R for efficient exploratory data analysis

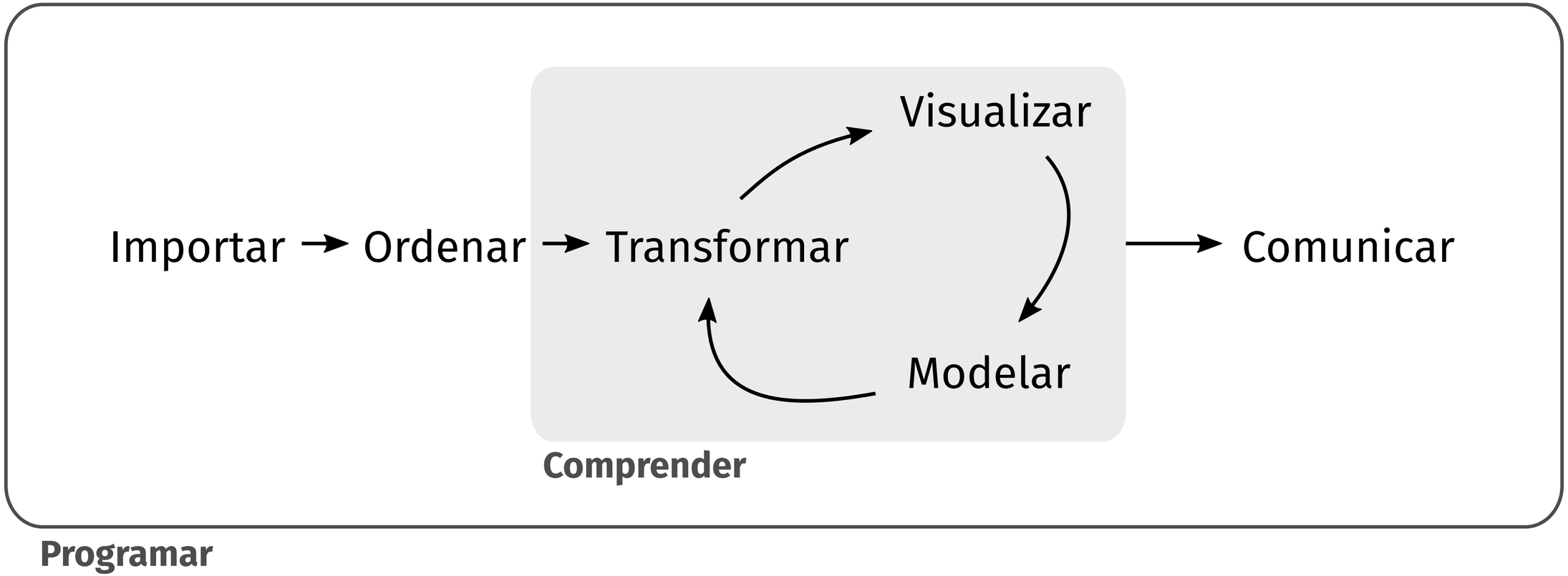
with `tydiverse`

Pablo Villarreal, Ph.D. Assistant Professor.

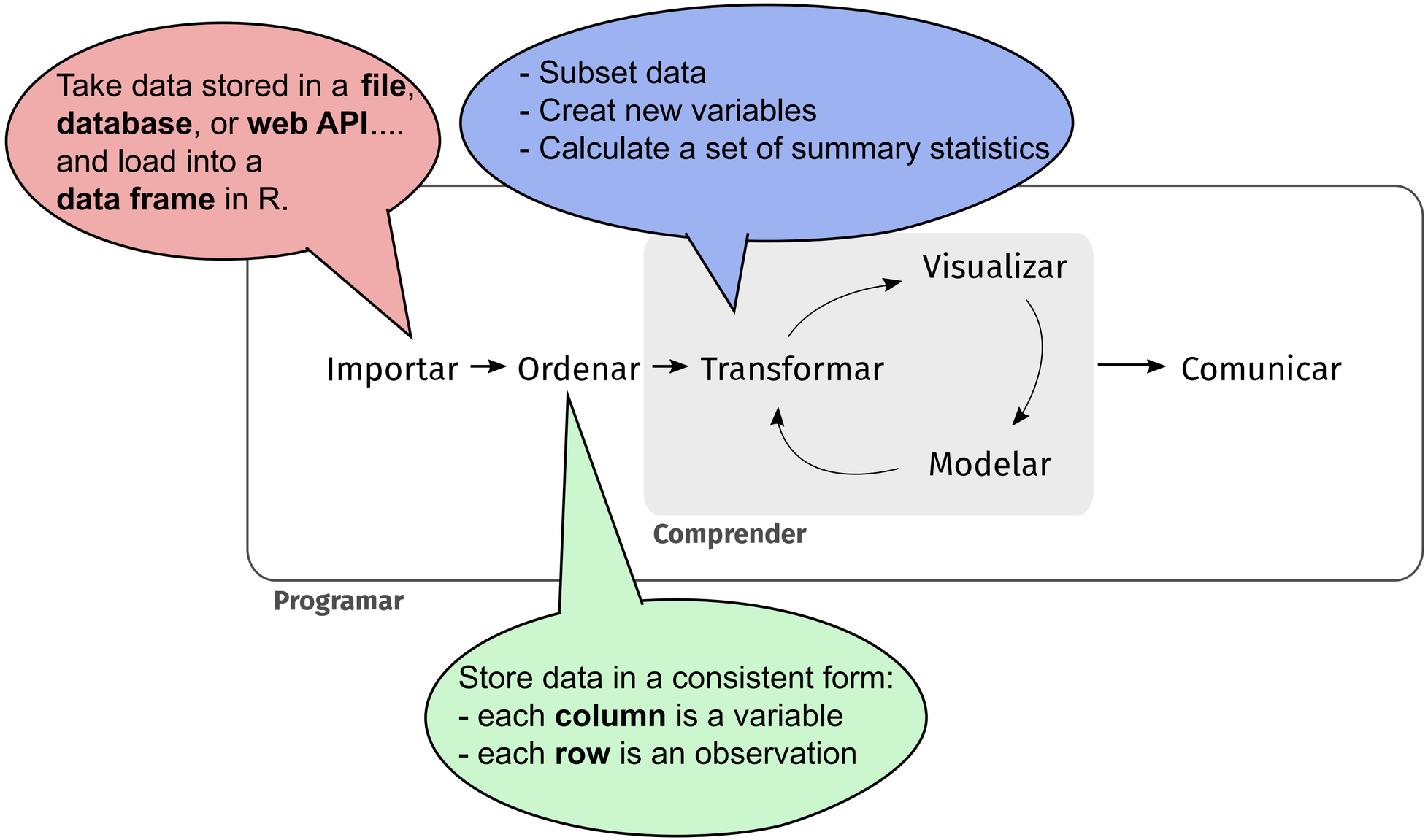
Center for Genomics and Bioinformatics Faculty of Science, Engineering, and Technology. Universidad Mayor

2025

What You Will Learn



What You Will Learn



Where?

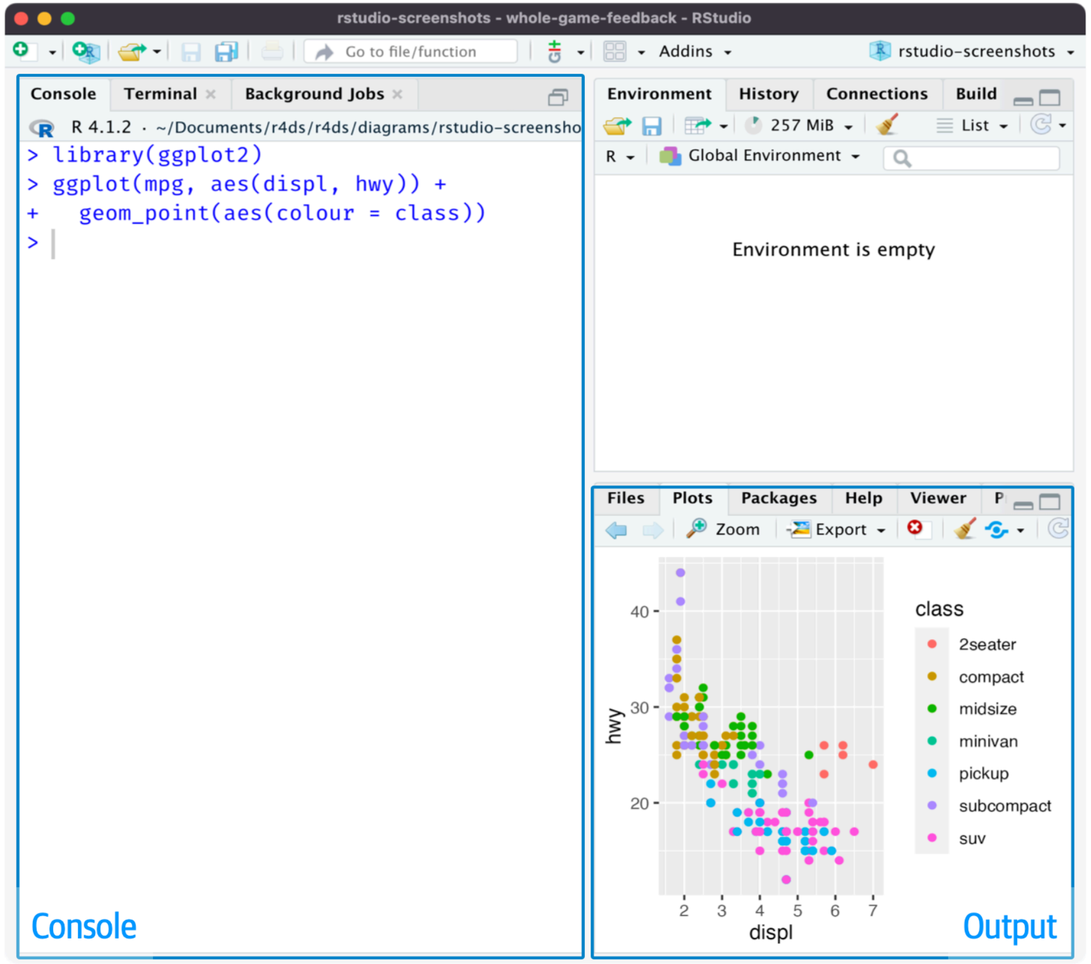
R

> Is **a free, open-source programming language for statistical computing and data visualization**

RStudio

> Is an integrated development environment, or IDE, for R programming, which you can download from https://posit.co/download/rstudio-desktop/

The **RStudio** IDE has two key regions: type R code in the console pane on the left, and look for plots in the output pane on the right.



Packages

An R **package** is a collection of functions, data, and documentation that extends the capabilities of base R.

