Tutorial R Markdown

Jacolien van Rij

Table of contents

Using R Markdown as analysis script

This tutorial describes how to use **R Markdown**. R Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. Example: this tutorial itself is generated with R Markdown.

For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

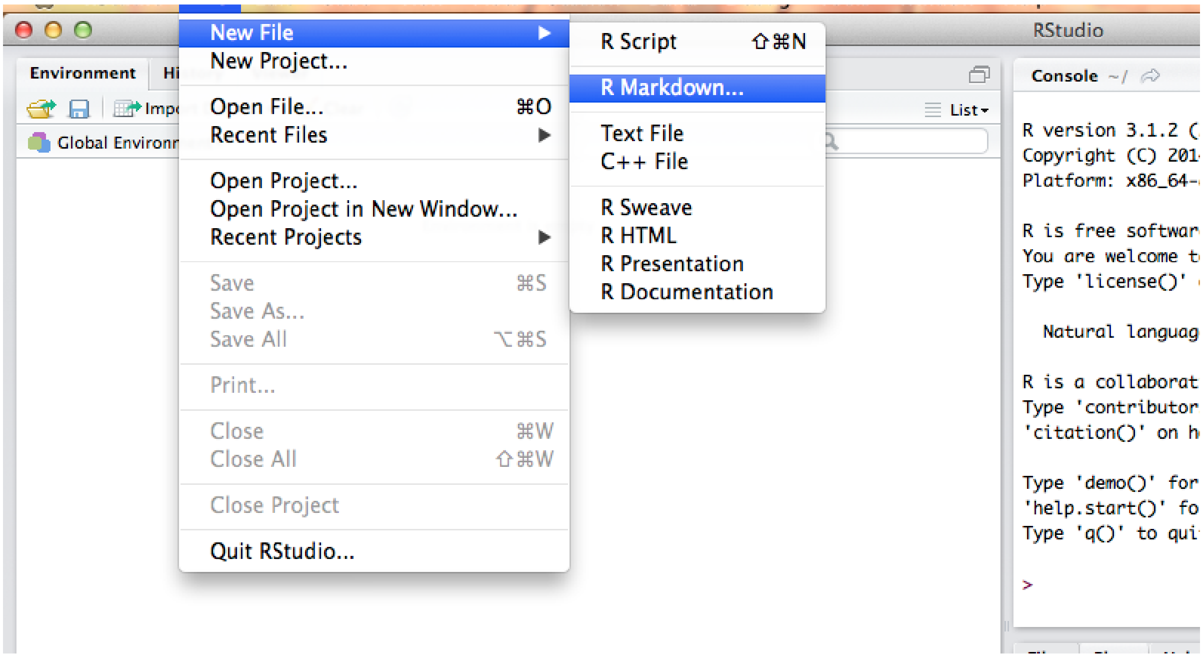
Some advantages of using R Markdown:

1. R code can be embedded in the report, so it is not necessary to keep the report and R script separately. Including the R code directly in a report provides structure to analyses.
2. The report text is written as normal text, so no knowledge of HTML coding is required.
3. The output is an HTML file that includes pictures, code blocks R output and text. No additional files are needed, everything is incorporated in the HTML file. So easy to send the report via mail, or place it as paper package on your website.
4. These HTML reports enhance collaboration: It is much more easy to comment on an analysis when the R code, the R output and the plots are available in the report.

