Title of your article

List of authors, François Birgand, Associate Professor

05 October, 2018

Contents

```
# This is template that shows some of the main features offered by R Markdown to write a
```

```
## [1] "bookdown::pdf_document2"
```

1 Template goals

The goal of this template is to show how to write an article in R Markdown. There are many advantages for using R Markdown: it is an integrated way to put together text, codes for figures, figure labels, tables, and references. Actually the template I am showing here, is not *strictly sensu* `R Markdown` but another package, which is called `bookdown`. `bookdown` is based on R markdown, but has some really great features about cross-references, which I find to be better than R Markdown proper. Additionally, papers written individually can eventually be assembled into a book with all the cross references done automatically.

The first thing you need to do is to read the R Markdown crash course I have written, as it will give you lots information very useful to understand what is happening in this template. The second thing you need to do is to take this **TemplateArticleBookdown.Rmd** document and try to `knit` it using the button that has whool and the 'Knit' word next to it in the R Studio window. Normally, all the functionalities should work, so you should be able to have the document appear directly. In the end it is the **TemplateArticleBookdown.Rmd** file that is important because you have access to all the code, regardless of the output.

So by now, you probably realize that the main features are the YAML header, the text and its associated *markup* alteration, and the code chunks. An additional important feature is equations, which are written a bit differently in `bookdown`, than in `R Markdown`. There is a paragraph just for this below.

2 Input File locations

I think it is best to create a project directory where all the necessary files are going to be needed. In this directory, you should have

- a project file in R Studio *.Rproj

- the *.Rmd files you are using
- the *.csl file you will need to define the formatting of the bibliography references
- the *.bibtex file in which the references of your articles are located (see example with this template)
- possibly a directory where you will store the pictures files
- all your *.csv files where you might have your data for plotting, but also csv files corresponding to info to be added in tables
- a template document for word (e.g., template.docx) if you want to use one. I have added one in the directory for references.

3 Output Files

The beauty of `R Markdown` and `bookdown` is that several outputs are possible. For most of what we do, html, pdf, and *.docs are good outputs. If you look in the YAML header, you will see three outputs like:

```
output:
  bookdown::pdf_document2:
    toc: true
    number_sections: true
  bookdown::html_document2:
    toc: false
    number_sections: false
  bookdown::word_document2:
    reference_docx: template.docx
    fig_width: 4
    fig_height: 3
    fig_caption: true
```

This means that you can 'knit' your document either into pdf, html, or word. For this, instead of clicking on the 'knit' button, click on the bottom arrow just right of it, and you will be able to choose the option you want. The reason for the "2" in '*_document2' is because the `documentclass` is an article, and not a book. Had it been a book, there would be no "2" added here. There are lots of options that can be added for each document type but the ones above are probably good by default.

You can actually remove the options under, e.g., `bookdown::word_document2:` and add `bookdown::word_document2: null` and the default template document in word will be chosen by default.

Upon 'knitting' your document, several files will be created in your directory, including word, html and pdf documents. If you see some in the TemplateRMarkdownReport directory, it is because somebody has tried knitting and has already created some output files.

3.0.1 Some default options in the YAML

In the YAML header of this document, in the `output` section description, you can see

```
output:
  bookdown::html_document2:
    toc: true
    number_sections: false
```

This means the table of content (toc) will show up, but the numbering of the section will not. Changing values to `TRUE` or `FALSE` will able/enable each option

3.1 Paragraph headers and numbering

For the different paragraph headers, use the number of necessary `#` to create as many subsections as needed.

