Report Template for a Reproducible Automated Data Analysis

Roman Kiselev

August 31, 2017

1 Introduction

This is an easy to use report template for reproducible research. The main idea is to combine `GNU Makefile`, LaTeX, `markdown` and `knitr` to generate beautifully-looking reproducible reports easily. Any number of independent `rmarkdown` documents can be combined with arbitrary number of plain markdown files to form a single PDF report.

2 How it works

The document is built up from `Rmd` and `md` documents stored in the folder `content`. All of them get converted into LaTeX and inserted together into the body of a LaTeX template file called `latex.template`. The template file determines the final look and feel of the report, so feel free to adjust it to fit your needs.

2.1 Under the hood

To get started, place your content into the (`r`)`markdown` files in the `content` folder. You can conveniently edit `rmarkdown` files using `RStudio`. When done, call `make`. This will run all `R` code chunks and convert `rmarkdownmarkdown`, then all markdown files are fed to `pandoc` that generates a single LaTeX file from them using the template file. Finally, the generated LaTeX report file is translated to PDF, calling `bibtex` if necessary. The order of the included files is determined by their names.

`GNU Make` automatically determines which files have been update since last run and processes only what has to be processed.

2.2 How to use

If you want to include images into your document, feel free to do it using the normal `markdown` syntax, like this `![caption](imgs/path_to_image)`. For more control, the

image width can be specified along with a LaTeX label, which allows to reference this image later (see Figure 1). Here is an example how to do this:

```
    ![\label{fig:statue}Statue of Nero, a Flemish comic, Hoeilaart
       (Flemish Brabant,
       Belgium)](imgs/Hoeilaart_station_Nero_beeld.JPG){ width=25% }
```

Figure 1: Statue of Nero, a Flemish comic, Hoeilaart (Flemish Brabant, Belgium)

For even more control, you can always embed LaTeX code directly into your `markdown` files. This means, you can use `figure` environment to make floating figures like this:


```
    \begin{wrapfigure}[4]{r}{25mm}
      \hfil \begin{overpic}[grid,width=17mm]
        {imgs/Hoeilaart_station_Nero_beeld.JPG}
        \put (2,84) {\huge \bfseries a}
      \end{overpic}
    \end{wrapfigure}
```

Citations are simple: `[@paper1]` or `@paper2` produce [1] and Jass [2].

2.3 R Markdown

This part comes from an `R Markdown` file. This means, that now I can include arbitrary chunks with `R` code here. For example, there are 32 rows in the `mtcars` dataset.

This code uses `kable` function to create a nicely-rendered table, like Table 1.

```
    kable(mtcars[1:3, ], caption="\\label{tbl:cars}First three rows
          from the ‘mtcars‘ dataset.")
```

Table 2: Floating table

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

Table 1: First three rows from the `mtcars` dataset.

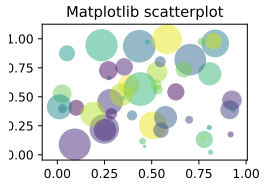
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1

We can ask `kable` to make fancy floating tables using LaTeX, like Table 2. They are quite customizable.

```
iris %>% head(4) %>% kable(format="latex", digits=2, row.names=T,
  caption="\\label{tbl:float_tbl}Floating table")
```

2.3.1 Python?!

Since `knitr` and `RStudio` support `Python` as well, we can insert `Python` chunks into the document! However, we have to care about image saving ourselves. Another problem is about environments: `R` uses one environment for all chunks in the document, but each `Python` chunk do not know anything about other `Python` chunks in the same document.




```
import numpy as np
from matplotlib import pyplot as plt

N = 50
x, y, colors = (np.random.rand(n) for n in [N, N, N])
area = np.pi * (15 * np.random.rand(N))**2  # 0 to 15 point radii

f = plt.figure(figsize=(3,2))
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.title("Matplotlib scatterplot")
plt.savefig("imgs/scatterplot.pdf")
```

2.4 Tricks with images

2.4.1 Size specification

If image is saved into PDF, it will have exactly the same size on paper as specified in `knitr` (either in the `YAML` header or in the chunk options). PNG seems to also work fine.

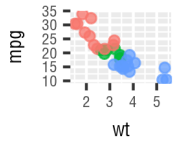


Figure 2: Tiny `ggplot2` figure, 1×0.8 inches, PNG

2.4.2 Pack several `ggplot2` or `lattice` images together

For `ggplot2`, use function `ggplotGrob()` from `gridExtra`, then pack images as you need using `cbind()` and `rbind()` – see example in Fig 3. If you need to combine `ggplot2` with `lattice`, use other funcs from `gridExtra`.

```
suppressMessages(library(gridExtra, warn.conflicts = F))
p <- ggplot(mtcars, aes(x=mpg, col=factor(cyl))) +
    scale_color_brewer("Set1", guide=F)
a <- p + geom_point(aes(y=hp))
b <- p + geom_point(aes(y=qsec))
t1 <- theme_classic()
t2 <- theme_wsj()
t3 <- theme_igray()

f <- ggplotGrob

rbind(cbind( f(a+t1), f(a+t2), f(a+t3) ),
      cbind( f(b+t1), f(b+t2), f(b+t3) )) %>% plot
```

2.4.3 Mathematical formulae in (r)`markdown`

You can insert formulas into a `markdown` files using dollar signs like this `$E=mc^2$` which gives $E = mc^2$. But make sure that you have no whitespaces between the formula and the surrounding `$`, and no digits follow immediately after the closing `$`. More details can be found in the pandoc manual, section tex_math_dollars.

