Building a Dataviz Design System for a Medical Research Project

RMedicine 2023 | 8th June 2023

Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/



What is a Dataviz Design System?

Dataviz-friendly brand guidelines

- Colours
- Fonts
- Simple rules



What is a Dataviz Design System?

Dataviz-friendly brand guidelines

- Purposeful colour semantics
- Dataviz-friendly text formatting
- Preferred `geom`s
- Accessibility

🎨 Graphs, tables, presentations, documents...



Why have a Dataviz Design System?

"So much more than pretty graphs"

- Clear visual identity
- Decision-making & trial-and-error energy saver
- `#rstats` 💙 Automate the "boring stuff" to focus on the stuff that needs your expertise



Building a dataviz design system

Functional

- Text hierarchy
- Number of colours + symbolism
- Accessibility

Aesthetic

- On-brand
- Personality
- Nice extras (background, toggles)

Implementation

- Documentation
- Automation
- Flexibility



Let's go!

Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/



Gathering specific requirements

- What kind of colours would you like to be associated with this project?
- How many colours do you need?
- Any colour semantics we should include?



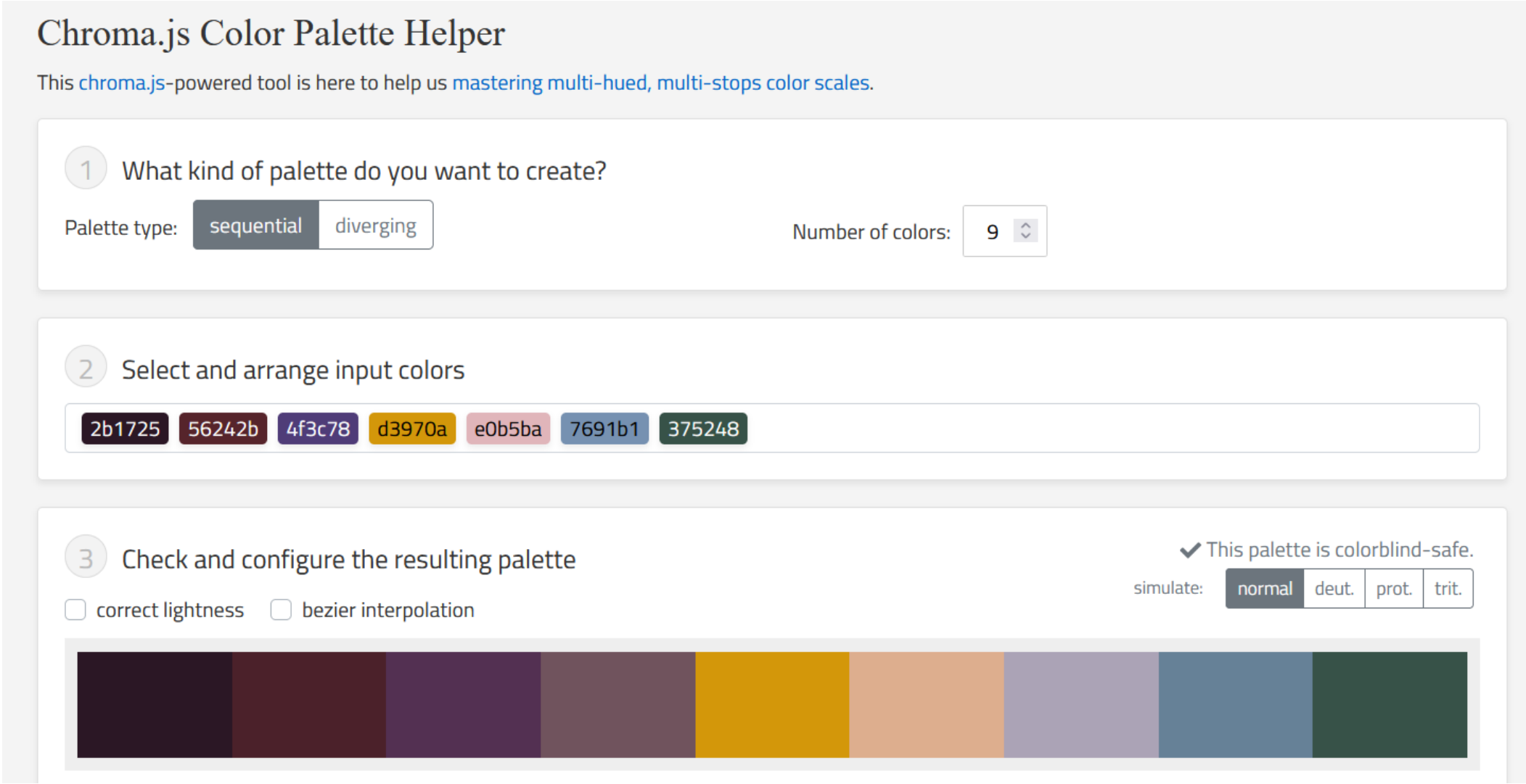
- What types of plots do you use a lot?
- How much personality would you like to convey in the text formatting?

Colour inspiration



- imagecolorpicker.com
- www.vis4.net/palettes - Chroma.js Color Palette Helper

Colour inspiration



Colour inspiration 

Chroma.js Color Palette Helper

This chroma.js-powered tool is here to help us mastering multi-hued, multi-stops color scales.

1 What kind of palette do you want to create?

Palette type: sequential | diverging

Number of colors: 32

2 Select and arrange input colors

2b1725 56242b 4f3c78 d3970a e0b5ba 7691b1 375248

3 Check and configure the resulting palette

This palette is colorblind-safe.

simulate: normal | deut. | prot. | trit.

correct lightness | bezier interpolation



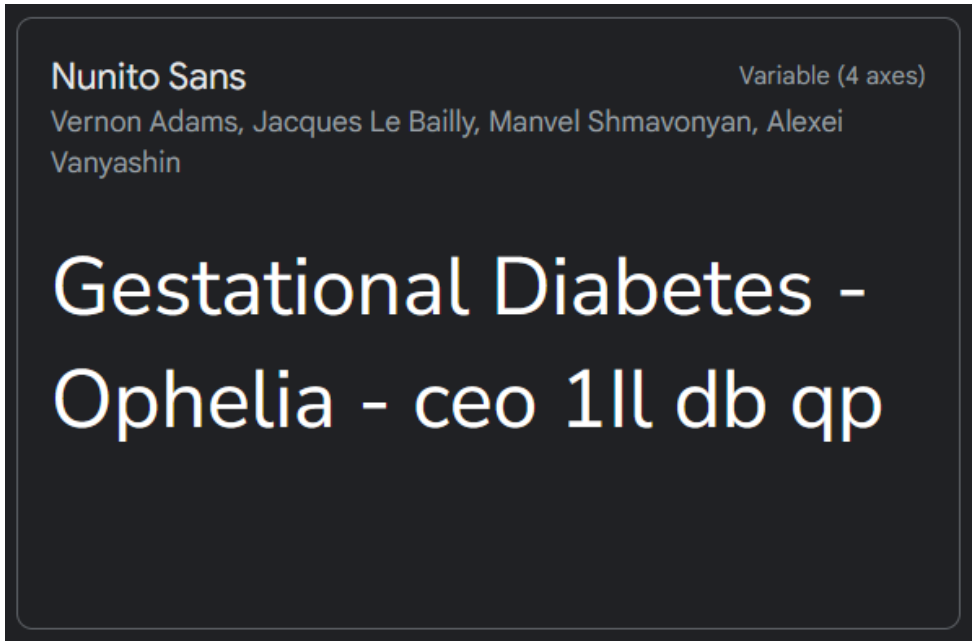
Font inspiration

- Make it easy for the end user!
  - Easy to find & install
  - `.ttf`
- Italics & bold
- Numbers & special characters
- Easy to read