How to generate an HTML report

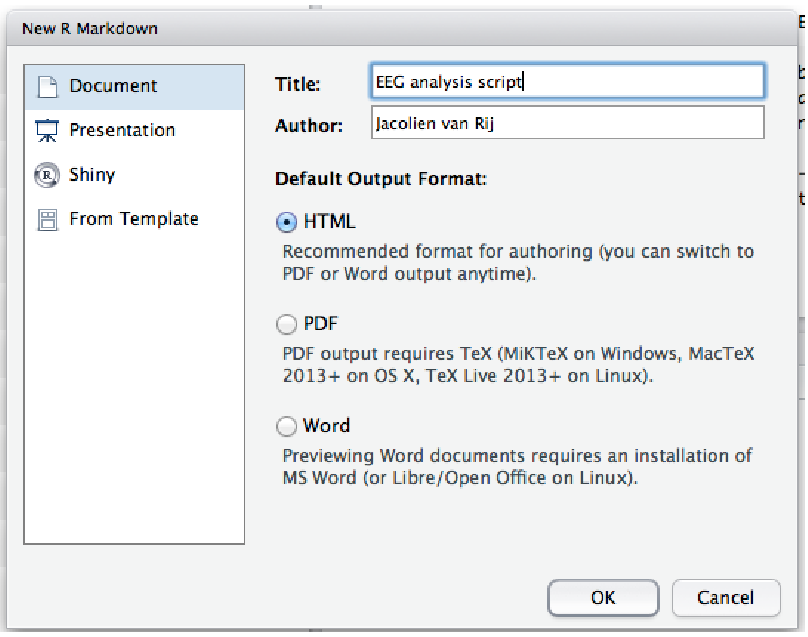
Method 1: Start a new R Markdown file in R Studio

Open R Studio, then go to File → New file → R Markdown.



```
R version 3.1.2 (
Copyright (C) 201
Platform: x86_64-

R is free softwar
You are welcome t
Type 'license()'

  Natural languag

R is a collaborat
Type 'contributor
'citation()' on h

Type 'demo()' for
'help.start()' fo
Type 'q()' to qui

>
```

Choose a title and author, but leave the rest as it is, and press the OK button:



Title: EEG analysis script

Author: Jacolien van Rij

Default Output Format:

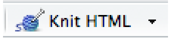
● HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

The R markdown file you opened, already contains example code.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document:



More info on how to format the HTML report in the section **Formatting the report**.

Method 2: If you really don't want to use R Studio…

This method requires that pandoc (http://pandoc.org) document converter software is installed (see instructions on the pandoc website).

After pandoc is installed, install the R package `rmarkdown` and it's dependencies. Type the following command in the R console:

```
install.packages("rmarkdown", repos="http://cran.us.r-project.org")
```

Download here a template of a markdown file (from R Studio) and store it in your working directory: download template (template.Rmd).

Open the file `template.Rmd` in a text editor, and add your R code in the code blocks, which start with ` ```{r}` ` and end with ` ``` `. Around the blocks you could type your report. Example code is provided in the file `template.Rmd`.

Run the following code in the R console to convert the `.Rmd` file to an HTML report (courtesy of Martijn Wieling):

```
# make sure to have the package rmarkdown updated and installed:
```

```
library(rmarkdown)
# generates html file with results
render('template.Rmd')
```

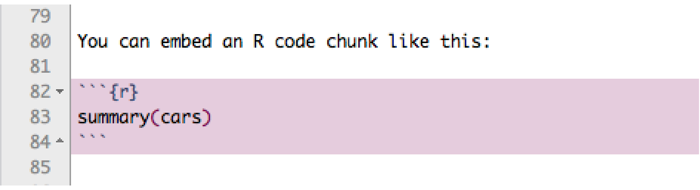
The `render` function will generate an HTML file in your working directory. You could view the HTML in your browser, or use the following code to view the result in R:

```
browseURL(sprintf('file://%s%s', 'template.html', file.path(getwd())))
```

More info on how to format the HTML report in the section **Formatting the report**.

Embedding R code

You can embed an R code chunk like this:



~~~
You can embed an R code chunk like this:

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

~~~

This will look as follows in the HTML file:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

The HTML file will first **echo** the commands in the R code block, then **eval**uates the R code, and **include** the output.

