

Introduction to `ggplot2` and plotting flowers

R Ladies Trondheim startup meeting

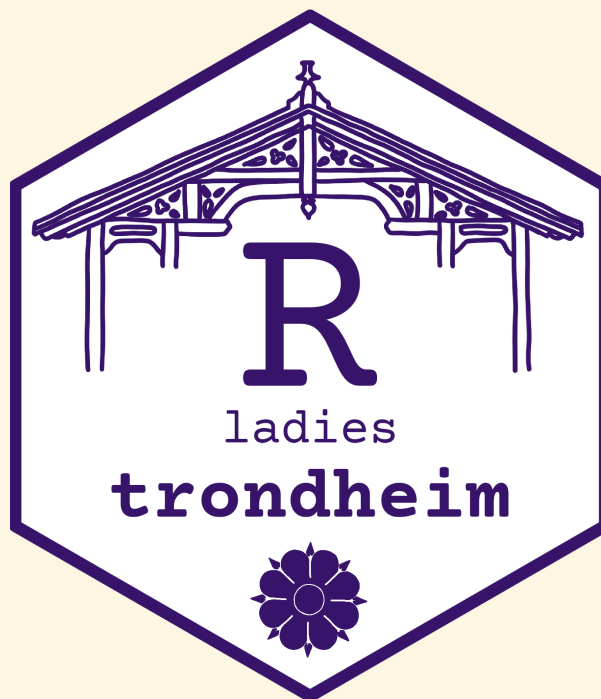
Emma Skarstein

Spring 2024

Goal for today

Thank you for coming! <3 Please help yourself to tea/coffee and cookies.

- What is R-Ladies Trondheim?
- Introduction to the package `ggplot2`
- Brief example of how to plot flowers in `ggplot2`
- Future of R Ladies Trondheim
- Talk at the tables: Who are you and why are you here?
- Cookies and mingling



Who am I?

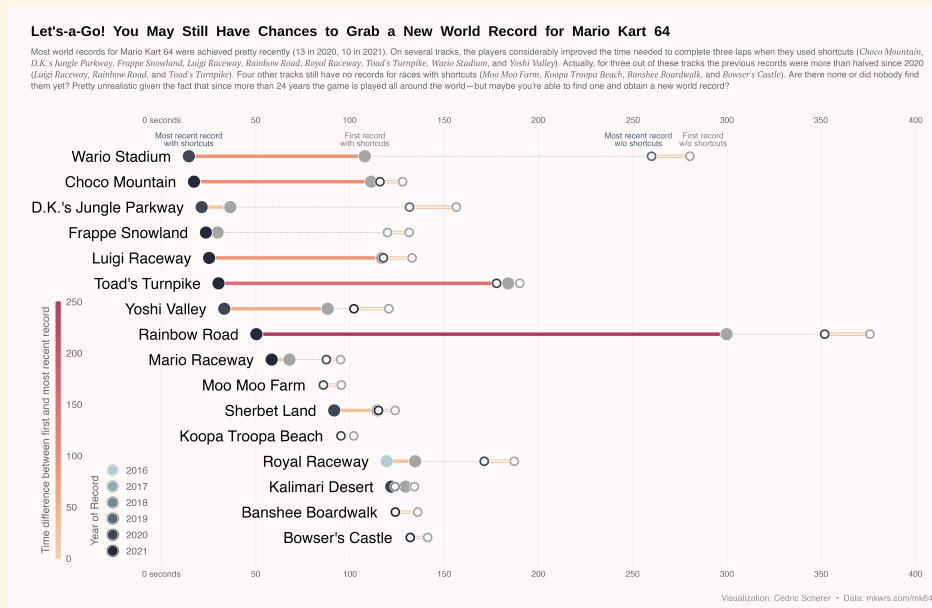
- Doing a PhD in statistics at the Department of mathematical sciences at NTNU
- Finished in September: feel sad about leaving the university, and maybe starting a job where I can't use R :'(
- Hope that R Ladies Trondheim can encourage connections between academia and industry, and across different departments at the university.