3.1.1 Paragraph1

Your beautiful prose goes here

3.1.2 Paragraph2

3.1.2.1 Subparagraph

Your beautiful prose goes here

```
### Paragraph1

Your beautiful prose goes here

### Paragraph2
#### Subparagraph

Your beautiful prose goes here
```

3.2 Figures, captions and references

Below in Figure 1 you can see how a chunk is written and how some of the figure parameters are added.

```
Below in Figure \\ref{fig:plot-example-data-fig1} you can see how a chunk is written and how some of th
```

This is borrowed from another document so the figure in itself is not essential, but the code to express how the figure should be displayed is

```
{r "plot-example-data-fig1", eval=TRUE, echo=TRUE, out.width = '60%', fig.align = 'center', fig.cap="10-
```

3.2.1 Figures generated by R codes

The first item between “” is the name of the chunk, which serves as the reference for the figure. Try to find a simple name which describes well that figure. The current label is not particularly good. You should find a better one. If two chunks have the exact same name, R Studio will not knit and will give you an error message: each chunk must have its unique identifier name.

The other options are rather self explanatory in there. `fig.cap=` gives the exact caption of the figure you want. Notice that the caption appears under the figure with an automatic number. You do not have to tweak it. It is automatic. The figure number is also automatically added in the text cross reference as in the `Figure \@ref(fig:plot-example-data-fig1)` code.

3.2.2 Figures with images

In the Figure 1 example, it is an `R` code was actually run. But if you want to add a picture, you need to use something a bit different. you use the code and you get the results in Figure 2

```
r "fig-pic", echo=FALSE, out.width = '70%', fig.align = 'center', fig.cap="Eight floating island mesoco
knitr::include_graphics("pictures/Activation_energy.png")
```

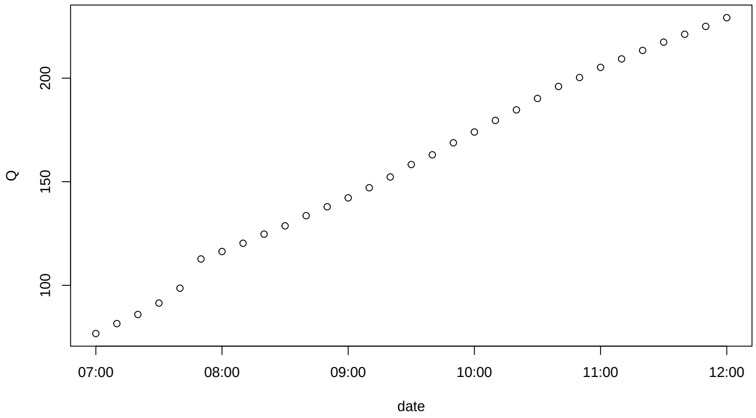

Figure 1: 10-min instantaneous flow rates as a function of time

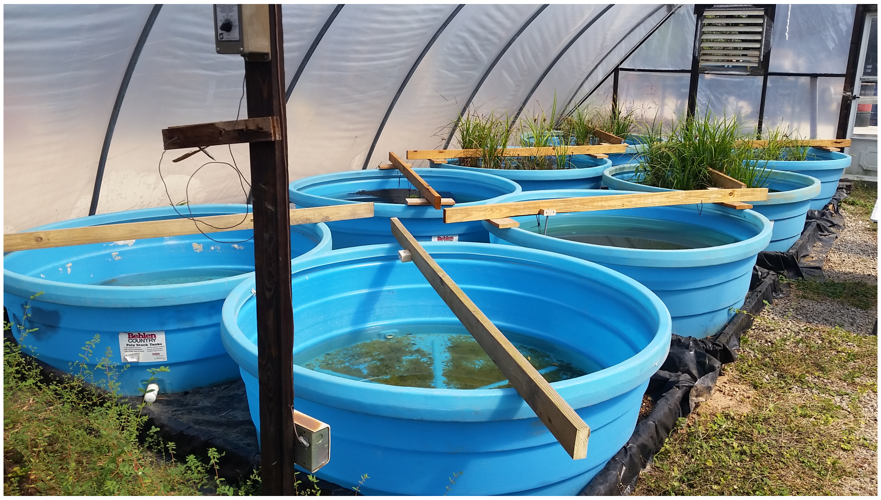

Figure 2: Eight floating island mesocosms *(FW 1)* to *(FW 8)*

Notice that it is possible to have some markup to the text in the figure caption. This works well for html and word output but not so for pdf output… I would stay away from markup in the figure captions as much as possible.