The tidyverse

You can install the complete tidyverse with a single line of code:

```
1  install.packages("tidyverse")
```

The tidyverse

> **Important**
>
> You will not be able to use the functions, objects, or help files in a package until you load it with `library()`.

```
1 library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Data

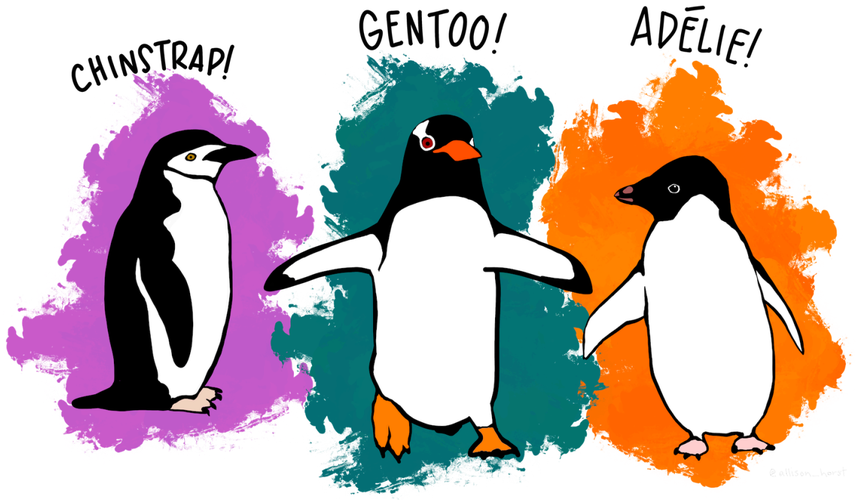
```
1 library(ggthemes)
2 library(gt)
3 library(palmerpenguins)
4
5 penguins
```

```
# A tibble: 344 × 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
 1 Adelie  Torgersen           39.1          18.7               181        3750
 2 Adelie  Torgersen           39.5          17.4               186        3800
 3 Adelie  Torgersen           40.3          18                 195        3250
 4 Adelie  Torgersen           NA            NA                  NA          NA
 5 Adelie  Torgersen           36.7          19.3               193        3450
 6 Adelie  Torgersen           39.3          20.6               190        3650
 7 Adelie  Torgersen           38.9          17.8               181        3625
 8 Adelie  Torgersen           39.2          19.6               195        4675
 9 Adelie  Torgersen           34.1          18.1               193        3475
10 Adelie  Torgersen           42            20.2               190        4250
# ℹ 334 more rows
# ℹ 2 more variables: sex <fct>, year <int>
```

Gorman KB, Williams TD, Fraser WR (2014). Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus

Meet the penguins

> The `penguins` data from the **palmerpenguins** package contains size measurements for 344 penguins from three species observed on three islands in the Palmer Archipelago, Antarctica.



Gorman KB, Williams TD, Fraser WR (2014). Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus

Data

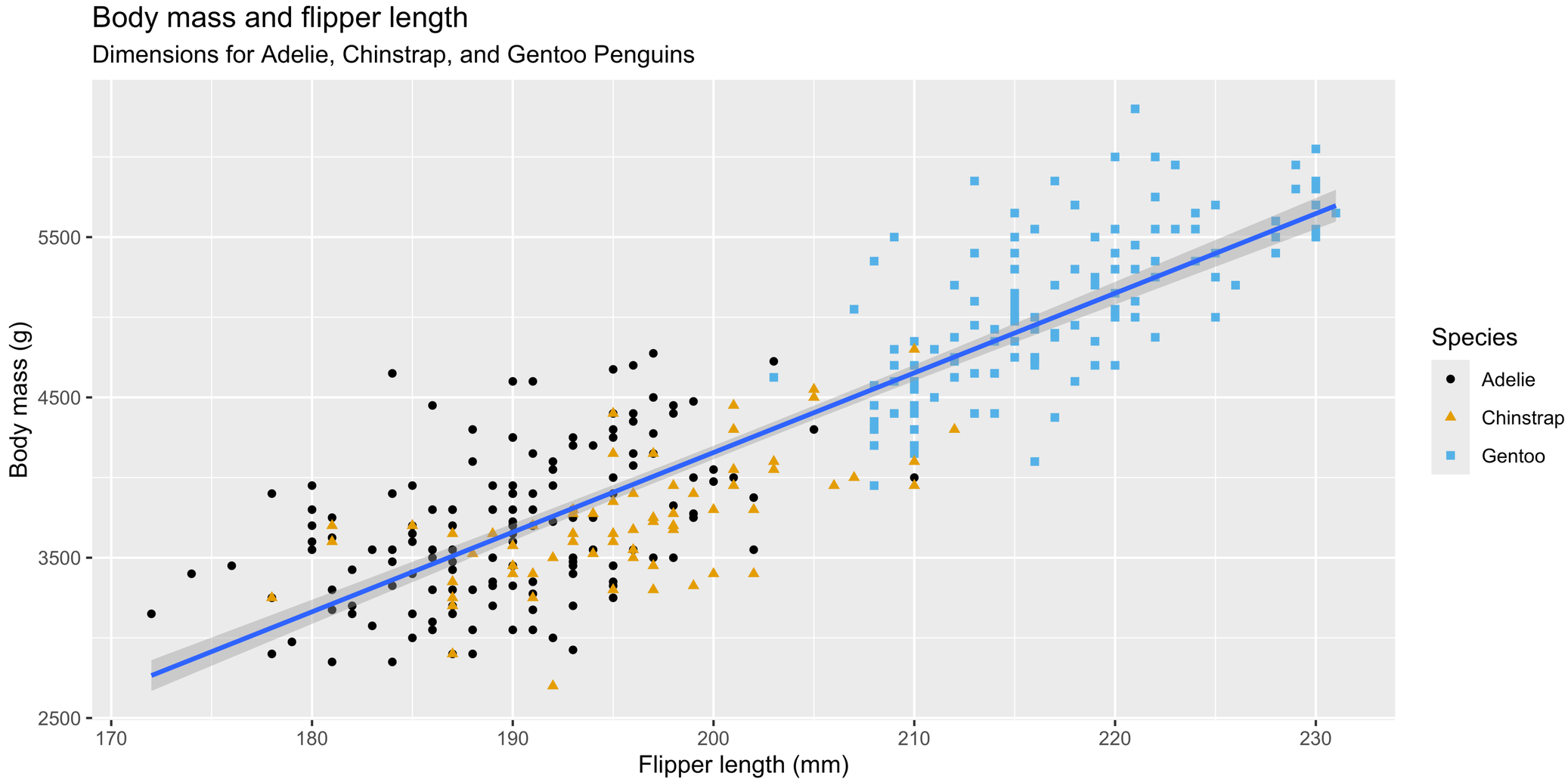
```r
library(knitr)
library(kableExtra)

penguins %>%
  head() %>%
  kable("html") %>%
  kable_styling(font_size = 30, full_width = FALSE)
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007

Ultimate goal

Our ultimate goal in this workshop is to recreate the following visualization displaying the relationship between flipper lengths and body masses of these penguins, taking into consideration the **species** of the penguin.