Get Started with `ggplot2`

Why ggplot2?

- It gives you a recognizable system - when you have understood the structure of `ggplot2` it is very easy to understand how to modify things to get exactly the plot you want.
- When you have learned how to tweak the layout it is really easy to create your own style.

Example: (here is a tutorial)



The components of ggplot2

<code>ggplot(data)</code>	The data that you want to build your graphic on.
---------------------------	---

<code>aes()</code>	Aesthetic mappings between your data and visual properties of your graphic.
--------------------	--

<code>geom_*()</code>	The geometric shapes representing the data.
-----------------------	--

```
ggplot(data, aes(x = ..., y = ..., color = ...)) +  
  geom_*()
```

First steps

Install the `ggplot2` package:

```
install.packages("ggplot2")
```

Load the package:

```
library(ggplot2)
```

For this tutorial, I'll use the `palmerpenguins` dataset, which you can install and load in the same way:

```
install.packages("palmerpenguins")
```

```
library(palmerpenguins)
```

A simple plot

Let's first take a look at the structure of our data.

```
head(penguins)
```

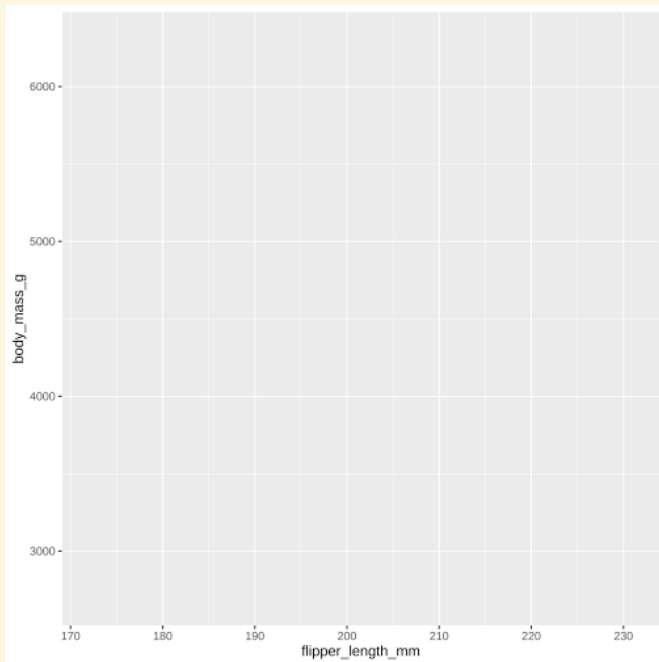
```
## # A tibble: 6 × 8
##   species island      bill_length_mm bill_depth_mm flipper_length_mm body_mas
##   <fct>    <fct>          <dbl>          <dbl>          <int>      <i
## 1 Adelie  Torgersen          39.1           18.7           181        3
## 2 Adelie  Torgersen          39.5           17.4           186        3
## 3 Adelie  Torgersen          40.3            18           195        3
## 4 Adelie  Torgersen          NA            NA            NA         NA
## 5 Adelie  Torgersen          36.7           19.3           193        3
## 6 Adelie  Torgersen          39.3           20.6           190        3
## # i 2 more variables: sex <fct>, year <int>
```

```
names(penguins)
```

```
## [1] "species"          "island"            "bill_length_mm"
## [4] "bill_depth_mm"    "flipper_length_mm" "body_mass_g"
## [7] "sex"              "year"
```