3.2.3 Figures with multiple images

It is possible to have several images right next to each other. `fig.show = 'hold'` is new here, and this is what allows to plot several pictures together. `out.width = '25%'` also details the width of both images. Images are stuck next to each other however with this method… The only option I have found is to add a white strip of a white rectangle in PowerPoint to the left image… Not great but it works.

```
{r "fig-2pic", echo=FALSE, out.width = '25%', fig.show = 'hold', fig.align = 'center', fig.cap="A: Eigh
knitr::include_graphics(c("pictures/mesocosms1.jpg","pictures/mesocosms2.jpg"))
```

This code, however, does not render in Word. I have thus inactivated the code so this document could be rendered in all three formats.

3.3 Tables, caption, and references

For tables, there is a very cluncky way that is described in R Markdown cheet sheet. However, it is hardly satisfying. I believe it is a lot easier to use packages that already generate the markdown code for us.

3.3.1 Tables in html

The best package I have found right now is `kableExtra`. All the parameters and possibilities are described by the kableExtra author Hao Zhu and things are regularly updated.

The best way is to have the values for a table stored in a dataframe or in a .csv file, which is then read in first before `kableExtra` can represent it.

If you knitted to pdf or word, you will not see the table I am describing in html here. To produce Table **??**, I used the code below:

```
{r "RemovalTable1", echo=FALSE, warning=FALSE}
RemovalTab1<-read.csv("table1.csv", header = TRUE)
fmt <- rmarkdown::default_output_format(knitr::current_input())$name #find the output format when knitt
if (fmt == "bookdown::html_document2"){
kableExtra::kable(RemovalTab1, align = "c", "html",
             caption = 'Removal values in three mescosms reports (fake values)',
             col.names = c("Reference","concentrations","type of water", "percentage removal")) %>%
      kable_styling(full_width = F) %>%
  column_spec(1, bold = T, border_right = T, width = "15em") %>%
  column_spec(2, width = "15em") %>%
  column_spec(3, width = "10em") %>%
  column_spec(4, width = "10em") %>%
  row_spec(1:4, background = "white")
}
```

The actual data stored in table1.csv is

```
Reference,concentrations,type of water, percentage removal
@Borne2014-ek,0 to 30 mg/L,urban water, 35%
@Chang2013-cv,10 to 20 mg/L,Ag water, 3%
@De_Stefani2011-zt,0.9 to 2.3 mg/L,Forest water, 22%
@Khan2013-qr,0.3 to 0.79 mg/L,Ag water, 5%
```

I suggest you keep the header in your table such that you can remember what column is what. But it seems that you have to specify again the col.names in kable again. I do not seem to find a way around that…?

To reference the table in your text, use, similarly to figures, `Table \@ref(tab:RemovalTable1)` to write Table **??**.

OK, this works OK for html, but not so great with word or pdf outputs

3.3.2 Tables in word

The best way to get tables in word is to actually render the document in html, copy and paste it directly into word. It works. It is not pretty but it does the job.

3.3.3 Tables rendered in pdf

This is not as smooth for pdf. It turns out that as I was writing this tutorial, I just could not find a way to make `kableExtra` work here… So I spent a lot of time trying to find ways around it to make it work. If you are reading an html document, you will not see the table, only the code. The problem with `kableExtra` is that, as far as I can tell, the markup text and reference citations are not rendered within the table… After reading quite a bit, I found that the package `pander` would work well, although it had one major drawback, which is that it would not allow the cross referencing by default… After a lot of time, I did find a way to make it work and this is the code that you see below:

```
{r "RemovalTablepdf", echo=FALSE, warning=FALSE}
fmt <- rmarkdown::default_output_format(knitr::current_input())$name #find the output format when knitti
if (fmt == "bookdown::pdf_document2"){
RemovalTab1<-read.csv("table1.csv", header = TRUE)
library(pander)
pander(RemovalTab1, align = "c", "latex",
            caption = '(\\#tab:RemovalTablepdf) Removal values in three mescosm reports (fake values)'
            col.names = c("Reference","concentrations","type of water", "percentage removal"),
            booktabs = T)
}
```

If you are using pdf as an output, you should be able to see Table 1 just below this paragraph.

