

Put Rmd Code in Appendix

Steve Dutky

9/26/2022

The goal: produce uncluttered knitted document.

I set the the chunk options to `echo=FALSE`. This keeps the code out of the output and, IMHO, produces a cleaner report.

I hoped that placing the code in an appendix at the end of the document, a viewer can easily copy it to their own Rmd and manipulate it as they see fit.

The problem: knitr does not cooperate

I try to do this by putting a chunk like `catRmd` below, but knitting prepends each line with “`##`”, which defeats being able to simply copy the code and truncates each line longer than some threshold, which every line here exceeds.

On the other hand, running the chunk directly from `rstudio` produces the output I had hoped to see.

The hope: can this be fixed?

Is there some way to go about this that avoids these problems? That would go a long way toward making my life so much easier.

The environment

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] forcats_0.5.2  stringr_1.4.1  dplyr_1.0.10  purrr_0.3.4
## [5] readr_2.1.2    tidyr_1.2.1    tibble_3.1.8  ggplot2_3.3.6
```

```
## [9] tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.1.2      xfun_0.32          haven_2.5.1
## [4] gargle_1.2.1          colorspace_2.0-3  vctrs_0.4.1
## [7] generics_0.1.3       htmltools_0.5.3   yaml_2.3.5
## [10] utf8_1.2.2            rlang_1.0.5        pillar_1.8.1
## [13] withr_2.5.0           glue_1.6.2         DBI_1.1.3
## [16] dbplyr_2.2.1          modelr_0.1.9       readxl_1.4.1
## [19] lifecycle_1.0.2      munsell_0.5.0      gtable_0.3.1
## [22] cellranger_1.1.0     rvest_1.0.3        evaluate_0.16
## [25] knitr_1.40            tzdb_0.3.0         fastmap_1.1.0
## [28] fansi_1.0.3           broom_1.0.1        scales_1.2.1
## [31] backports_1.4.1      googlesheets4_1.0.1 jsonlite_1.8.0
## [34] fs_1.5.2              hms_1.1.2          digest_0.6.29
## [37] stringi_1.7.8         grid_4.2.1         cli_3.4.0
## [40] tools_4.2.1           magrittr_2.0.3     crayon_1.5.1
## [43] pkgconfig_2.0.3      ellipsis_0.3.2     xml2_1.3.3
## [46] reprex_2.0.2          googledrive_2.0.0 lubridate_1.8.0
## [49] assertthat_0.2.1     rmarkdown_2.16     httr_1.4.4
## [52] rstudioapi_0.14      R6_2.5.1           compiler_4.2.1
```

Appendix: Rmarkdown Code

```
## ---
## title: "Put Rmd Code in Appendix"
## author: "Steve Dutky"
## date: "9/26/2022"
## output:
##   pdf_document: default
##   html_document: default
## ---
##
## ‘‘{r setup, include=FALSE}
## knitr::opts_chunk$set(echo = FALSE)
## ‘‘
##
## ‘‘{r init, include=FALSE}
## library(tidyverse)
## ‘‘
##
## ### The goal: produce uncluttered knitted document.
## I set the the chunk options to echo=FALSE. This keeps the code out of the output and, IMHO, produces
##
## I hoped that placing the code in an appendix at the end of the document, a viewer can easily copy it
##
## ### The problem: knitr does not cooperate
## I try to do this by putting a chunk like catRmd below, but knitting prepends each line with "##", which
##
## On the other hand, running the chunk directly from rstudio produces the output I had hoped to see.
##
## ### The hope: can this be fixed?
## Is there some way to go about this that avoids these problems? That would go a long way toward making
```

```
##
## ### The environment
##
## ```{r sessionInfo}
## sessionInfo()
## ```
##
## ### Appendix: Rmarkdown Code
## ```{r catRmd,asis=TRUE}
## rmd<-read_lines("catRmd.Rmd")
## cat(paste(rmd,"\n"))
## ```
```