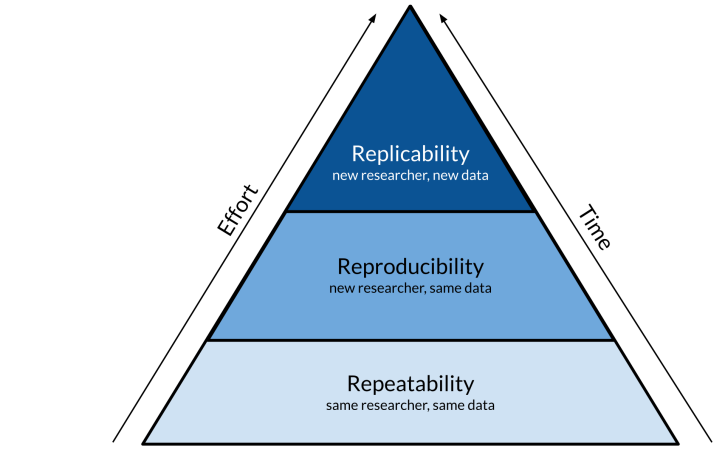
Reproducibility

Reproducibility

What's reproducibility?

A different analyst re-performs the analysis with the same code and the same data and obtains the same result.

Reproducibility vs Repeatability vs Replicability



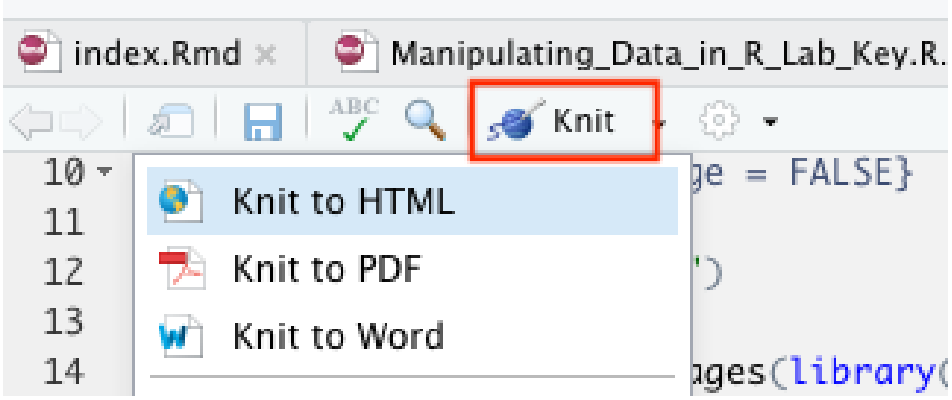
Based off of a figure from Essawy et al, 2020 https://doi.org/10.1016/j.envsoft.2020.104753

Reproducibility

- Clean your environment regularly
- Use RMarkdown
- Check the knit of your RMarkdown regularly
- Tell your future self and others what you did!
- Print session info!

RMarkdown

Clicking the knit button, will knit your document to create different types of reports. The default is html.



Code Chunks

```{r, echo = FALSE}
x<-2
```

```{r, eval = FALSE}
x<-2
```

```{r, message = FALSE}
x<-2
```

After knitting:
1) **echo** - determines if your code should be shown or not
- TRUE = code is **shown** (default)
- FALSE = code is not shown (but might be run…depends on eval)
2) **eval** - determines if your code should be evaluated (run) or not
- TRUE = code is **run** (default)
- FALSE = code is not run (but might be shown…depends on echo)
3) **message** = FALSE -suppresses messages when your run your code

RMarkdown syntax

Before:

```
# Header - biggest font created by hashtag and space
## SubHeader Second Biggest created by 2 hashtags and space

**bold** text
*italicized* text

`code` referenced outside of a chunk needs backticks
```

After knit:

Header - biggest font created by hashtag and space

SubHeader Second Biggest created by 2 hashtags and space

bold text *italicized* text

`code` referenced outside of a chunk needs backticks

RMarkdown syntax

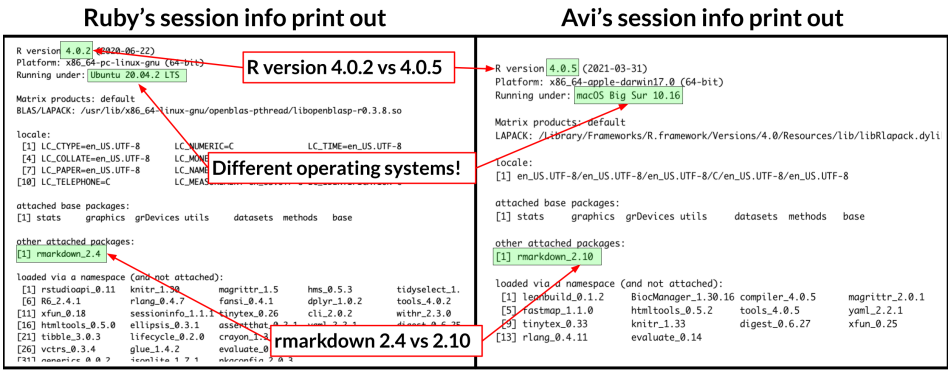
Go to File > Help > Cheatsheets > R Markdown Cheatsheet