Font inspiration

- Make it easy for the end user!
  - Easy to find & install
  - `.ttf`
- Italics & bold
- Numbers & special characters
- Easy to read



Implementation

Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/



Colours

Anchor colours

```
ophelia_colours <- list(
  deep_purple = "#2B1725",
  dark_red = "#56242b",
  purple =  "#4f3c78",
  gold = "#d3970a",
  pink = "#e0b5ba",
  light_blue = "#7691b1",
  dark_green = "#375248",
  na_value = "#6A686F",
  ... # and a few more derived from these
)
```




Colours

Several palettes, derived from the anchor colours

```
ophelia_palettes <- list(
  default = c(ophelia_colours$deep_purple,
              ophelia_colours$dark_red,
              ophelia_colours$purple,
              ophelia_colours$gold,
              ophelia_colours$pink,
              ophelia_colours$light_blue,
              ophelia_colours$dark_green),
  cool_colours = c(ophelia_colours$dark_green,
                   ophelia_colours$light_blue,
                   ophelia_colours$pale_green),
  warm_colours = c(ophelia_colours$deep_purple,
                   ophelia_colours$pink,
                   ophelia_colours$gold),
  neg_to_pos = c(ophelia_colours$deep_purple,
                 ophelia_colours$pale_pink,
                 ophelia_colours$bright_gold),
  greens = c(ophelia_colours$dark_green,
             ophelia_colours$pale_green),
  purples = c(ophelia_colours$deep_purple,
              ophelia_colours$pale_purple)
)
```




Colours

Feed the palettes into a bespoke function that uses ggplot scale functions

```
scale_colour_ophelia <- function(palette = "default",
                                 continuous = FALSE,
                                 .colours = ophelia_colours,
                                 .palettes = ophelia_palettes,
                                 ...) {

  if(continuous == FALSE) {

    ggplot2::discrete_scale(palette = grDevices::colorRampPalette(.palettes[[palette]]),
                            aesthetics = "colour",
                            scale_name = .palettes[[palette]],
                            na.value = .colours$na_value,
                            ...)

  } else {

    ggplot2::scale_colour_gradientn(colours = .palettes[[palette]],
                                    na.value = .colours$na_value,
                                    ...)
  }

}
```




Colours

... with a few extra touches



Usage

```
scale_colour_ophelia(
  palette = "default",
  reverse = FALSE,
  continuous = FALSE,
  .aesthetic = "colour",
  .colours = ophelia_colours,
  .palettes = ophelia_palettes,
  ...
)
```

Arguments

`palette`	The Ophelia palette you want to use. Choose from "default", "cool_colours", "warm_colours", "neg_to_pos", "greens", or "purples".
`reverse`	Logical. Default is `FALSE`. `TRUE` reverses the palette (e.g. makes "neg_to_pos" start with the positive end rather than the negative end of the colour palette).
`continuous`	Logical. Default is `FALSE`. Change to `TRUE` when applying the colour scale to a continuous variable.
`.aesthetic`	Default is "colour". Can be changed to "fill", but it's best to use `scale_fill_ophelia()` to apply the colour palettes to the fill aesthetic.
`.colours`	Used to access `ophelia_colours`. Please leave as is.
`.palettes`	Used to access `ophelia_palettes`. Please leave as is.
	Additional arguments to pass to `ggplot2::continuous_scale()`



Colours

Text colours: find a “starting” colour the ties in with all the palettes

```
monochromeR::generate_palette(
  ophelia_colours$dark_red,
  blend_colour = ophelia_colours$purple,
  n_colours = 5,
  view_palette = TRUE)
```




```
[1] "#56242B" "#54283A" "#532D49" "#513259"
```

Colours

Text colours: feed it into monochromeR to generate a dark text colour

```
monochromeR::generate_palette(
  "#532D49",
  "go_darker",
  n_colours = 5,
  view_palette = TRUE)
```




```
[1] "#532D49" "#42243A" "#311B2B" "#21111D"
```

Colours

Text colours: feed **that** dark text colour into monochromeR again to generate a light text colour

```
monochromeR::generate_palette(
  "#10090E",
  "go_lighter",
  n_colours = 8,
  view_palette = TRUE)
```



```
[1] "#10090E" "#2B2529" "#464145" "#615D60"
"#7D797C" "#989597" "#B3B1B3"
[8] "#CFCDCE"
```




Plot theme

`theme_ophelia()`

- Removing clutter
- Adding custom fonts and text hierarchy
- Custom margins, relative text sizes, `ggtext::element_markdown` or `ggtext::element_textbox_simple`, ...
- Effortlessly achieve a consistent aesthetic beyond the colour scheme



Plot theme

... with a few option toggles



Usage

```
theme_ophelia(
  title_font = "Crimson Pro",
  base_font = "Nunito Sans",
  void = FALSE,
  show_grid_lines = TRUE,
  base_text_size = 11,
  background_colour = TRUE,
  .colours = ophelia_colours
)
```

Arguments

`title_font`	The default title is \"Crimson Pro\". To change to a different font (e.g. to match up with an external publication requirement), simply change as desired (e.g. \"Times New Roman\"). Make sure you've installed the fonts you want to use on your device first!
`base_font`	The default font is \"Nunito\"). If you want to use a different font (e.g. to match up with an external publication requirement), simply change as desired (e.g. \"Arial\"). Make sure you've installed the fonts you want to use on your device first!
`void`	Logical (`TRUE`/`FALSE`). If `TRUE`, all grid lines and axes are removed. This is useful when creating pie/donut charts.
`show_grid_lines`	Logical (`TRUE`/`FALSE`). If `FALSE`, all grid lines are removed but the axis text is retained.
`base_text_size`	Base text size in pt. The relative size of the title / subtitle text and of the margins is derived from this.
`background_colour`	Logical. The default (`TRUE`) adds a light colour to the background in line with the ophelia colour palette. Change to `FALSE` for a white background.
`.colours`	Used to read in `ophelia_colours`. Please leave as is!





