# Module 3 Pre-Read - Markdown

Saturday, February 3, 2024 11:45

https://www.markdownguide.org/cheat-sheet/

https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes
https://www.overleaf.com/learn/latex/Mathematical_expressions
https://bookdown.org/yihui/blogdown/html.html
https://bookdown.org/yihui/blogdown/css.html

## Anatomy of an R Markdown document

R Markdown documents consist of three components:

1. `yaml` header
   - specifies meta options for the document
   - Title, authors, output format, etc...
2. Plain text written in **markdown**
   - plain writing + section headers + tables + ...
3. `R` code
   - Inline with writing using single tick
   - Whole "chunk" of code using three ticks

YAML = Yet Another Markup Language

```
---
title: VRC01 neutralization report
author: David Benkeser
output: html_document
---
# Exploring markdown
Markdown is a way of writing plain text that, upon rendering, looks formatted.
For example, we can __bold text__ or *italicize* it.
We can also ~~strikethrough text~~.
Sometimes `monospace font` is nice for formatting as well.
## Subheadings are fun too
### Even smaller subheadings are also possible
We can create bullet points:
* that is pretty helpful
* for listing things
We can create numbered lists:
1. item number 1
2. item number 2
We can also [include hyperlinks](www.emory.edu).


# Including R code
There are two options for inputting code in R Markdown.


## Code chunks

```{r, name-of-chunk}
print("hello from R!")
```
```{r, read-data}
# replace with your own
abs_file_path <- "~/Dropbox/Emory/Teaching/DSTK/intro_repro_workflo/lectures/04_rmarkdown/"
full_file_path <- paste0(
  abs_file_path, "hiv_project/data/vrc01_data.csv"
)
data <- read.csv(full_file_path, header = TRUE)
head(data)
```
## Inline R expressions
We can use inline `R` expressions to report numbers in the text. For example, `vrc01_data.csv` has `r nrow(data)` rows and `r ncol(data)` columns.
```

YAML header is contained within the --- dashes
Render type of document via `output`

Monospace font basically makes the writing look like code form

From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_rmarkdown/hiv_project/tmp_reports/01_example_report.Rmd>

A single backtick will run r code in the single line

More advanced YAML headers

```
date: " `r format(Sys.Date(), '%m-%d-%Y') ` "

output:
    highlight: is how the code displays in the compiled r markdown (I like zenburn)
    toc: table of contents, can be true or false
```

## Output formats

There are many output formats. Here are a few useful ones (with links to further documentation):

| output | Description |
|---|---|
| `html_document` | html template with many preset themes |
| `pdf_document` | pdf using LaTex |
| `word_document` | MS Word document (can create custom theme) |
| `ioslides_presentation` | HTML5 presentation slides |
| `beamer_presentation` | pdf presentation slides with beamer |
| `powerpoint_presentation` | MS Powerpoint presentation |

## yaml header

R Markdown documents begin with a **yaml header**.

- "meta-data" for the document
- options that control **how the document is rendered**

Options can be nested.

- don't forget your `:`'s!
- two spaces, **no tabs**!

Options can include inline `R` code.

## Code chunk options

Each code chunk gets a label.

- avoid spaces, underscores, periods; hyphens are safe

Each chunk has **options**. Options can be included by either:

- comma separated list between
  - `{r, chunk-label, option1 = foo, option2 = bar }`
- special demarcated comment
  - `{r, chunk-label}`
  - `#| option1 = foo,`
  - `#| option2 = bar`
- or as yaml
  - `{r, chunk-label}`
  - `#| option1: foo`
  - `#| option2: bar`

## Code chunk options

Common options for **controlling display** of results.

- For a full list see here.

| Option | Action |
|---|---|
| `eval` | Run the code included in the chunk? |
| `echo` | Show the code chunk in the rendered document? |
| `warning` | Print warning messages generated by code? |
| `error` | Print and keep running after errors? |
| `message` | Print messages generated by code? |
| `include` | Show the code and results in the rendered document? |

Including raw latex and html in document
We can create bullet points:
* that is pretty helpful
* for listing things
We can also use LaTeX to do this: \begin{itemize}
 \item LaTeX is pretty helpful
 \item for listing things too!
 \item This only works if rendering to pdf
\end{itemize}
We can create numbered lists:
1. item number 1
2. item number 2
We can also use html to do this:
<ol>
 <li>item number 1 (in html)</li>
 <li>item number 2 (in html)</li>
 <li>this works both in pdf and html</li>
</ol>
From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_

HTML for html docs
LATEX for pdf docs

_rmarkdown/hiv_project/tmp_reports/03_example_report.Rmd>

## knitr and pandoc

- `rmarkdown`
  - user-friendly wrapper around `knitr` and `pandoc`
- `knitr`
  - an `R` package that executes code chunks
  - "knits" results back to document
- `pandoc`
  - a general purpose document converter
  - command line tool, no GUI
  - install directly or included in R Studio IDE installation

Knitr runs r code in a fresh clean session

Pandoc converts documents between different types