Final Project

Specific guidelines and example on website

Turn in through CoursePlus:
1) RMarkdown file
2) html file

Session Info



If Avi and Ruby have discrepancies in their results, the session info print out gives a record which may have clues to why that might be!

Image by Candace Savonen

Session info helps

```
sessionInfo()
```

```
R version 4.2.0 (2022-04-22)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.4 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3

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] emo_0.0.0.9000         patchwork_1.1.1        here_1.0.1
 [4] plotly_4.10.0          directlabels_2021.1.13 scales_1.2.0
 [7] knitr_1.39             esquisse_1.1.1         readxl_1.4.0
[10] naniar_0.6.1           broom_0.8.0            jhur_0.2.1
[13] lubridate_1.8.0        forcats_0.5.1          stringr_1.4.0
[16] dplyr_1.0.9            purrr_0.3.4            tidyr_1.2.0
[19] tibble_3.1.7           ggplot2_3.3.6          tidyverse_1.3.1
[22] readr_2.1.2

loaded via a namespace (and not attached):
 [1] fs_1.5.2            bit64_4.0.5         httr_1.4.3
 [4] rprojroot_2.0.3     phosphoricons_0.1.2 tools_4.2.0
 [7] backports_1.4.1     bslib_0.3.1         utf8_1.2.2
[10] R6_2.5.1            lazyeval_0.2.2      DBI_1.1.2
[13] colorspace_2.0-3    withr_2.5.0         tidyselect_1.1.2
[16] bit_4.0.4           curl_4.3.2          compiler_4.2.0
[19] cli_3.3.0           rvest_1.0.2         xml2_1.3.3
[22] labeling_0.4.2      sass_0.4.1          quadprog_1.5-8
[25] digest_0.6.29       foreign_0.8-82      rmarkdown_2.14
[28] rio_0.5.29          pkgconfig_2.0.3     htmltools_0.5.2
[31] dbplyr_2.1.1        fastmap_1.1.0       highr_0.9
[34] htmlwidgets_1.5.4   rlang_1.0.2         rstudioapi_0.13
[37] shiny_1.7.1         jquerylib_0.1.4     farver_2.1.0
[40] generics_0.1.2      jsonlite_1.8.0      crosstalk_1.2.0
[43] vroom_1.5.7         zip_2.2.0           magrittr_2.0.3
[46] Rcpp_1.0.8.3        munsell_0.5.0       fansi_1.0.3
[49] lifecycle_1.0.1     visdat_0.5.3        stringi_1.7.6
[52] yaml_2.3.5          grid_4.2.0          parallel_4.2.0
[55] datamods_1.3.2      promises_1.2.0.1    crayon_1.5.1
[58] haven_2.5.0         hms_1.1.1           pillar_1.7.0
[61] clisymbols_1.2.0    reprex_2.0.1        glue_1.6.2
[64] evaluate_0.15       data.table_1.14.2   modelr_0.1.8
[67] vctrs_0.4.1         tzdb_0.3.0          httpuv_1.6.5
[70] cellranger_1.1.0    gtable_0.3.0        assertthat_0.2.1
[73] xfun_0.31           openxlsx_4.2.5      mime_0.12
[76] xtable_1.8-4        later_1.3.0         viridisLite_0.4.0
[79] shinyWidgets_0.7.0  ellipsis_0.3.2
```

DRY code

DRY (https://web.archive.org/web/20131204221336/http://programmer.97things.oreilly.com/wiki/index.php/Don't_Repeat_Yourself) is an acronym: “Don’t repeat yourself” [@Smith2013].

> “I hate code, and I want as little of it as possible in our product.” - Jack Diedrich

More resources

These are just some quick tips, for more information:

- The RMarkdown book (https://bookdown.org/yihui/rmarkdown/)
- Jenny Bryan’s organizational strategies (https://www.stat.ubc.ca/~jenny/STAT545A/block19_codeFormattingOrganization.html).
- Write efficient R code for science (https://www.earthdatascience.org/courses/earth-analytics/automate-science-workflows/write-efficient-code-for-science-r/).
- Reproducibility in Cancer Informatics course (https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/introduction.html)