A simple plot

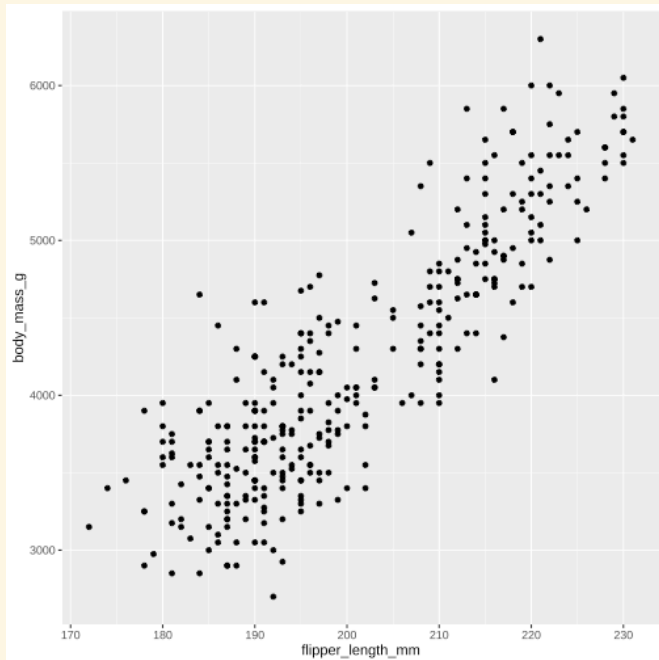
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g))
```



Let's say we want to plot the flipper length against body mass. We use the `ggplot()` function, and specify the data set, and what we want on the x- and y-axis. If we run this, we see the initialized plot.

A simple plot - Adding points

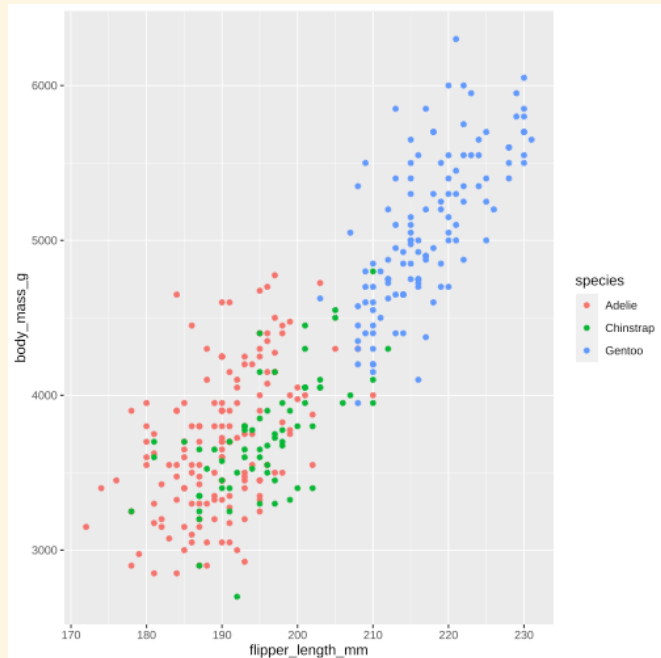
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point()
```



To get the actual data on the plot, we need to specify `geom_point()`. The layers of a `ggplot` are added together with a `+`.