{ophelia}

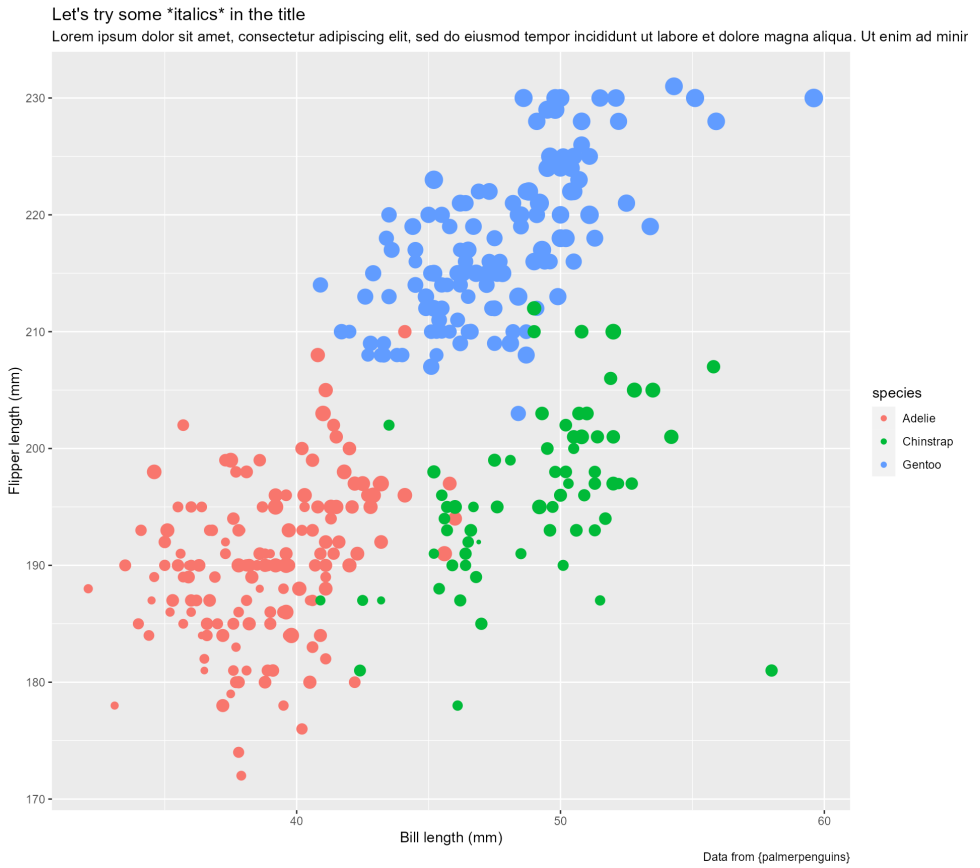
Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/

Two extra lines



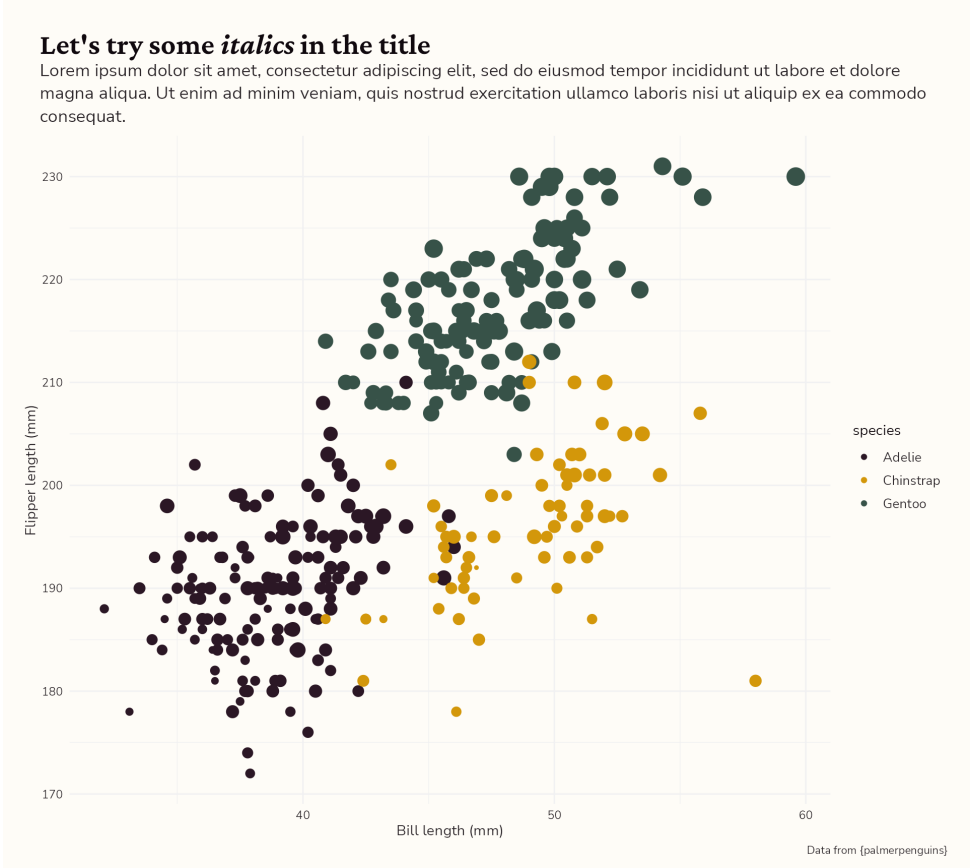
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 colour = species,
                 size = body_mass_g)) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Let's try some *italics* in
       subtitle = "Lorem ipsum dolor sit ame
       caption = "Data from {palmerpenguins
  guides(size = "none")
```



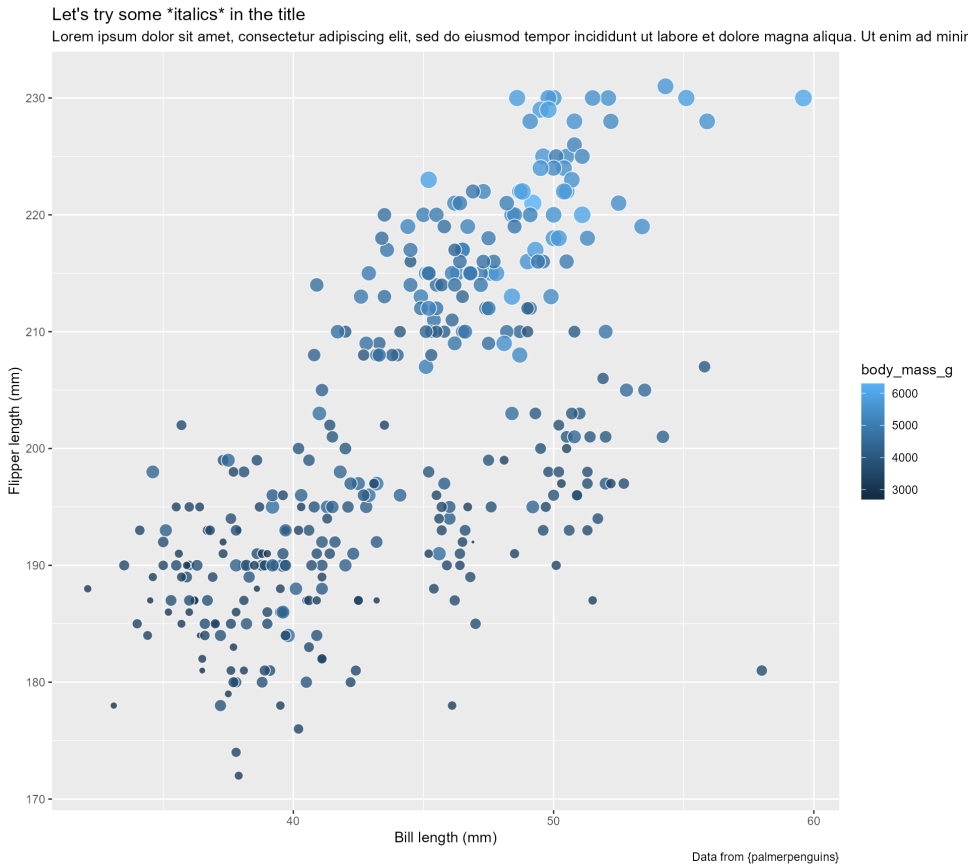
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 colour = species,
                 size = body_mass_g)) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Let's try some *italics* in
       subtitle = "Lorem ipsum dolor sit ame
       caption = "Data from {palmerpenguins
  guides(size = "none") +
  scale_colour_ophelia() +
  theme_ophelia()
```



