Higher, further, faster with Marvelous R Markdown

@thomas_mock

The Heroine's Journey (sans spoilers)

- Act 1: Story is setup (Here's the tool)
- Act 2: Complication arises (Here's the problem)
- Act 3: Heroine finds resolution (Here's a solution)

RMarkdown Taxonomy

Literate Programming

Data Products

Control Documents

Templating

Literate Programming

Goal: Capture code, text/comments, and output in a single document

Literate Programming

> A programming paradig introduced by Donald Knuth in which a computer program is given an explanation of its logic in a **natural language**, such as English, interspersed with snippets of macros and traditional **source code**, from which compilable source code can be generated.

Wikipedia/Literate_programming

```
---
title: "Penguins"
data: 2020-08-11
output: html_document
---

```{r setup, include=FALSE}

library(ggplot2)
library(dplyr)
library(palmerpenguins)

smaller <- penguins %>%
  filter(body_mass_g <= 4000)
```

The Adelie penguins are smaller than the other type
of penguins. The plot generated above indicated a
distribution of penguins weighing less than 4 kilog

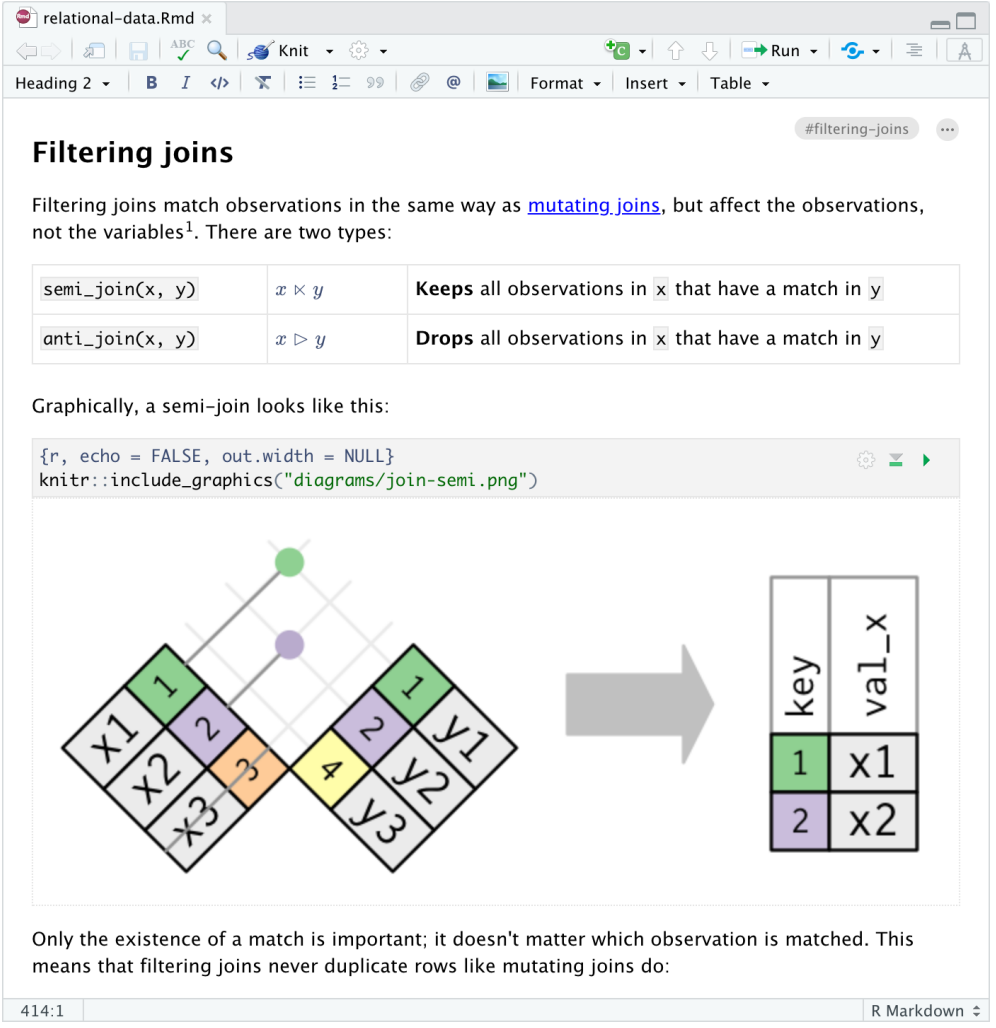
```{r echo=FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

Visual RMarkdown

Live as of RStudio 1.4 (released in Jan-21)!!!

- Visual editing for all of Pandoc markdown
- Extensive support for citations
- Scientific and technical writing features, including LaTeX
- Writing productivity w/ real time spell-checking
- Tight integration with source editing
- Rich keyboard support and can use the ⌘ / shortcut to insert just about *anything*

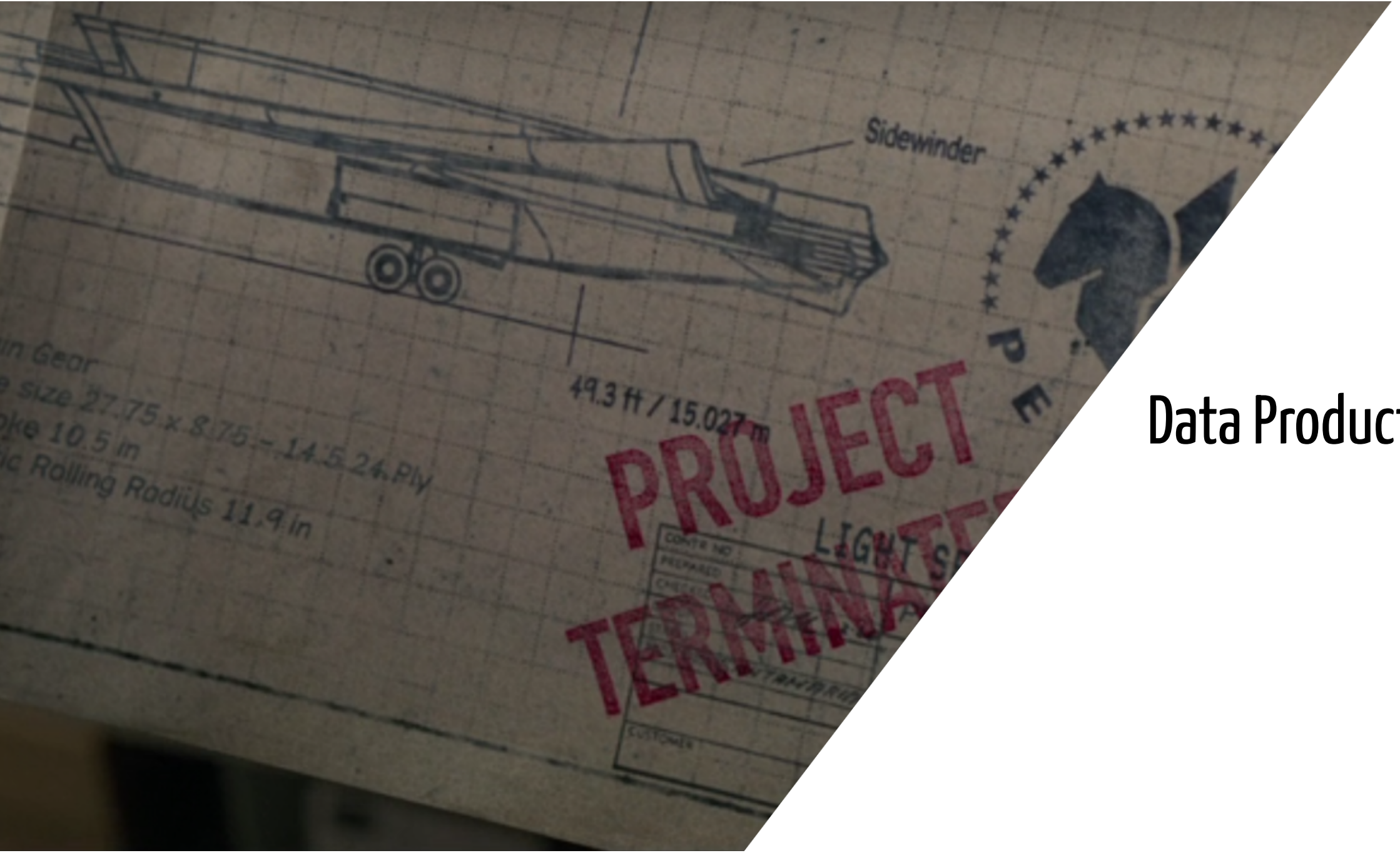
Visual Editor Guide



Not *just* for R

- Python via `{reticulate}`
- SQL
- CSS or Javascript
- Bash, Rcpp, Stan, and others

All together a total **52** possible language engines from `{knitr}`.



Data Product

Goal: Use R to generate final output for consumption

Data Products

- Reports - HTML, PDF, RTF, Github document
- Presentations - `xaringan`, Powerpoint, `reveal.js`, beamer
- Dashboards - `flexdashboard` either as static or with `shiny`
- Entire websites - `blogdown`, `distill`
- Books via `bookdown`
- HTMLWidgets - `DT`, `reactable`, `plotly`, `crosstalk` and more!

Control Document

Goal: Modularize data science tasks, use RMarkdown to control code flow

Knit with Parameters

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
params:
  species: Adelie
---
```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)
smaller <- penguins %>%
  filter(species == params$species,
         !is.na(body_mass_g))
```
We have data about `r nrow(penguins)` penguins. Only
`r nrow(penguins) - nrow(smaller)` are classified as
`r params$species`. The distribution of the
`r params$species` penguins are shown below:
```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

Reference .R files

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
---
```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)
knitr::read_chunk(‘penguins.R’)
```

```{r, smaller-penguins, echo = FALSE}
```

We have data about `r nrow(penguins)`
penguins.

```{r, plot-penguins, echo = FALSE}
```
```

penguins.R

```
## ---- smaller-penguins
smaller <- penguins %>%
  filter(species == "Adelie”,
         !is.na(body_mass_g))
## ---- plot-penguins
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```

Child Documents

adelie-report.Rmd

```
---
output: html_document
---
```{r, echo = FALSE}
smaller <- penguins %>%
  filter(species == "Adelie",
         !is.na(body_mass_g))
```
We have data on `r nrow(penguins)` penguins.
The distribution of the Adelie
penguins are shown below:

```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

Parent Document

Uses the child document by name.

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
---


```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)
```

```{r, child=c("adelie-report.Rmd”)}
```
```

Logical Child documents

report.Rmd

```
---
output: html_document
---

```{r, echo = FALSE}
smaller <- penguins %>%
  filter(species == "Adelie",
         !is.na(body_mass_g))
```
We have data on `r nrow(penguins)` penguins.
The distribution of the Adelie
penguins are shown below:

```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
---


```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)

sp <- "Chinstrap"
```

```{r, child=if (sp == "Adelie”) "report.Rmd”
```
```

Uses report code only if specific condition is met.

Blastula Emails

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
---

```{r setup, include = FALSE}
library(tidyverse)
library(blastula)
```
```{r penguin-plot, echo = FALSE}
palmerpenguins::penguins %>%
  filter(species == "Adelie",
         !is.na(body_mass_g)) %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```

```{r connect_email, echo = FALSE}
render_connect_email(
  input = "penguin-email.Rmd”) %>%
  attach_connect_email(
    subject = "Penguin Report")
```
```

`penguin-email.Rmd`

```
---
title: "Penguins Report"
date: 2020-08-11
output: blastula::blastula_email
---


```{r packages, include = FALSE}
library(tidyverse)
library(palmerpenguins)
```


Important update on the state of
the penguins!

```{r penguin-plot, echo = FALSE}
```
```

When parent document is rendered, it generates a HTML email, can include arbitrary R output like `ggplot2` or even tables.

Templating

Goal: Don't repeat yourself, generate input templates or output documents from code

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
params:
  species: Adelie
---

```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)

smaller <- penguins %>%
  filter(species == params$species,
         !is.na(body_mass_g))
```
We have data about `r nrow(penguins)` penguin
The distribution of the
`r params$species` penguins are shown below:

```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

```
rmarkdown::render(
  “penguin.rmd”,
  params = list(species = “Gentoo”)
)
```

Uses report code with new parameter.

```
---
title: "Penguins"
date: 2020-08-11
output: html_document
params:
  species: Adelie
---

```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)

smaller <- penguins %>%
  filter(species == params$species,
         !is.na(body_mass_g))
```
We have data about `r nrow(penguins)` penguin
The distribution of the
`r params$species` penguins are shown below:

```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
```

```
render_fun <- function(penguin){
  rmarkdown::render(
    input = "penguins-report.rmd",
    params = list(species = penguin),
    output_file = glue::glue(
      "{penguin}-report.html"
      )
  )
}

distinct(penguins, as.character(species)) %>%
  pull() %>%
  purrr::walk(render_fun)
```

```
Adelie-report.html
Chinstrap-report.html
Gentoo-report.html
```

Loop within a doc

```
---
title: "Penguin Report"
output: html_document
---

```{r penguin function, include=FALSE}
library(tidyverse)
library(palmerpenguins)
library(glue)
knitr::opts_chunk$set(echo=FALSE, fig.width=6
  message=FALSE)
source("multiplot.R")
penguins <- palmerpenguins::penguins %>%
  filter(!is.na(bill_length_mm),
   !is.na(flipper_length_mm))
```

```{r loop-output, results="asis"}
penguins %>%
  distinct(species) %>%
  pull(species) %>%
  as.character() %>%
  walk(multiplot)
```
```

multiplot.R

```
multiplot <- function(penguin_name){
  glue("  \n### {penguin_name}  \n  \n") %>%

  df_pen <- penguins %>%
    filter(as.character(species) == penguin_n

  flipper_len <- df_pen %>%
    summarize(mean = mean(flipper_length_mm))
    pull(mean) %>%
    round(digits = 1)

  glue::glue("There are {nrow(df_pen)} observ

  plot_out <- df_pen %>%
    ggplot(aes(x = bill_length_mm,
               y = flipper_length_mm)) +
    geom_point() +
    labs(x = "Bill Length",
         y = "Flipper Length",
         title = penguin_name)

  print(plot_out)

  cat("  \n  \n")
}
```

whisker vs glue...

... is {one} vs {{two}}

```
# logic templating
glue::glue("There are {nrow(mtcars)} rows in the mtcars dataset")
```

```
## There are 32 rows in the mtcars dataset
```

```
rows_in <- nrow(mtcars)
whisker::whisker.render('There are {{rows_in}} rows in the mtcars dataset')
```

```
## [1] "There are 32 rows in the mtcars dataset"
```

```
# logicless templating
whisker::whisker.render('There are {{nrow(mtcars)}} rows in the mtcars dataset')
```

```
## [1] "There are  rows in the mtcars dataset"
```

```
---
title: "{{species}} Penguin"
output: html_document
---
```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)

smaller <- penguins %>%
  filter(species == “{{species}}”,
         !is.na(body_mass_g))
```
We have data about `r nrow(penguins)` penguin
The distribution of the {{species}} penguins

```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
{{long_prose}}
```

```
---
title: "Adelie Penguin"
output: html_document
---
```{r setup, include = FALSE}
library(tidyverse)
library(palmerpenguins)

smaller <- penguins %>%
  filter(species == Adelie, !is.na(body_mass_
```
We have data about `r nrow(penguins)` penguin
The distribution of the Adelie penguins are s
```{r, echo = FALSE}
smaller %>%
  ggplot(aes(body_mass_g)) +
  geom_histogram(binwidth = 100)
```
Mating
The mating season begins with the Antarctic s
The penguins create nests by piling little st
Once the egg is laid in December the parents
The parent that stays behind does not eat dur
Once the hatched chick is about 3 weeks old b
The downy chicks gather into a group called a
They will start to hunt at about 9 weeks old
```

`whisker` function

```r
use_penguin_template <- function(species, long_prose){

  raw_rmd <- readLines("penguin-whisker.Rmd")

  filled_rmd <- whisker::whisker.render(raw_rmd)

  writeLines(
    text = filled_rmd,
    con = glue::glue("{species}-report.rmd")
  )
}
```

```
species <- "Adelie"

long_prose <- "### Mating\n\nThe mating season begins with the Antarctic spring in October.
The penguins create nests by piling little stones in circles.
Once the egg is laid in December the parents take turns incubating the egg and going to hunt.
The parent that stays behind does not eat during their turn with the egg.
Once the hatched chick is about 3 weeks old both parents will abandon it, returning to the sea to hunt.
The downy chicks gather into a group called a crèche to keep each other warm.
They will start to hunt at about 9 weeks old once their down has been replaced with waterproof feathers."
```

```r
use_penguin_template(species, long_prose)
```

RMarkdown + RStudio Connect

RStudio Connect is a hosting and execution platform for Shiny, RMarkdown, Plumber (also Jupyter, Flask, Dash, & Streamlit)

Can execute/schedule RMarkdown for all sorts of things like:

- Self-service parameterized RMarkdown for non-tech users
- Extract Transform Load from SQL, APIs or Spark for example
- Automated reports with logging/history
- Long model training steps and save model upon completion
- Send blastula emails conditionally or on a schedule

docs.rstudio.com/connect/user/rmarkdown/

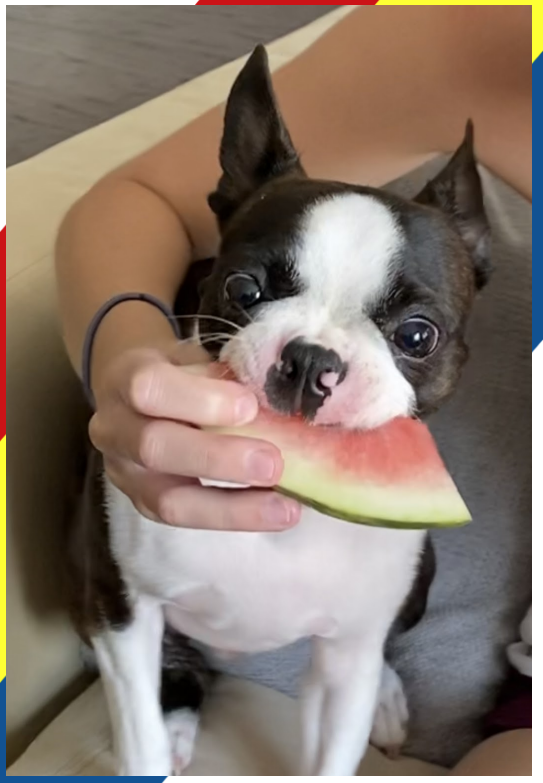
You have the power, now use it!

]

Follow-up reading

- Emily Riederer's RMD Driven Development
- Sharla Gelfand's RMD templating
- My Meta RMarkdown blogpost
- The fantastic RMarkdown Cookbook
- RMarkdown Definitive Guide book
- Rethinking Reporting with Automation within the Insurance Industry
- Using R to Enhance Clinical Reporting within the Life Sciences industry
- One of my favorites, Avoid Dashboard Fatigue

BONUS HOWARD SLIDE



LOVES WATERMELON