Data Visualization

2024-02-05

Today's gameplan

- We will break down `ggplot()` in painstaking detail.
  - I will try to address many common causes of error.
- I will then cut you loose to practice.

If you prefer typing along on your own instead of having the code written for you

- The `.r` script is mostly blank. You can fill in things as you go.
  - I will pause to give people a chance to type a few times in the lecture.
- Alternatively, if you prefer, you can just run my chunks in the `.rmd` as you follow along.

Let's load in a few important packages

```
library(tidyverse)
library(palmerpenguins)  #has a dataset we are interested
library(ggthemes)  #gives us more options to mess with ggp
library(plotly)  #for interactive plots
```

Bringing in our data and peeking at it

- With `palmerpenguins` loaded, we can `data(penguins)`.
- `glimpse()` is a cleaner version of `str()`, take a look

```
data(penguins)
glimpse(penguins)
str(penguins)
```

Breaking down `ggplot()` step-by-step.

- note that by using `ggplot()` I just get a blank screen

```
ggplot(penguins)
```

Defining our axes.

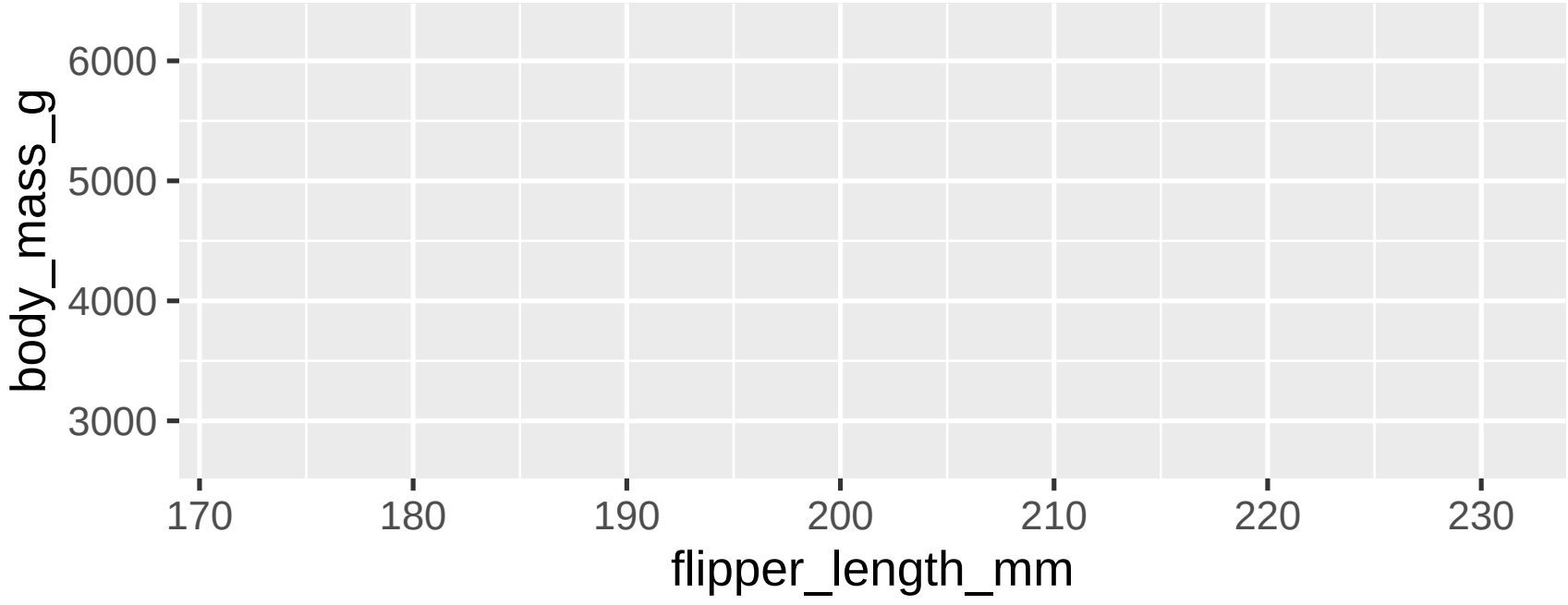
- Suppose I want to compare flipper length to body mass
- ggplot() syntax looks like `ggplot(data, mapping = aes(xvarname,yvarname))`
- the `mapping` argument is **always** defined using an `aes()` command.
- What would I put in `aes()` to get my axes ready?

```
ggplot(penguins, mapping = aes())
```

The axes defined

- Note that now we have these tick marks and lines since we've told ggplot what our axes are.
- Next, we want to have ggplot add dots to our graph. What do we do?

