

Title of the Article

First Author's name^{1*}

Second Author's name²

Abstract

Summary of the article.

Keywords

keyword1, keyword2, etc

¹Department / University

Street address, Zip code, Country.

²Department / University

Street address, Zip code, Country.

*Corresponding author: name@company.com, <https://www.company.com>

Contents

1	Introduction	1
2	R Markdown	1
2.1	Motivation	1
2.2	How to do it	1
3	Code	1
3.1	R code	1
3.2	Tables	2
3.3	Figures	2
3.4	Captions	2
3.5	Lists	2
3.6	Math	2
3.7	Cross-references	3
3.8	Bibliography	3
3.9	LaTeX preamble	3
3.10	Forcing line breaks	3
3.11	Languages	4
4	Document types	4
4.1	PDF Document	4
4.2	HTML document	4
4.3	Word document	4

1. Introduction

This template allows writing articles in Markdown¹ format. It directly produces well-formatted articles for self-archiving (deposit on HAL for example) or in other formats, for example HTML.

2. R Markdown

Markdown is a very simple language for producing various types of documents: HTML, PDF, and Word among others. Its documentation is available at the RStudio website².

Markdown is extended by Bookdown³, which allows for

¹<https://ericmarcon.github.io/travailleR/chap-rediger.html>

²<http://rmarkdown.rstudio.com/articles.html>

³<https://bookdown.org/yihui/bookdown/>

book writing and more efficient syntax for articles. This document is made with Markdown in RStudio: knitr processes the Markdown code, passes it to Pandoc for transformation into LaTeX, finally LaTeX compiles it into PDF.

2.1 Motivation

Markdown is very easy to learn.

Markdown allows you to integrate your R code for a *reproducible* result.

Markdown allows to produce, without rewriting the text, a document in different formats: HTML, LaTeX or Word for example.

2.2 How to do it

In RStudio, create a new document of type Document R Markdown. The wizard allows you to choose between different formats.

Click on *From template*: from templates installed by packages. The memoIR package templates are displayed: choose *Stylish Article*.

It is better to create an RStudio project to benefit from all the possibilities: *File / New Project* then use the wizard to create a project from an existing folder.

Write the document in RStudio.

Clicking the **Knit** button in RStudio generates the document in the requested format.

3. Code

The main features of Markdown are summarized here.

3.1 R code

R code is included in code chunks:

```
head(cars)

##   speed  dist
## 1     4     2
## 2     4    10
## 3     7     4
## 4     7    22
## 5     8    16
## 6     9    10
```

Table 1. Table created by R

Sepal length	Width	Petal length	Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

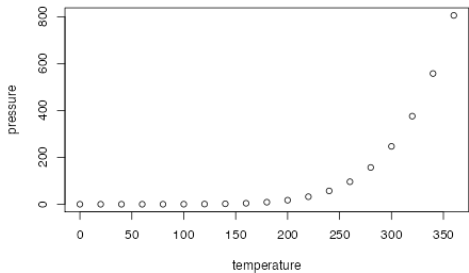


Figure 1. Figure title

3.2 Tables

The horizontal – and vertical separators | allow you to draw a table according to Markdown syntax, but this is not the best method.

Tables can also be produced by R code. The content of the table is in a dataframe. The `kable()` function in the *knitr* package prepares the table for display and passes the result to the `kable_styling` function in the *kableExtra* package for final formatting.

The caption is specified by the `caption` argument and referencing is possible because the table receives a label whose name is `tab:` followed by the name of the code snippet (table 1). Always use the `booktabs = TRUE` argument so that the thickness of the separator lines is optimal in LaTeX. The `bootstrap_options = "striped"` option provides more readable tables in HTML.

In LaTeX, tables can have the width of the column and possibly span multiple pages, or use the width of the page, like table 2).

This table contains mathematics: the `escape = FALSE` option is necessary.

Finally, the `full_width = FALSE` option adjusts the width of the table to its content instead of occupying all the available width. It must be `TRUE` for correct formatting of two-column tables in LaTeX.

3.3 Figures

```
plot(pressure)
```

Figures can be created by the R code (figure 1). With Bookdown, a label is associated with each figure: its name



Figure 2. A figure from a file

is `fig:xxx` where `xxx` is the name of the R code snippet. Cross-references are made with the command `\@ref(fig:xxx)`.

A figure can use the full width of the page by adding the following options to the header of the code snippet that generates it: `fig.env="figure"` and `out.extra=""`.

Existing figures are integrated into a piece of code by the `include_graphics` function, see figure 2.

Systematically place these files in the `images` folder for the automation of GitHub pages.

3.4 Captions

Figure and table captions can be long, include formatted text, maths, references... The only limit is they cannot contain more than a single paragraph. Such captions must be stored in a separate paragraph starting with `(ref:ChunkName)` and a space. The text of the caption follows.

In the figure chunk heading, the caption is called in the `fig.cap` field:

```
{r ChunkName, fig.cap="(ref:ChunkName)"} 
```

In tables, the `caption` argument of the `kable()` function is used the same way.

3.5 Lists

Lists are indicated by `,` `+` and `-` (three hierarchical levels) or numbers `1.`, `i.` and `A.` (numbered lists). Indentation of lists indicates their level: `,` `+` and `-` may be replaced by `-` at all levels, but four spaces are needed to nest a list into another.

- First element of a list
 - sub-list
- Second element
- Continuation of the list

Leave an empty line before and after the list, but not between its items.

