpp` chunk, such as the one below. I don't know yet how to see the output from std::cout.

```cpp
#include <iostream>
// [[Rcpp::export]]
int myfunction(){
  std::cout << " HEJ" << std::endl;
  return 0;
}
```

The Rcpp::export line makes `myfunction()` available to the R environment, so it can be called like

```r
myfunction()
```

```
## [1] 0
```

More about R Markdown

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
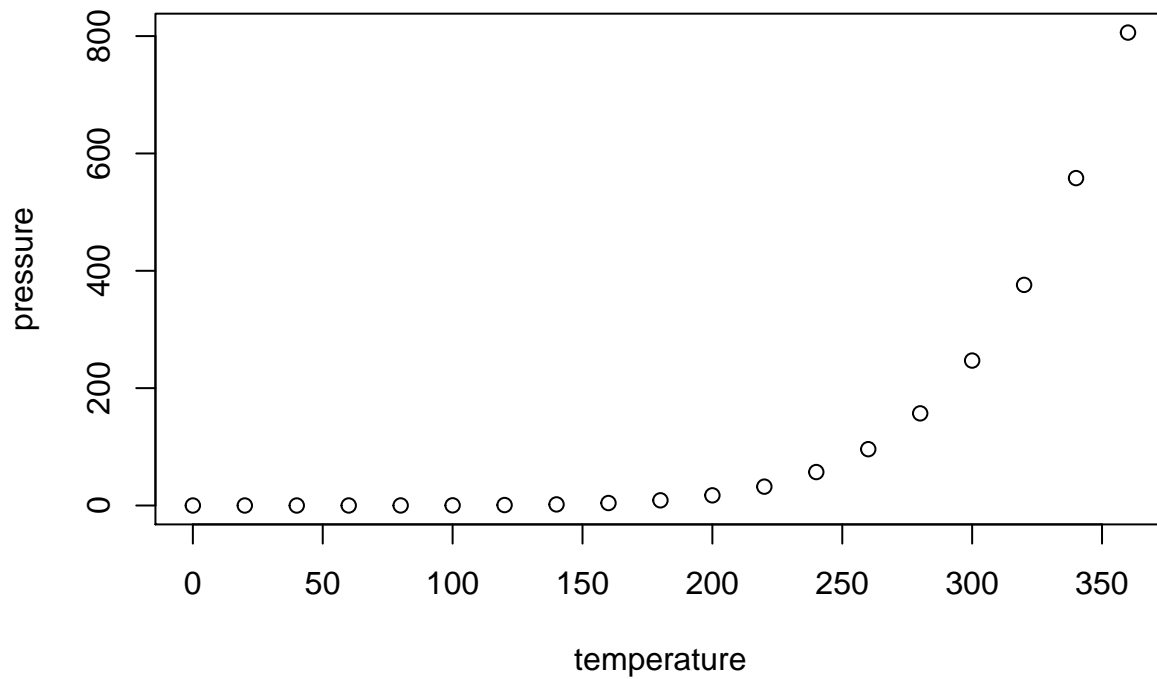summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.