Title of the Article

First Author's name[1]*[iD]

Second Author's name[2]

Abstract
Summary of the article. Mandatory.
Use double line feed for a new paragraph.

Keywords
keyword1, keyword2, etc

JEL
Code1, Code2, etc, MayBeDeleted

Acknowledgements
Acknowledgements.
May be deleted.

[1]Department / University
  Street address, Zip code, Country.
[2]Department / University
  Street address, Zip code, Country.
*Corresponding author: name@company.com, https://www.company.com

Contents

1. Introduction

This template allows writing articles in Markdown[1] 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 website[2].

Markdown is extended by Bookdown[3], which allows for 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.

---

[1]https://ericmarcon.github.io/WorkingWithR/chap-rediger.html

[2]http://rmarkdown.rstudio.com/articles.html
[3]https://bookdown.org/yihui/bookdown/

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

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
```

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 `kbl()` function in the *kableExtra* package (which enhances the original `kable()` function from *knitr*) prepares the table for display and passes the result to the `kable_styling` function for final formatting.
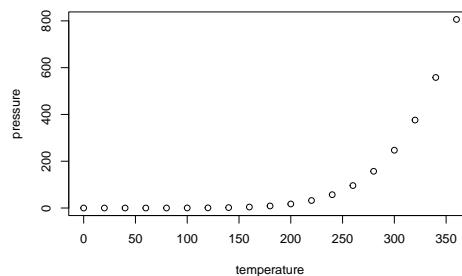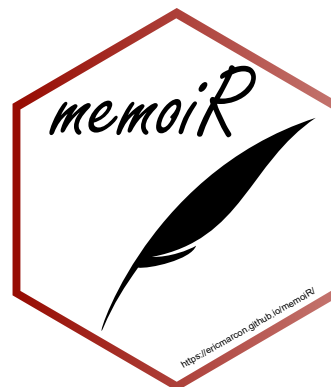
```
names(iris) <- c("Sepal length", "Width", "Petal length",
    "Width", "Species")
kableExtra::kbl(head(iris), caption = "Table created by R",
    longtable = TRUE, booktabs = TRUE) |>
    kableExtra::kable_styling(bootstrap_options = "striped",
        full_width = FALSE)
```

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 (`longtable = TRUE`), or use the width of the page (`longtable = FALSE`), 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

Figure 1. Figure title

Figure 2. A figure from a file

formatting of two-column tables, i.e. with `longtable = FALSE` in LaTeX.

The content of table cells can be formatted following the Markdown syntax, with some limits due to *kableExtra*: the argument `format="markdown"` is necessary in `kbl()` or LaTeX output will ignore formatting, but this is not compatible with `full_width = TRUE` in `kable_styling()`. See *kableExtra*'s documentation to format rows or columns globally without using Markdown (e.g. function `row_spec()`. The header of table 2 is set to bold this way.

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 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.

In the HTML output, figure 2 is located in the

Table 2. Disturbance intensity of 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	$[35\% - 56\%]$

margin of the document thanks to the advanced formatting possibilities of the **distill**[4] package.

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 `kbl()` 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:

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

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}.

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

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 available[5].

For PDF documents, the bibliography is managed by natbib. The style is declared in the header:

biblio-style: chicago

It can be changed as long as the appropriate `.bst` file (by default: `chicago.bst`) is included in the project.

3.9 LaTeX preamble

LaTeX commands can be added in the preamble of the produced LaTeX file, for example to load additional packages. These commands are in the `preamble:` section of the Markdown file header.

The default commands allow to show the use of the hyphenation command:

---

[4]https://rstudio.github.io/distill/

[5]https://github.com/citation-style-language/styles

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

Other commands can be added as needed. Warning:

- Comments are not allowed.
- Complex commands (e.g. `\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

Languages are declared in the document header.

The main language of the document (`lang`) changes the name of some elements, such as the table of contents. The change of language in the document (one of `otherlangs`) is managed in LaTeX but not in HTML by inserting on a new line the following command:

```
\selectlanguage{english}
```

The current language has an effect only in LaTeX output: a space is added before double punctuation in French, the size of spaces is larger at the beginning of sentences in English, etc. The `\selectlanguage` command is simply ignored in HTML.

Language codes are used in the header, such as `en-US` but language names are necessary in `\selectlanguage{}`. Name matches are listed in table 3 of the polyglossia package documentation[6].

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.

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 Document and all formats from the **rmdformats** packages are other possibilities.

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.* Boca Raton, Florida: Chapman and Hall/CRC.

Xie, Y., J. Allaire, and G. Grolemund (2018). *R Markdown: The Definitive Guide.* Boca Raton, Florida: Chapman and Hall/CRC.

---

[6]http://mirrors.ctan.org/macros/unicodetex/latex/polyglossia/polyglossia.pdf