

R Markdown

David Aceituno

Contents

R Markdown	1
Math	1
Latex	1
Tikz	1
Tensors	2
Rank 2 (matrix)	2
Rank 3	2
Rank 4	2
Matrix Product States	3
Rcpp	3
More about R Markdown	3
Including Plots	4

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```
```

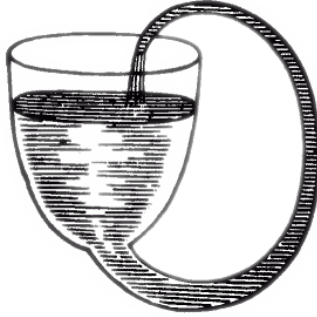


Figure 1: Hydrostatic paradox

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}
```

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

### Rank 3

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

$$j - \boxed{\Gamma}^i - k$$

### Rank 4

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

$$k - \boxed{\theta}^{ij} - l$$

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

$$\alpha \begin{array}{c} j \\ | \\ \boxed{\theta} \\ | \\ k \end{array} \beta = \alpha \begin{array}{c} j \\ | \\ \boxed{\Gamma^A} \end{array} \gamma + \gamma \begin{array}{c} k \\ | \\ \boxed{\Gamma^B} \end{array} \beta \quad (1)$$

## Matrix Product States

An MPS can be drawn using

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

$$\alpha \begin{array}{c} \sigma_1 \\ | \\ \boxed{A^1} \end{array} \begin{array}{c} \sigma_2 \\ | \\ \boxed{A^2} \end{array} \begin{array}{c} \sigma_3 \\ | \\ \boxed{A^3} \end{array} \begin{array}{c} \sigma_4 \\ | \\ \boxed{A^4} \end{array} \beta$$

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}`

$$\alpha \begin{array}{c} \sigma_1 \\ | \\ \boxed{\Gamma^1} \end{array} \begin{array}{c} \Lambda^1 \end{array} \begin{array}{c} \sigma_2 \\ | \\ \boxed{\Gamma^2} \end{array} \begin{array}{c} \Lambda^2 \end{array} \begin{array}{c} \sigma_3 \\ | \\ \boxed{\Gamma^3} \end{array} \begin{array}{c} \Lambda^3 \end{array} \begin{array}{c} \sigma_\Lambda \\ | \\ \boxed{\Gamma^4} \end{array} \beta$$

## 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`.

```
#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

```
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>.

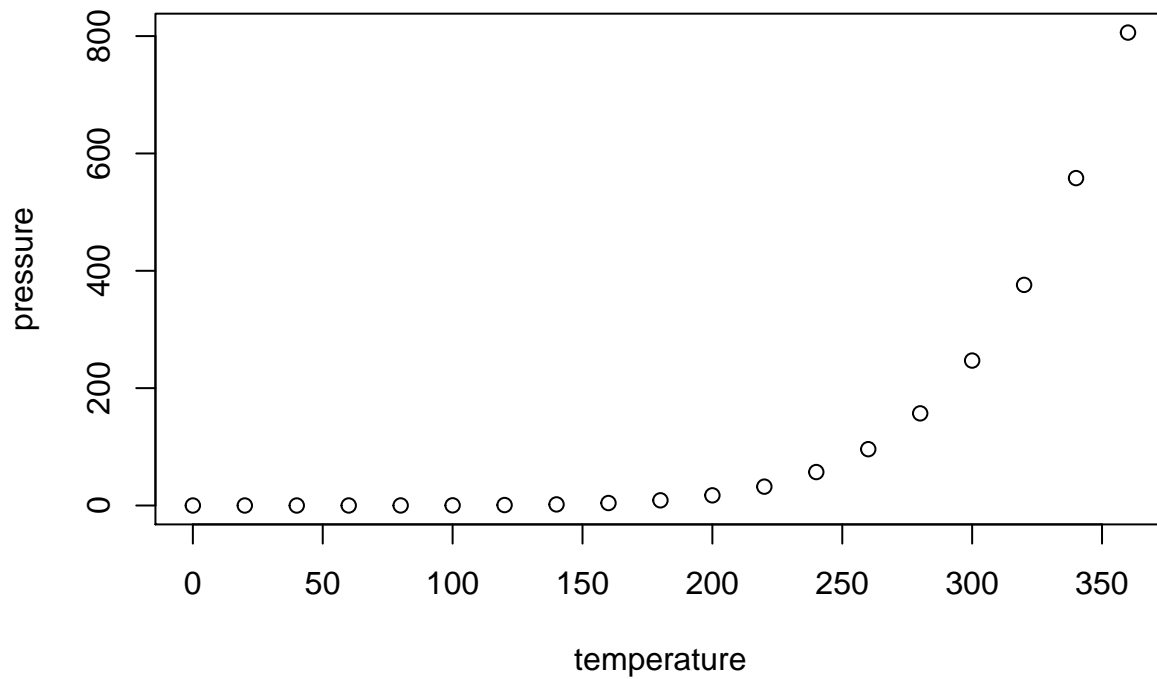
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.