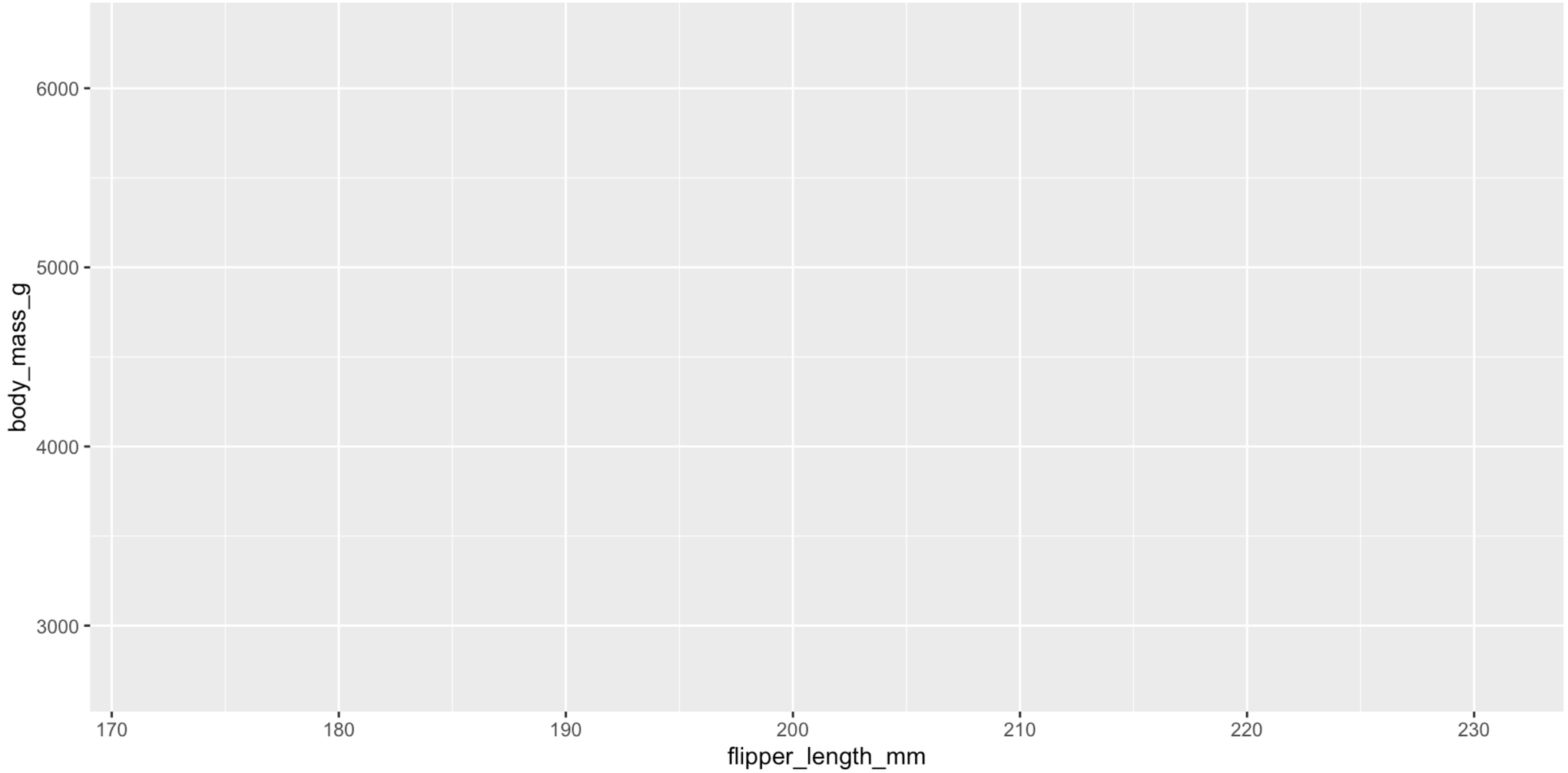


Creating a ggplot

```
1  ggplot(data = penguins)
```

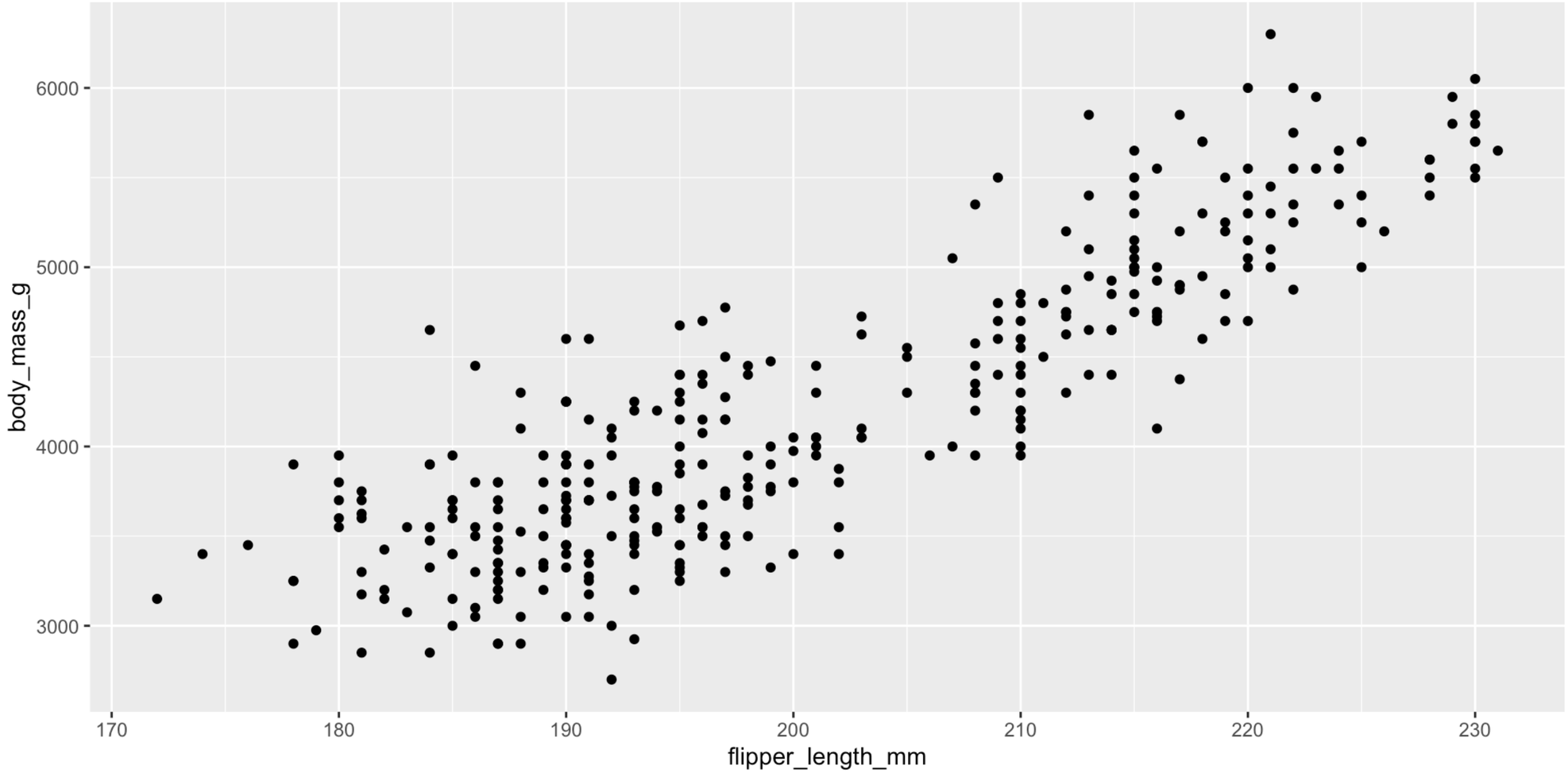
Creating a ggplot

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



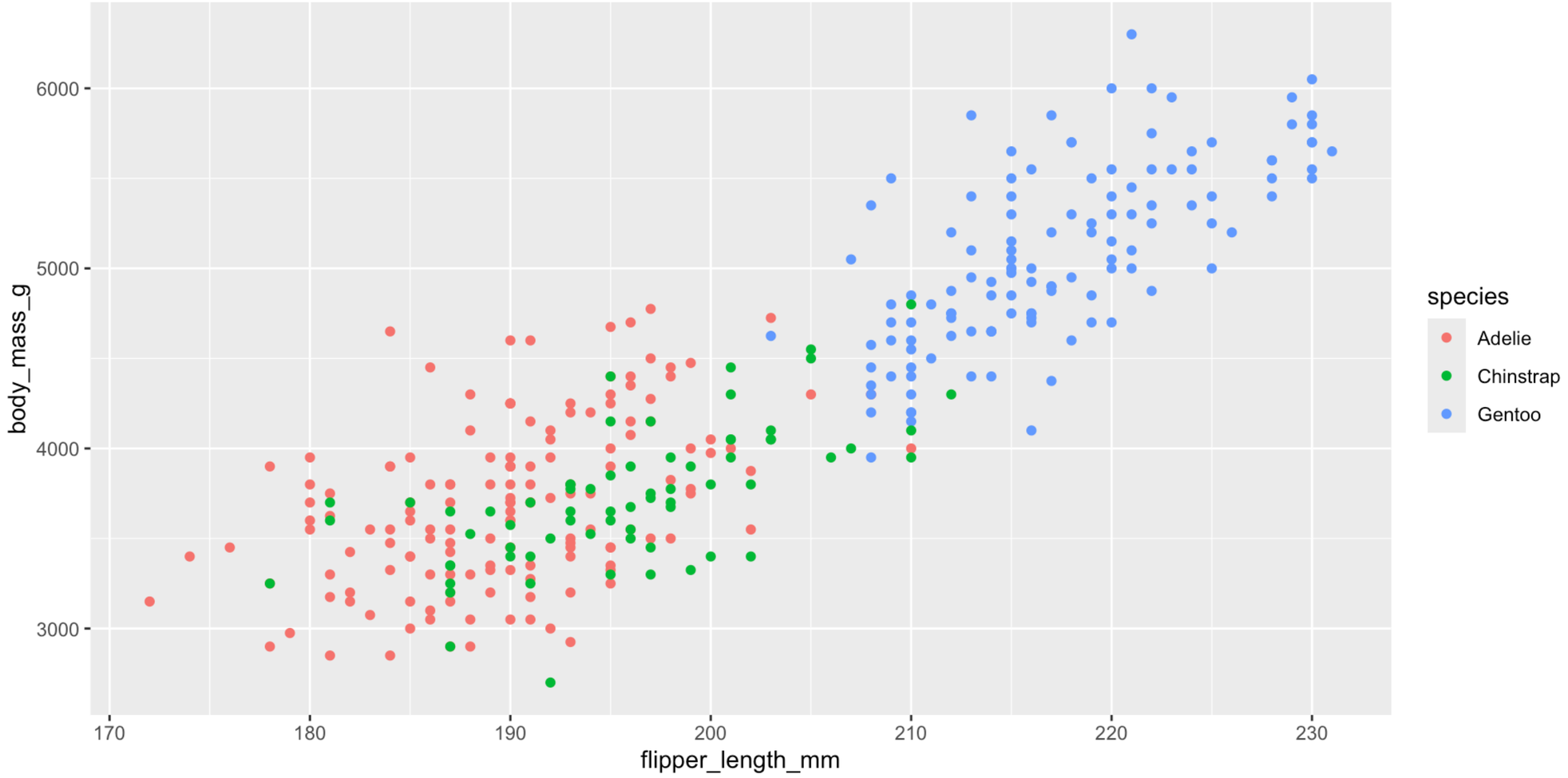
Creating a ggplot

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



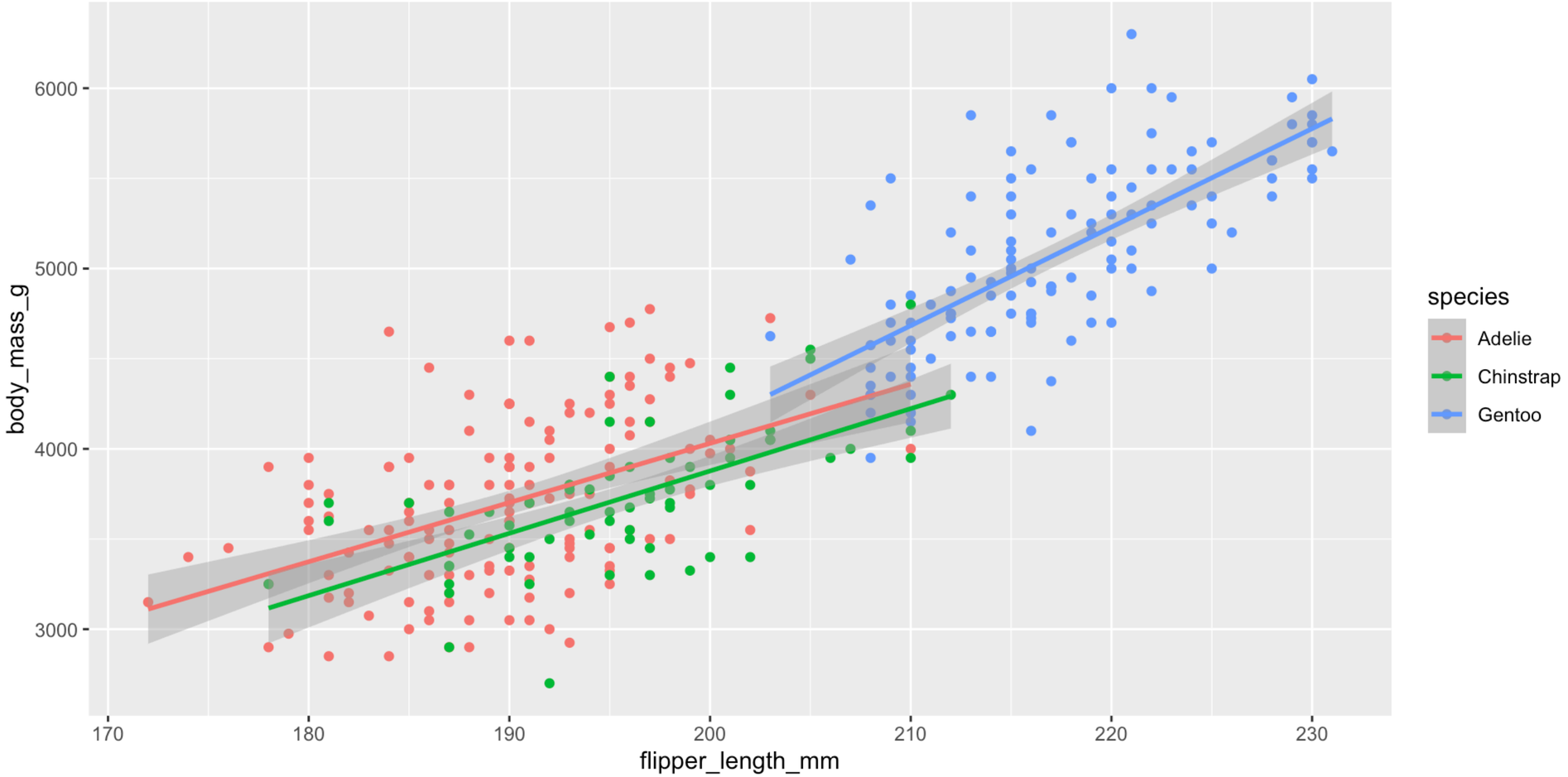
Creating a ggplot

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



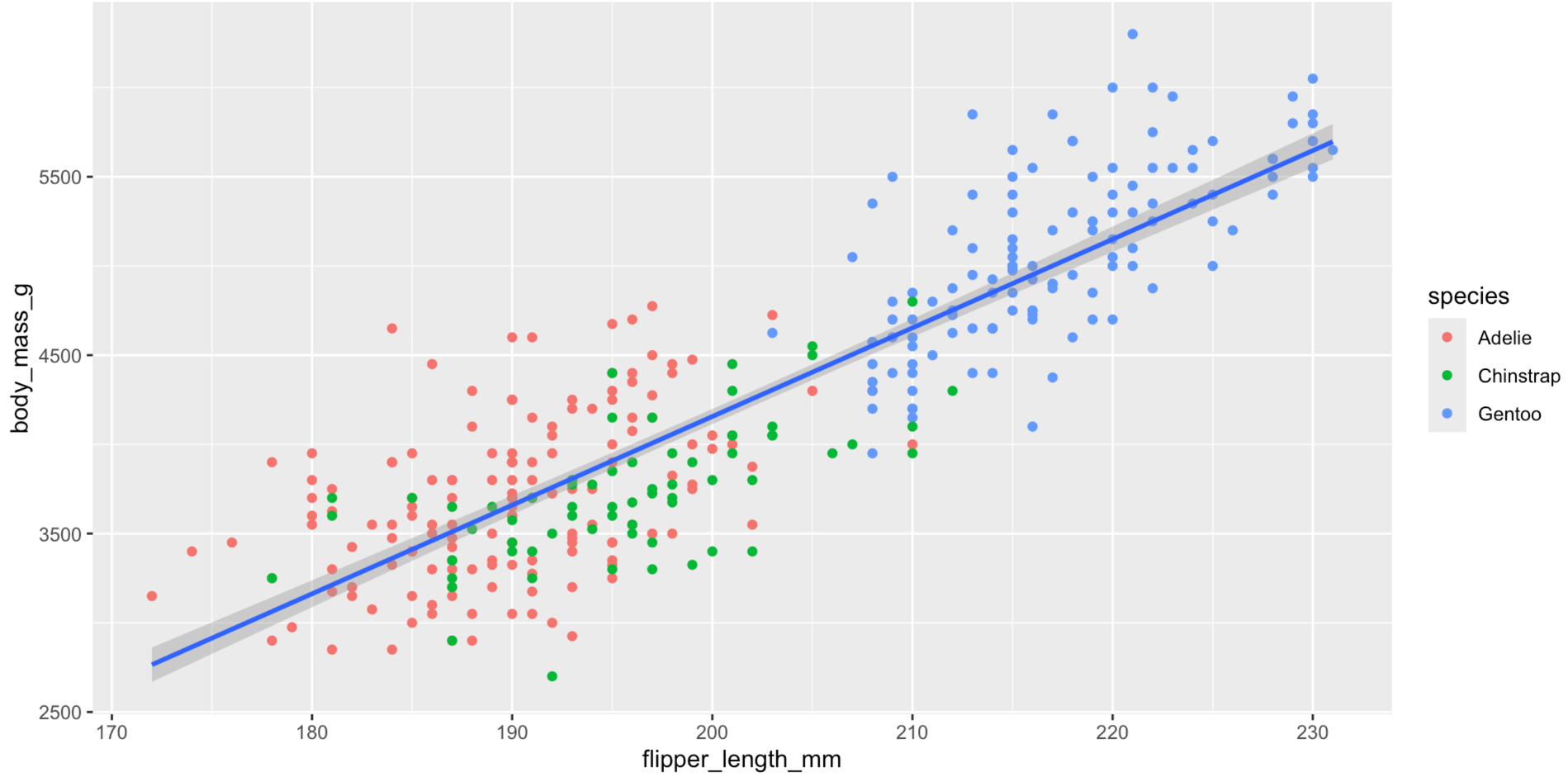
Creating a ggplot

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



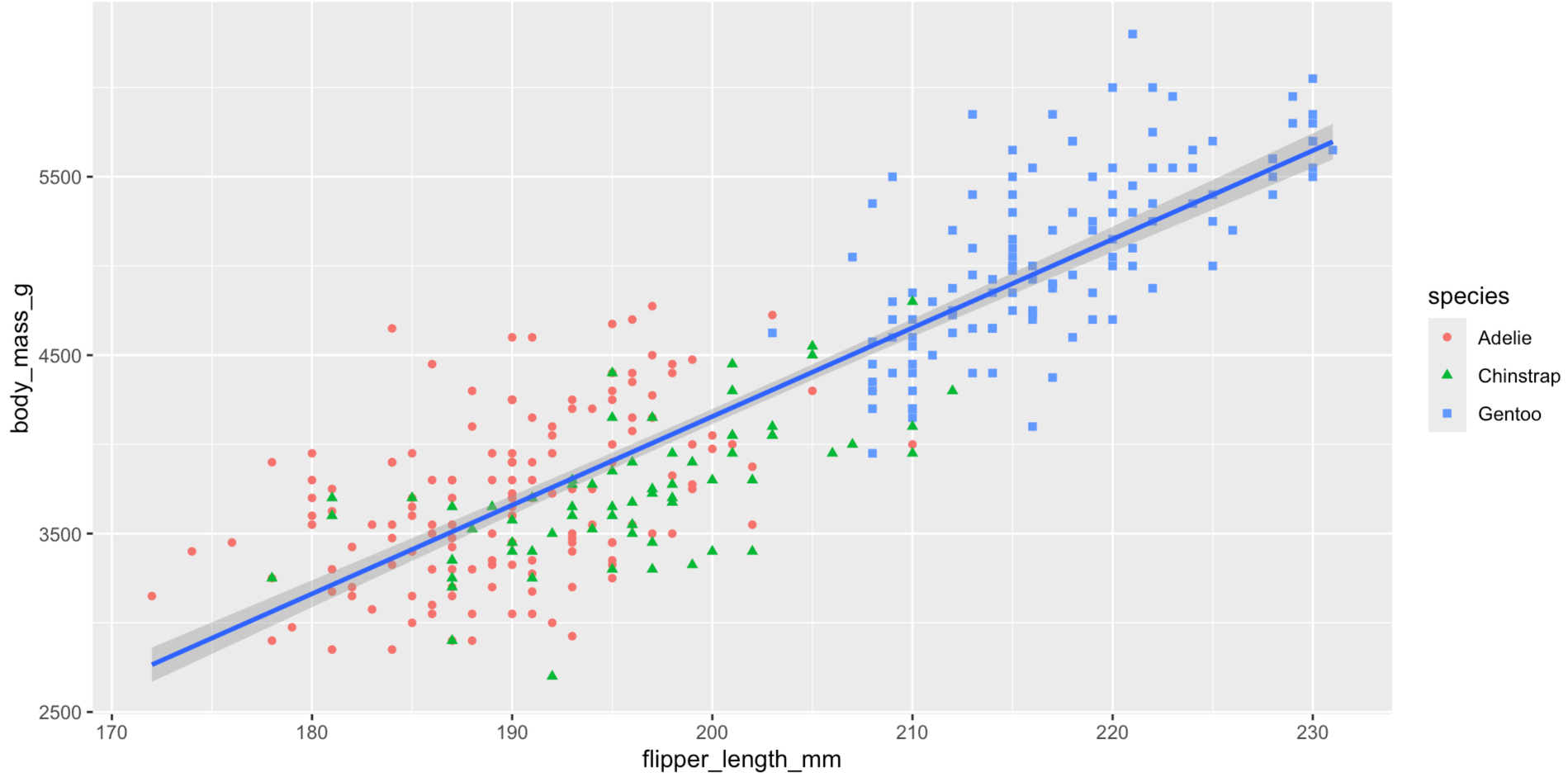
Creating a ggplot

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



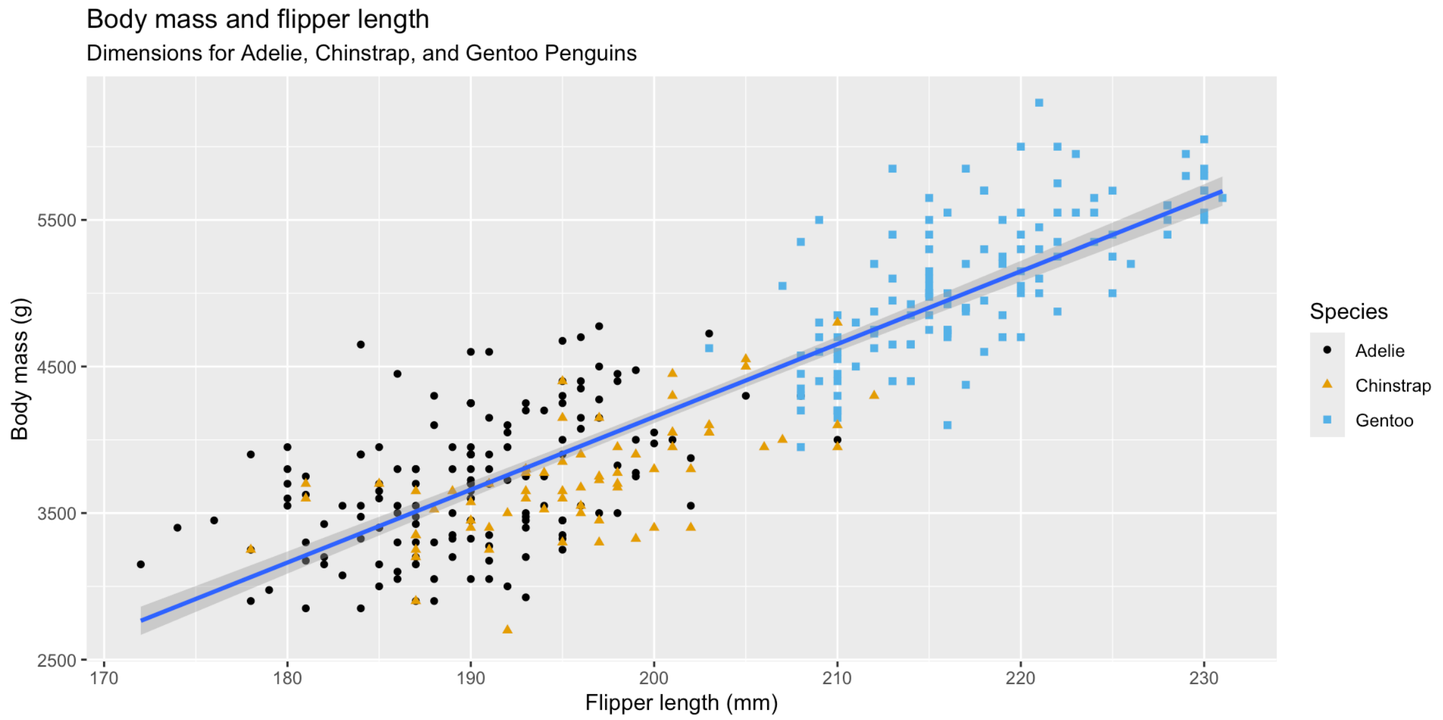
Creating a ggplot

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



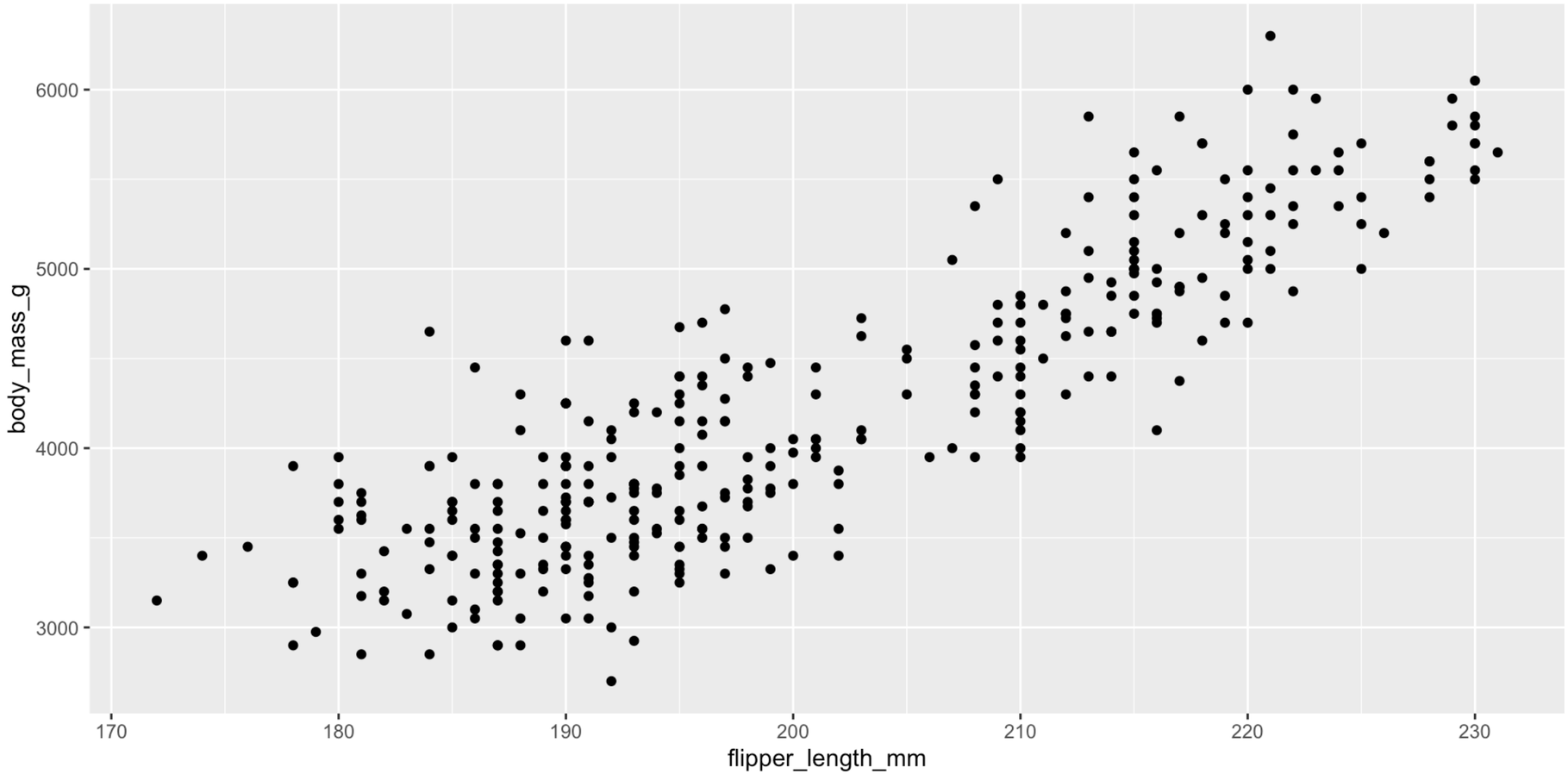
Creating a ggplot

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point(aes(color = species, shape = species)) +
  geom_smooth(method = "lm") +
  labs(
    title = "Body mass and flipper length",
    subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
    x = "Flipper length (mm)", y = "Body mass (g)",
    color = "Species", shape = "Species"
  ) +
