A Very Short Introduction to R Markdown by Example

Johannes Binswanger

November 24, 2016

What this document is about

Let's start this text with an important warning. In the so-called YAML header that configures this `Rmd` file, you see the `date` item. If you happen to use a German date format there with a period after the number, like `24. November 2016`, then the compilation to *pdf* will not work. However, it works if you set it to `24\\. November 2016`. It can easily be that the date is automatically set to a German format if you open a new Rmarkdown file in RStudio.

If you want to spellcheck your text, put the cursor at the beginning of the chunk of text that you want to check and then press the little green checking symbol with "ABC" on top in the taskbar in the upper left corner of the RMarkdown document. The rest is self-explaining.

The most important elements for text editing

Links

Let's start with links. You use a link by placing it inside `<>`. A very useful link goes to the R Markdown cheatsheets. The one in English can be found under https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf. If you find this lengthy link to clumsy, here is a more elegant example:

- Please find the link to the German cheatsheet here.

A very useful link is this one!

More on headings

You can generating heading x by placing x hashtags (e.g. `##` for heading 2) in front of the heading text.

More formatting elements

You can put text in *italics* by writing `*italics*`. And you put it in **bold** with `**bold**`. If you want to have code appear as `code`, put it inside these little apostrophes ‘ ‘.

A unnumbered list looks like this:

- First item
- Second item
- Third item

Make sure that there is an empty line before the list! Otherwise, it looks different. A numbered list looks like this

1. First item

2. Second item
3. Third item

Again, include an empty line before the list

Mathematical expressions

You can use mathematical expressions by using Latex syntax (see e.g. https://www.sharelatex.com/learn/Mathematical_expressions). Let's copy a chunk of text from that webpage:

> The well known Pythagorean theorem $x^2 + y^2 = z^2$ was proved to be invalid for other exponents. Meaning the next equation has no integer solutions:
>
> $$x^n + y^n = z^n$$

Note that I used the blockquote formatting for the quote from the sharelatex webpage. The blockquote syntax is just >, extremely simple! Again, insert an empty line before.

You can also do it like this:

$$\int_{-\infty}^{\infty} xf(x)dx = 1. \tag{1}$$

How to include R

Now it is time to include output from R.

Here is a chunk of code that is visible in the output document. (**Adjust the working directory to your situation!**)

```
# Load data.
# We first use the full (absolute) path,
# in order to avoid setwd()
# ADJUST IT TO YOUR SITUATION

load("D:/Dropbox/Mac&Surf/Programmierkurs Dropb/Data/dataForAnalysis.RData")

# And some output
names(dataAussen)
```

```
## [1] "Date"   "D0"     "D1"     "D2"     "Value"  "timeID"
```

```
unique(dataAussen$D1)
```

```
## [1] "Total"
## [2] "Maschinen, Apparate und Elektronik"
## [3] "Praezisionsinstrumente, Uhren, Bijouterie"
## [4] "Chemikalien"
## [5] "Textilien, Bekleidung, Schuhe"
```

```
## [6] "Fahrzeuge"
## [7] "Metalle"
## [8] "Uhren"
## [9] "Praezisionsinstrumente"
```

You can also try to use `setwd()`:

```
setwd("D:/Dropbox/Mac&Surf/Programmierkurs Dropb/Data")
load("dataForAnalysis.RData") # Load data

# And some output
ls()
```

```
## [1] "dataAussen" "dataXrates"
```

However, `setwd()` does not always work. If neither the full path for loading the data works, nor `setwd()`, then put your markdown file in the same folder as where you have your data.

In any case, changing the working directory via `setwd()` always produces a warning message. Nevertheless, I personally use it whenever it is convenient (and when it happens to work).

Here is something very important to realize:

> **The working directory of R Markdown is quite separate from R. R Markdown often has no clue what the current working directory is of your R session. It's really two separate things.**

Now let's repeat the first chunk of code, but with an additional argument added to `{r}`, namely `{r, echo = FALSE}`:

```
## [1] "Date"   "D0"     "D1"     "D2"     "Value"  "timeID"
```

```
## [1] "Total"
## [2] "Maschinen, Apparate und Elektronik"
## [3] "Praezisionsinstrumente, Uhren, Bijouterie"
## [4] "Chemikalien"
## [5] "Textilien, Bekleidung, Schuhe"
## [6] "Fahrzeuge"
## [7] "Metalle"
## [8] "Uhren"
## [9] "Praezisionsinstrumente"
```

You see that the code that generates the output is no longer shown. You can also do exactly the opposite, i.e. show the code but suppress the output using `{r, eval = FALSE}`

```
load("dataForAnalysis.RData") # Load data

# And some output
names(dataAussen)
unique(dataAussen$D1)
```

No output is shown in this case.