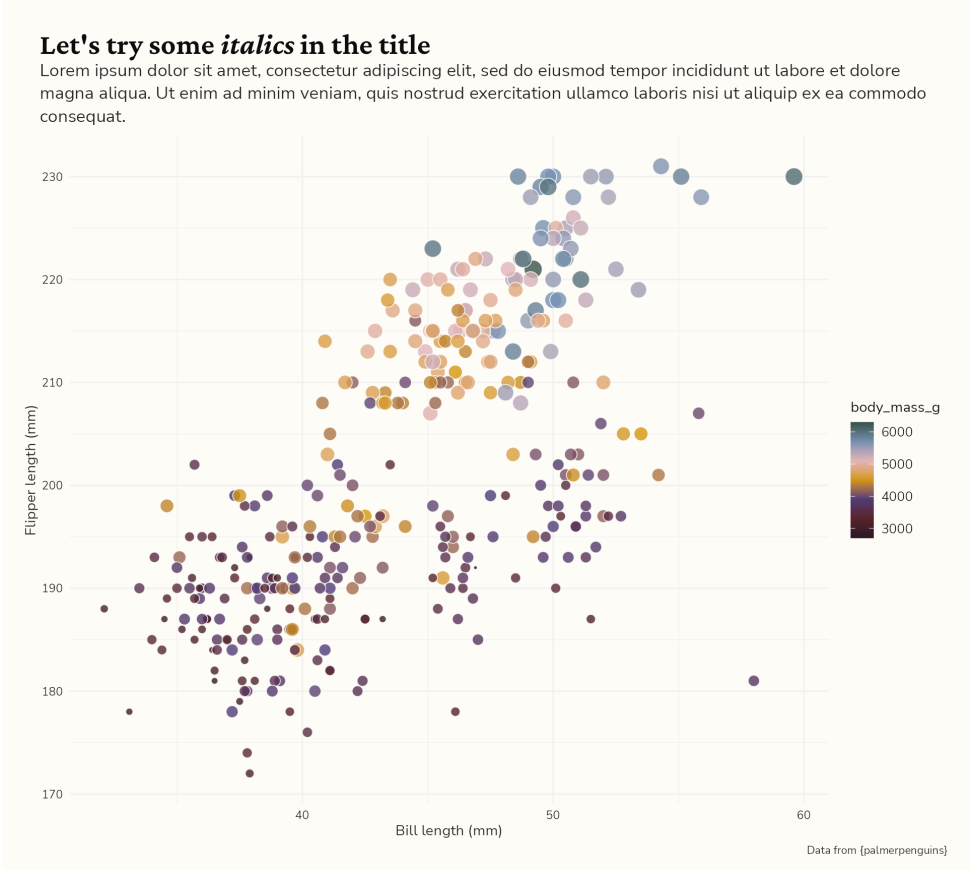
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 fill = body_mass_g,
                 size = body_mass_g),
             shape = 21,
             colour = "white",
             alpha = 0.8) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Let's try some *italics* in
       subtitle = "Lorem ipsum dolor sit ame
       caption = "Data from {palmerpenguins
  guides(size = "none")
```



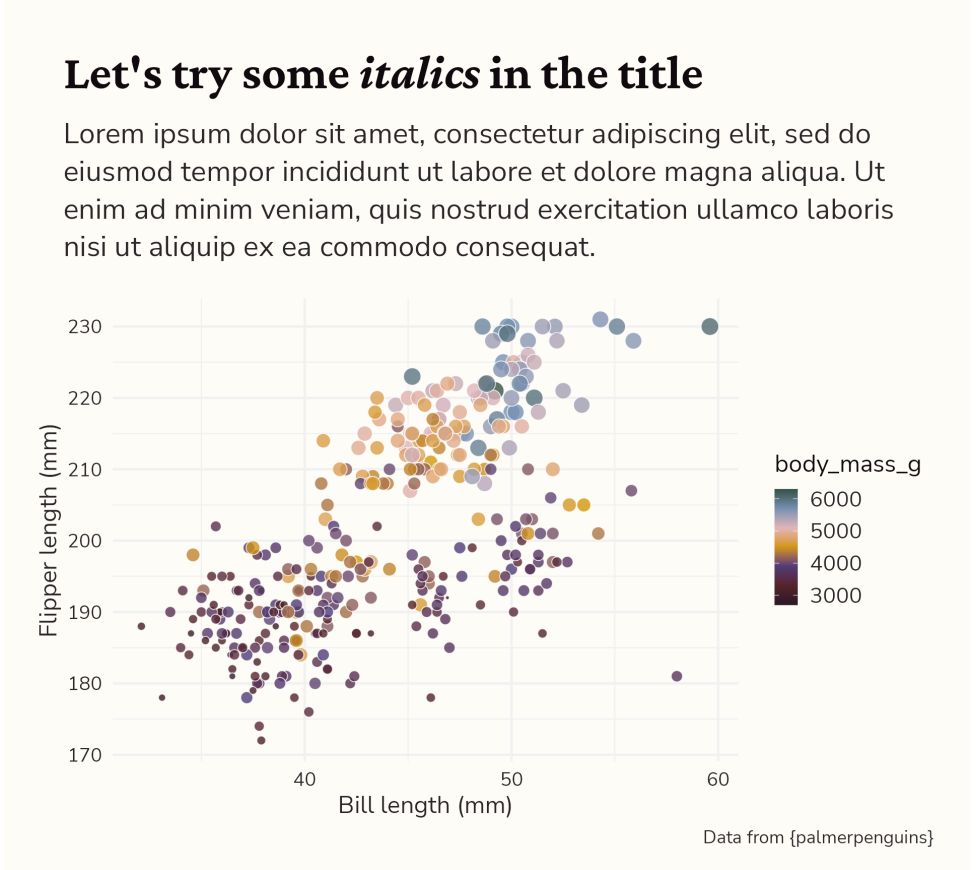
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 fill = body_mass_g,
                 size = body_mass_g),
             shape = 21,
             colour = "white",
             alpha = 0.8) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Let's try some *italics* in
       subtitle = "Lorem ipsum dolor sit ame
       caption = "Data from {palmerpenguins
  guides(size = "none") +
  scale_fill_ophelia(continuous = TRUE) +
  theme_ophelia()
```