Table 1: Removal values in three mescosm reports (fake values)

Reference	NO_3^- concentrations	type of water	%age removal
Borne (2014)	0 to 30 mg/L	urban water	35%
Chang et al. (2013)	10 to 20 mg/L	Ag water	3%
De Stefani et al. (2011)	0.9 to 2.3 mg/L	Forest water	22%
Khan, Melville, and Shamseldin (2013)	0.3 to 0.79 mg/L	Ag water	5%

3.4 Equations and their referencing

Equations are a little bit different in `bookdown` than they are in `R Markdown`. Overall, it uses all the magic developed in Latex. So if you want to make complicated stuff, the first thing I would do is to consult the latex equation tutorials. It is possible to write these two equations:

$$\begin{align} H_2S \rightleftharpoons & \quad HS^- + H^+ \tag{1} \\ HS^- \rightleftharpoons & \quad S^{2-} + H^+ \tag{2} \end{align}$$

using this code

```
\begin{align}
H_2S & \rightleftharpoons & HS^- + H^+ \label{eq:H2S} \\
HS^- & \rightleftharpoons & S^{2-}+ H^+ \label{eq:HS}
\end{align}
```

Several things here. Each equation has its own reference. So if I want to refer to equation (1), I write equation `\@ref(eq:H2S)`, very similarly to what is used for figures and tables.

4 Literature citations

This is one of the best things about `bookdown`, you essentially have to do nothing. . . ! The first thing you need is to have your citation list exported into a `bibtex` format. Most referencing software provide that. Then, to cite like this (Borne 2014; De Stefani et al. 2011), you use `[@Borne2014-ek; @De_Stefani2011-zt]`. For most citation formatting dictated in the `*.csl` file, this will appear as the citation in parentheses. But for another format such as `nature.csl`, then the citation will appear as an exponent. Sometimes, you absolutely want a comma between the authors and the date, right? Do not waste time on how you want your citation to look like, it is probable that there is a `*.csl` from some journal which can already do the job (e.g., put citations in exponents, add a comma between the authors and the date, etc.).

There are two things to know. If you want the authors not to be in parentheses, like in Borne (2014), just use `like in @Borne2014-ek`. Now if you just want the year saying "these authors in (2014)", use `"these authors in [-@Borne2014-ek]"`.

And the last thing, the reference list always appears at the end after the last markdown text. Happy `bookdown` writing!!

References

Borne, Karine E. 2014. "Floating Treatment Wetland Influences on the Fate and Removal Performance of Phosphorus in Stormwater Retention Ponds." *Ecol. Eng.* 69 (August): 76–82. doi:10.1016/j.ecoleng.2014.03.062.

Chang, Ni-Bin, Zhemin Xuan, Zachary Marimon, Kamrul Islam, and Martin P Wanielista. 2013. "Exploring Hydrobiogeochemical Processes of Floating Treatment Wetlands in a Subtropical Stormwater Wet Detention Pond." *Ecol. Eng.* 54 (May): 66–76. doi:10.1016/j.ecoleng.2013.01.019.

De Stefani, G, D Tocchetto, M Salvato, and M Borin. 2011. "Performance of a Floating Treatment Wetland for in-Stream Water Amelioration in NE Italy." *Hydrobiologia* 674 (1): 157–67. doi:10.1007/s10750-011-0730-4.

Khan, Sher, Bruce W Melville, and Asaad Shamseldin. 2013. "Design of Storm-Water Retention Ponds with Floating Treatment Wetlands." *J. Environ. Eng.* 139 (11): 1343–9. doi:10.1061/(ASCE)EE.1943-7870.0000748.