Bookdown with a single document

Han Oostdijk

23 January 2020

Intro

With this document I am experimenting with the **bookdown** package (Xie, 2020), which was built on top of R Markdown and **knitr** (Xie, 2015). At the moment I see no reason to compile a book, but I think it useful to have the features of **bookdown** available for producing a single `html` or `pdf` document.

After I had finished my experiments I tried to document what I had done and why. Unhappily I could not retrace the document (not changed by me) that served as a starting point. But most probably it must have been authored by Yihui Xie. I have said it earlier, but I appreciate his work very much.

I changed the original document (that was a kind of template that showed some possibilities) to be able to:

- produce a text that would display well in both `html` and `pdf` format. I could not do this without including some switches related to the output format using `knitr::is_latex_output` and `knitr::is_html_output`
- produce tables and figure with a caption **below** the table or figure with a possiblity to reference them
- use `css` files to ensure that the table entries are not too close to the borders and that the tables with caption are **centered** in the `html` output. See the `mycss.css` file
- use text references to refer to a text (optionally with special `html` or `latex` characters). This is used here only for the tables. Note that a text reference must be in a separate paragraph with empty lines above and below it
- integrate the possibilities that the `myknit` function (from the HOQCutil package) (Oostdijk, 2019) offers to produce documents with a version number
- use references to a bibliography and sections of text (see the files `book.bib` and `packages.bib`)
- change the font of the `pdf` document to a non-serif one (see the `preamble.tex` file)
- try out methods to make the document self describing concerning the creation of tables and figures. In doing this I relied heavily on the concept of zero-width space. See the `.Rmd` file in a text editor that is capable of showing zero-width spaces such as the RStudio IDE

So the following text is a mix of the original document and changes that I made to test if these objective could be met.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

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.

Note that I changed the document to include the `knit` statement in the `YAML` prefix. That is why it is no longer possible to indicate if the `html` or ‘`pdf` version should be knitted: clicking the button just executes the first output format specified. If you want to produce the second one, you have to change the order by manually ’cut and paste’.

Including a table

Coding like this:

~~~
(ref:taba) My *first* table is a very small table

```{r cars,results='asis'}
 print(
  xtable::xtable(summary(cars)
    ,caption='(\\#tab:cars) (ref:taba)'
  )
  ,include.rownames =F
  ,html.table.attributes = 'width=70%'
 )
```
The resulting table can be found in Table \@ref(tab:cars).
~~~

prints the code, the table and the following output:

```
print(
  xtable::xtable(summary(cars)
    ,caption='(\\#tab:cars) My first table is a very small table'
    )
  ,include.rownames =F
  ,html.table.attributes = 'width=70%'
  )
```

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

Table 1: My *first* table is a very small table

The resulting table can be found in Table 1.

end of table output

Note that the `width` attribute only works in html output. And in case of latex output the placement of tables and figures is determined by the latex software. Therefore we can not garantee that the table output is indeed displayed before the **end of table output** line.

The code used can be packed in the following function:

```
print_df <- function(df,cap,
     include.rownames=F,html.table.attributes = '',
     type='latex',comment=T) {
  chunk_opts=knitr::opts_current$get()
  caption = paste0('(\\#tab:',chunk_opts$label,') ',cap )
  print(
  xtable::xtable(df,caption=caption)
```

```
  ,type = getOption("xtable.type",type) # set in setup chunk
  ,include.rownames =include.rownames
  ,html.table.attributes = html.table.attributes
  ,comment=getOption("xtable.comment",comment) # set in setup chunk
  )
}
```

Note that I have set the options `xtable.type` and `xtable.comment` in the `knitr` setup chunk: the first depending on the type of document that is produced (so depending on the first output format in the `YAML` header) and the second fixed to `FALSE`. Even while it is fixed here, I wanted to leave open the option to use the function elsewhere and therefore parameterised these values.