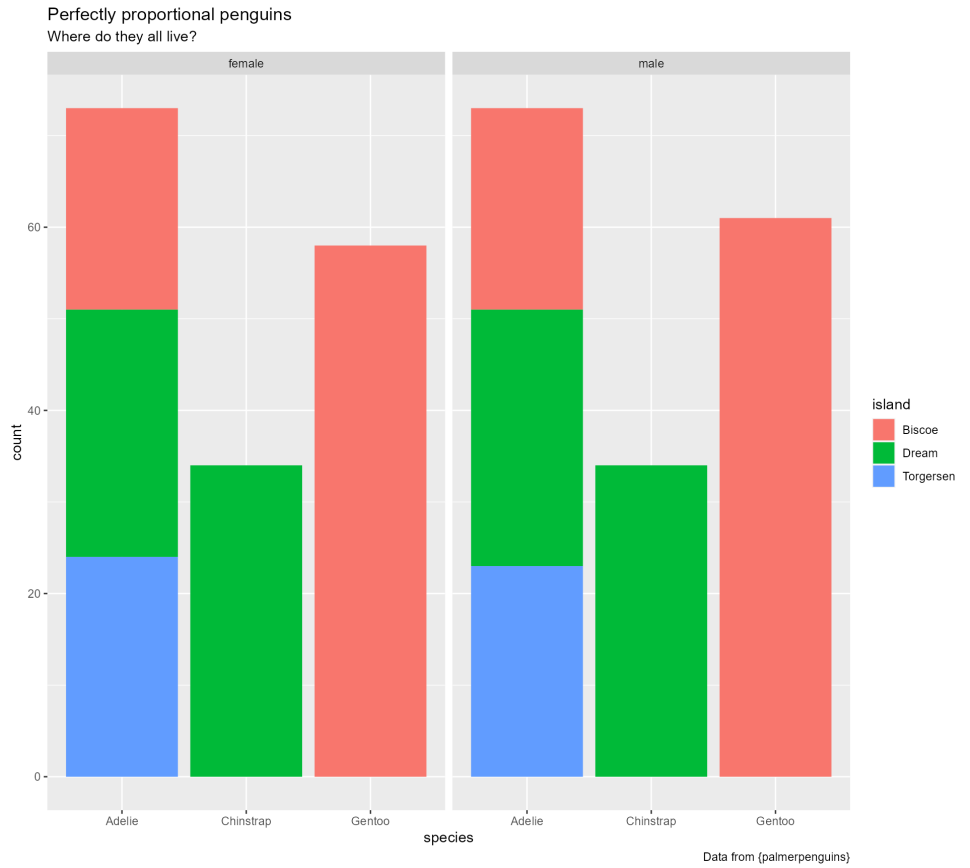
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 fill = body_mass_g,
                 size = body_mass_g),
             shape = 21,
             colour = "white",
             alpha = 0.8) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Let's try some *italics* in
       subtitle = "Lorem ipsum dolor sit ame
       caption = "Data from {palmerpenguins
  guides(size = "none") +
  scale_fill_ophelia(continuous = TRUE) +
  theme_ophelia(base_text_size = 18)
```



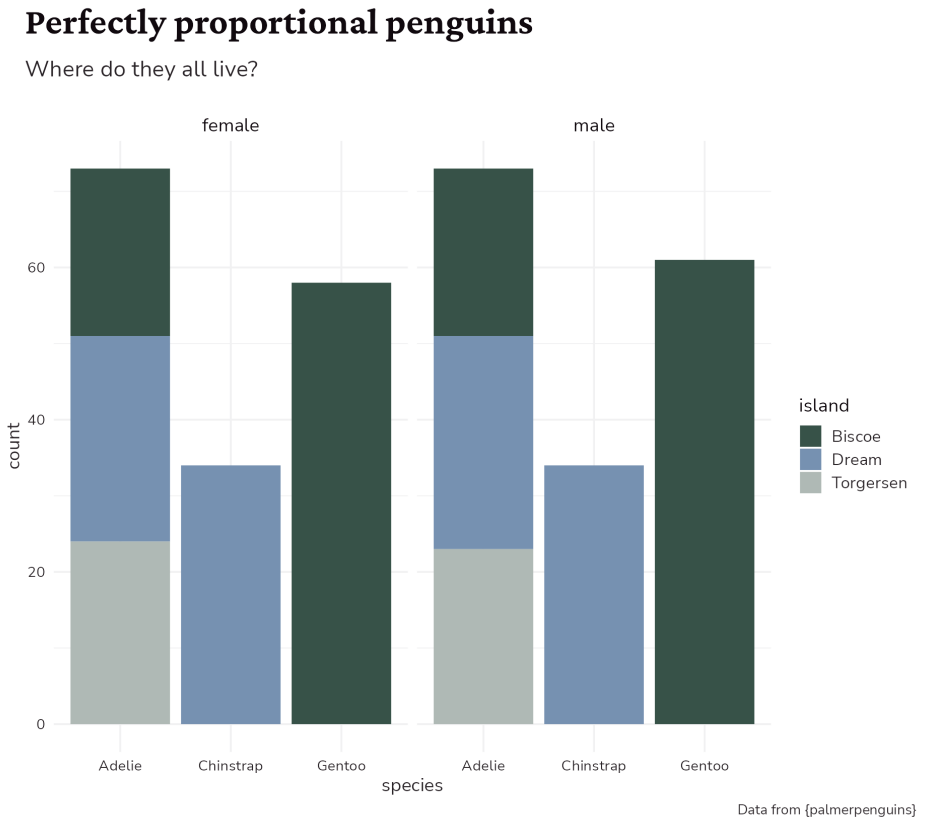
Two extra lines