Arguments

Three arguments can change this behavior:

- The argument **echo** specifies whether the R commands are included (default is `TRUE`). Adding `echo=FALSE` in the opening line of the R code block will not include the commmand: ` ```{r, echo=FALSE} `

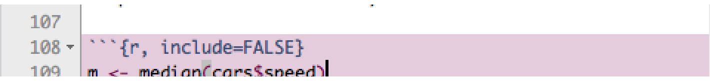
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

- The argument **eval** specifies whether the R commands is evaluated (default is `TRUE`). Adding `eval=FALSE` in the opening line of the R code block will not evaluate the commmand: ` ```{r, eval=FALSE} `. Now only the command is shown, but no output.

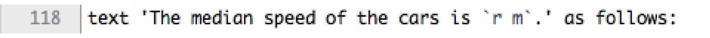
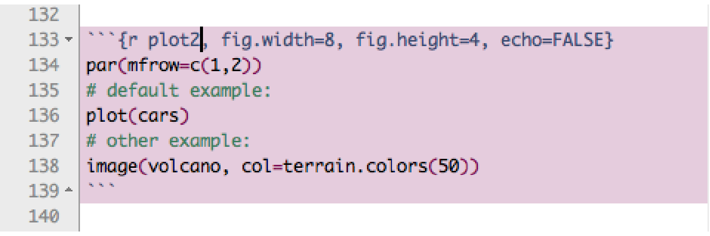
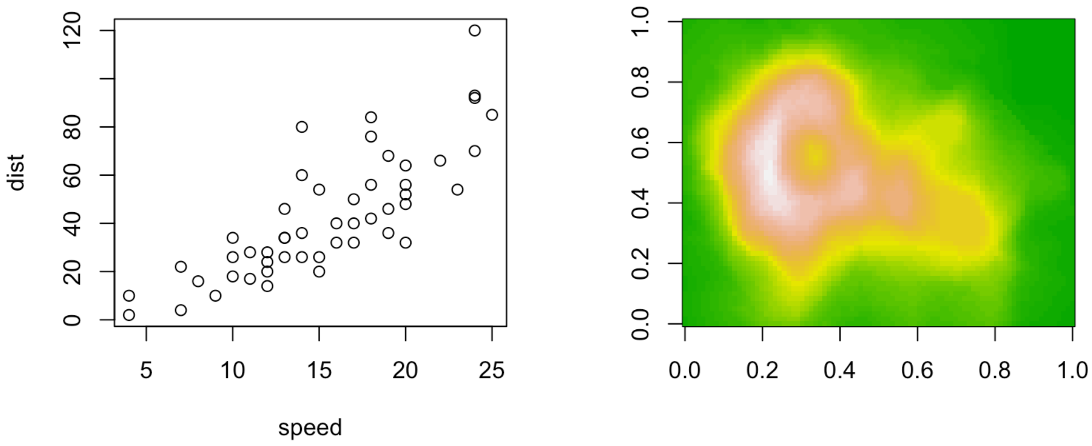
```
summary(cars)
```

- The argument **include** specifies whether the output is included (default is `TRUE`). Adding `include=FALSE` in the opening line of the R code block will not include the commmand: ` ```{r, include=FALSE} `. Now the command and the output are both not shown, but the statement is evaluated.

For example:

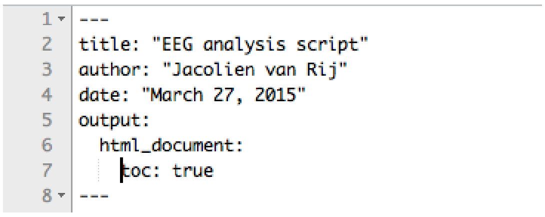


~~~
```{r, include=FALSE}
m <- median(cars$speed)
~~~

```
m <- median(cars$speed)
```
```

We can now include the value of the variable `m` in the text, because the statement is evaluated, although not presented in the output. We can inlcude the following text 'The median speed of the cars is 15.' as follows:



```
text 'The median speed of the cars is `r m`.' as follows:
```

It is good practice to provide a name for each R code block, which helps with debugging. Add a name nameblock directly behind the `r` : ` ```{r nameblock, echo=FALSE} `. Note that the name should be unique.