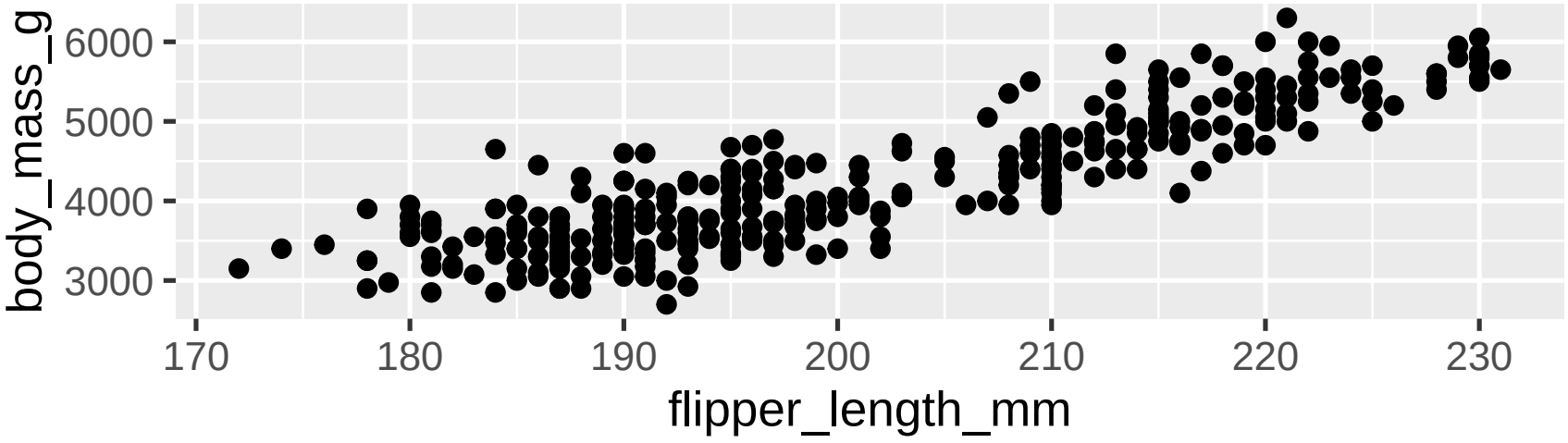
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g))
```



Adding the feature we want to display

- Now we have dots on our graph, representing pairs of values for each observation of penguin.
- Take a moment to see if you can color the dots by species.
  - two ways: inside the initial `aes()`, add `color=species` OR
  - inside `geom_point()`, add an `aes()` with `color=species` in it
  - the first way is better! I'll show you why soon.

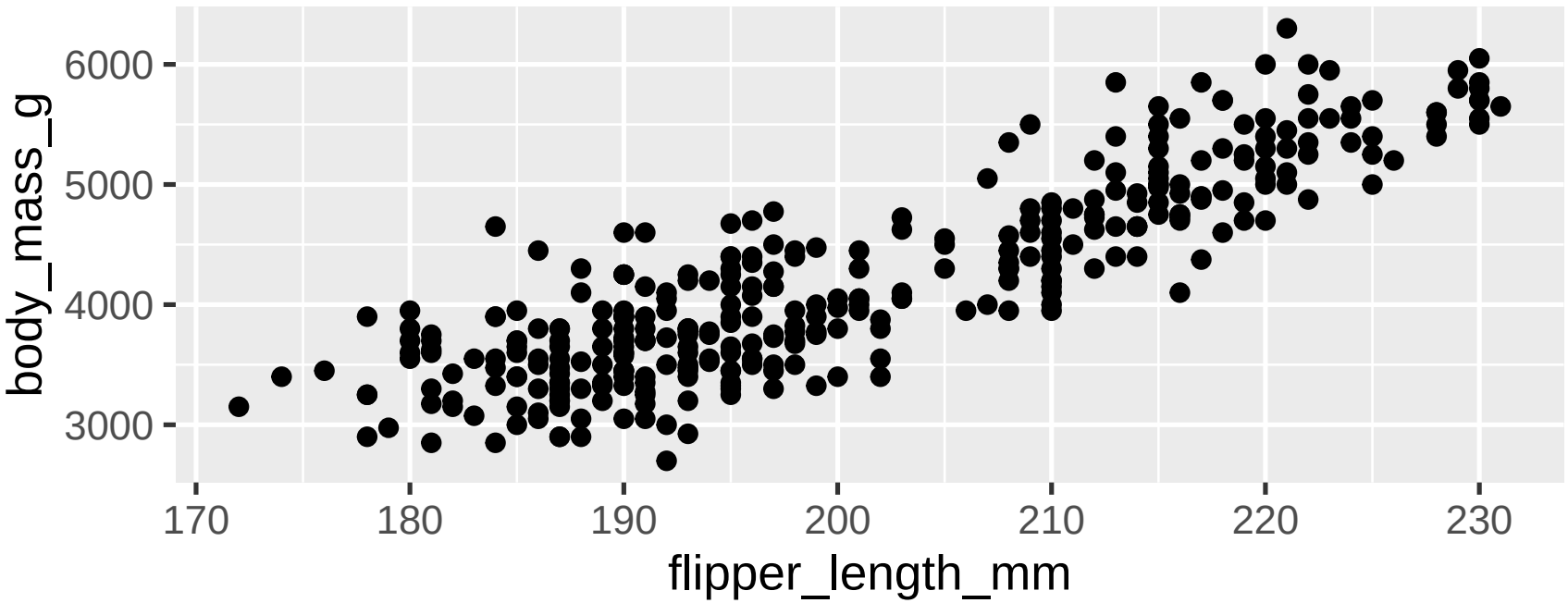
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g)) + geom_point()
```



A sidenote about ggplot syntax

- Note that this line below will not color them.
- I did not need to put x = and y = inside `aes()` because the default first argument is x, and the default second argument is y.
- **There is no default third argument in `aes()`, so you must say `color=species` for it to work