A simple plot - Color by species

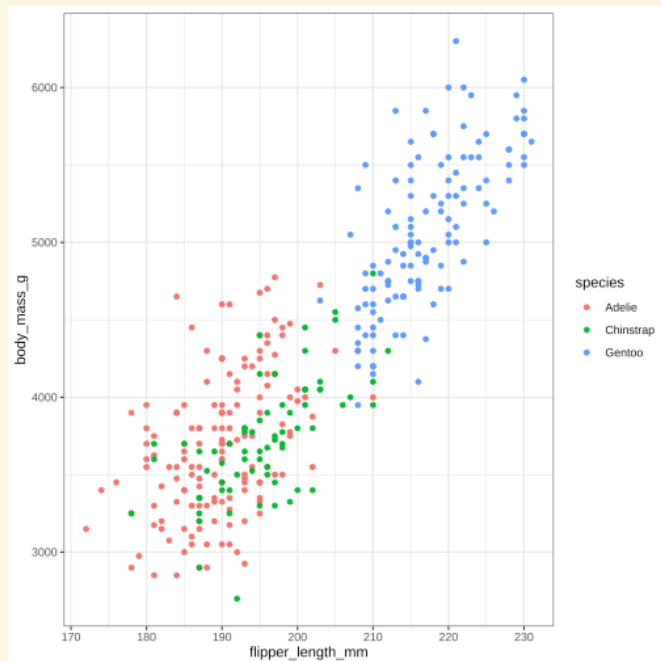
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species))
```



To color the points by species, add `aes(color = species)` inside `geom_point()`.

A simple plot - Themes

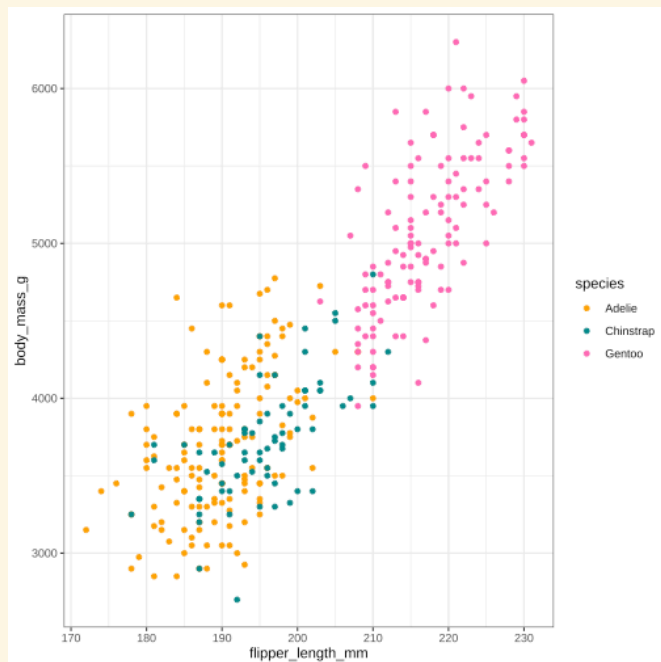
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species)) +  
  theme_bw()
```



We can make changes to the layout of the plot by specifying a different theme. The default is `theme_gray()`, let's change it to `theme_bw()`.

A simple plot - Color palettes

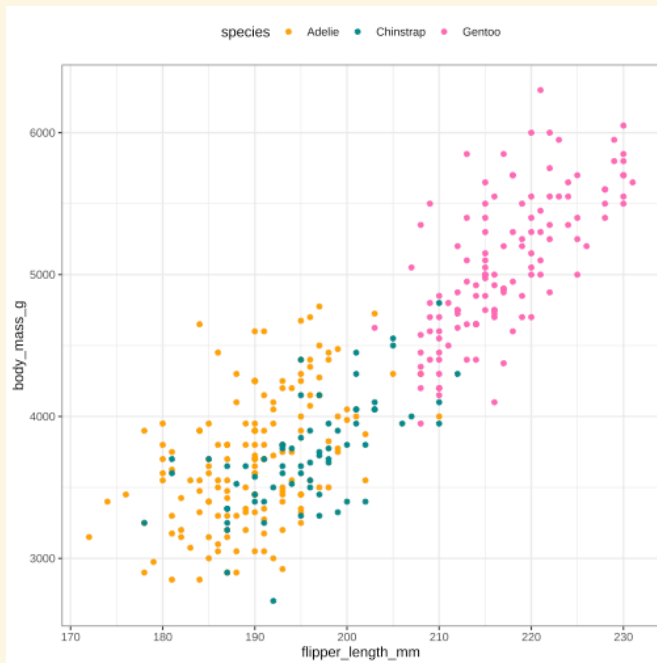
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species)) +  
  scale_color_manual(values = c("orange", "darkcyan", "hotpink")) +  
  theme_bw()
```



The color palette can be changed through `scale_color_...()`. In this case, we want to specify manual values for the colors, but you could also use ready-made color palettes, **there are lots of them!**

A simple plot - Theme options

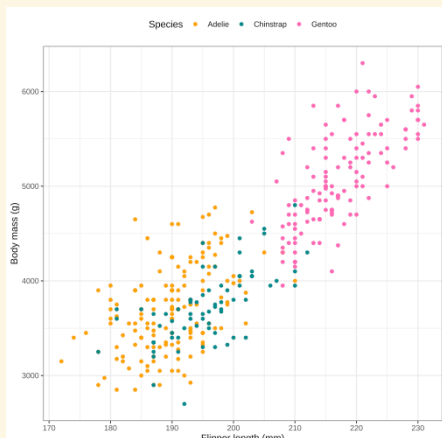
```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species)) +  
  scale_color_manual(values = c("orange", "darkcyan", "hotpink")) +  
  theme_bw() +  
  theme(legend.position = "top")
```



In the `theme()` function you can change stuff like background color, font, text sizes and colors, and much more, basically anything that has to do with the layout. There we move the legend to be on the top.

A simple plot - Axis and legend titles