  scale_color_colorblind()
```



ggplot2 calls

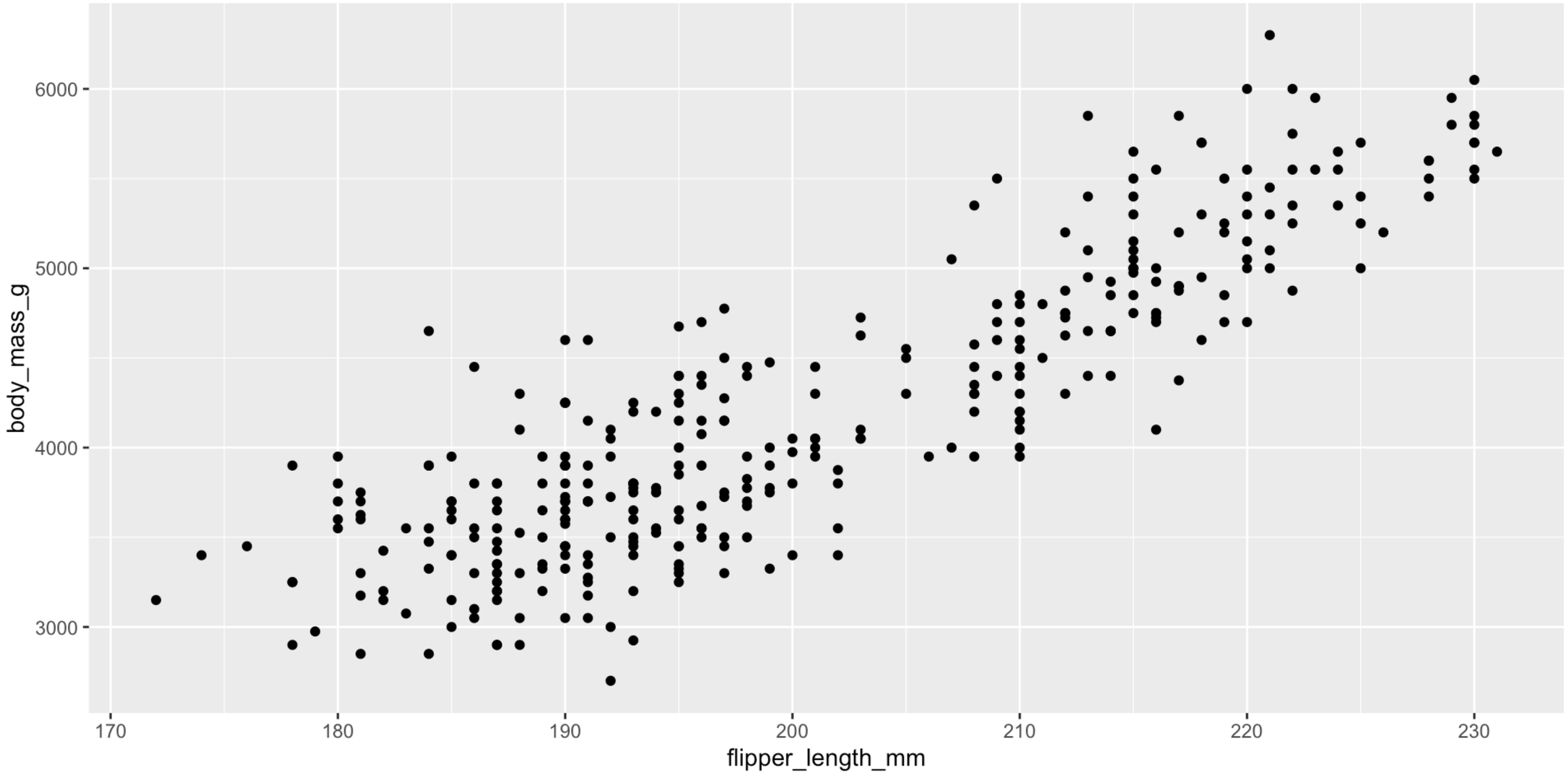
```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point()
```



Rewriting the previous plot more concisely yields:

ggplot2 calls

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

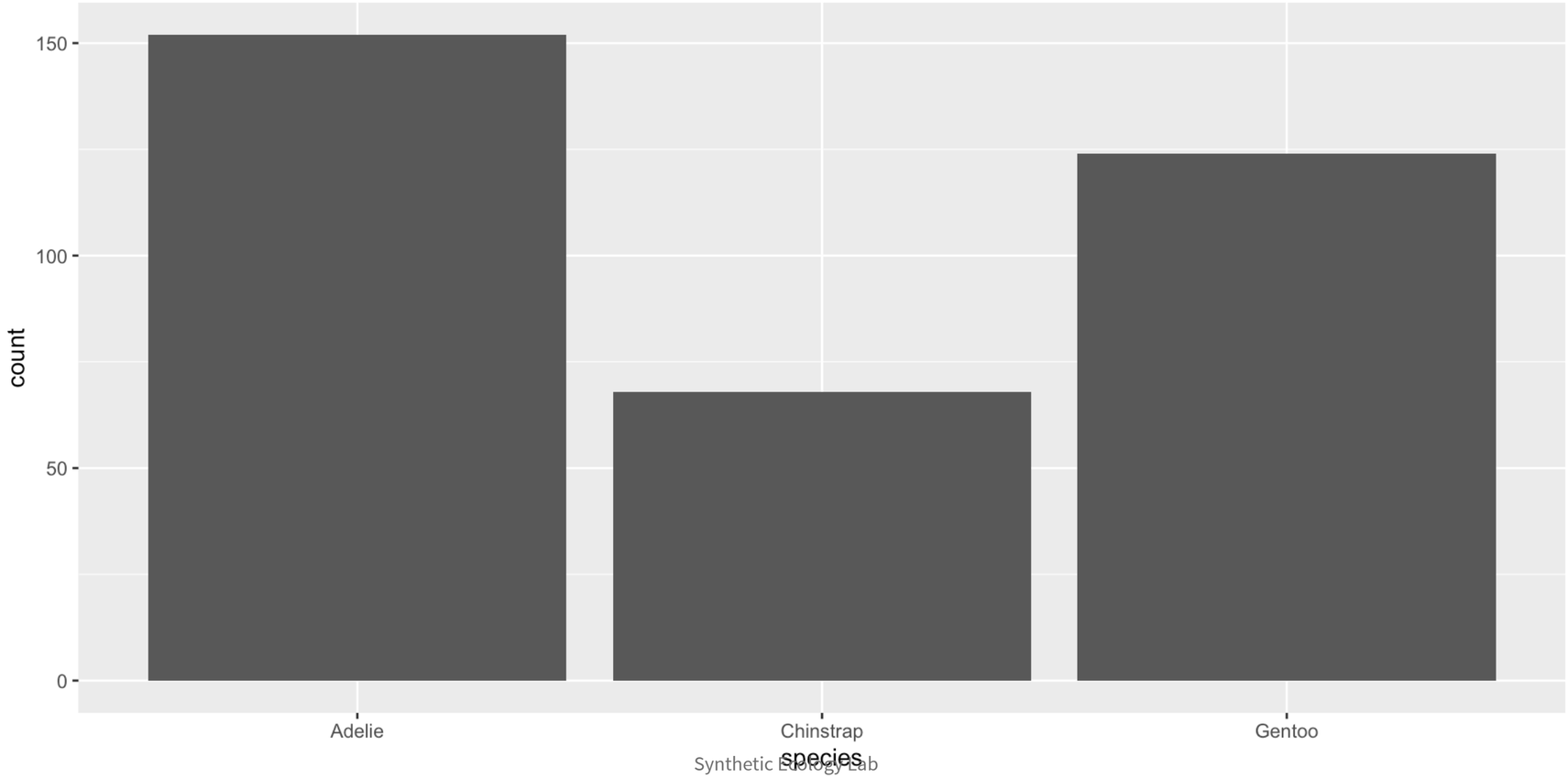


Visualizing distributions

A categorical variable

A variable is categorical if it can only take one of a small set of values. To examine the distribution of a categorical variable, you can use a bar chart. The height of the bars displays how many observations occurred with each x value.