Including output from R in the text

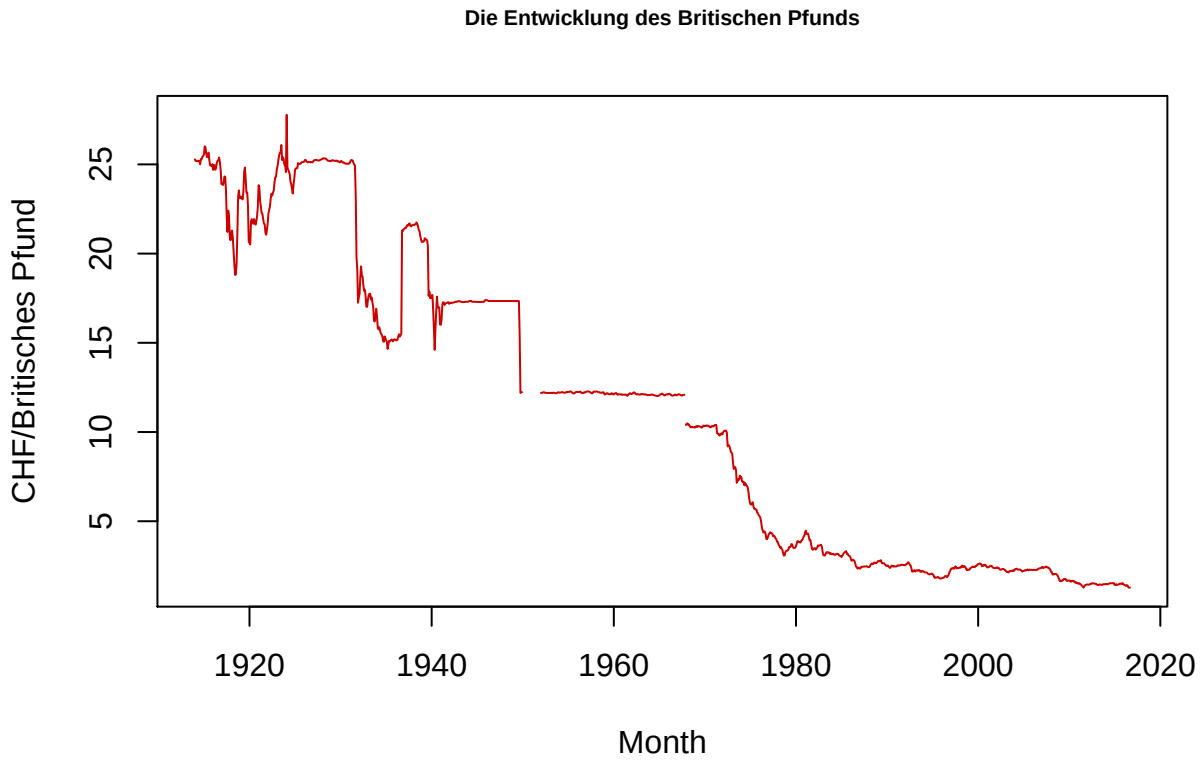
Suppose we want to talk of the mean of a variable in the text. We can directly include R expressions in the text using `<apostrophe> r <expression> <apostrophe>`. The following blockquote contains an example.

> The mean of the variable CHF/Britisches Pfund is 11.0870803. Or, rounded to only 2 digits, this is 11.09.

Note that you can still interactively evalue R expressions inside an R Markdown document by selecting the expression and hitting `control/command + Enter`. That is very convenient.

Including graphs

Here is how to include a graph (without the generating code):



More sophisticated formatting

So far, our document looks reasonably good in both html and pdf. If you want to go for more sophisticated formatting, you often have to make a choice. The reason is that you will need syntax that is specific to either html or latex. Quite often, latex and pdf is the better choice, or at least the more convenient one. You will see examples of how to use latex code in `Lec7_AnalysInMarkdown.Rmd` and `getAnalysis.R`. Important instances are tables and colored text.