```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species)) +  
  scale_color_manual(values = c("orange", "darkcyan", "hotpink")) +  
  labs(color = "Species",  
       x = "Flipper length (mm)", y = "Body mass (g)") +  
  theme_bw() +  
  theme(legend.position = "top")
```



We can change the axis and legend titles by using `labs()`. The reason we specify `color = "Species"` is that we want the legend for color to have this title. `labs()` is also where you can add a title, subtitle and caption.

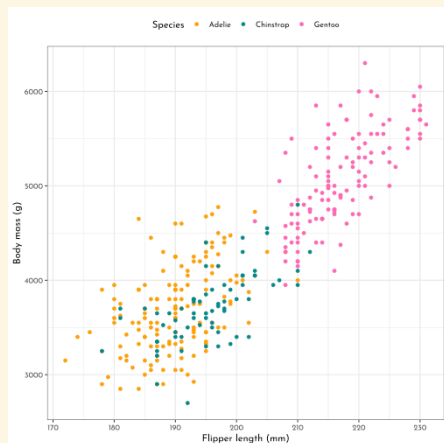
A simple plot - Fonts

```
library(showtext)
showtext_auto()
font_add_google(name = "Josefin Sans", family = "Josefin Sans")
```

The package `showtext` is great for changing fonts. You can change the font to any font from [Google Fonts](#).

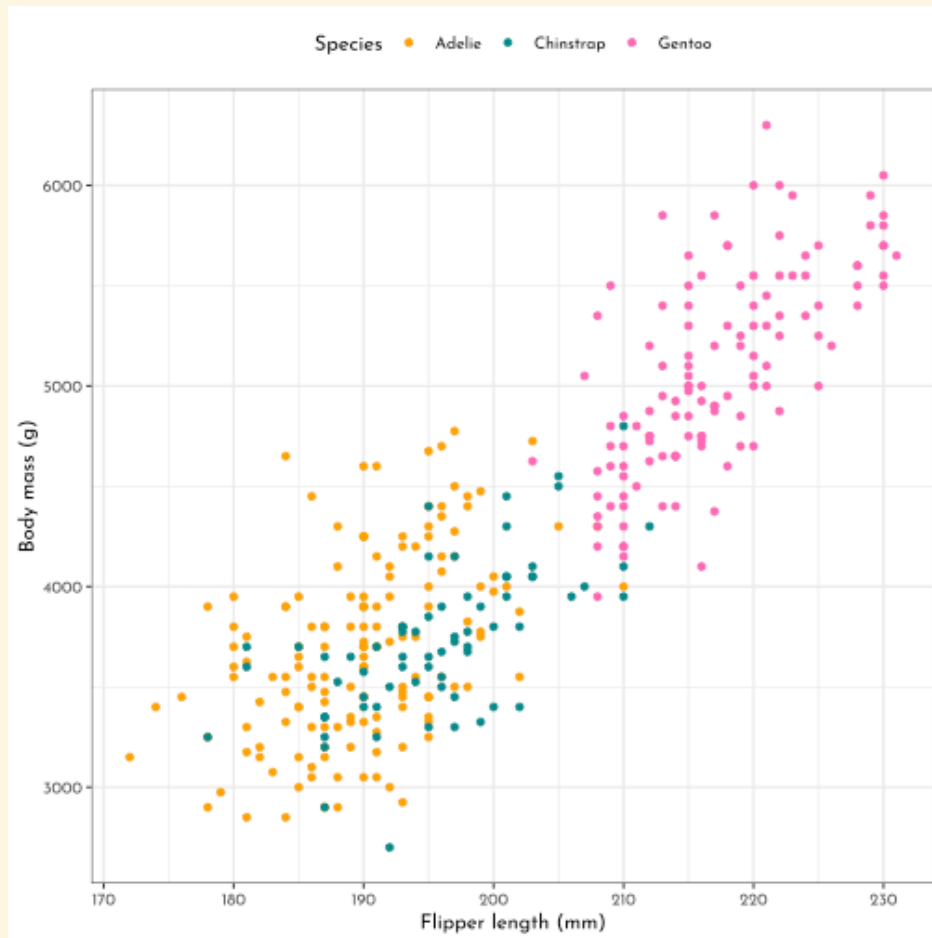
A simple plot - Fonts