```
1 ggplot(penguins, aes(x = species)) +
2   geom_bar()
```

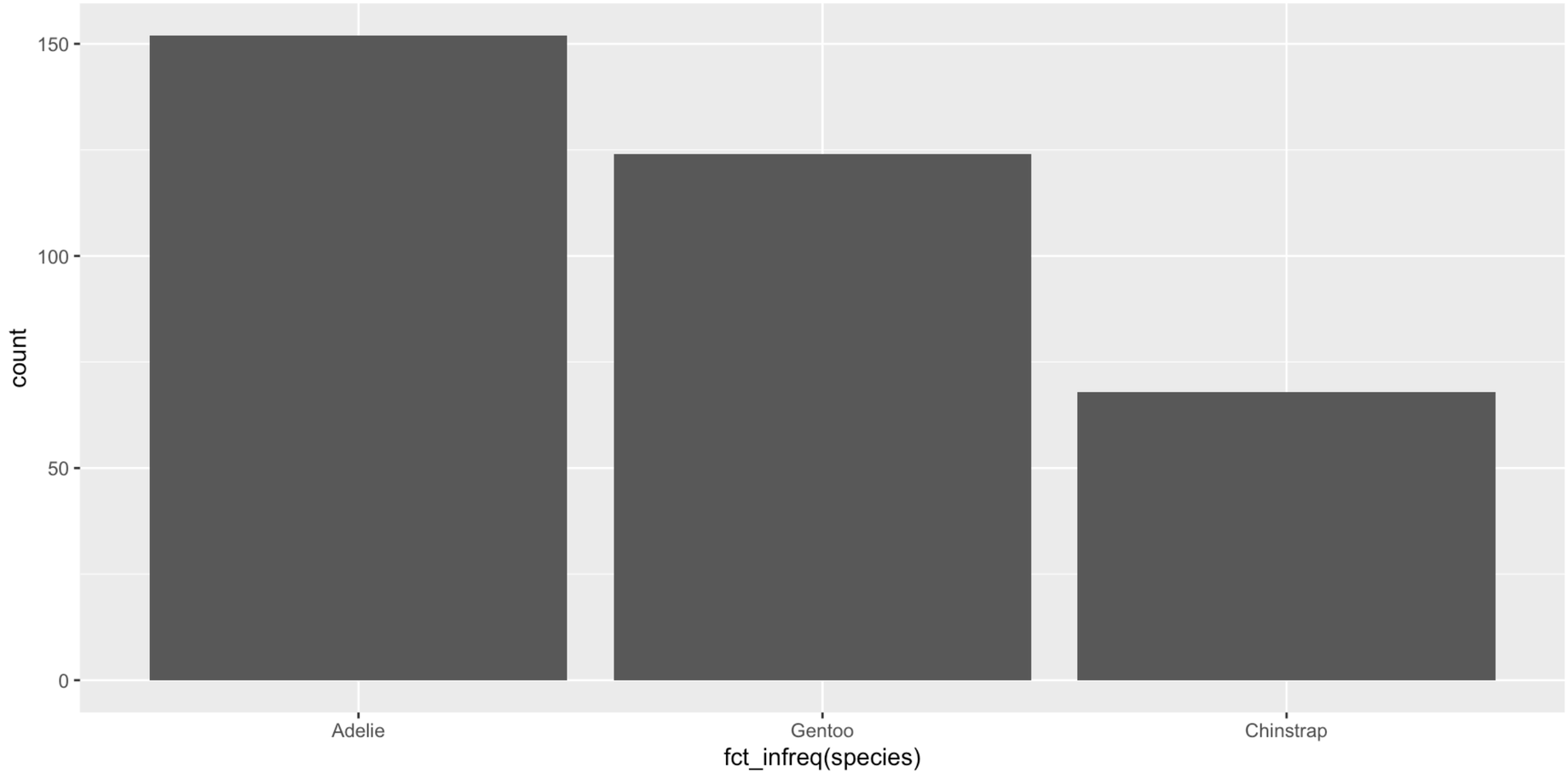


Visualizing distributions

A categorical variable

A variable is categorical if it can only take one of a small set of values.

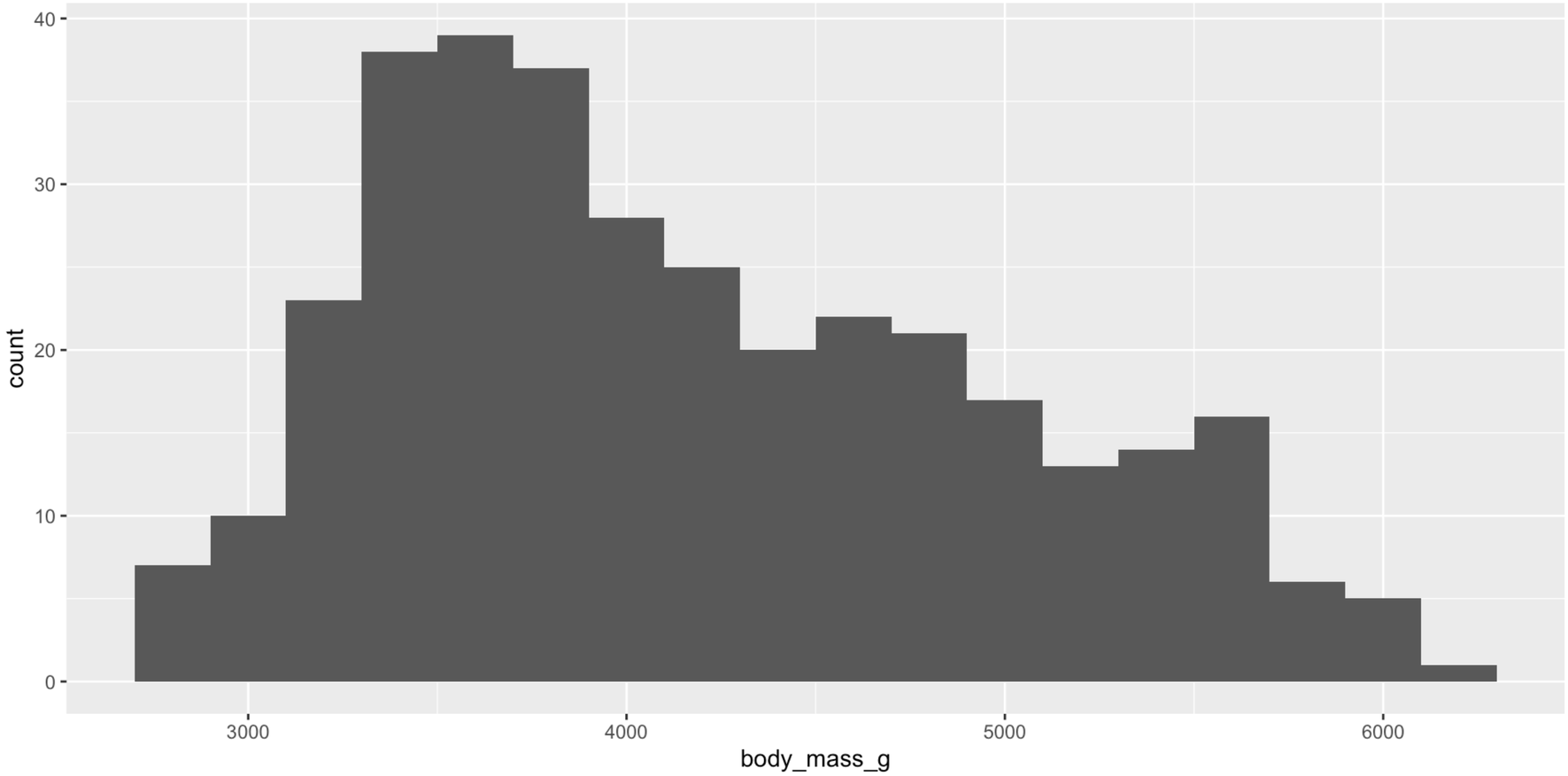
```
1 ggplot(penguins, aes(x = fct_infreq(species))) +
2   geom_bar()
```



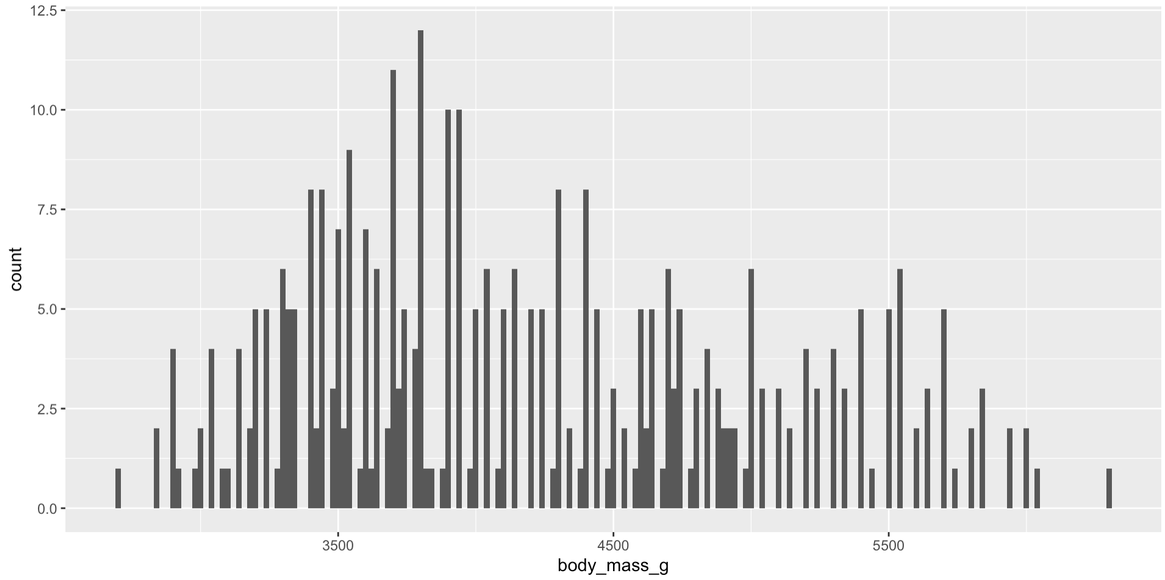
A numerical variable

A variable is numerical (or quantitative) if it can take on a wide range of numerical values, and it is sensible to add, subtract, or take averages with those values. Numerical variables can be continuous or discrete. One commonly used visualization for distributions of continuous variables is a histogram.

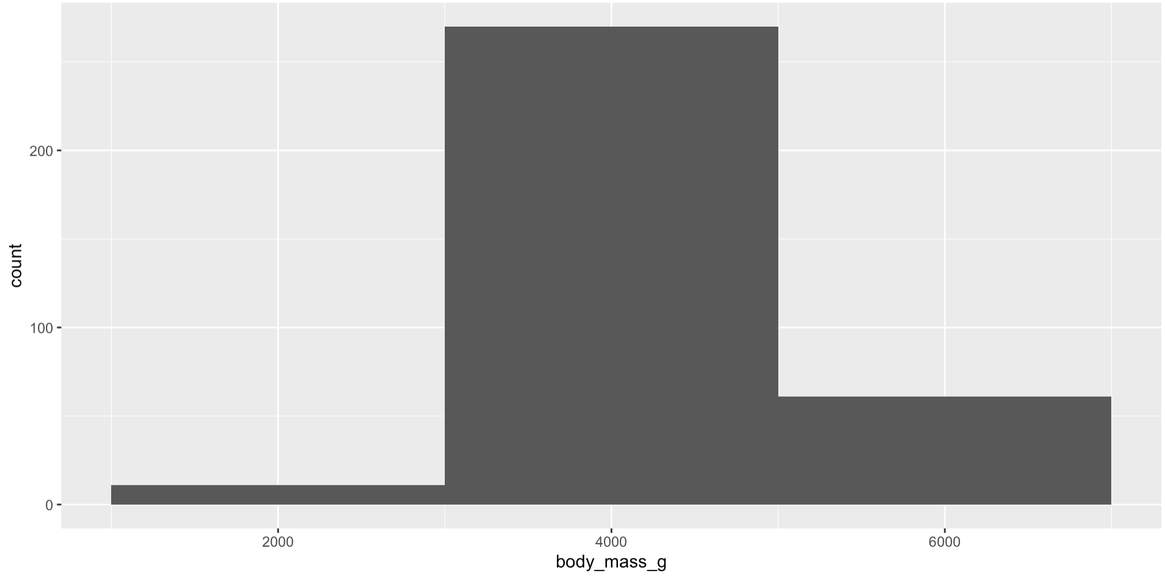
```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 200)
```