This function can then be called in the following way:

~~~
(ref:taba2) Same contents as Table \@ref(tab:cars) but different width

```{r cars2,results='asis'}
print_df(summary(cars),cap='(ref:taba2)',html.table.attributes = 'width=50%')
```
The resulting table can be found in Table \@ref(tab:cars2).
~~~

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

Table 2: Same contents as Table 1 but different width

The resulting table can be found in Table 2.

Note that in the LaTeXoutput the width is not changed by html.table.attributes!

Including a plot

You can also embed plots, for example:

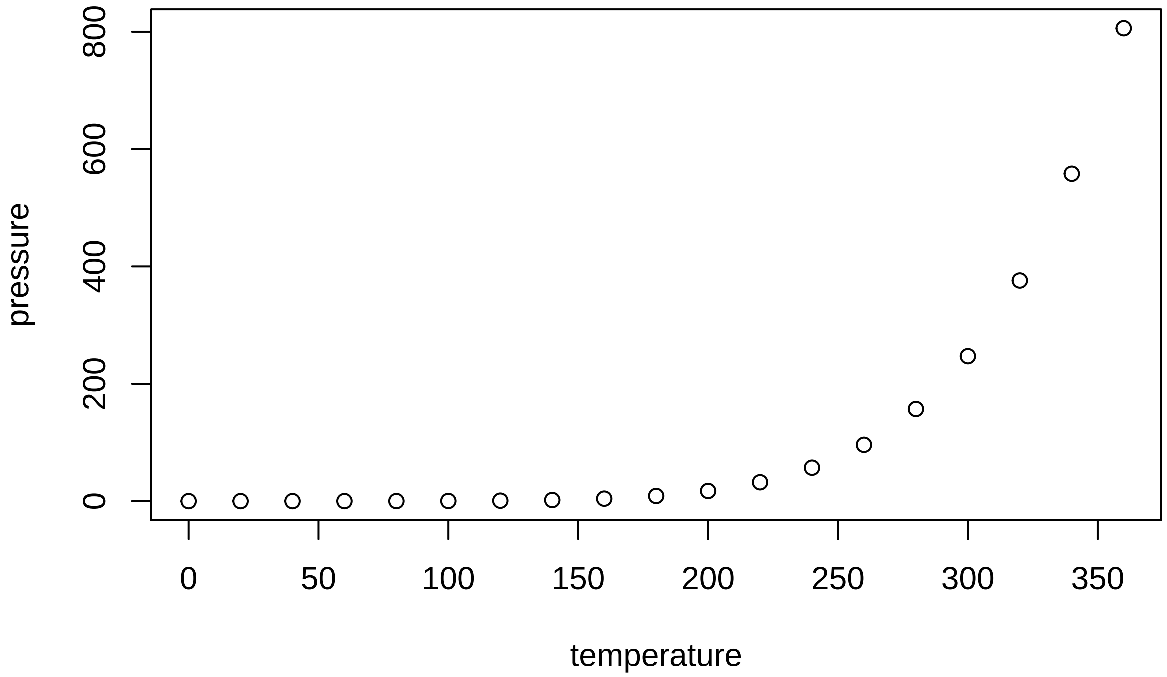

Figure 1: My *first* figure

See Figure 1 for the plot.

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

Including a plot (code included)

We will plot it again and now show how to code the plotting in such a way that we:

- see the code that is executed (by specifying the `echo = TRUE` parameter)
- specify the caption by setting the `fig.cap` parameter directy to a character string without using text references (which is also possible)
- are able to refer to the plot (by using \@ref(fig:pressure2))

