scdoc: long scientific documents within R Studio

How to use

Joseph Larmarange

2014-01-17

With *scdoc*, you can combine several powerful tools to write a long scientific document directly within *R Studio* using *markdown*. Compared to the native support of *R markdown* in *R Studio*, *scdoc* adds:

- automatic numbering of titles, tables and figures;
- automatically generated list of contents, list of tables and list of figures;
- add automatically the appropriate number to internal links to a title, a table or a figure;
- smart punctuation;
- production of an HTML version with a modern design, an easy navigation and footnotes presented as sidenotes;
- production of a PDF version with a classic design and page numbers added to internal links.

R, R Studio, knitr, pandoc, Prince XML

table of contents

1 About *scdoc*

scdoc is developed by Joseph Larmarange on GitHub :
https://github.com/larmarange/scdoc

For questions, bugs or feature requests:
https://github.com/larmarange/scdoc/issues

2 Pre-requisite

To works properly, *scdoc* needs that several tools are installed on your computer.

2.1 R Studio

First, *scdoc* is principally designed to be used with the *R Studio*, an integrated development environment (IDE) for *R*. *R Studio* is available for Windows, Mac OS and Linux on http://www.rstudio.com/ide/.

2.2 pandoc

scdoc uses *pandoc* to convert a *markdown* file to *HTML*. *pandoc* could be downloaded on http://johnmacfarlane.net/pandoc/installing.html. *pandoc* needs to be accessible as command-line tool[1]. NB: due to changes in *pandoc*, please use a recent version (≥ 1.12).

1. On Windows, you will probably have to add manually the path to *pandoc* to the system `path` variable. To test if *pandoc* is available as command-line, open a *command prompt* and type `pandoc --help`. If you get an error (command not recognized), search for the directory where is located the file `pandoc.exe` (something like `C:\Users\mon_nom\AppData\Local\Pandoc\`). Add this directory to the Windows system `path` (see http://www.computerhope.com/issues/ch000549.htm).

2.3 Prince XML

Prince XML is used to generate the PDF version of your document.[2] It could be downloaded from http://www.princexml.com/download/. *Prince XML* should also be available as a command-line tool[3].

2.4 zotxt

You can insert *Zotero* references within your R markdown file. For that, you will need to install *zotxt* and *pandoc-zotxt*:

- Install the Firefox add-on *zotxt* (https://addons.mozilla.org/fr/firefox/addon/zotxt/)
- *Python* should be available on your computer (to install it, cf. http://www.python.org/download/)
- Install *pip* http://www.pip-installer.org/en/latest/installing.html
- Type and execute the command `pip install pandoc-zotxt` (in a prompt window or with the function `system` within R)

3 Installing scdoc

In *R*, execute the following command:

```
R> if (!require(devtools)){
       install.packages('devtools')
       library(devtools)
   }
   install_github("scdoc","larmarange")
```

4 Utilisation

When staring a new R session, execute the following commands:

```
R> require(scdoc)
   scdoc()
```

That's all! Edit your *R markdown* (`.Rmd`) in *R Studio*. Click on **Knit HTML** to produce the HTML[4] (and PDF if required) version of your document.

2. Generating a PDF is optional, see section 5 page 4.

3. On Windows, the directory `C:\Program Files (x86)\Prince\Engine\bin` should be added to the system `path`.

4. The HTML file is self-contained and can be easily shared.

5 Parameters of the document

Note: you can use this document as a model for your owns.

5.1 YAML metadata bloc

Several options can be define at the beginning of your document in a YAML bloc. To be recognized by *scdoc*, this bloc[5] must start by a line containing only `---` and end by a line containing only `...`. All parameters are optional.

Table 1. Parameters of the YAML metadata bloc

lang	Lang of your document in ISO alpha 2 format, `'en'` for English, `'fr'` for French…
title	Title of your document
author	Author(s) (several entries allowed)
date	Date of your document (formatted as YYYY-MM-DD, to generate automatically the date of the current day, indicate ``` `r format(Sys.time(), "%Y-%m-%d")` ```)
tags	Key-words
abstract	Abstract (should start by `|` for a abstract with several paragraphs, each paragraph being indented)
toc-title	Title for the table of contents
tof-title	Title for the table of figures
tot-title	Title for the table of tables
prefix-fig	Prefix for figures numbering (default: *Figure*)
prefix-table	Prefix for tables numbering (default: *Table*)
pdf	To generate a PDF version of your document with *Prince XML*, this option should be equal to `'prince'`