```
palmerpenguins::penguins %>%
  filter(!is.na(sex)) %>%
  ggplot(aes(x = species,
             fill = island),
         stat = "count") +
  geom_bar() +
  labs(title = "Perfectly proportional pengu
       subtitle = "Where do they all live?",
       caption = "Data from {palmerpenguins
  facet_grid(. ~ sex)
```



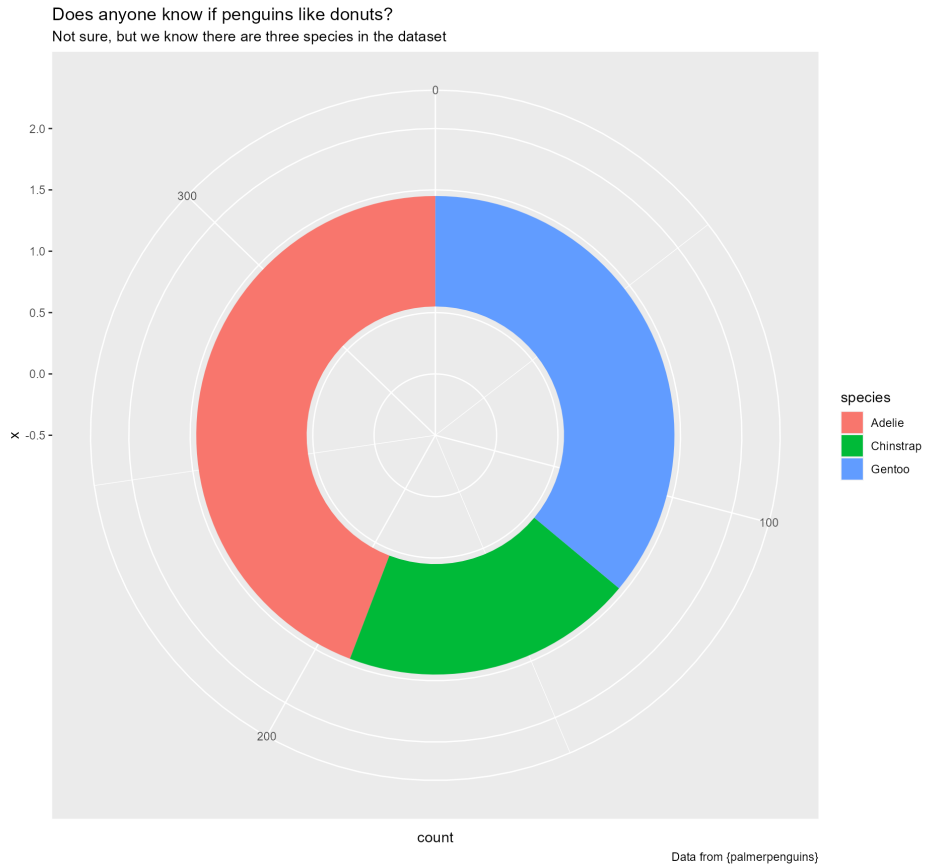
Two extra lines

```
palmerpenguins::penguins %>%
  filter(!is.na(sex)) %>%
  ggplot(aes(x = species,
             fill = island),
         stat = "count") +
  geom_bar() +
  labs(title = "Perfectly proportional pengu
       subtitle = "Where do they all live?",
       caption = "Data from {palmerpenguins
  facet_grid(. ~ sex) +
  scale_fill_ophelia(palette = "cool_colours
  theme_ophelia(background_colour = FALSE,
                base_text_size = 14)
```



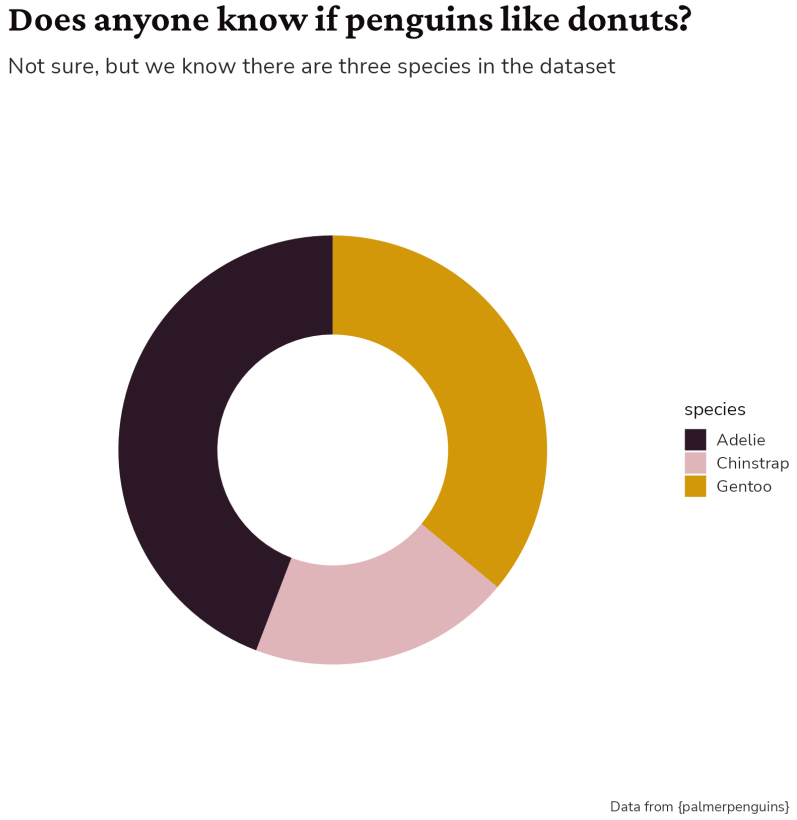
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot(aes(x = 1,
             fill = species),
         stat = "count") +
  geom_bar() +
  xlim(c(-0.5, 2)) +
  coord_polar(theta = "y") +
  labs(title = "Does anyone know if penguins
       subtitle = "Not sure, but we know the
       caption = "Data from {palmerpenguins
```