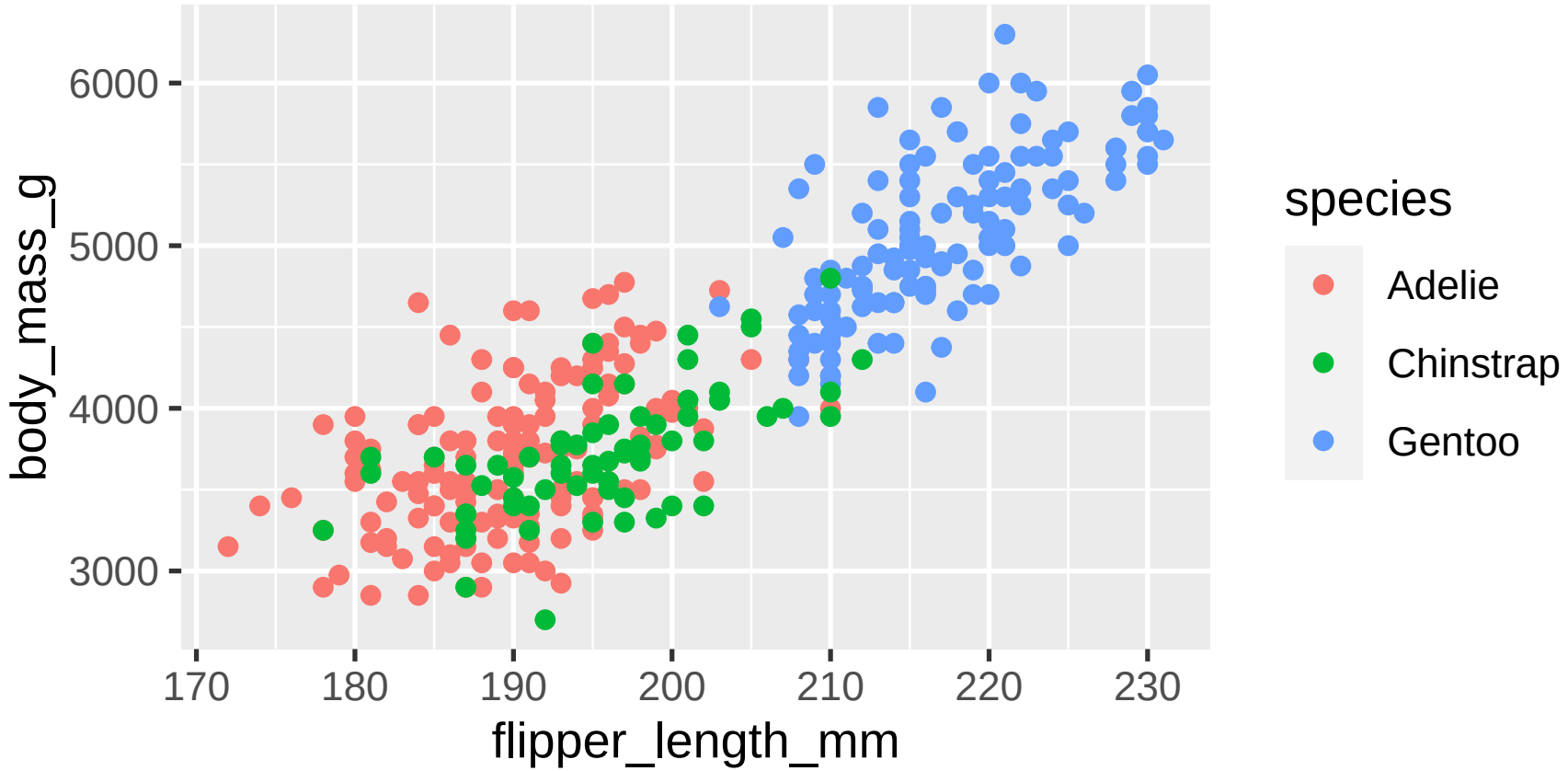
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g, species)) + geom_point()
```



Two ways of making the same plot

- The same point applies to the second line here.
- If you do `geom_point(aes(species))` you will get an error.

```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g, color = species)) +
    geom_point()
```



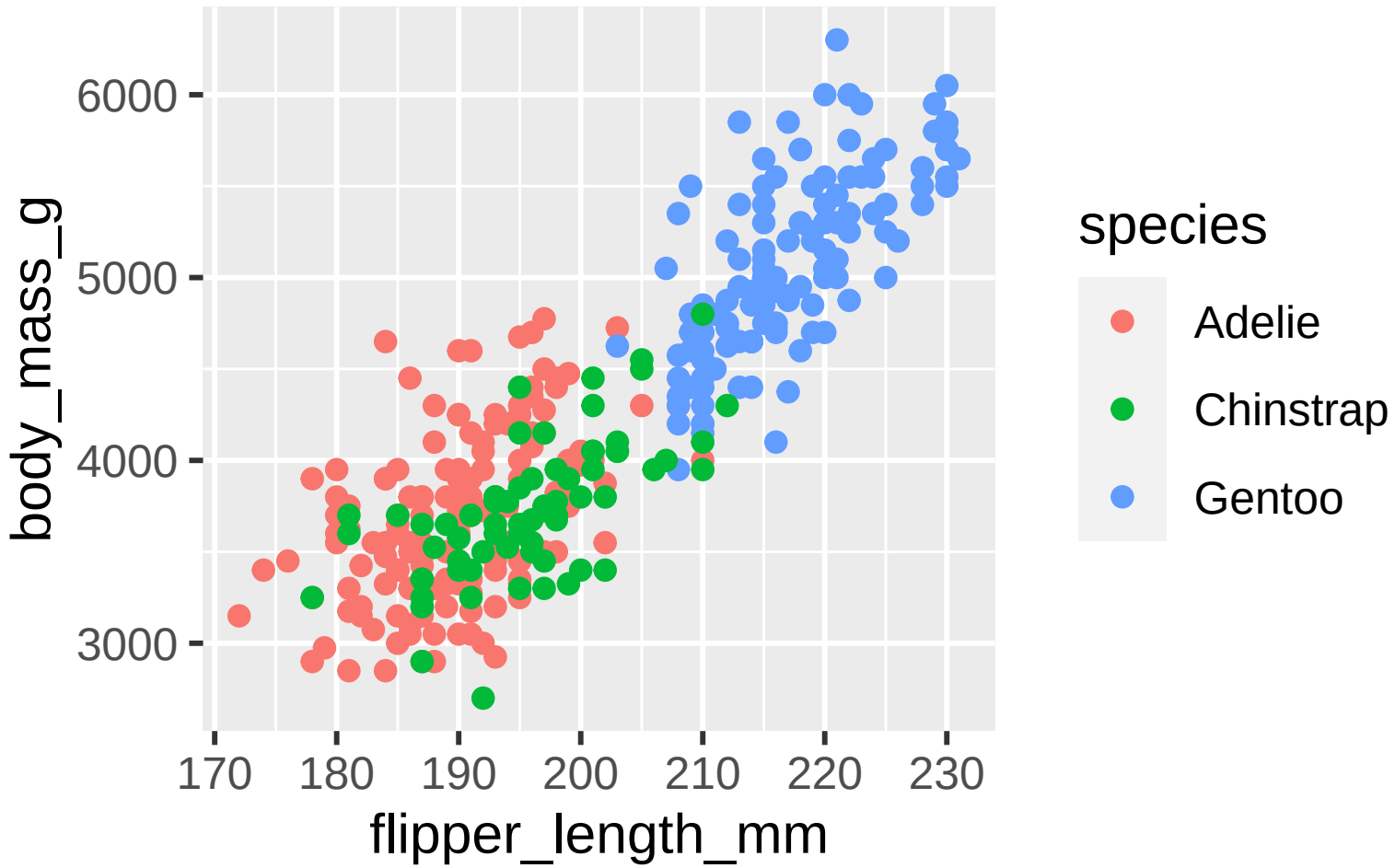
What is going on when ggplot reads your color variable?

- When a categorical variable is mapped to an aesthetic, `ggplot` will automatically assign a unique value to each factor within the category.
- This is called "scaling".

Let's add UNIQUE trendlines for each species.

- What command will I add to this line? Try it out.

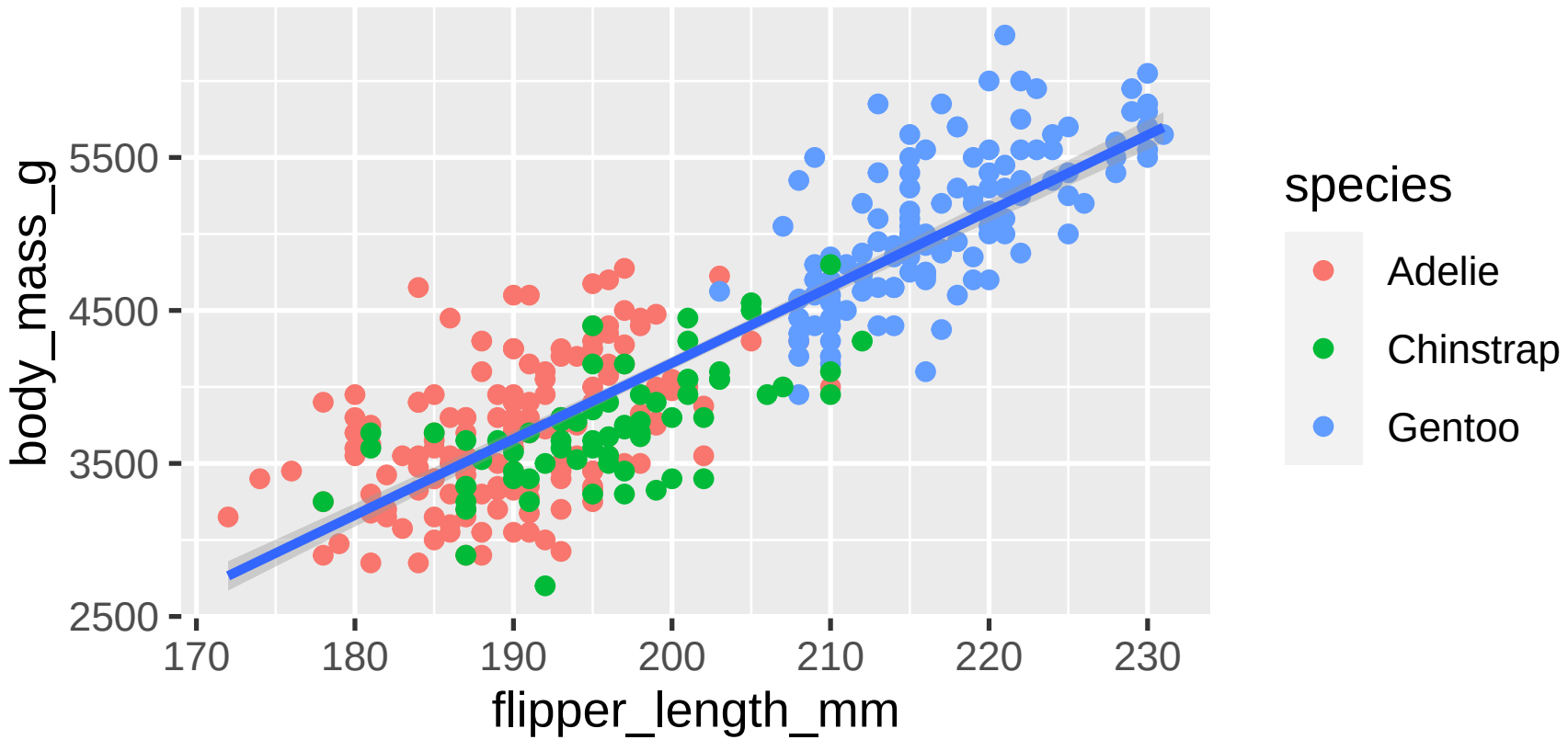
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g, color = species)) +
    geom_point()
```



geom_smooth(method="lm")

- I add a line and tell it to use the method "lm".
- This creates a linear model with Y as the dependent variable and X as the independent variable.

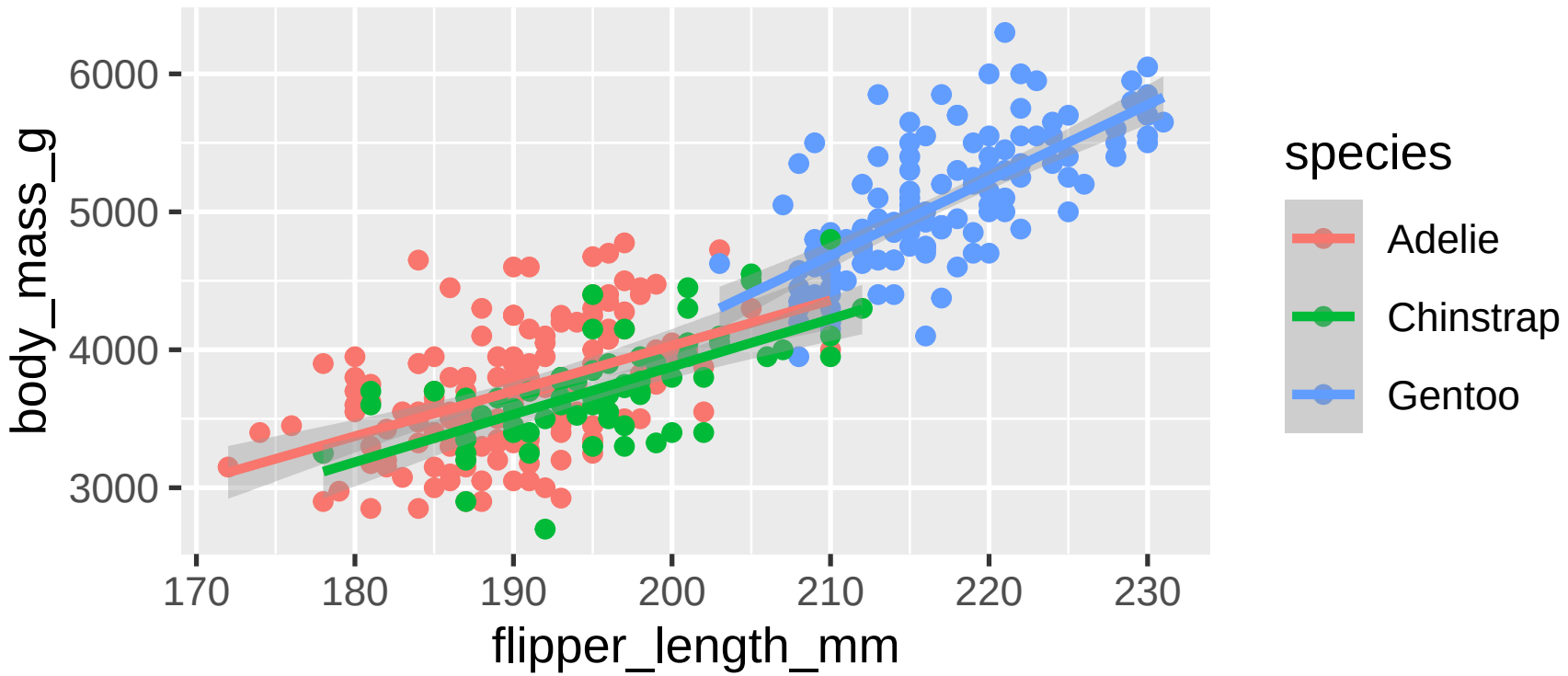
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g)) + geom_point(aes(color = species)) +
    geom_smooth(method = "lm")
```



The problem with grouping within your "geoms"

- `geom_smooth()` operates independently of `geom_point()`, it just reads the options in `ggplot()`
- If I do my scaling inside `geom_point()`, `geom_smooth()` will just smooth out all the coordinates it reads in `ggplot()`

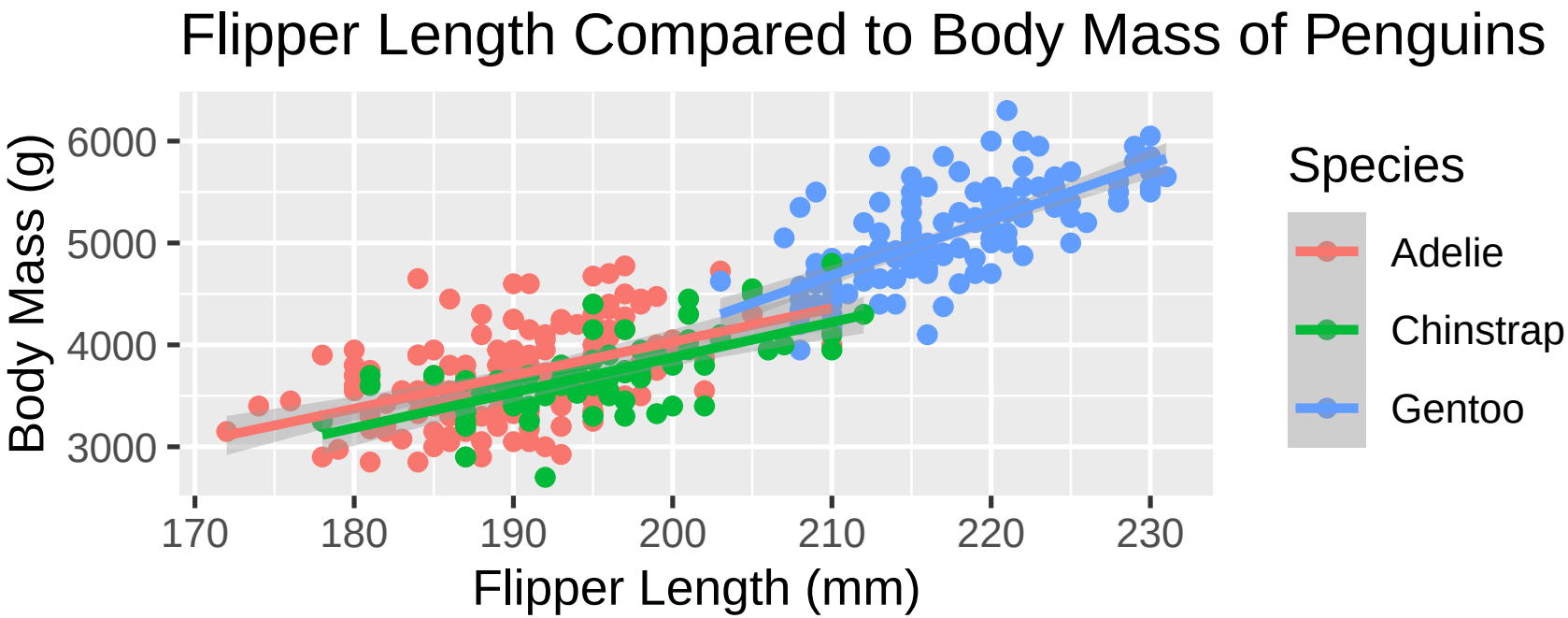
```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g, color = species)) +
    geom_point() + geom_smooth(method = "lm")
```



Proper labelling of graphs

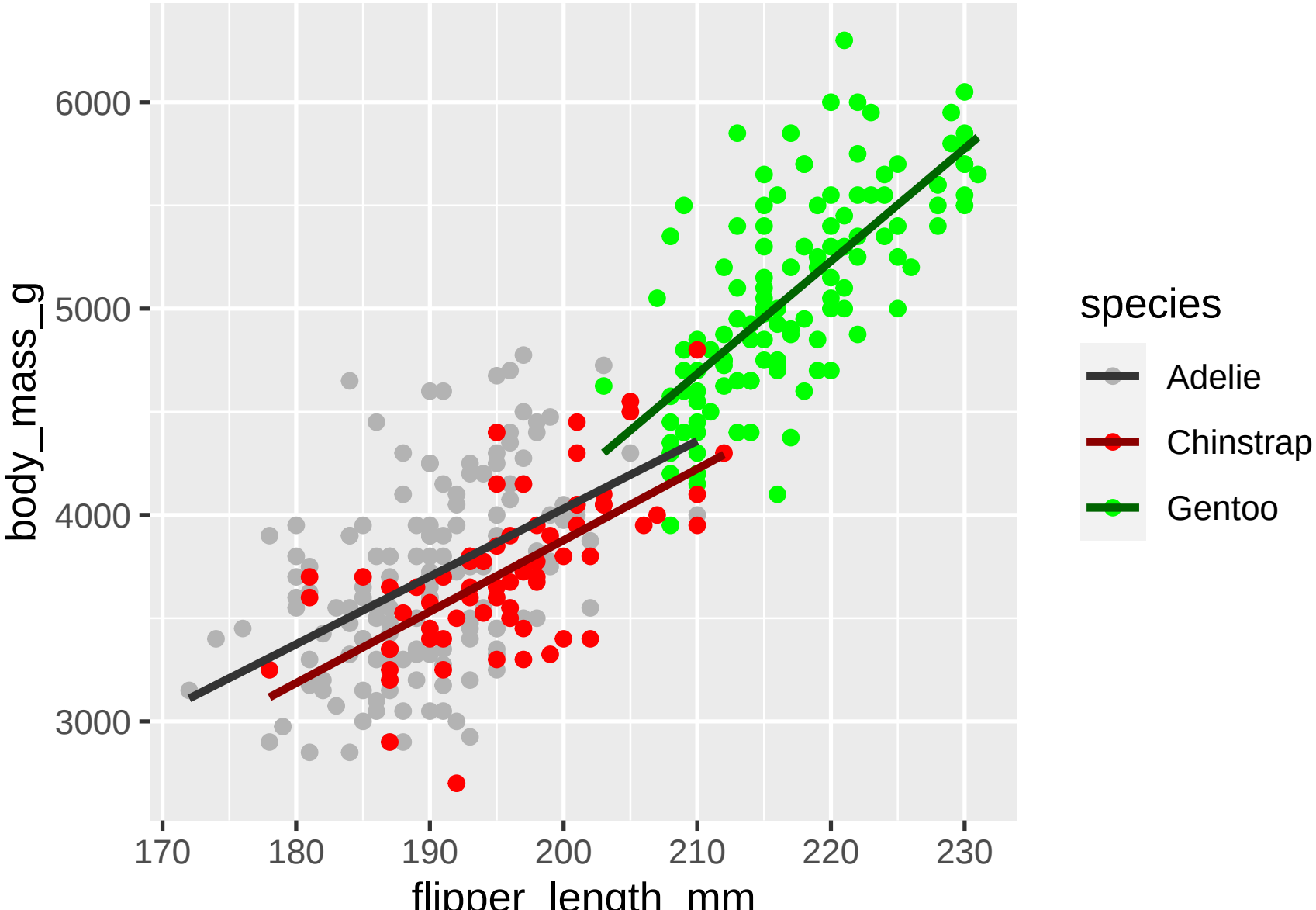
- within a labs() function, I can define the names of all of our aesthetics [aes()]

```
ggplot(penguins, mapping = aes(flipper_length_mm,
    body_mass_g, color = species)) +
    geom_point() + geom_smooth(method = "lm") +
    labs(x = "Flipper Length (mm)",
        y = "Body Mass (g)", title = "Flipper Length Compar
        color = "Species")
```



We can even define where the legend goes

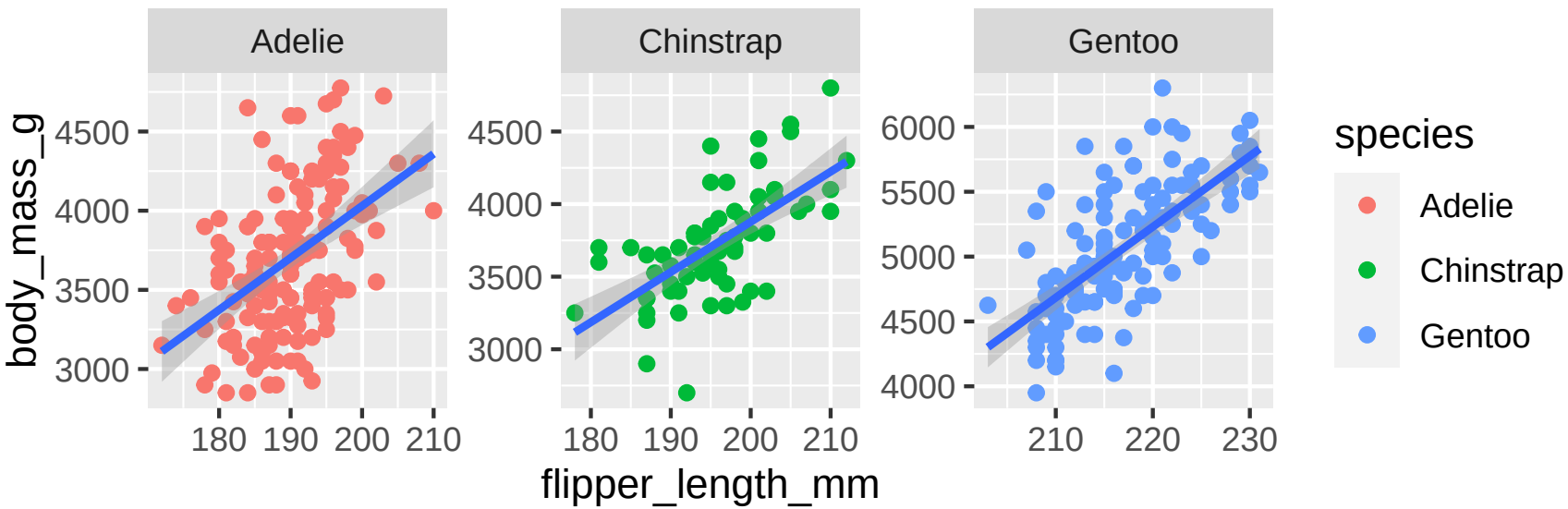
- `+theme(legend.position = c(0.15, 0.8))` start the legend 15% right and 80% up in the plot



Faceting with the Penguins Dataset

-`facet_wrap()` is used to create separate plots for each level of the `species` variable. - The `~` symbol indicates that we're using the `species` variable for faceting. - `scales = "free"` allows each facet to have its own scale, making comparisons easier.