5.2 Other parameters

It is recommanded to add the following code at the begining of your document:

````
```{r configuration, echo=FALSE, message=FALSE}
opts_chunk$set(
  comment = NA, # No ## put before source code output
  dev = "svg", # Plots in SVG (usually renderring better in PDF)
  cache = TRUE, # Cache
  fig.width = 8, fig.height = 6 # Default size for plots
)
options(
  knitr.table.format = "html", # Tables directly in HTML (kable)
  xtable.type = "html", # Tables in HTML (xtable)
  xtable.caption.placement = "top" # Caption at the top of tables (xtable)
)
knit_hooks$set(plot = hook_plot_html) # Figures directly in HTML
windowsFonts(sans=windowsFont("Open Sans")) # Default font for plots
```
````

5. See as an example https://github.com/larmarange/scdoc/blob/master/How_to_use.Rmd.

Options defined through `opts_chunk$set` are optional. You can adapt them according to your needs. For plots, we recommand SVG as the result is better when producing a PDF file (SCG is also correctly interpreted by modern browsers).

`options(knitr.table.format = "html")` and `knit_hooks$set(plot = hook_plot_html)` are required for providing adequately an identifier and a title to tables and figures (see section 9 page 7 and section 10 page 9 for more details). `xtable.type` and `xtable.caption.placement` are required only if you use *xtable* package.

`windowsFonts(sans=windowsFont("Open Sans"))` is optional and allows you to specify the default font used by R for plots.

6 Markdown

As *scdoc* uses *pandoc* to generate an HTML file, you can use all markdown extensions implemented in *pandoc*, see http://johnmacfarlane.net/pandoc/README.html#pandocs-markdown. You can also write directly HTML code that will be preserved.

7 Titles and internal links

Titles are automatically numbered. You can specify that a specific title should not be numbered by adding the specific CSS class `.unnumbered`.

```
## Unnumbered title {.unnumbered}
```

will produce

Unnumbered title

To create an internal link to a title, you need first to attribute an unique identifier to it:

```
## Title with an identifier {#id_title}
```

will produce

7.1 Title with an identifier

Use this identifier to create an internal link.

```
See [section](#id_title).
```

will produce

> See section 7.1 page 5.

You can notice that the number of the title has been automatically added to the link (and the page number in the PDF version).

8 Include R chunks

8.1 Code bloc

````
```{r}
summary(cars)
```
````

will produce

```
R> summary(cars)
```

```
     speed           dist
 Min.   : 4.0   Min.   :  2
 1st Qu.:12.0   1st Qu.: 26
 Median :15.0   Median : 36
 Mean   :15.4   Mean   : 43
 3rd Qu.:19.0   3rd Qu.: 56
 Max.   :25.0   Max.   :120
```

Another example:

````
```{r}
summary(cars)
str(cars)
sum(cars$speed)
```
````

```
R> summary(cars)
```

```
     speed           dist
 Min.   : 4.0   Min.   :  2
 1st Qu.:12.0   1st Qu.: 26
 Median :15.0   Median : 36
 Mean   :15.4   Mean   : 43
 3rd Qu.:19.0   3rd Qu.: 56
 Max.   :25.0   Max.   :120
```

```
R> str(cars)
```

```
'data.frame':   50 obs. of  2 variables:
 $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

```
R> sum(cars$speed)
```

```
[1] 770
```

For more details, see http://yihui.name/knitr/.

8.2 Inline code

You can insert R code directly in a text with the following syntax:
`` `r … some R code … ` ``

```
The sum of 3 plus 4 is equal to 7.
```

will produce

> The sum of 3 plus 4 is equal to 7.

9 Tables

Tables are automatically numbered and a table of tables is automatically generated. Prefix for tables numbers and the title of the list of tables are customizable (see chapter 5.1 page 4).

Several functions and R packages can produce correctly formatted tables in HTML. For each package, we detail how to define an identifier and/or a caption. An identifier is required to create an internal link, for example:

```
See [table](#example_kable).
```

will produce

> See table 2 page 8.

9.1 Function kable

You need a recent version of *knitr* (≥ 1.5.15) to be able to define a caption. The last version of *knitr* can be installed with the following command:

```
R> require(devtools)
   install_github('knitr', 'yihui')
```

First, you need to calculate your table (with a function like `table` or `xtabs`).

```
R> data(Titanic)
   d <- as.data.frame(Titanic)
   tab <- xtabs(Freq~Class+Survived, data=d)
```

Then, you need to call the function `kable` in a chunck with the option `results='asis'` in order that the HTML code produced by `kable` will be include as is. It is also important to force *knitr* to generate tables in HTML (`options(knitr.table.format = "html")`, see section 5.2 page 4).

The `caption` argument (optional) allows to define a title for the table. To add an identifier, you will use the `table.attr` argument. Finally, you can specify text alignment for each column with `align`.