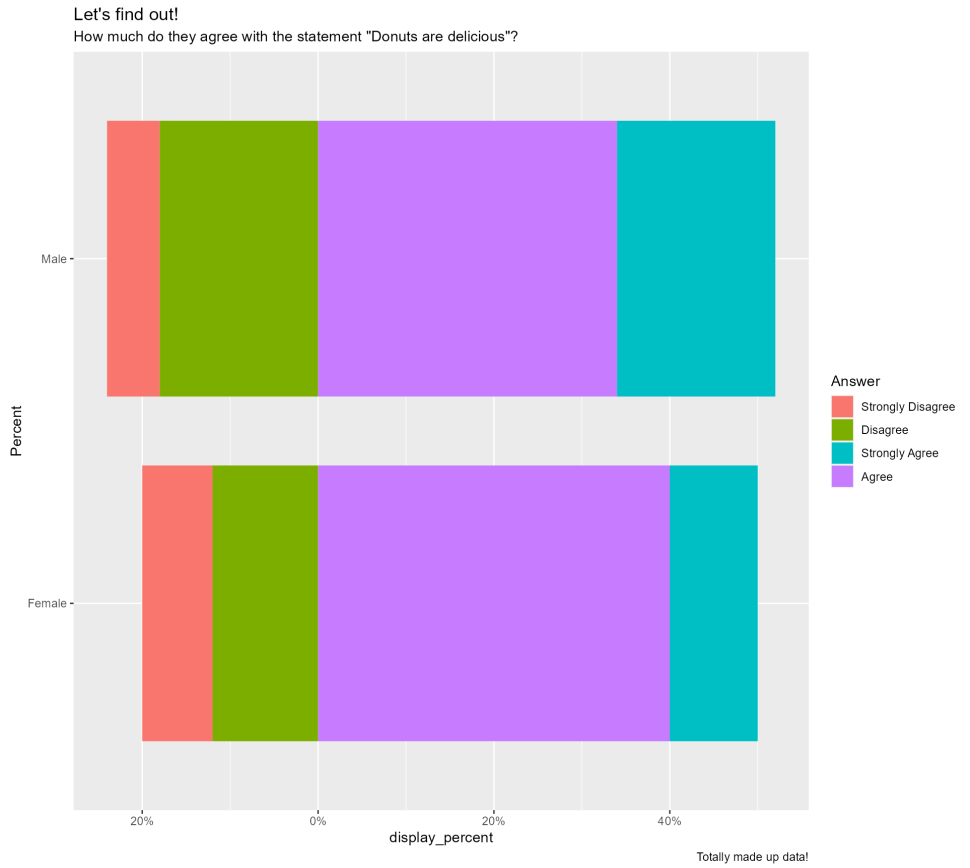
Two extra lines

```
palmerpenguins::penguins %>%
  ggplot(aes(x = 1,
             fill = species),
         stat = "count") +
  geom_bar() +
  xlim(c(-0.5, 2)) +
  coord_polar(theta = "y") +
  labs(title = "Does anyone know if penguins
       subtitle = "Not sure, but we know the
       caption = "Data from {palmerpenguins
  scale_fill_ophelia("warm_colours") +
  theme_ophelia(void = TRUE,
                base_text_size = 14,
                background_colour = FALSE)
```



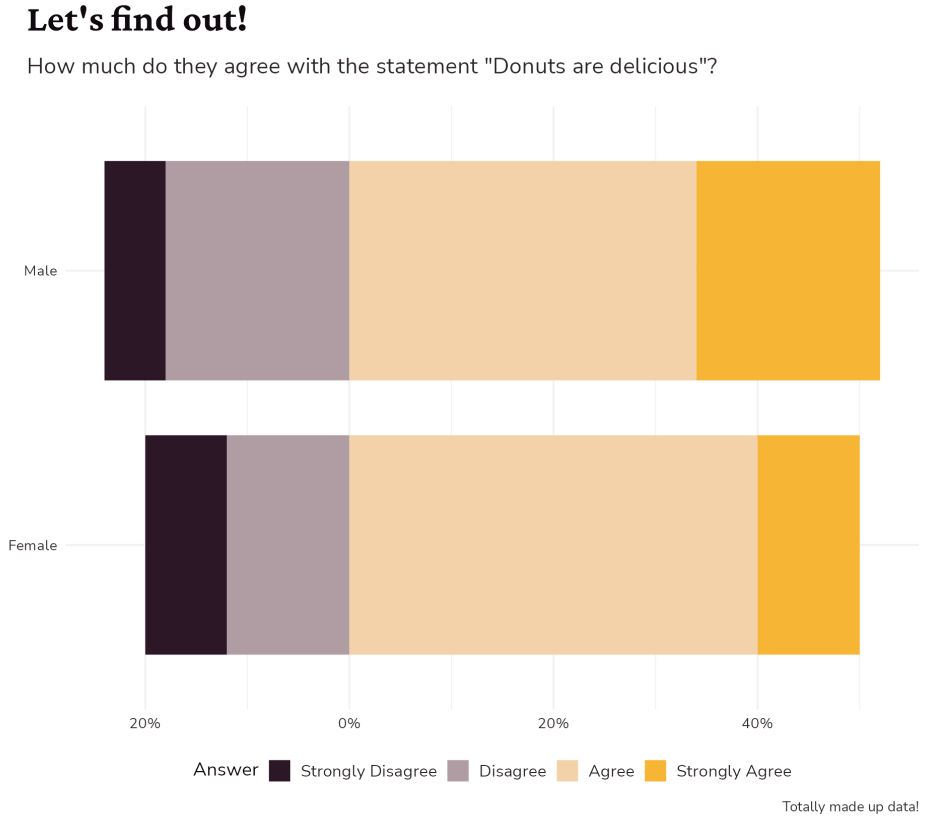
Two extra lines

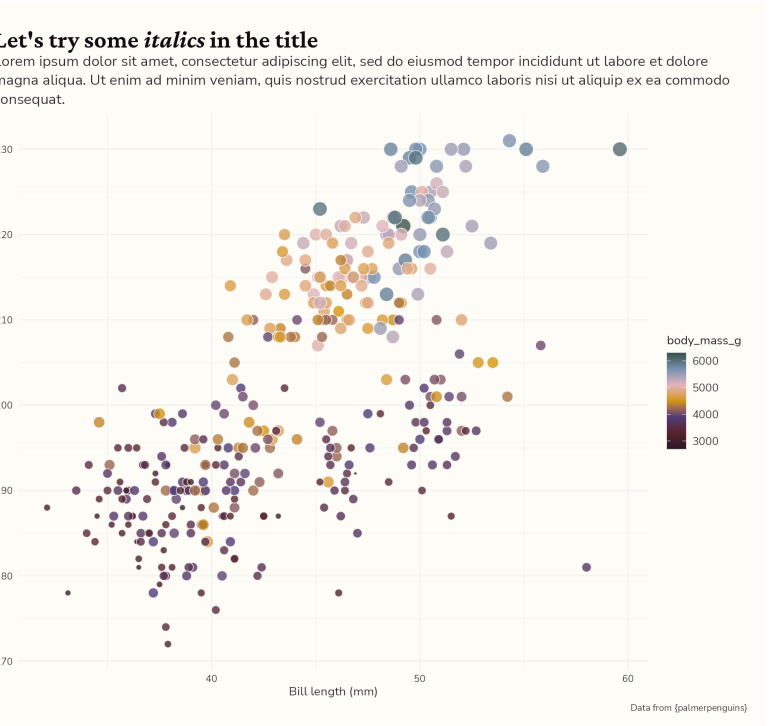
```
tibble(answer = factor(rep(c("Strongly Disa
                             "Agree", "Stro
                      levels = c("Strongly
                                 "Strongly
       percent = c(8, 12, 40, 10,
                   6, 18, 34, 18),
       group = sort(rep(c("Male", "Female"),
  mutate(display_percent = case_when(grepl(
                                     TRUE ~
  ggplot() +
  geom_col(aes(x = group,
               fill = answer,
               y = display_percent),
           width = 0.8) +
  labs(title = "Let's find out!",
       subtitle = "How much do they agree w
       caption = "Totally made up data!",
       x = "Percent",
       fill = "Answer") +
  scale_y_continuous(labels = function(x) pa
  coord_flip()
```

