

Flex Knit

Han Oostdijk

file created : 18Feb2018

Synopsis

This document presents the R-function `myknit` that enables dynamically changing the output files of the knit-process by using the undocumented `knit` yaml option.

Acknowledgement

The idea for manipulating yaml information in the way done here was found in a [Stack Overflow article](#) (option 2) authored by *mathematical-coffee*.

Introduction

When creating a new document I always have to try various versions of the yaml metadata block before I have a version that works for me. I need an easy way to keep these various inputs and outputs separated. The standard way (the `knit` button) always creates/overwrites the same output file. The `rmarkdown::render` function that is called as a result of clicking the button allows specifying a name for the output file. I found (see acknowledgement) a method to call this function with the undocumented (why?) `knit` yaml option. The `knit` option specifies what is executed when the `knit` button is clicked: a direct call to `rmarkdown::render` or a user function.

I use the construction `knit: (function (...) { source('myknit.r'); myknit(...) })` with the user written function `myknit`. The `myknit` function supports some additional options in the yaml metadata block and uses these to construct output names.

Functionality of the `myknit` function

By specifying additional options in the yaml metadata block the user can do one or more of the following things:

- specify an alternative name for the output file
- append a version indicator to the name of the output file
- force that the proper extension is given to the name of the output file (when not specified)
- create a file with the yaml metadata block that was specified with a name including the version indicator
- create a file with the yaml metadata block after processing with a name including the version indicator
- create a copy of the input file with a name including the version indicator

How to use the functionality of the `myknit` function

The `myknit` function is controlled by additional options in the yaml metadata block. To distinguish these options from the regular ones, they are prefixed with `hoqc_`. The additional options (with defaults in parentheses) are:

- `hoqc_output` ('') : the name of the output document file
- `hoqc_version` ('') : suffix to be given to files to indicate version
- `hoqc_force_ext` (F) : should a filename extension be forced for document and yaml files? (TRUE or FALSE)
- `hoqc_yaml` ('') : the name of file containing the original yaml metadata block
- `hoqc_yaml_new` ('') : the name of file containing the new (modified) yaml metadata block

- `hoqc_copy` (' ') : the name of file containing a copy of the input file

The output document will in principle be named `hoqc_output` if this is specified; otherwise it is named after the input file. If `hoqc_output` is specified without an extension and `hoqc_force_ext == T` then the output name will be given an extension of `pdf` or `html` depending on the (first encountered, not commented) `document_type`. `hoc_version` will be inserted after the basename of the file.

The original yaml metadata block will be written to an output file with name `hoqc_yaml` if this is specified. After reading the original yaml metadata block comment lines (beginning with `#`) are removed. The modified yaml metadata block will be written to an output file with name `hoqc_yaml_new` if this is specified. In both yaml file names the `hoc_version` will be inserted and a `txt` extension will be given when an extension was not specified and `hoqc_force_ext == T`.

A copy of the input file is created with name `hoqc_copy` if this is specified. The `hoc_version` will be inserted and a `Rmd` extension will be given when an extension was not specified and `hoqc_force_ext == T`.

In the modified metadata block the additional options are included (added if it already exists) in the `params` block. The name of the output files are expanded to the full path. The next paragraph shows how the `params` block can be used to include e.g. the yaml metadata block in the document by specifying `r params$hoqc_yaml` within backticks.

A usage scenario for working with LaTeX could be:

- set `hoqc_output` and `hoqc_copy` e.g. to `./output/flexkit` and `keep_tex` to `yes`
- set `hoc_version` to `_v1` and knit the file.
- set `hoc_version` to `_v2`, make some changes and knit again

Intermediate folders will be created if necessary.

In the output subfolder one now can find for both versions the saved inputs (`flexknit_v1.Rmd` and `flexknit_v2.Rmd`) and the corresponding `tex` files and `pdf` files (with version indicators). When something is not working as expected one can compare the various versions to see what is wrong. In this way I saw that the yaml options for `geometry` and `params` do not allow blanks between the options and the colon.

Listing of the yaml metadata blocks

As an example the original yaml (in `D:/data/R/rmd_pdf_examples/output/yaml_v1.txt`) for this document will be shown:

```
cat(paste0(readLines(params$hoqc_yaml),collapse = '\n'))
```

```
---
title           : "Flex Knit"
author          : "Han Oostdijk"
date            : "file created : r format(Sys.time(), '%d%b%Y')"
output          :
  pdf_document  :
    keep_tex    : yes
  # html_document :
  # keep_md      : yes
classoption     : portrait # or landscape
# do not place blanks between geometry and ':' or the option will be ignored !
geometry: [
  left=0.6in ,
  right=0.5in ,
  top=1in ,
  bottom=1in
```

```

]
# comment following lines if you do not want to use sans serif monotype
header-includes : [
  '\usepackage[scaled]{DejaVuSansMono}' ,
  '\renewcommand*{\familydefault}{\sfdefault}'
]
urlcolor          : blue
hoqc_version      : _v1
# hoqc_filenames can have a path
hoqc_output       : './output/Flex Knit'
hoqc_yaml         : './output/yaml'
hoqc_yaml_new     : './output/yaml_new'
hoqc_copy         : './output/flexknit.Rmd'
hoqc_force_ext    : yes
knit              : (function (...) { source('myknit.R'); myknit(...) })
# do not place blanks between params and ':' !
params:
  parml           : myfirstparm
---

```

and also the modified yaml (in D:/data/R/rmd_pdf_examples/output/yaml_new_v1.txt) :

```
cat(paste0(readLines(params$hoqc_yaml_new),collapse = '\n'))
```

```

---
title              : "Flex Knit"
author             : "Han Oostdijk"
date               : "file created : r format(Sys.time(), '%d%b%Y')"
output             :
  pdf_document     :
  keep_tex         : yes
classoption        : portrait # or landscape
geometry: [
  left=0.6in ,
  right=0.5in ,
  top=1in ,
  bottom=1in
]
header-includes    : [
  '\usepackage[scaled]{DejaVuSansMono}' ,
  '\renewcommand*{\familydefault}{\sfdefault}'
]
urlcolor           : blue
knit               : (function (...) { source('myknit.R'); myknit(...) })
params:
  parml            : myfirstparm
  hoqc_output      : 'D:/data/R/rmd_pdf_examples/output/Flex Knit_v1.pdf'
  hoqc_yaml        : 'D:/data/R/rmd_pdf_examples/output/yaml_v1.txt'
  hoqc_yaml_new    : 'D:/data/R/rmd_pdf_examples/output/yaml_new_v1.txt'
  hoqc_force_ext   : 'yes'
  hoqc_version     : '_v1'
  hoqc_copy        : 'D:/data/R/rmd_pdf_examples/output/flexknit_v1.Rmd'
---

```

Session Info

R version 3.4.1 (2017-06-30)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 16299)

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

other attached packages:

[1] stringr_1.2.0 magrittr_1.5 fs_1.1.0

loaded via a namespace (and not attached):

[1] compiler_3.4.1 backports_1.1.2 rprojroot_1.3-2 htmltools_0.3.6
[5] tools_3.4.1 yaml_2.1.16 Rcpp_0.12.14 rmarkdown_1.8.10
[9] stringi_1.1.6 knitr_1.19 digest_0.6.14 evaluate_0.10.1