Embedding plots

You can also embed plots with specific size, for example:



````
```{r plot2, fig.width=8, fig.height=4, echo=FALSE}
par(mfrow=c(1,2))
# default example:
plot(cars)
# other example:
image(volcano, col=terrain.colors(50))
```
~~~

Wil result in:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

## Running models on the server or analyses that take time

If you're running analyses that take a long time, or you're running the models on a server:

Include the script for the analysis in your R Markdown file, but make sure the model does not evaluate. Save the model, and load it for further use. For example, include something like this:

```
# Setting an if-statement to false, will prevent the evaluation of this part.
if(FALSE){
  # run this code on the server:
  model <- gam(Y ~ s(Time), data=dat)
  save(model, file="Models/model.rda", compress="xz")
}
# Now download the model, and store it in the directory Models
# Load the model:
load("Models/model.rda")
# ... and continue with your script.
```

# Formatting the report
~~~

The website http://rmarkdown.rstudio.com/authoring_pandoc_markdown.html (http://rmarkdown.rstudio.com/authoring_pandoc_markdown.html) provides more information on how to include lists, pictures, etc. Here only a few things are discussed.

- header 1: `# header 1`
- header 2: `## header 2`
- header 3: `### header 3`
- header 4: `#### header 4`
- **bold** text: `**bold** text`
- italics text: `*italics* text`
- `code` text: `` `code` text ``
- inlcude a link (www.sfs.uni-tuebingen.de): `[link](www.sfs.uni-tuebingen.de)`
- Including a picture: `![R Studio Logo](img/R Studio Logo.png)`
- Table of contents: you need to change the settings of the R Markdown file. Go to the top of the file and change the line `output: html_document` as follows:



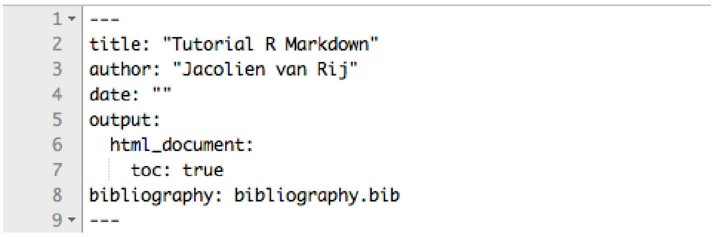
```
---
title: "EEG analysis script"
author: "Jacolien van Rij"
date: "March 27, 2015"
output:
  html_document:
    toc: true
---
```

Although no HTML coding is required, HTML could optionally be used to format the report. The HTML tags are interpreted by the browser after the `.Rmd` file is converted into an HTML file. Some examples:

- color word: `<span style="color:red">color</span> word`
- mark word: `<span style="background:yellow">mark</span> word`
- quick way of adding more space after text block: `<p></br></p>`

# References

It is possible to include a BibTex bibliography file in the settings of the R Markdown file:



```
---
title: "Tutorial R Markdown"
author: "Jacolien van Rij"
date: ""
output:
  html_document:
    toc: true
bibliography: bibliography.bib
---
```

Download here an example of a Bibtex bibliography file: download bibliography (bibliography.bib).

## In-line citations

R Markdown accepts in-line citations. The sentence '...as Lastname and MyLastname (2012) proposed (see Last Name and other Last Name 2015)...' was generated by the following text:

```
...as @ref2 proposed [see @ref1]...
```

See (http://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html)http://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html (http://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html) for more info on this topic.

## Reference list

The references are listed at the end of the HTML file. An example of an automatically generated reference list on the basis of bibliography.bib

The references are listed at the end of the HTML file. An example of an automatically generated reference list on the basis of bibliography.bib (bibliography.bib):

Last Name, First Name, and FirstName-Two other Last Name. 2015. “Article Title.” Journal of Something 2 (3): 95–150.

Lastname, Name, and My Name MyLastname. 2012. “Other Article Title.” New Journal of Something Else 5: 1–10.