3.6 Math

Equations in LaTeX format can be inserted in line, like $A = \pi r^2$ or isolated like

$$e^{i\pi} = -1$$

They can be numbered, see equation (1), using the `equation` environment:

$$A = \pi r^2. \tag{1}$$

Table 2. Intervention table, summary of the disturbance intensity for the 4 plot treatments in Paracou.

Treatment	Timber	Thinning	Fuelwood	%AGB lost
Control				0
T1	DBH \geq 50 cm, commercial species, \approx 10 trees/ha			[12% – 33%]
T2	DBH \geq 50 cm, commercial species, \approx 10 trees/ha	DBH \geq 40 cm, non-valuable species, \approx 30 trees/ha		[33% – 56%]
T3	DBH \geq 50 cm, commercial species, \approx 10 trees/ha	DBH \geq 50 cm, non-valuable species, \approx 15 trees/ha	40 cm \leq DBH \leq 50 cm, non-valuable species, \approx 15 trees/ha	[35% – 56%]

3.7 Cross-references

Figures and tables have an automatically generated label, identical to the name of the code snippet prefixed with `fig:` and `tab:`.

For equations, the label is added manually by the code `(\#eq:xxx)` before the end of the equation.

Sections can be tagged by ending their title with `{#yyy}`.

Bookmarks can also be placed freely in the text with the command `(ref:zzz)`.

In all cases, the call to the reference is made by the command `\@ref(ref:zzz)`.

3.8 Bibliography

Bibliographic references included in the `references.bib` file can be called by `[@CitationKey]`, in parentheses (Xie, 2016), or without square brackets, in the text, as Xie *et al.* (2018).

The bibliography is processed by Pandoc when producing Word or HTML documents. The bibliographic style can be specified, by adding the line

`csl:file_name.csl`

in the document header and copying the `.csl` style file to the project folder. More than a thousand styles are available⁴.

For PDF documents, the bibliography is managed by LaTeX. The style is included in the template: it is the one from *Methods in Ecology and Evolution*. It cannot be changed, to ensure the homogeneity of the documents produced.

To prepare the submission of a manuscript to a journal, it will be necessary to open the intermediate `.tex` file produced by Pandoc and to copy the contents of the environment into the template proposed by the journal, which will take care of the formatting.

3.9 LaTeX preamble

LaTeX commands can be added in the preamble of the produced LaTeX file, for example to load additional packages.

⁴<https://github.com/citation-style-language/styles>

These commands are in the `preamble:` section of the Markdown file header.

The default commands allow:

- To use the degree character (example: 20°C):

```
\usepackage{textcomp}
\DeclareUnicodeCharacter{B0}{textdegree}
```

- To obtain two-column tables by redefining the `table` environment and loading the `tabu` package necessary for *kableExtra*:

```
\usepackage{tabu}
\renewenvironment{table}{%
  \begin{table*}}{%
  \end{table*}%
  \ignorespacesafterend
}
```

- To show the use of the hyphenation command:

```
\hyphenation%
{bio-di-ver-si-ty sap-lings}
```

Other commands can be added as needed. Warning:

- Comments are not possible.
- Complex commands (such as `renewenvironment`) must be entered on a single line otherwise they will be destroyed by knitr at the first knitting in HTML.

3.10 Forcing line breaks

Hyphenation is handled automatically in LaTeX. If a word is not hyphenated correctly, add its hyphenation in the preamble of the file with the command `hyphenation` (words are separated by spaces, hyphenation locations are represented by dashes).

If LaTeX can't find a solution for the line break, for example because some code is too long a non-breaking block, add the LaTeX command `\break` to the line break location. Do not leave a space before the command. The HTML document ignores LaTeX commands.

3.11 Languages

English and French are supported, to be declared in the document header.

The chosen language has an effect only in LaTeX output: a space is added in front of double punctuation in French, the size of spaces is bigger at the beginning of a sentence in English, etc.

LaTeX templates use the LaTeX package *babel* which recognizes the language names *English* and *french*. Pandoc on the other hand uses the IETF codes *fr-FR* or *en-US* for limited language support in HTML. Both parameters must therefore be entered in the document header.

4. Document types

This template is designed to work with the Stylish Article template in LaTeX and produce documents in PDF, HTML or Word format. Use the list of choices in the *Knit* button to choose the output format.

The Word document can then be modified to comply with the instructions to journal authors: double-spaced, font, etc.

4.1 PDF Document

The document is formatted for self-archiving of well-formatted articles.

4.2 HTML document

The GitBook template is optimized for on-screen reading. While writing, prefer knitting to HTML format for its speed of execution. A download button is available in the document menu bar: it will work if the document is also knitted in PDF format and if the file name is entered in the download field of the YAML header.

The HMTL Book template is an alternative.

4.3 Word document

Its content can be formatted or copied into a template. The standard text styles are “First Paragraph” and “Body Text”.

The advantage of the Word format is to produce a manuscript for journals that do not support LaTeX. The bibliographic style of the journal is most likely available in *.csl* format, which minimizes manual preparation.

The last line of this template (R code snippet) must be kept to display the title *References* (to be translated into the document language if necessary) in HTML format. The level 1 title *References* must be added manually to Word files.

References

- Xie, Y. (2016) *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida.
- Xie, Y., Allaire, J. & Grolemund, G. (2018) *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida.