

Higher, further, faster with Marvelous R Markdown

@thomas_mock

bit.ly/marvelRMD



The Heroine's Journey (sans spoilers)

Act 1: Story is set up
(Here's the tool)

Act 2: Complication arises
(Here's the problem)

Act 3: Heroine finds resolution
(Here's a solution)



Slides at bit.ly/marvelRMD

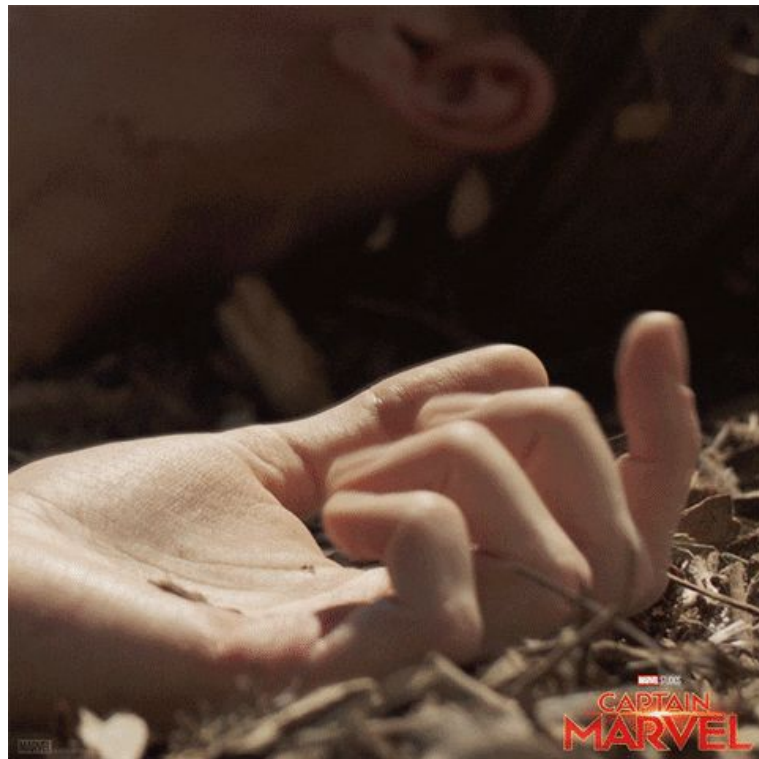
But coding is not a linear journey

Less about :

Beginner < Intermediate < Expert

More about:

Expanding your toolset for the right job
at the right time



Slides at bit.ly/marvelRMD

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 paradigm 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.

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

YAML

```
```{r setup, include = FALSE}  
library(ggplot2)
library(dplyr)
library(palmerpenguins)

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

Code

Text

We have data about ``r nrow(penguins)`` penguins. Only ``r nrow(penguins) - nrow(smaller)`` are larger than 4000 grams. The distribution of the remainder is shown below:

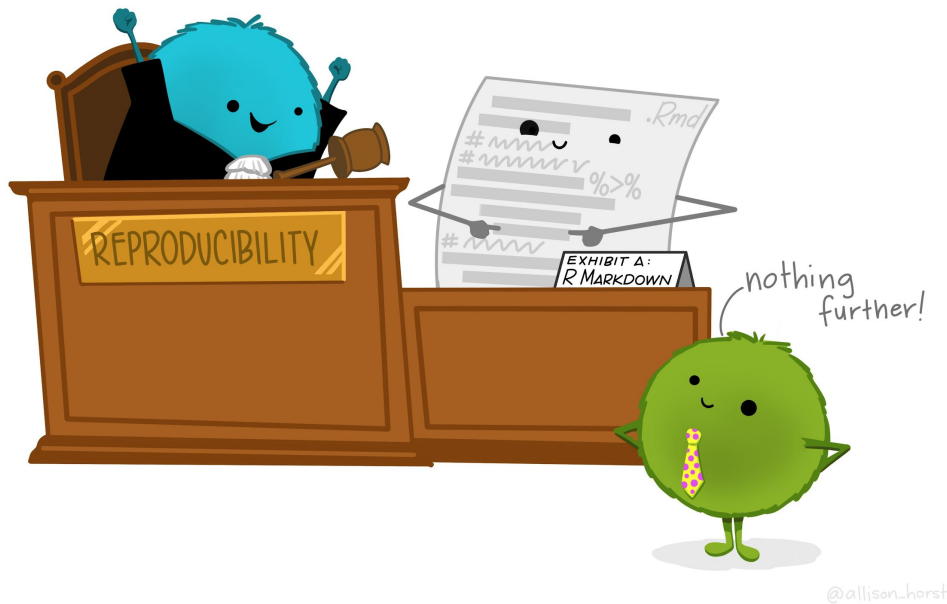
```
```{r, echo = FALSE}  
smaller %>%
 ggplot(aes(body_mass_g)) +
 geom_histogram(binwidth = 100)
```
```

Code

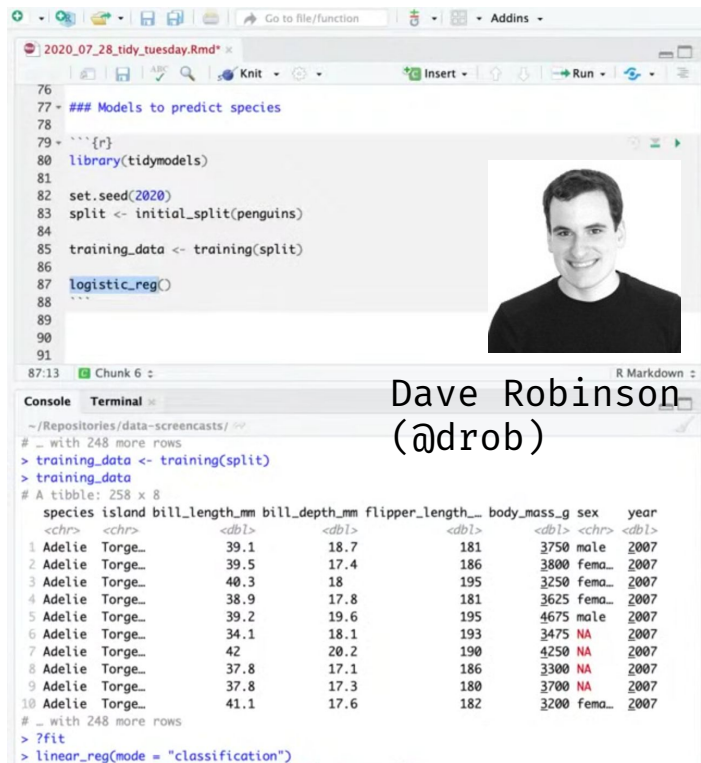
M.V.P of Reproducibility

R Markdown

- Output is self-documenting (code is embedded)
- Has to run successfully to save/knit output
- Is diffable *AND* human readable in version control



TidyTuesday screencasts exemplars



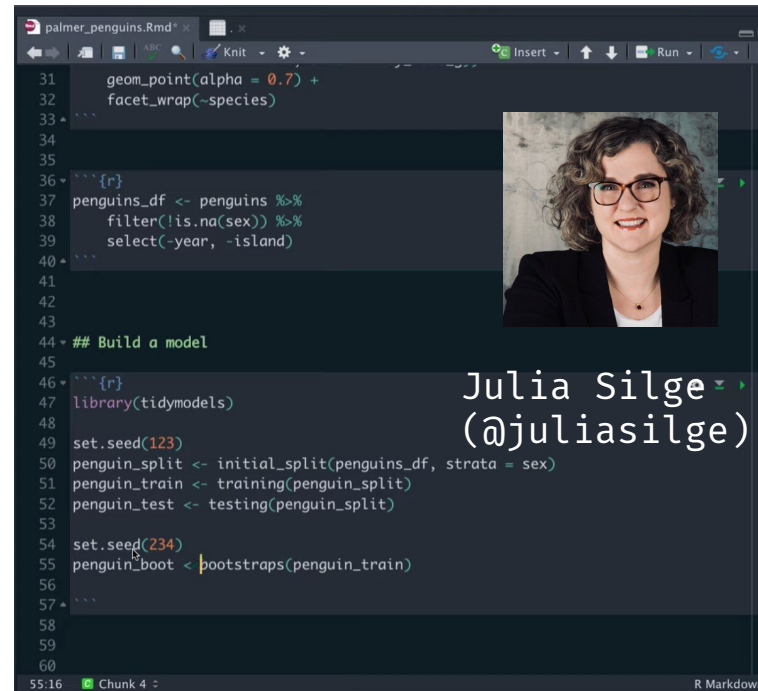
```
## Models to predict species
library(tidymodels)
set.seed(2020)
split <- initial_split(penguins)
training_data <- training(split)
logistic_reg()
```

87:13 Chunk 6

Console

```
~/Repositories/data-screencasts/
# ... with 248 more rows
> training_data <- training(split)
> training_data
# A tibble: 258 x 8
  species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex year
  <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>
1 Adelie Torge... 39.1 18.7 181 3750 male 2007
2 Adelie Torge... 39.5 17.4 186 3800 fema... 2007
3 Adelie Torge... 40.3 18 195 3250 fema... 2007
4 Adelie Torge... 38.9 17.8 181 3625 fema... 2007
5 Adelie Torge... 39.2 19.6 195 4675 male 2007
6 Adelie Torge... 34.1 18.1 193 3475 NA 2007
7 Adelie Torge... 42 20.2 190 4250 NA 2007
8 Adelie Torge... 37.8 17.1 186 3300 NA 2007
9 Adelie Torge... 37.8 17.3 180 3700 NA 2007
10 Adelie Torge... 41.1 17.6 182 3200 fema... 2007
# ... with 248 more rows
> ?fit
> linear_reg(mode = "classification")
```

Dave Robinson
(@drob)



```
geom_point(alpha = 0.7) +
facet_wrap(~species)

library(tidymodels)
set.seed(123)
penguin_split <- initial_split(penguins_df, strata = sex)
penguin_train <- training(penguin_split)
penguin_test <- testing(penguin_split)
set.seed(234)
penguin_boot <- bootstraps(penguin_train)
```

55:16 Chunk 4

Console

```
~/Repositories/data-screencasts/
# ... with 248 more rows
> training_data <- training(split)
> training_data
# A tibble: 258 x 8
  species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex year
  <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>
1 Adelie Torge... 39.1 18.7 181 3750 male 2007
2 Adelie Torge... 39.5 17.4 186 3800 fema... 2007
3 Adelie Torge... 40.3 18 195 3250 fema... 2007
4 Adelie Torge... 38.9 17.8 181 3625 fema... 2007
5 Adelie Torge... 39.2 19.6 195 4675 male 2007
6 Adelie Torge... 34.1 18.1 193 3475 NA 2007
7 Adelie Torge... 42 20.2 190 4250 NA 2007
8 Adelie Torge... 37.8 17.1 186 3300 NA 2007
9 Adelie Torge... 37.8 17.3 180 3700 NA 2007
10 Adelie Torge... 41.1 17.6 182 3200 fema... 2007
# ... with 248 more rows
> ?fit
> linear_reg(mode = "classification")
```

Julia Silge
(@julasilge)

<https://www.youtube.com/watch?v=z57i2GVcdww>

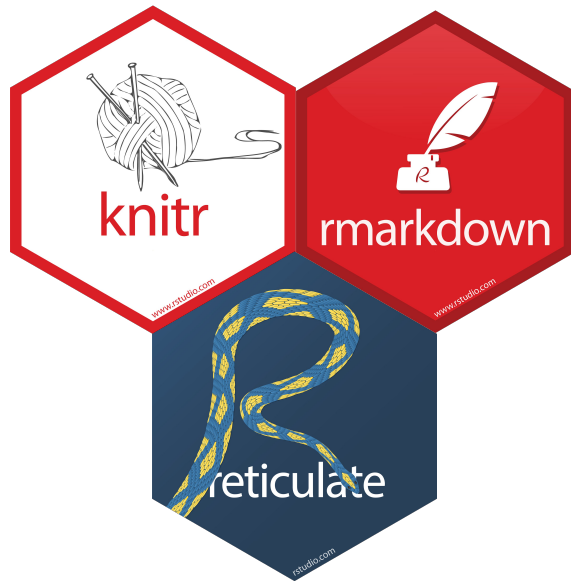
<https://www.youtube.com/watch?v=ImpXawPNCfM>

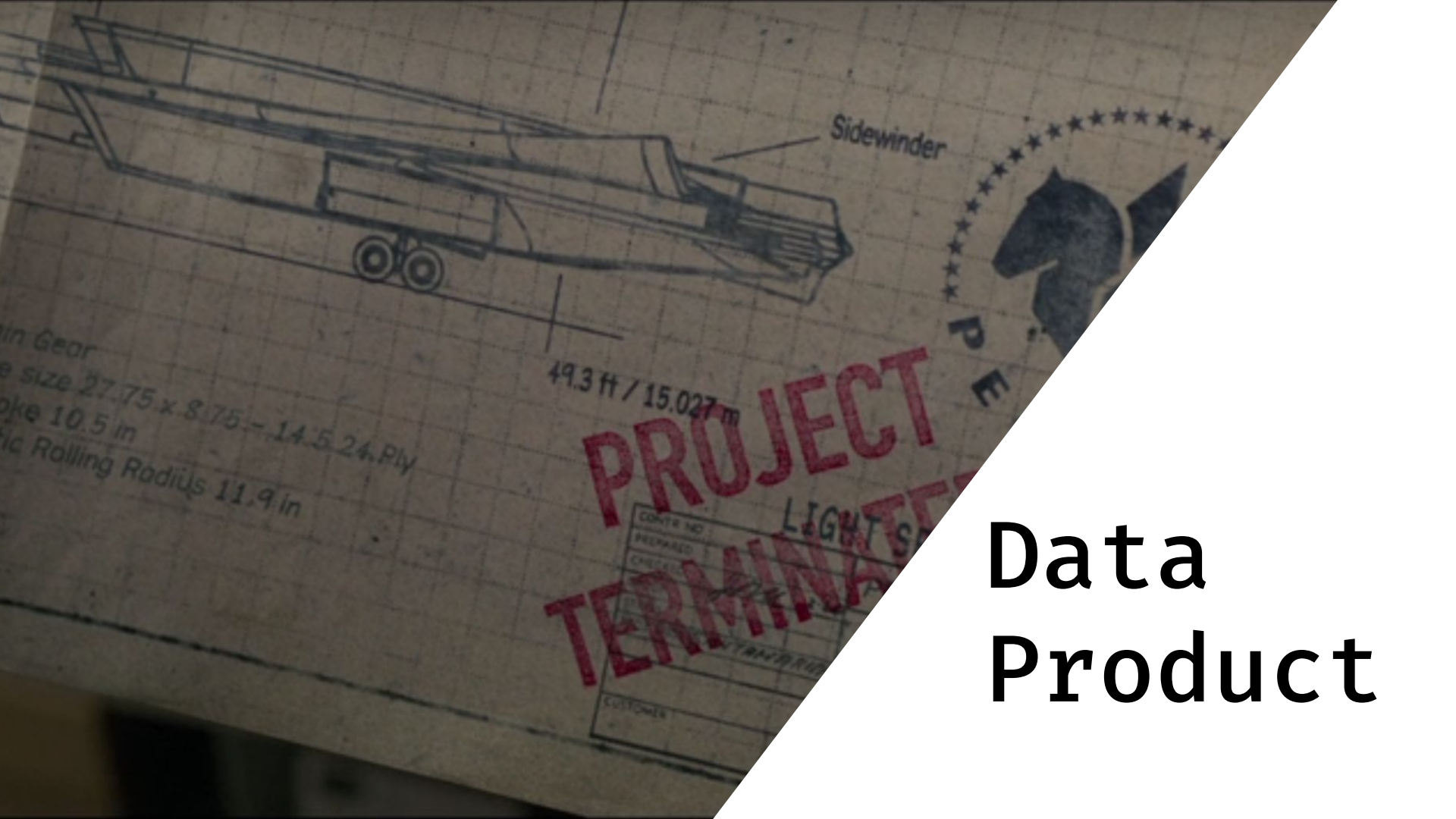
Slides at bit.ly/marvelRMD

Not just for R!

- Python natively or via reticulate
- SQL
- CSS or Javascript
- Bash, Rcpp, Stan and others

All together a total of **52** possible language engines from knitr





Data
Product



Goal: Use R to generate final output
for consumption

Data Product(s)

Reports - HTML, PDF, Word, Markdown, RTF, Tufte handout, GitHub document

Presentations - Xaringan, Powerpoint, ioslides, reveal.js, beamer

Dashboards - flexdashboard as standalone or with Shiny

Entire websites - Blogdown, Hugodown, distill, R Markdown site

Books - bookdown

HTMLWidgets - DT, reactable, plotly, crosstalk and more

Crosstalk

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

```{r setup, include=FALSE}
library(tidyverse)
library(reactable)
library(plotly)
library(crosstalk)
library(htmltools)
```

```{r, echo = FALSE}
small_df <- palmerpenguins::penguins %>%
 select(species, sex, flipper_length_mm, body_mass_g) %>%
 na.omit()
```

```
pen_df <- SharedData$new(small_df)
```

Crosstalk data

```
plotly_graphic <- plot_ly(pen_df,
 x = ~flipper_length_mm,
 y = ~body_mass_g,
 color = ~species,
 text = ~species) %>%
 add_markers() %>%
 layout(xaxis = x, yaxis = y) %>%
 highlight(on = "plotly_selected",
 off = "plotly_doubleclick",
 persistent = FALSE))
```

reactable table

```
reactable_table <- reactable(pen_df,
 columns = list(
 species = colDef("Species"),
 sex = colDef("Sex", cell = function(value){
 if (value == "male") "Male" else "Female"}),
 flipper_length_mm = colDef("Flipper Length"),
 body_mass_g = colDef("Body Mass"))))
...
```

```
```{r, echo = FALSE, warning=FALSE, message=FALSE}
x <- list(title = "Flipper Length (mm)", range = c(170, 240))
y <- list(title = "Body Mass (g)", range = c(2500, 6500))
div(
  h1("Penguin Data"),
  h3("Filter by: Sex, Species, or Body Mass"),
  br(),
  bscols(
    widths = c(2, 10),
    list(
      filter_checkbox("sex", "Sex", pen_df, ~sex),
      filter_checkbox("species", "Species", pen_df, ~species),
      filter_slider("mass", "Body Mass (g)", pen_df, ~body_mass_g)
    ),
    plotly_graphic,
  br(),
  reactable_table
)
```

crosstalk



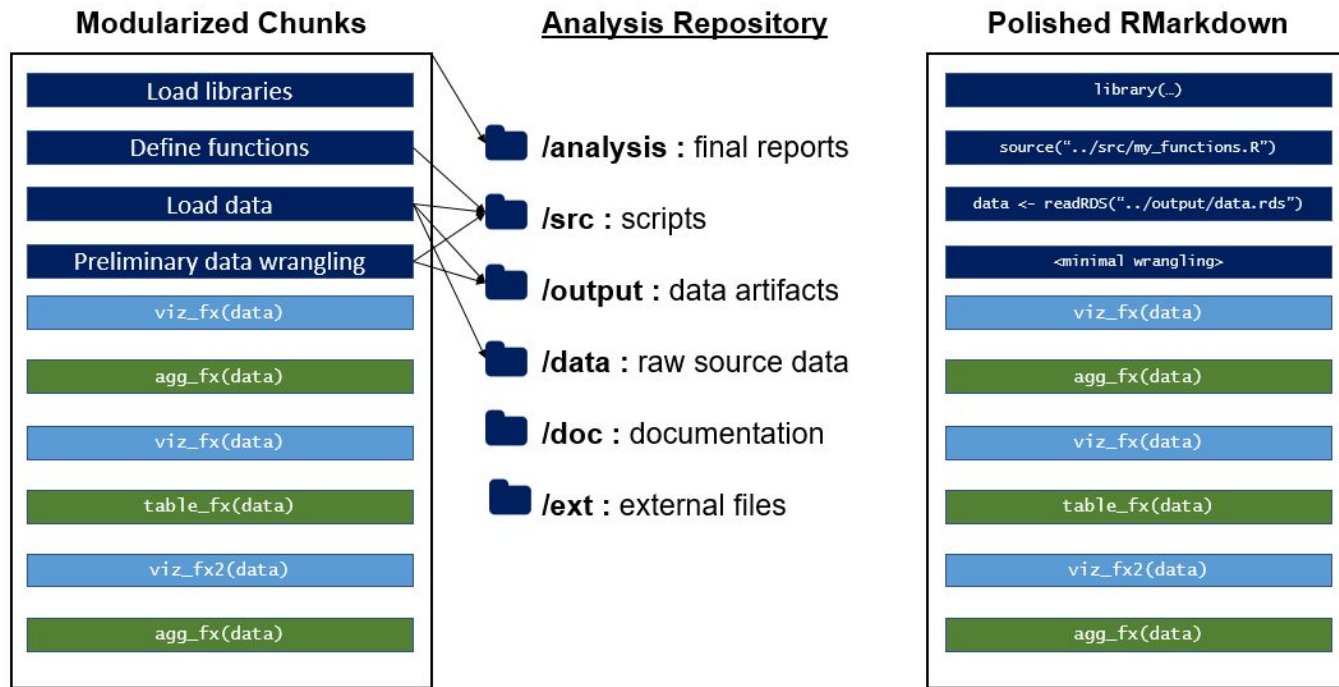
Control Document





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

Meta RMarkdown Driven Development



Knit with Parameters

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

```
params:  
  species: Adeline
```

Create in YAML

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

```
smaller <- penguins %>%
 filter(species == params$species,
 !is.na(body_mass_g))

```

Reference in Code

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:

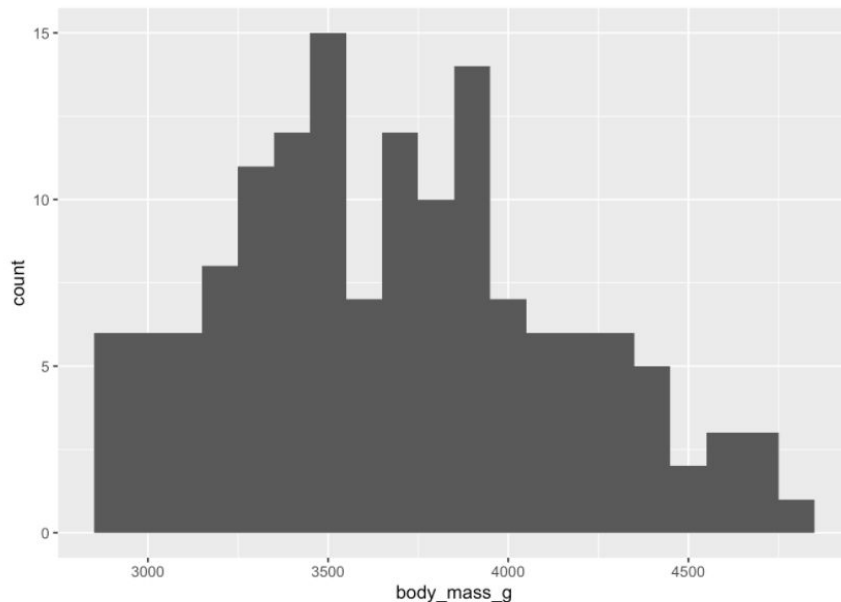
Reference in Line

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

Penguins

2020-08-11

We have data about 344 penguins. Only 198 are classified as Adeline. The distribution of the Adeline penguins are shown below:



Resource Links: [Ex. 1](#), [Ex. 2](#), [Ex. 3](#)

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')
```
```

Read .R code

Reference by name

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

We have data about ``r nrow(penguins)`` penguins. Only ``r nrow(penguins) - nrow(smaller)`` are classified as Adelie. The distribution of the Adelie penguins are shown below:

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

penguins.R

```
## ---- smaller-penguins
```

Label block of code

```
smaller <- penguins %>%  
  filter(species == "Adelie",  
         !is.na(body_mass_g))
```

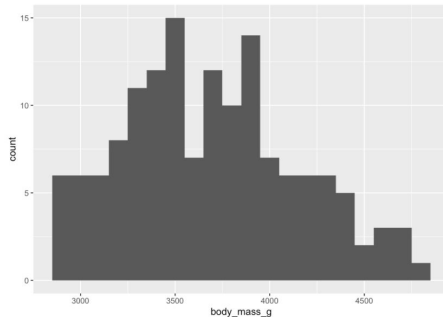
```
## ---- plot-penguins
```

```
smaller %>%  
  ggplot(aes(body_mass_g)) +  
  geom_histogram(binwidth = 100)
```

Penguins

2020-08-11

We have data about 344 penguins. Only 198 are classified as Adelie. The distribution of the Adelie penguins are shown below:



Child documents

adelie-report.Rmd

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

Uses all code

```
```${r, echo = FALSE}  
smaller <- penguins %>%
 filter(species == "Adelie",
 !is.na(body_mass_g))
```
```

We have data about `nrow(penguins)` penguins. Only `nrow(penguins) - nrow(smaller)` are classified as Adelie. 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)
```
```

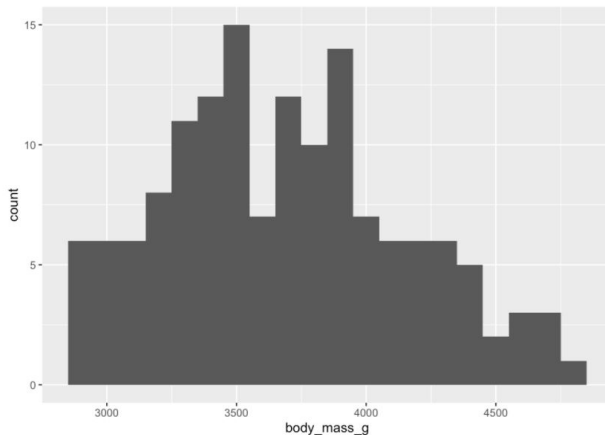
Uses report code

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

## Penguins

2020-08-11

We have data about 344 penguins. Only 198 are classified as Adelie. The distribution of the Adelie penguins are shown below:



# Logical Child documents

## adelie-report.Rmd

```

output: html_document

```

Uses all code

```
```{r, echo = FALSE}  
smaller <- penguins %>%  
  filter(species == "Adelie",  
         !is.na(body_mass_g))  
```
```

We have data about `nrow(penguins)` penguins. Only `nrow(penguins) - nrow(smaller)` are classified as Adelie. 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)
```

```
species_sel <- "Chinstrap"
```

Logical

```
```{r, child=if (species_sel == "Adelie") "a-report.Rmd"}```
```

Uses report code conditionally

Penguins

2020-08-11

# Blastula emails

```

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

```

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

Referenced code

```
```\{r penguin-plot, echo = FALSE}  
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"  
  )  
---
```

Generates email

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

penguin-email.Rmd

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

```

Important update c  
the penguins!

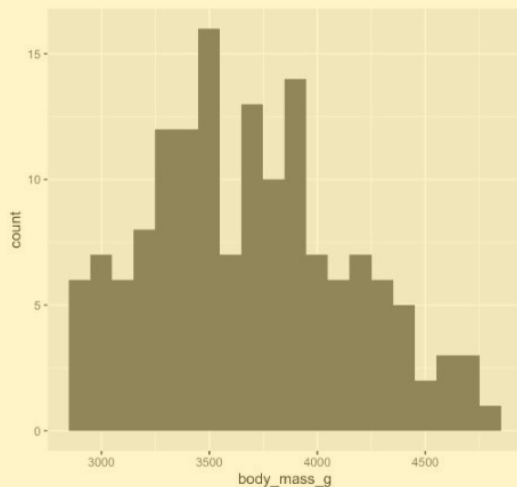
```
```\{r penguin-plot  
---
```

This is an email preview for RStudio Connect
Use `attach_connect_email(preview = FALSE)` to attach without this preview.

EMAIL SUBJECT: Penguin Report

Penguin Report

Important update on the state of the penguins!



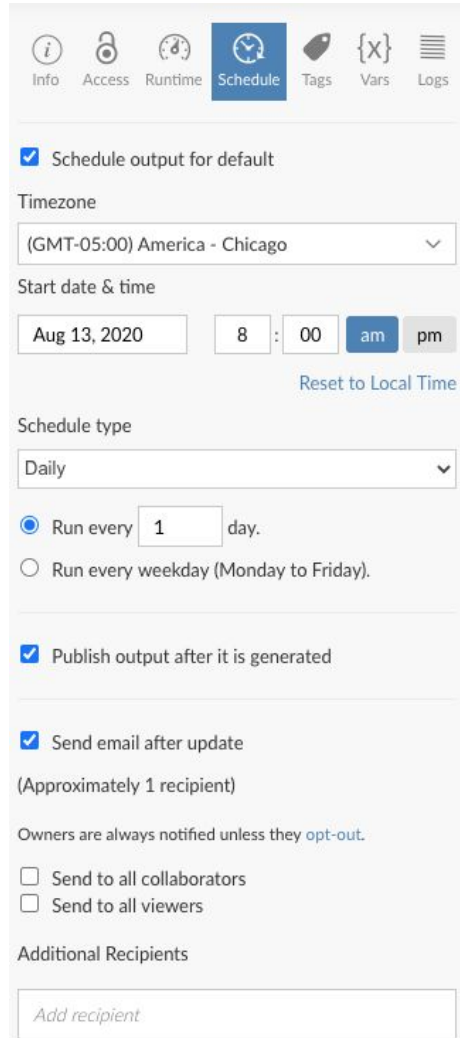
RMarkdown + RStudio Connect

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](#)

<https://docs.rstudio.com/connect/user/rmarkdown/>



The screenshot shows the 'Schedule' configuration page in RStudio Connect. At the top, there is a navigation bar with icons for Info, Access, Runtime, Schedule (which is highlighted), Tags, Vars, and Logs. Below the navigation bar, the 'Schedule' section is active. It includes a checkbox for 'Schedule output for default' which is checked. The 'Timezone' is set to '(GMT-05:00) America - Chicago'. The 'Start date & time' is set to 'Aug 13, 2020' at '8:00 am'. There is a link to 'Reset to Local Time'. The 'Schedule type' is set to 'Daily'. Under 'Run every', the 'Run every 1 day' option is selected. There is also an option for 'Run every weekday (Monday to Friday)'. Below this, there is a checkbox for 'Publish output after it is generated' which is checked. Another checkbox for 'Send email after update' is also checked, with a note '(Approximately 1 recipient)'. A note states 'Owners are always notified unless they opt-out.' There are two checkboxes for email notifications: 'Send to all collaborators' and 'Send to all viewers', both of which are unchecked. At the bottom, there is a section for 'Additional Recipients' with a text input field containing the placeholder 'Add recipient'.

Info Access Runtime **Schedule** Tags Vars Logs

☒ Schedule output for default

Timezone
(GMT-05:00) America - Chicago

Start date & time
Aug 13, 2020 8 : 00 am pm
[Reset to Local Time](#)

Schedule type
Daily

☒ Run every 1 day.
☐ Run every weekday (Monday to Friday).

☒ Publish output after it is generated

☒ Send email after update
(Approximately 1 recipient)

Owners are always notified unless they opt-out.

☐ Send to all collaborators
☐ Send to all viewers

Additional Recipients



Templating



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

Render with Params

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

Create in YAML

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

```
smaller <- penguins %>%
 filter(species == params$species,
 !is.na(body_mass_g))
...
```

We have data about `nrow(penguins)` penguins. Only `nrow(penguins) - nrow(smaller)` are classified as `params$species`. The distribution of the `params$species` penguins are shown below:

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

Reference in Code

Reference in Line

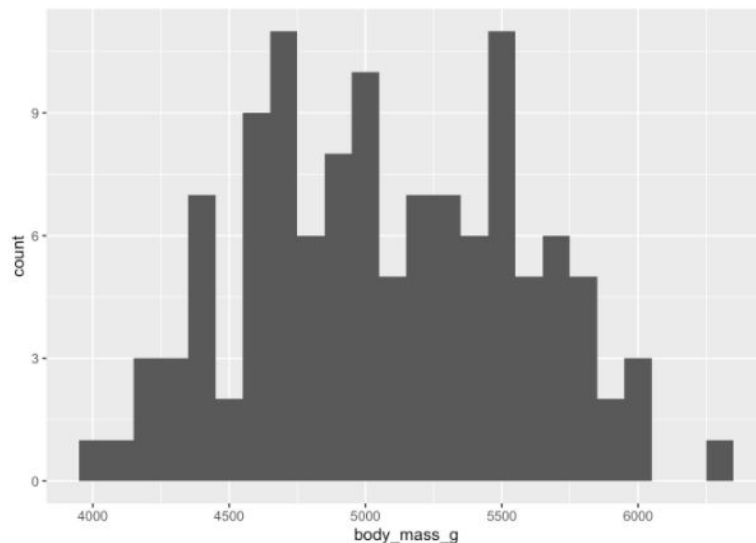
```
rmarkdown::render(  
  "penguin.rmd",  
  params = list(species = "Gentoo")  
)
```

Uses report code w/ new parameter

Penguins

2020-08-11

We have data about 344 penguins. Only 225 are classified as Gentoo. The distribution of the Gentoo penguins are shown below:



Slides at bit.ly/marvelRMD

Render with Params

```
---
title: "`r params$species` Penguins"
date: 2020-08-11
output: html_document
params:
  species: Adelia
---
```

Create in YAML

```
```{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 in Line

Uses report code w/ new parameter

```
for (penguin in unique(penguins$species)) {
  rmarkdown::render(
    input = "penguins-report.rmd",
    params = list(species = penguin),
    output_file = glue::glue("{penguin}-report.html")
  )
}
```

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

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

	 Adelia-report.html	584 KB	Aug 13, 2020, 11:45 AM
	 Chinstrap-report.html	672 KB	Aug 13, 2020, 11:45 AM
	 Gentoo-report.html	672.3 KB	Aug 13, 2020, 11:45 AM

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")` **Source function**

```
penguins <- palmerpenguins::penguins %>%
 filter(!is.na(bill_length_mm),
 !is.na(flipper_length_mm))
...
```

**Generate raw output**

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

Loop across input w/ function

multiplot.R

```
multiplot <- function(penguin_name){  
  glue(" \n### {penguin_name} \n \n") %>% cat()  
  
  df_pen <- penguins %>%  
    filter(as.character(species) == penguin_name)  
  
  flipper_len <- df_pen %>%  
    summarize(mean = mean(flipper_length_mm)) %>%  
    pull(mean) %>%  
    round(digits = 1)  
  
  glue::glue("There are {nrow(df_pen)} observations of  
{penguin_name} penguins. The average flipper length is  
{flipper_len}. \n") %>% cat()  
  
  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**

```
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"
```

Whisker for #TidyTuesday

```
use_tidyuesday_readme <- function(title_subject, dataset_name){  
  
  raw_readme <- readLines(here::here("static/readme_template.md"))  
  
  title_subject <- title_subject  
  
  dataset_name <- dataset_name  
  
  core_url <- "https://raw.githubusercontent.com/rfordatascience/tidyuesday/master/data/"  
  
  year <- lubridate::year(Sys.Date())  
  tues_date <- tidyuesdaymeta::next_tuesday()  
  week_num <- tidyuesdaymeta::next_week_num()  
  
  filled_readme <- whisker::whisker.render(raw_readme)  
  
  writeLines(  
    filled_readme,  
    here::here(  
      lubridate::year(lubridate::today()),  
      tidyuesdaymeta::next_tuesday(),  
      "readme.md"  
    )  
  )  
}
```

Fill the template

Whisker output

```
### Get the data here
```

```
`{r}
```

```
# Get the Data
```

```
# Read in with tidyuesdayR package
```

```
# Install from CRAN via: install.packages("tidyuesdayR")
```

```
# This loads the readme and all the datasets for the week of interest
```

```
# Either ISO-8601 date or year/week works!
```

```
tuesdata <- tidyuesdayR::tt_load('2020-08-11')
```

```
tuesdata <- tidyuesdayR::tt_load(2020, week = 33)
```

```
avatar <- tuesdata$avatar
```

Filled readme.md

```
# Or read in the data manually
```

```
avatar <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidyuesday/master/data/2020/2020-08-11/avatar.csv')
```

```
...
```

```
### Data Dictionary
```

```
# `avatar.csv`
```


Whisker inputs

```
---  
title: "{{species}}" Penguin"  
output: html_document  
---
```

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

Pass short text

```
smaller <- penguins %>%
 filter(species == "{{species}}",
 !is.na(body_mass_g))
...
```

We have data about `r nrow(penguins)` penguins. Only  
`r nrow(penguins) - nrow(smaller)` are classified as `{{species}}`.  
The distribution of the `{{species}}` penguins are shown below:

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

`{{long_prose}}`

Pass long text

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

```
smaller <- penguins %>%
 filter(species == Adelie, !is.na(body_mass_g))
...
```

We have data about `r nrow(penguins)` penguins. Only  
`r nrow(penguins) - nrow(smaller)` are classified as `Adelie`. The  
distribution of the `Adelie` penguins are shown below:

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

Mating

The 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.

whisker function

```
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."
```

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

usethis::use_template

Creates a file from data and a template found in a package. Provides control over file name, the addition to .Rbuildignore, and opening the file for inspection.

```
use_template(
  template,
  save_as = template,
  data = list(),
  ignore = FALSE,
  open = FALSE,
  package = "usethis"
)
```

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](#)
- RMarkdown [Cookbook](#)
- RMarkdown [Definitive Guide](#)

Slides at bit.ly/marvelRMD



BONUS HOWARD SLIDE



LOVES WATERMELON