```
ggplot(penguins, aes(x = flipper_length_mm,
    y = body_mass_g)) + geom_point(aes(color = species)) +
    geom_smooth(method = "lm") + facet_wrap(~species,
    scales = "free")
```



Making an interactive plot

```
p <- ggplot(penguins, aes(x = flipper_length_mm,
    y = body_mass_g, color = species)) +
    geom_point() + geom_smooth(method = "lm")
theme_minimal()
```

```
## List of 97
##  $ line                           :List of 6
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "eleme
##  $ rect                           :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ linewidth    : num 0.5
##   ..$ linetype     : num 1
```

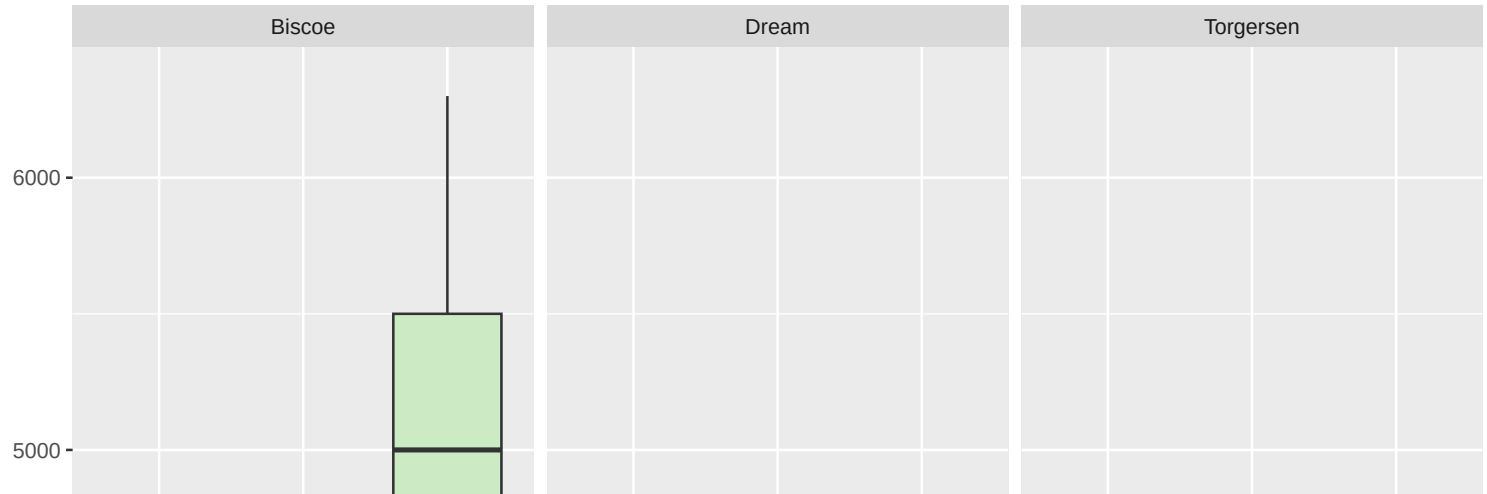
Exercise: Penguin Body Mass Exploration

Let's dive deeper into the `palmerpenguins` dataset by creating a visualization that explores the body mass of penguins across different species and islands.

Instructions:

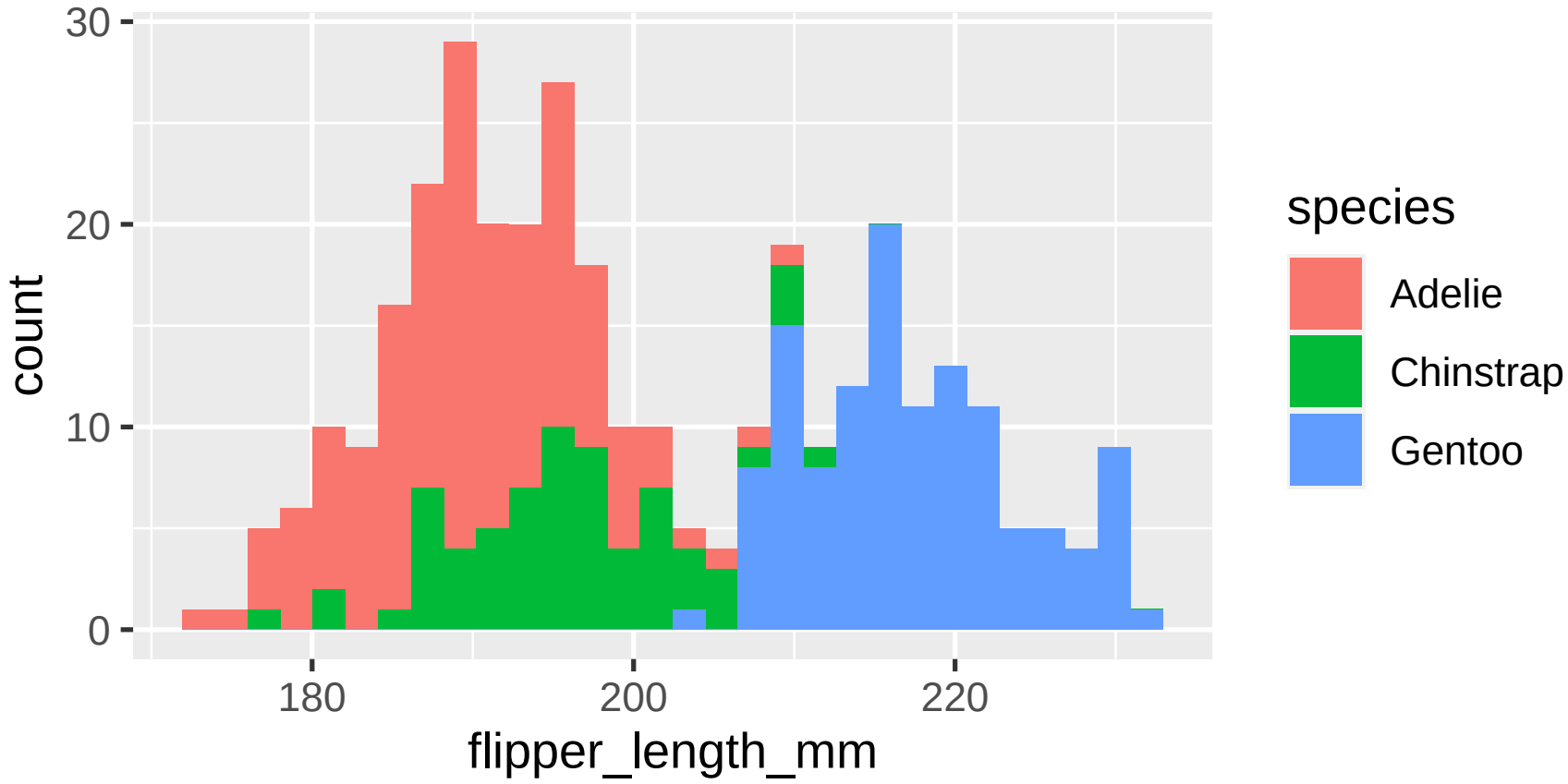
- Utilize the dataset to compare body mass across species and islands.
  - Create a boxplot of body mass for each species faceted across the three islands.

```
ggplot(penguins, aes(x = species, y = body_mass_g,
    fill = species)) + geom_boxplot() +
    facet_wrap(~island) + scale_fill_brewer(palette = "Past
```



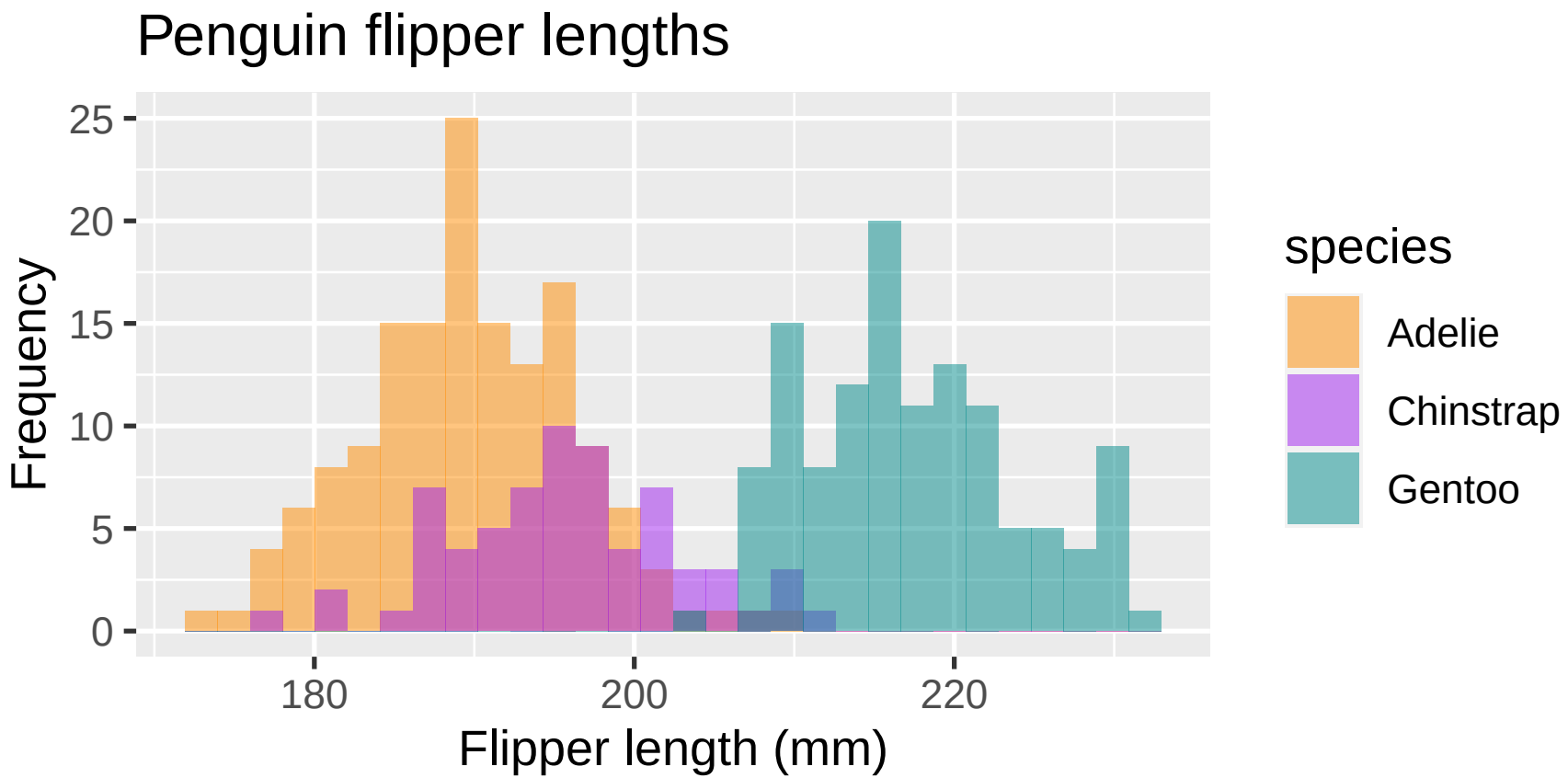
Histograms

```
ggplot(data = penguins, aes(flipper_length_mm)) +
    geom_histogram(aes(fill = species))
```



A more advanced histogram

```
ggplot(data = penguins, aes(flipper_length_mm)) +
    geom_histogram(aes(fill = species),
        alpha = 0.5, position = "identity") +
    scale_fill_manual(values = c("darkorange",
        "purple", "cyan4")) + labs(x = "Flipper length (mm)",
    y = "Frequency", title = "Penguin flipper lengths")
```



Next class

- In-class application where you'll be given a visualization task.
- It will be shorter than last application