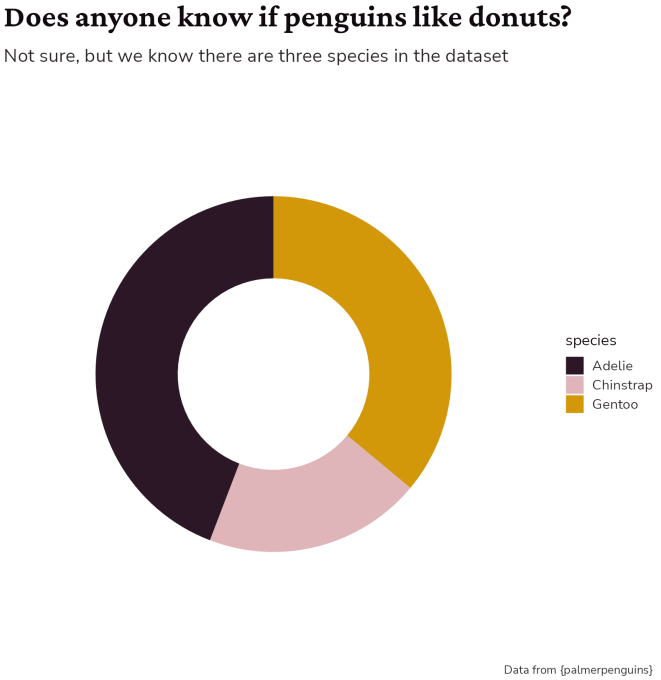


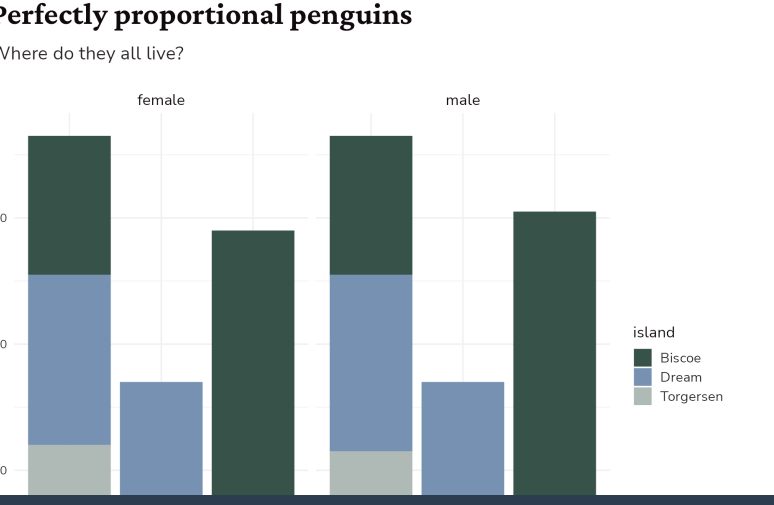
Two extra lines

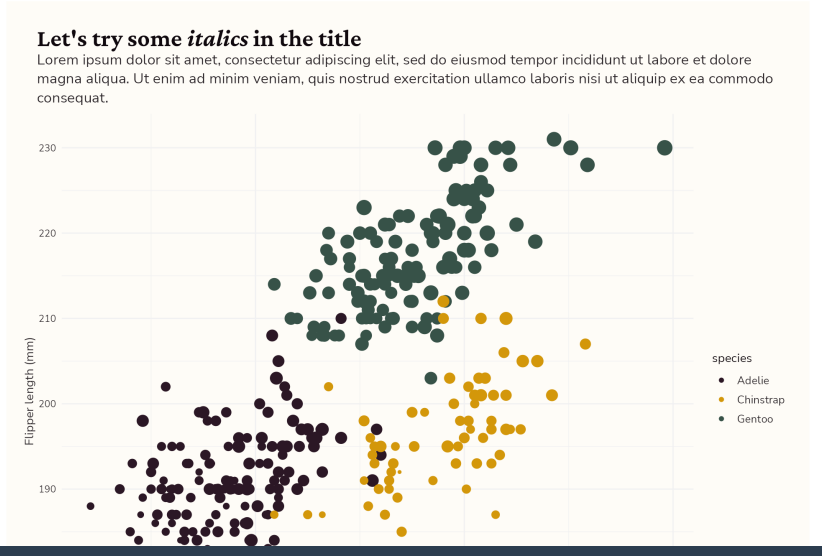
```
tibble(answer = factor(rep(c("Strongly Disa
                             "Agree", "Stro
                      levels = c("Strongly
                                 "Strongly
       percent = c(8, 12, 40, 10,
                   6, 18, 34, 18),
       group = sort(rep(c("Male", "Female"),
  mutate(display_percent = case_when(grepl(
                                     TRUE ~
  ggplot() +
  geom_col(aes(x = group,
               fill = answer,
               y = display_percent),
           width = 0.8) +
  labs(title = "Let's find out!",
       subtitle = "How much do they agree wi
       caption = "Totally made up data!",
       x = "Percent",
       fill = "Answer") +
  scale_y_continuous(labels = function(x) pa
  coord_flip() +
  scale_fill_ophelia(
    palette = "neg_to_pos",
```

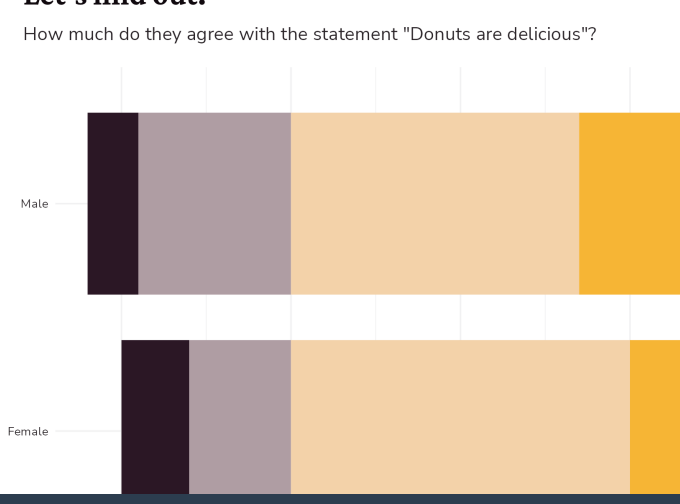












Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/

Next steps

- Add a table-styling function
- Finalise documentation
- Share package with the whole team
  - remotes::install_github(`"cararthompson/ophelia"`, `"TopSecretAuthorisationKey"`)
- Provide `ggplot` training and support in using the package
- Keep an eye out for Ophelia plots in the wild!



Over to you!

hello@cararthompson.com

Tw/Li: @cararthompson

☕💻

Dataviz Commissions
Dataviz Design Systems
Training & Consultations

Recording: cararthompson.com/talks/rmedicine2023-dataviz-design-system/