Instructions for Reports: Using the rmd2pdf-report framework

Paul Johnson, Director, CRMDA <crmda@ku.edu>
Brent Kaplan, GRA, CRMDA <crmda@ku.edu>

May 4, 2018

1 Introduction

1.1 Insert author information in the yaml header

It is necessary for the author to edit the yaml header of this document to specify the title, author data, address and the name of a logo file.

2 We use a template file

The look and feel of the document will be controlled by a template file. It is specified in the header as "theme/report-boilerplate.tex".

Until we were very far down the path which depended on the template file, we did not understand fully the limitations that the template imposes in markdown documents. The template file will "take over" the document production and it will block the other settings that users might expect to adjust within the document. In particular, users are not allowed to adjust the **toc** settings in the YAML header because the template will overwhelm those settings.

However, the same parameters can be altered by specifying them in the functions that process the document. Our function `rmd2pdf` will allow all of the arguments and they will override the settings in the template. It is also possible to override those settings by command line parameters with our script **`rmd2pdf.sh`**, as we will explain in the next section.

If this document is edited in Rstudio, the "knit" function will probably not work as intended because the users's settings in the YAML header will be ignored. The "knit" feature of Rstudio is not aware of the procedure needed to override the template settings. In order to override the template, arguments to the compiler functions must be edited. Please see the next section for details.

3 To compile this document

The document can be compiled either with a shell script that is provided with this document, **`rmd2pdf.sh`**, or by using a function of the same name that is included in the package. The ins-andouts of this are described more fully in the main vignette for the package.

If the user starts an R session, she can run the `rmd2pdf` function with many optional arguments, as follows:

```
> rmd2pdf("skeleton.Rmd", toc=FALSE, type="report",
    template="theme/report-boilerplate.tex")
```

From the command line, the same file can be compiled using notation that is nearly identical

```
$ ./rmd2pdf.sh --toc=FALSE  --type='"report"'
   --template='"theme/report-boilerplate.tex"' skeleton.Rmd
```

Note the single quotes protect the double quotes. Although our functions are designed to use our style, it is possible to compile a document with one's own template. By editing the "theme/report-boilerplate.tex" file, or by renaming that file and adjusting the previous argument, it fairly obvious how this can be customized.

Users who do not want to bother with the command line arguments can edit the `rmd2pdf.sh` script and change the defaults for the parameters in the obvious way at the top of the file.

The `rmd2pdf` function (and shell script) will also generate a "purled" copy of the R code chunks. The term "purled" is equivalent to "tangled" in the Sweave chunk engine. The user can specify either `purl=TRUE` or `tangle=TRUE`, the script will treat them as equivalent.

4 Formatting Input

4.1 Much LaTeX Syntax is allowed

This is not true only for math, but also other LaTeXenvironments.

This is explained in the `crmda` package vignette `Rmarkdown`. Not all LaTeXmarkup will work well, but most will.

Careful proof reading of output is essential. The markdown to PDF conversion does not warn us of unrecognized LaTeX code. The result is not an error, but rather "empty white space" where the user expects LaTeXoutput.

Use `\[` and `\[` for display equations. Do not use the double dollar signs:

$$\Sigma_{gt} = \Lambda_{gt}\Psi_{gt}\Lambda'_{gt} + \Theta_{gt}$$

4.2 Document customization: essentials

The function `rmd2pdf` (same as the script) supplies settings, including a LaTeXtemplate called "report-boilerplate.tex". That boilerplate is available in the R package and a copy is placed in the `theme` folder when the document is compiled the first time (there is an R chunk above called `themecopy` which does that work). After that, the author is allowed to revise the "report-boilerplate.tex" file and the revised version will be in effect. The `themecopy` stanza can be removed after the document is compiled for the first time, it will not replace a user-edited file from the theme directory with a new copy. We usually leave that chunk, so that we can delete the theme folder and it will be replaced when the document is compiled.

4.3 Customizations requiring more LaTeXpackages

In addition, if you insert LaTeXfeatures that require packages that are not currently in the template `report-boilerplate.tex`, then those packages can be inserted into the preamble by YAML header markup like so:

```
header-includes:
-  \usepackage{xcolor}
-  \usepackage{amsmath}
-  \usepackage{amssymb}
-  \usepackage{fancybox}
```

We have added many packages in the template already, and we have tested that, even when the template parameter is used, the header-includes values are taken into account by the compiling process.

5 R code chunks

In our report style, the author will not generally insert visible code chunks, so almost always the chunk will have the flag `include=FALSE` or, if the chunk is included, the code will not be echoed, but perhaps a LaTeXmark-up table or a figure may be placed into the document.

The process for doing this depends on the document type. As explained in the `crmda` vignette `code_chunks`, the appearance of code chunks–whether they are revealed in the document at all–is controlled by many options are avaiable for code chunks.

One approach might be to use one document to create graphs or tables, which are then to be saved in a folder (such as our tmpout folder, which is used in this document). This chunk code will create the output file "tmpout/p-hist-1.pdf"

~~~
```{r hist, include=F}
x <- rnorm(1000)
hist(x, main = "A Histogram", xlab = "Random Normal Data, N = 1000")
```
~~~

There are several R packages intended to create "ready to publish" LaTeXtables. One of the oldest and most venerable of these is `xtable`, which we use here to create a simple table that displays as a cross tabulation table.

When the LaTeXoutput is going directly into the document, the chunk flag is "results='asis'" and the echo parameter should be FALSE. The default configuration for xtable is to create tables that are floating LaTeXobjects.

See the help pages for `xtable` and `print.xtable` to find out all of the possible arguments. If one does intend to have the output go directly into the document, without any hand editing, it is almost certainly necessary to specify a large number of arguments.

It may be more workable to write the LaTeXfile on disk and then double-check its contents before manual inclusion in the document. If a LaTeX table file has been created from another document, we do not recommend "cutting and pasting" into this document. Instead, use "\input{filename}".

The following code chunk will save the same LaTeXmarkup table in a file.

Table 1: Ten Lines from One Data Frame

x	y
-1.21	0.41
0.28	-0.47
1.08	0.07
-2.35	-0.50
0.43	-0.83
0.51	0.17
-0.57	-0.90
-0.55	0.17
-0.56	0.35
-0.89	-0.05

6 Session Info

Reports do not include the R session replication information, generally speaking. However, compiling the document will produce a record-keeping file in which the session information is saved. This will be in the current working directory.