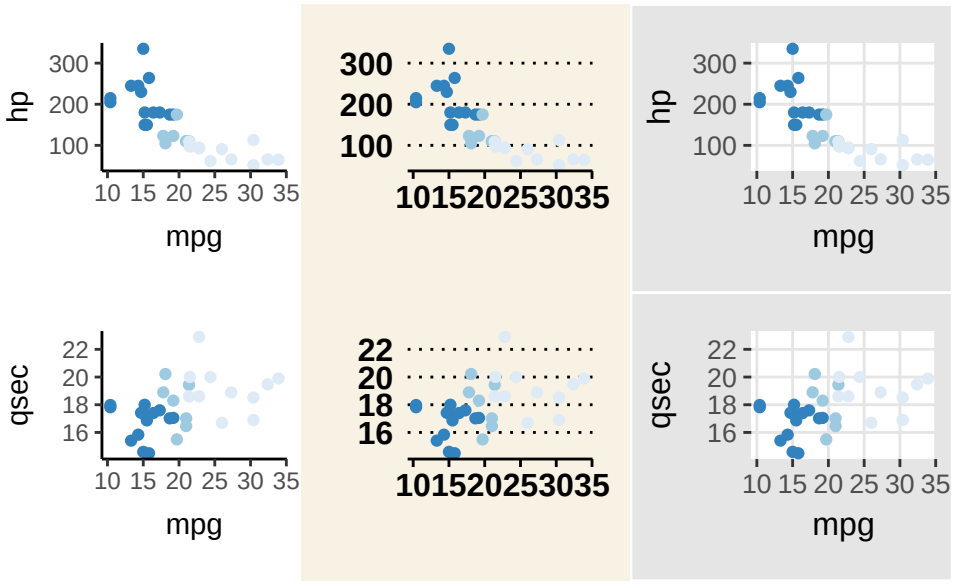


Figure 3: Alignment of plots using `ggplot2`, `rbind/cbind` and `ggplotGrob`.

3 Use of pure LaTeX

You can integrate a LaTeX file into this report. To do this, include it with the command `\input{content/filename.tex}` into one of your (R)md-documents.

> **Note,** however, that these TeXdocuments are not tracked for changes from your `Makefile`.

¡TeX can gènérate almöst all the accents[1] and speçial symbols used in Western [2] långuages! Likewise, its arsenal of mathematical symβols, introduced below, is formidable.

3.0.4 Mathematical Formulae

TeX is good at typesetting mathematical formulas like $x-3y=7$ or $y_{i+1}=x_i^{2n}-\sqrt{5}x_i^n+1$. Remember that a letter like x is a formula when it denotes a mathematical symbol, and should be treated as one.

Mathematical formulas may also be displayed. A displayed formula is one-line long; multiline formulas require special formatting instructions. The following formulae demonstrate many constructions you might find useful. Refer to equation (1), which is probably true, while equations (2-4) are silly. Note that the `equation` and `eqnarray` environments number the equations, but `eqnarray*` doesn't.

$$x_{i+1} = N^{i+1}(x_0) = N(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}$$

[1] This is an example of a footnote.

$$\frac{\partial u}{\partial t} + \nabla^4 u + \nabla^2 u + \frac{1}{2}|\nabla u|^2 \;=\; c^2$$

$$a^p + b^p \neq c^p \quad \text{for } p > 2 \;\; \text{(see proof in margin)} \tag{1}$$

$$\lim_{n\to\infty} x_n \geq \pi$$

$$\forall x \in \mathcal{O} \;\; \exists \delta \quad \text{such that} \quad |y - x| < \delta \;\Rightarrow\; y \in \mathcal{O}$$

$$\Psi' = \frac{d}{d\phi}\begin{pmatrix} \phi_2 \\ \phi_3 \\ 1 - \phi_2 - \phi_1^2/2 \end{pmatrix} \qquad \Theta = \begin{pmatrix} 0 & 1 & 0 \\ -\theta_1\psi_1 - \psi_2 & 0 & \psi_3 \\ -\phi_1 & -1 & 0 \end{pmatrix}$$

$$\begin{aligned} \int_0^\infty e^{-x^2}\,dx &= e^{-\left(\int_0^\infty x\,dx\right)^2} && (2)\\ &= e^{-\infty} \qquad \text{(bogus)} && (3)\\ &= 0.38 - 1.7i \qquad \text{(not!)} && (4) \end{aligned}$$

$$\begin{aligned} \sum_{n=1}^{k} \frac{1}{n} &\approx \ln k + \gamma \\ &= (\ln 10)(\log_{10} k) + \gamma \\ &\approx 2.3026 \log_{10} k + 0.57772 \end{aligned}$$

Unary operators "plus" and "minus" – just use exponentiation:

$$^{+}0.168 \text{ or } ^{-}1.168$$

References

[1] I.P. Freely. A small paper. The journal of small papers, -1, 1997. to appear.

[2] Hugh Jass. A big paper. The journal of big papers, MCMXCVII, 7991.