Coding like this:

~~~
figb = "My *second* figure"
```{r pressure2, echo=TRUE, fig.cap=figb}
plot(pressure)
```
See Figure \\ref{fig:pressure2} for the plot.
~~~

prints the conde, the plot and the following output:

```r
plot(pressure)
```

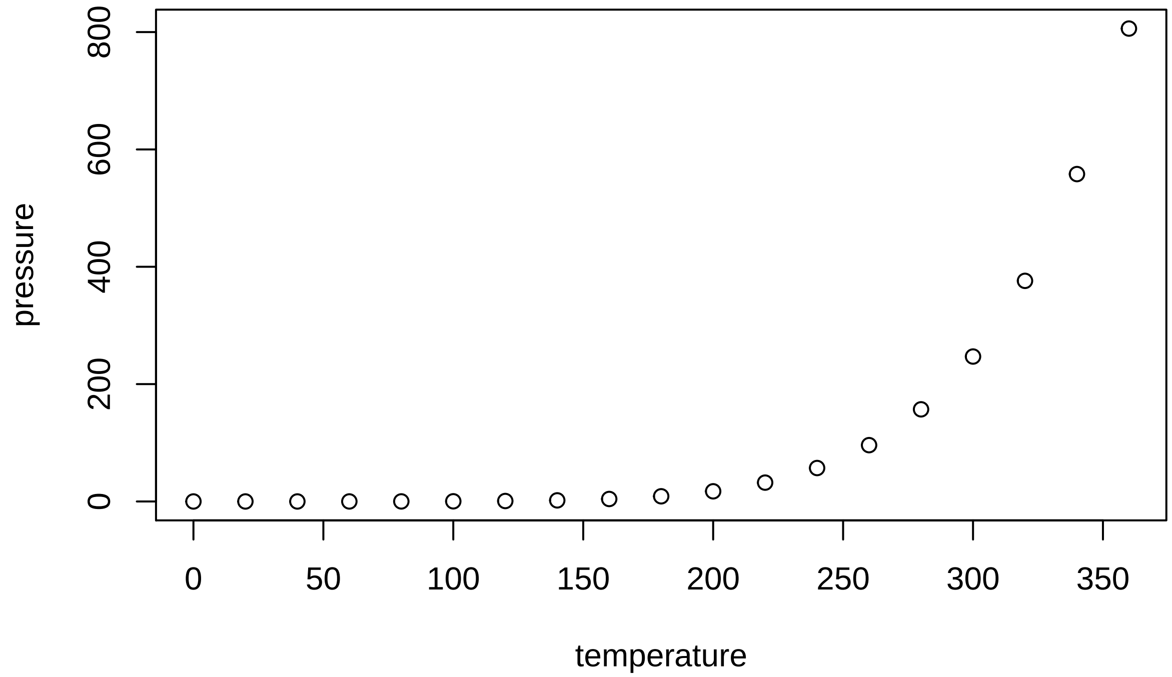

Figure 2: My *second* figure

See Figure 2 for the plot.

end of plot output

For details about the packages used see Intro and Session Info.

Session Info

```
# R version 3.6.0 (2019-04-26)
# Platform: x86_64-w64-mingw32/x64 (64-bit)
# Running under: Windows 10 x64 (build 18362)
#
# Matrix products: default
#
# locale:
# [1] LC_COLLATE=English_United States.1252
# [2] LC_CTYPE=English_United States.1252
# [3] LC_MONETARY=English_United States.1252
# [4] LC_NUMERIC=C
# [5] LC_TIME=English_United States.1252
#
# attached base packages:
# [1] stats     graphics  grDevices utils     datasets  methods   base
#
# loaded via a namespace (and not attached):
#  [1] Rcpp_1.0.3      bookdown_0.17   digest_0.6.23   HOQCutil_0.1.15
#  [5] xtable_1.8-4    magrittr_1.5    evaluate_0.14   highr_0.8
#  [9] rlang_0.4.2     stringi_1.4.5   fs_1.3.1        rmarkdown_2.0
# [13] tools_3.6.0     stringr_1.4.0   glue_1.3.1      xfun_0.10
# [17] yaml_2.2.0      compiler_3.6.0  htmltools_0.4.0 knitr_1.27
```

References

Oostdijk, H. (2019). *HOQCutil: Utilities by Han Oostdijk*. R package version 0.1.15.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.17.