```r
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 20)
```

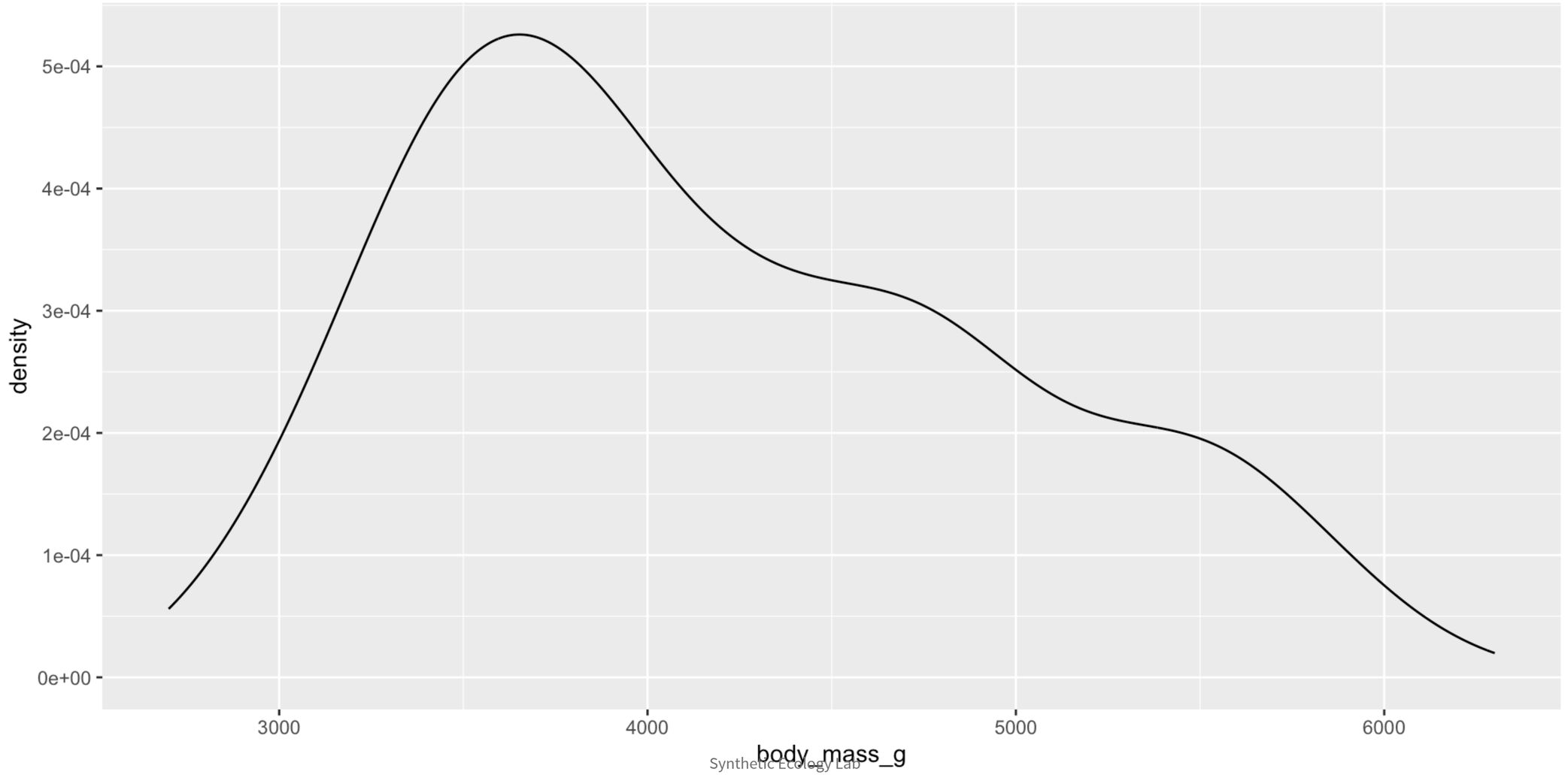



```r
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 2000)
```


count
200
100
0
2000
4000
6000
body_mass_g

An alternative visualization for distributions of numerical variables is a density plot. A density plot is a smoothed-out version of a histogram and a practical alternative, particularly for continuous data that comes from an underlying smooth distribution.

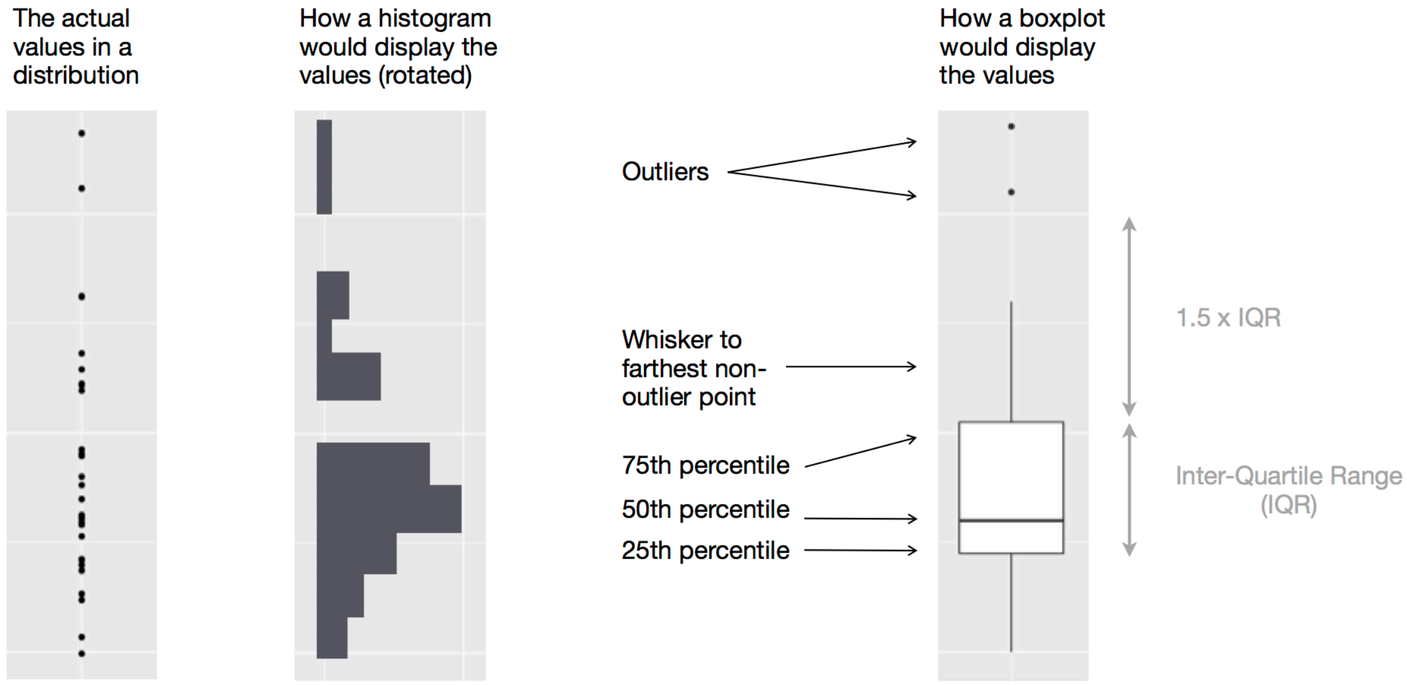
```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_density()
```



Visualizing relationships

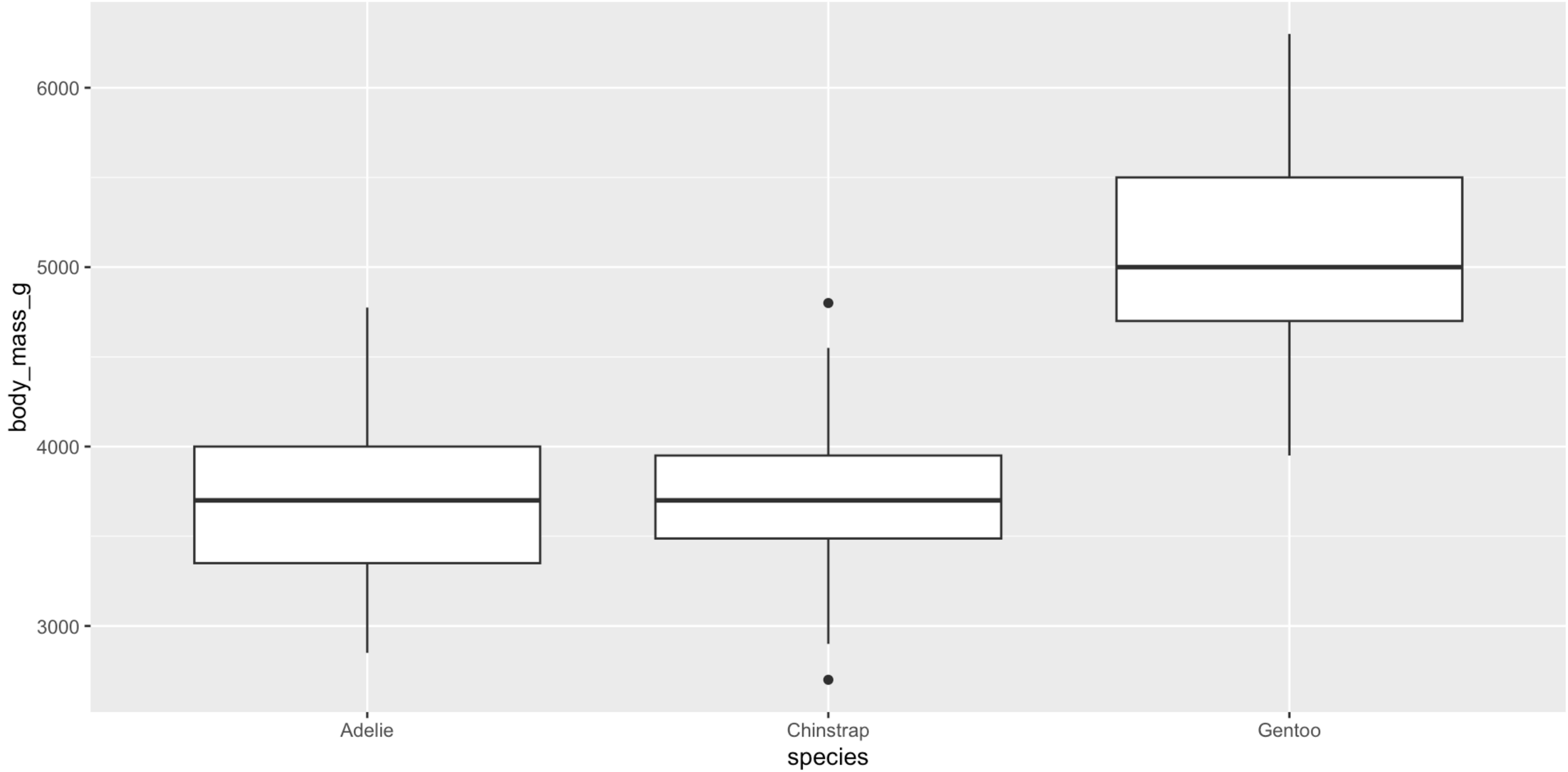
A numerical and a categorical variable

To visualize the relationship between a numerical and a categorical variable we can use side-by-side box plots. A boxplot is a type of visual shorthand for measures of position (percentiles) that describe a distribution. It is also useful for identifying potential outliers.



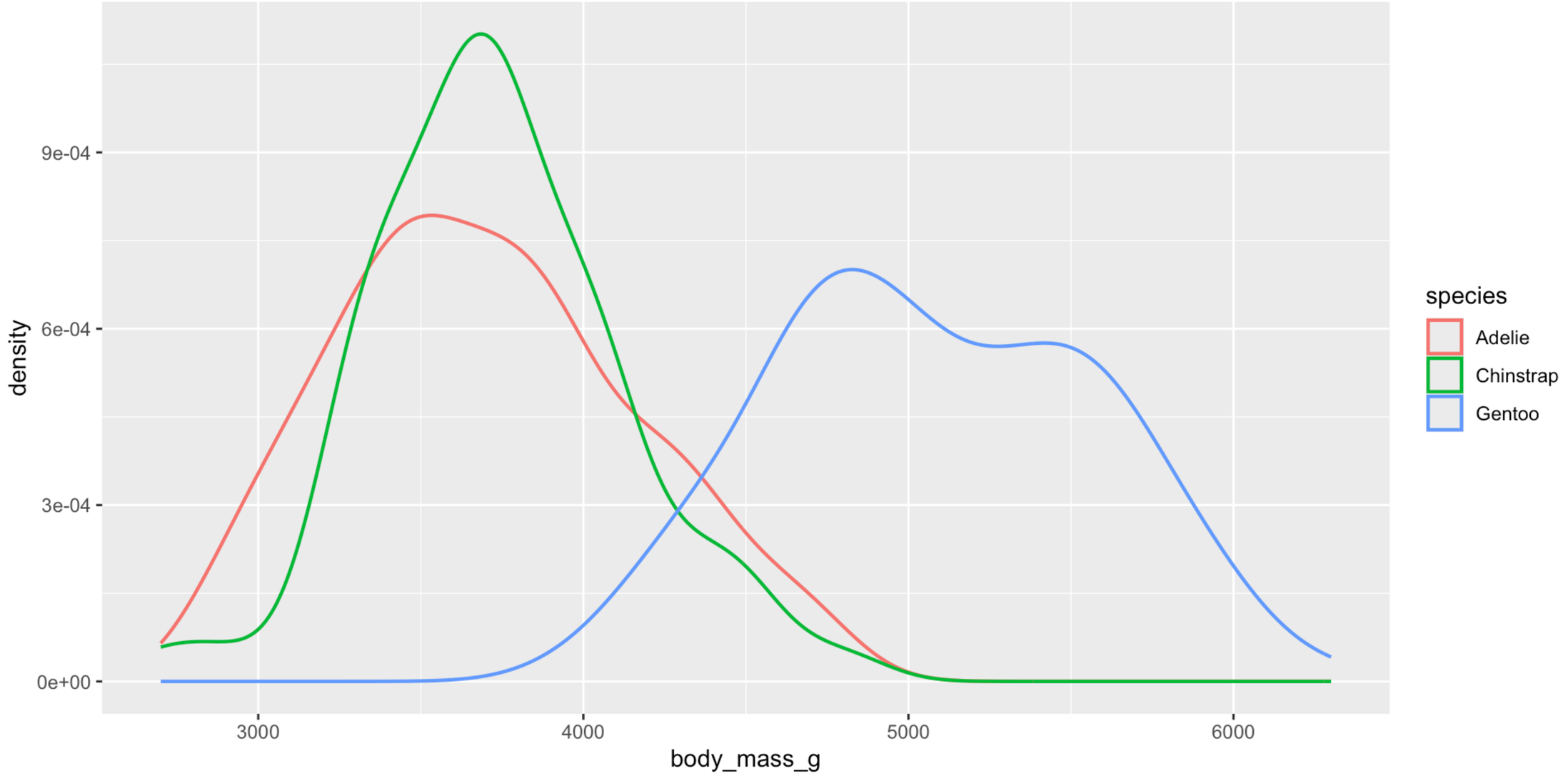
Let’s take a look at the distribution of body mass by species using geom_boxplot():