```
ggplot(data = penguins,  
       aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species)) +  
  scale_color_manual(values = c("orange", "darkcyan", "hotpink")) +  
  labs(color = "Species",  
       x = "Flipper length (mm)", y = "Body mass (g)") +  
  theme_bw() +  
  theme(legend.position = "top",  
        text = element_text(family = "Josefin Sans"))
```



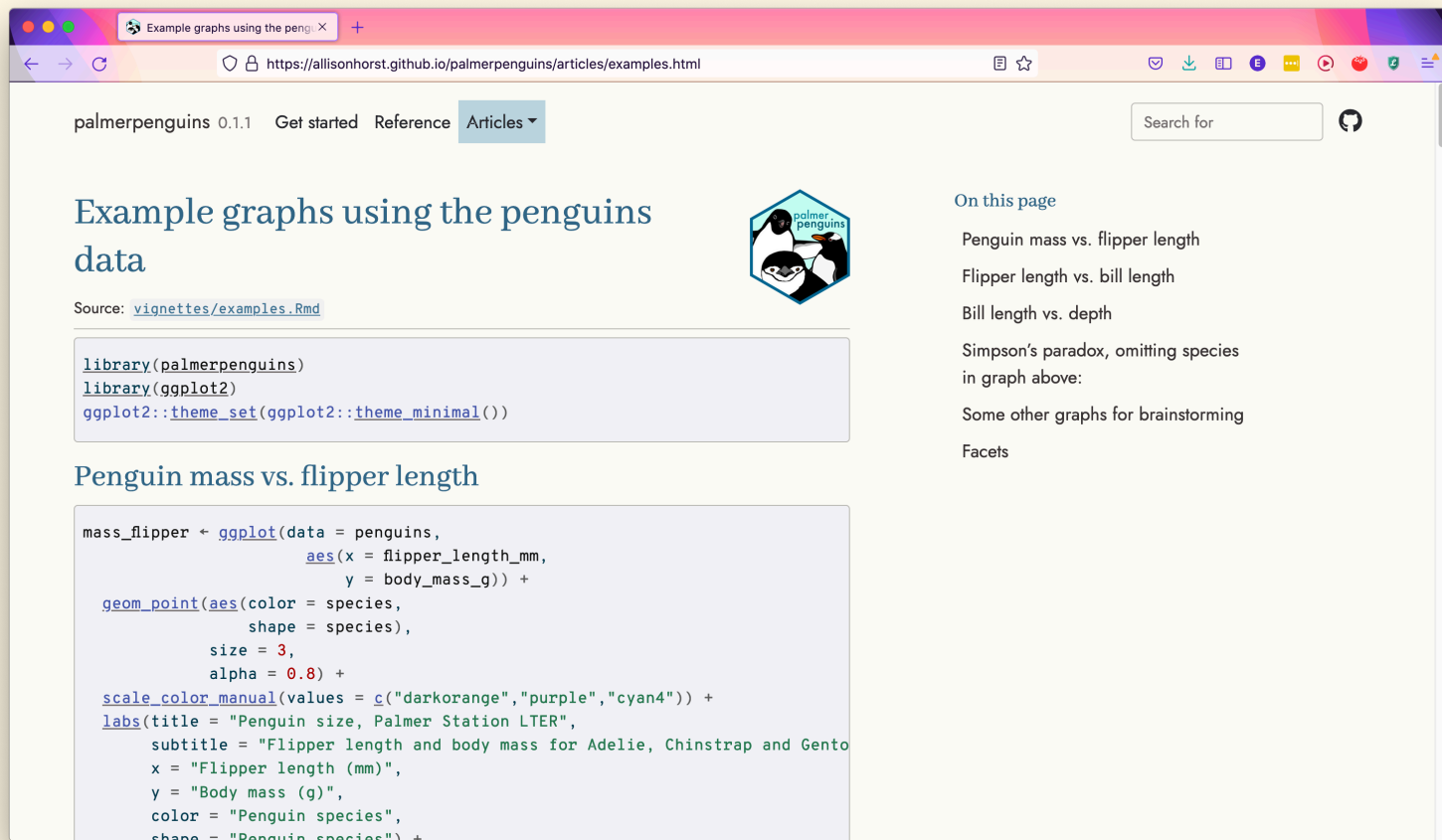
I used Josefin Sans here to make the change obvious. A nice go-to font is Open Sans.

Final penguin figure



More penguin plots

You can see some more examples of `ggplot2` plots with the `palmerpenguins` data [here](#).



The screenshot shows a web browser window with the URL `https://allisonhorst.github.io/palmerpenguins/articles/examples.html`. The page title is "Example graphs using the penguins data". The page content includes a source link to `vignettes/examples.Rmd`, a code block for loading the `palmerpenguins` data and setting the `ggplot2` theme, and a section titled "Penguin mass vs. flipper length" with a corresponding `ggplot2` code snippet. The code snippet is as follows:

```
mass_flipper <- ggplot(data = penguins,
  aes(x = flipper_length_mm,
    y = body_mass_g)) +
  geom_point(aes(color = species,
    shape = species),
    size = 3,
    alpha = 0.8) +
  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
  labs(title = "Penguin size, Palmer Station LTER",
    subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gento
    x = "Flipper length (mm)",
    y = "Body mass (g)",
    color = "Penguin species",
    shape = "Penguin species") +
```

On the right side of the page, there is a sidebar titled "On this page" with a list of links: "Penguin mass vs. flipper length", "Flipper length vs. bill length", "Bill length vs. depth", "Simpson's paradox, omitting species in graph above:", "Some other graphs for brainstorming", and "Facets".

ggplot2 resources

- [The R graph gallery](#): different graphs in base R and ggplot2, many examples and tutorials.
- [ggplot2 cheatsheet](#)
- Tutorial by Cédric Scherer, "[Graphic design with ggplot2](#)"
 - A really great reference, goes into *great* detail of all the options of ggplot2

Plotting flowers

Next steps for R Ladies Trondheim

Next steps for R Ladies Trondheim

- What kind of meetings are useful? Workshops, talks, co-working, just chatting?
- Potential talks/workshops:
 - CV in RMarkdown
 - Making slides using Xaringan
 - Using R in data journalism
 - Writing academic articles using Quarto
 - Making your own website in R
 - Creating R packages
 - Reproducible research reports using RMarkdown

Slides created via the R package `xaringan`, all code is available on GitHub:
<https://github.com/emmaSkarstein/R-Ladies-Trondheim>