```
R> kable(tab, table.attr = 'id="example_kable"',
         caption = "Table generated with kable",
         align = c('l','r','r'))
```

Table 2. Table generated with kable

	No	Yes
1st	122	203
2nd	167	118
3rd	528	178
Crew	673	212

9.2 xtable package

The *xtable* package can also generate HTML tables. It is recommanded to specify the options `xtable.type` and `xtable.caption.placement` at the beginning of your document (see section 5.2 page 4).

As for `kable`, `xtable` should be called in a chunk with the option `results='asis'`. The syntax to generate a table is:

```
R> require(xtable)
   xtable(tab, caption="Table generated with xtable",
          label="example_xtable",
          align=c("lrr"))
```

Table 3. Table generated with xtable

	No	Yes
1st	122.00	203.00
2nd	167.00	118.00
3rd	528.00	178.00
Crew	673.00	212.00

9.3 tables package

You need a recent version of *tables* (0.7.67 or more). The last version could be installed with the following command:

```
R> install.packages("tables", repos="http://R-Forge.R-project.org")
```

First, the table will be calculated with the function `tabular` (see `?tabular` for more details):

```
R> require(tables)
   tab2 <- tabular( (Species + 1) ~ (n=1) + Format(digits=2)*
            (Sepal.Length + Sepal.Width)*(mean + sd), data=iris )
```

The table will be converted in HTML with the function `html`. This function needs to be called in a chunk with the option `results='asis'`. A specificity of *tables* is that the title need to be defined with the function `table_options` before calling `html`. It is recommanded to reset this option to `NULL` just after to avoid undesired title for the foloowing tables.

```
R> table_options(HTMLcaption = 'Tableau généré avec tabular et htm
   html(tab2, id='exemple_tabular')
```

Table 4. Tableau généré avec tabular et html

		Sepal.Length		Sepal.Width	
Species	**n**	**mean**	**sd**	**mean**	**sd**
setosa	50	5.01	0.35	3.43	0.38
versicolor	50	5.94	0.52	2.77	0.31
virginica	50	6.59	0.64	2.97	0.32
All	150	5.84	0.83	3.06	0.44

```
R> table_options(HTMLcaption = NULL)
```

10 Plots

According to HTML 5, the tag `<figure>;` should be used to embed images or other medias. One figure can contain several plots. The title of a figure should be defined with `<figcaption>;`.

Figures are automatically numbered and a table of figures is automatically generated. Prefix for figures numbers and the title of the list of figures are customizable (see chapter 5.1 page 4).

`<figure>;` and `<figcaption>;` will be directly written in the markdown file. (NB: this approach will work only if the option `knit_hooks$set(plot = hook_plot_html)` has been defined at the begining of your document, see section 5.2 page 4.)

````
<figure id="example_fig">
```{r, echo=FALSE}
plot(cars)
```
<figcaption>Figure title</figcaption>
</figure>;
````

will produce

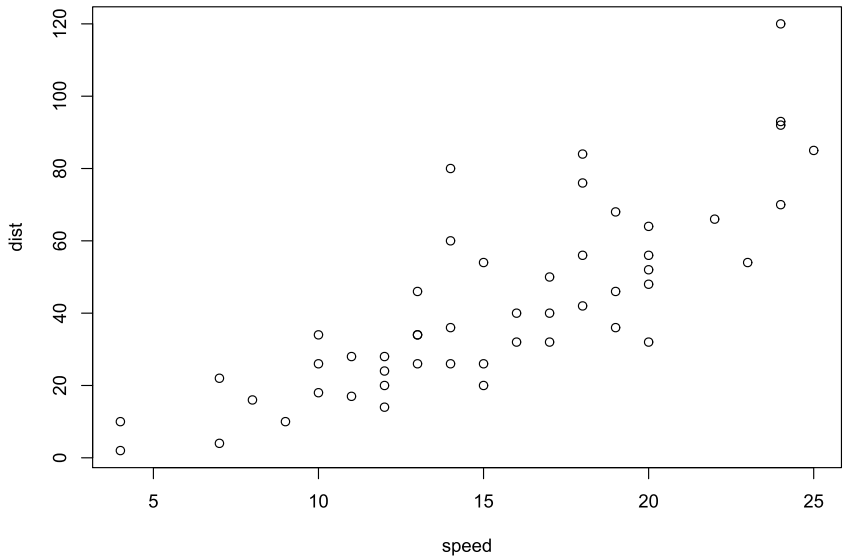


Figure 1. Figure title

Adding an identifier is required if you want to create an internal link. For example:

```
See [figure](#example_fig).
```

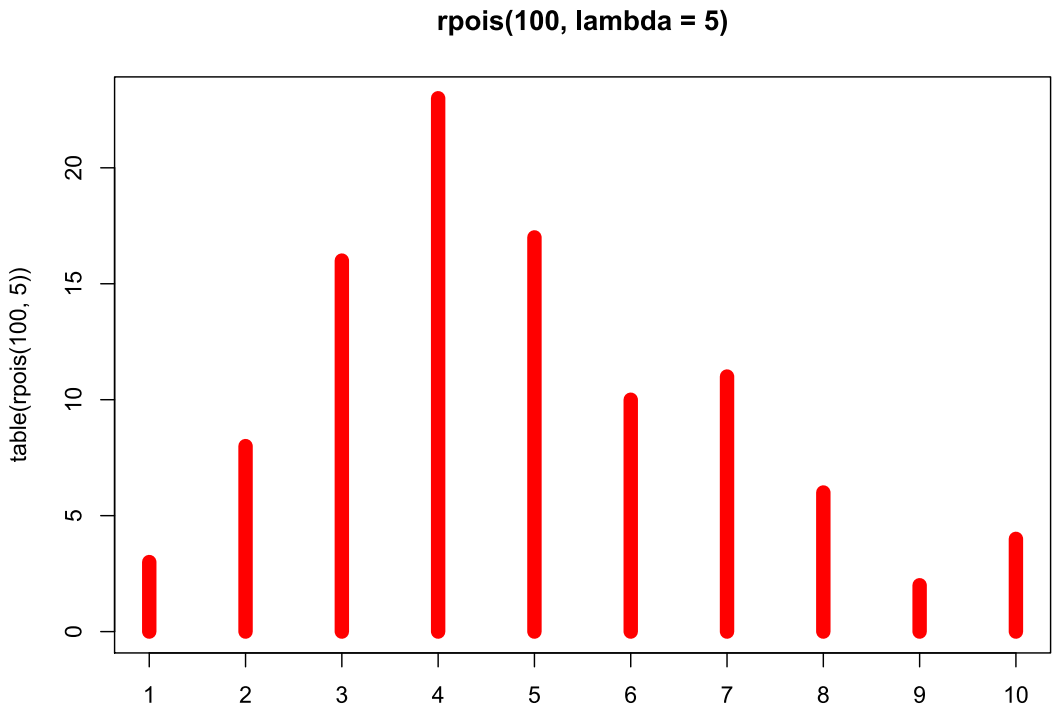
will produce

> See figure 1 page 10.

A plot created oustide a `<figure>;` tag will not be added to the table of figures.

````
```{r, echo=FALSE}
plot(table(rpois(100, 5)), type = "h", col = "red", lwd = 10,
     main = "rpois(100, lambda = 5)")
```
````

will produce



11 Footnotes

```
This is a text^[with a footnote].
```

will produce

> This is a text[6].

Several syntaxes can be used with *pandoc* to generate footnotes (see http://johnmacfarlane.net/pandoc/README.html#footnotes for more details).

12 Equations

You can write LaTeX equations.

```
$\alpha+\beta=\gamma$
$$\int_0^\infty e^{-x^2} dx=\frac{\sqrt{\pi}}{2}$$
```

6. with a footnote

will produce

$$\alpha + \beta = \gamma$$
$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Equations are transformed in MathML. If there is at least one equation in the document, MathJax will be automatically loaded to display properly equations with all browsers.

13 Smart punctuation

Smart punctuation option of *pandoc*[7] is used. *pandoc* will produce typographically correct output, converting straight quotes (`""`) to curly quotes (“”), `---` to em-dashes (—), `--` to en-dashes (–), and `...` to ellipses (…). Nonbreaking spaces are inserted after certain abbreviations, such as “Mr.”.

```
--  ---   ...  "test"
```

will produce

– — … “test”

14 Références bibliographiques

On peut ajouter une référence bibliographique au texte. Par exemple, le pacakge *TraMineR* permet d’analyser des séquences (Gabadinho et al. 2011). Pour plus de détails sur la syntaxe, voir http://johnmacfarlane.net/pandoc/README.html#citations.

Il faut soit que les sources bibliographiques soit ajouter au format YAML dans votre fichier *R markdown*.

Une autre option consiste à utiliser *zotxt* pour lier votre document à votre base Zotero. Pour cela, vous devrez ajouer l’option `- zotxt:'yes'` à votre YAML d’en-tête (voir section 5.1 page 4). Il importera également que Zotero soit ouvert lorsque vous lancer **Knit HTML**.

Enfin, vous pouvez utilisez la fonction *zotxt2yaml* pour extraire avec *zotxt* les références citées dans votre document et générer le YAML correspondant.

7. http://johnmacfarlane.net/pandoc/README.html#smart-punctuation

References

Gabadinho, Alexis, Gilbert Ritschard, Nicolas S. Müller, and Matthias Studer. 2011. "Analyzing and Visualizing State Sequences in R with TraMineR." *Http://www.Jstatsoft.Org/V40/I04/Paper* (April 7).

List of figures

List of tables