```
1 ggplot(penguins, aes(x = species, y = body_mass_g)) +
2   geom_boxplot()
```



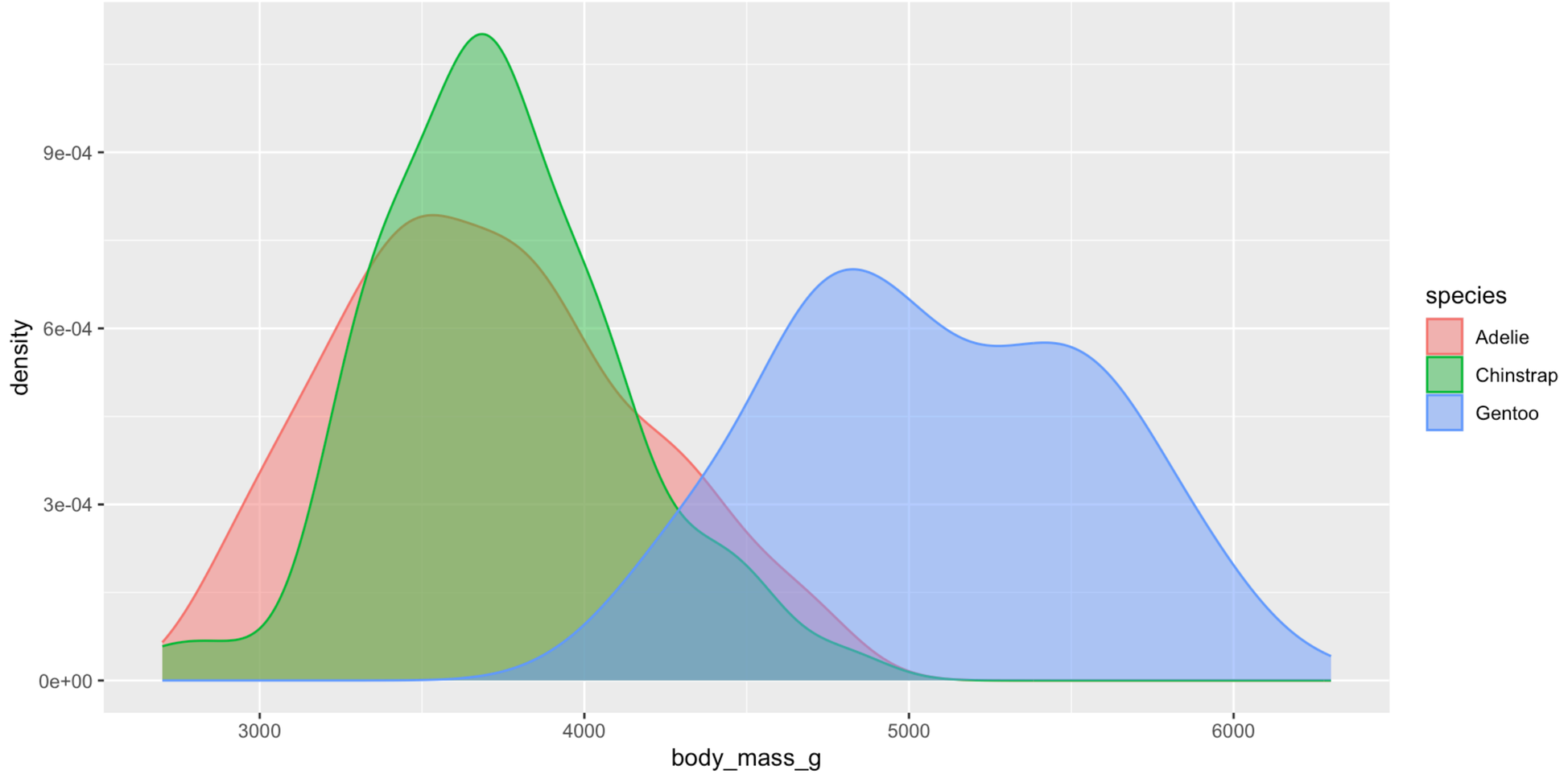
Alternatively, we can make density plots with geom_density().

```
ggplot(penguins, aes(x = body_mass_g, color = species)) +
  geom_density(linewidth = 0.75)
```



We can map species to both color and fill aesthetics and use the alpha aesthetic to add transparency to the filled density curves. This aesthetic takes values between 0 (completely transparent) and 1 (completely opaque). In the following plot it's set to 0.5.

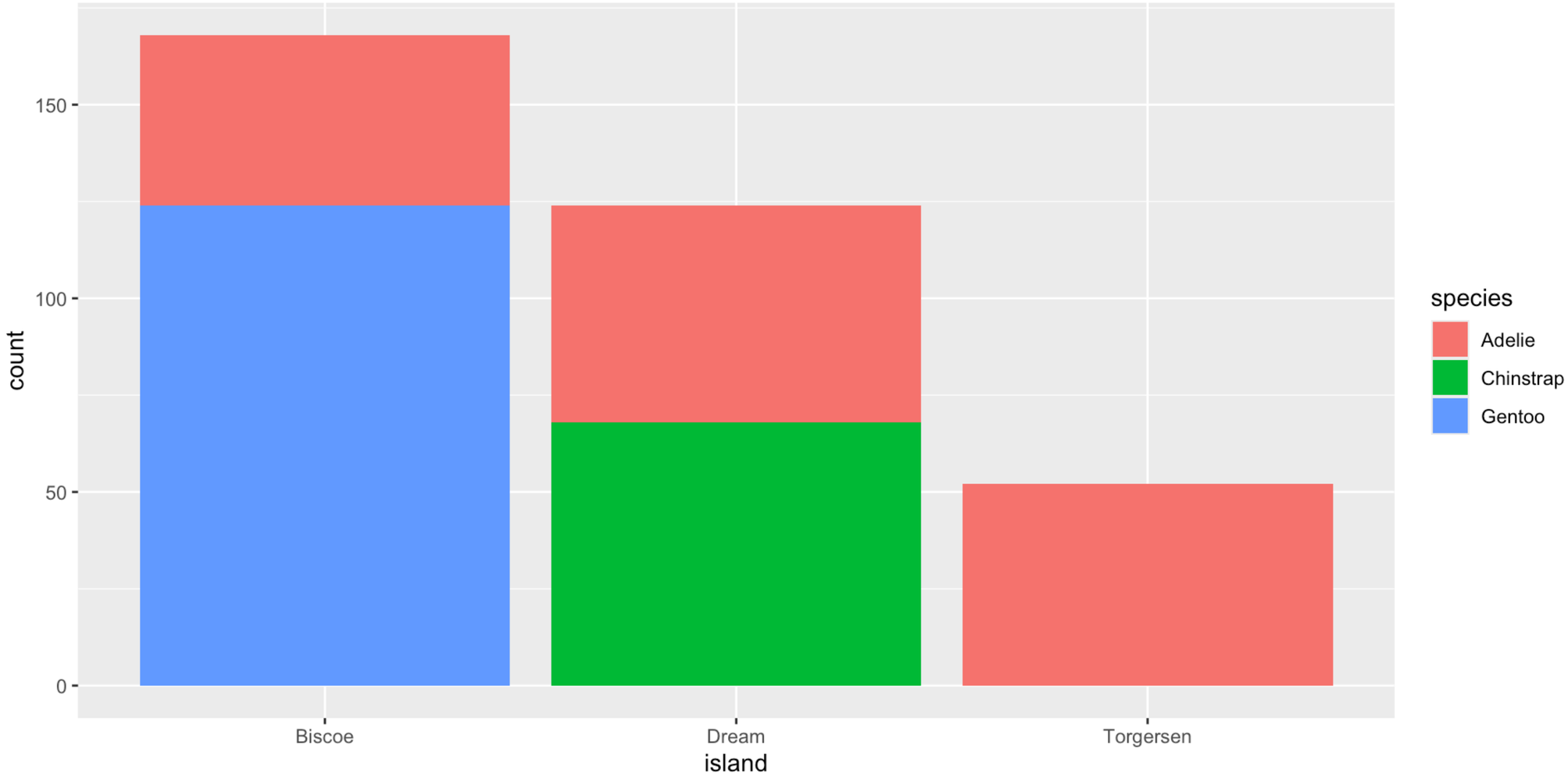
```
ggplot(penguins, aes(x = body_mass_g, color = species, fill = species)) +
  geom_density(alpha = 0.5)
```



Two categorical variables

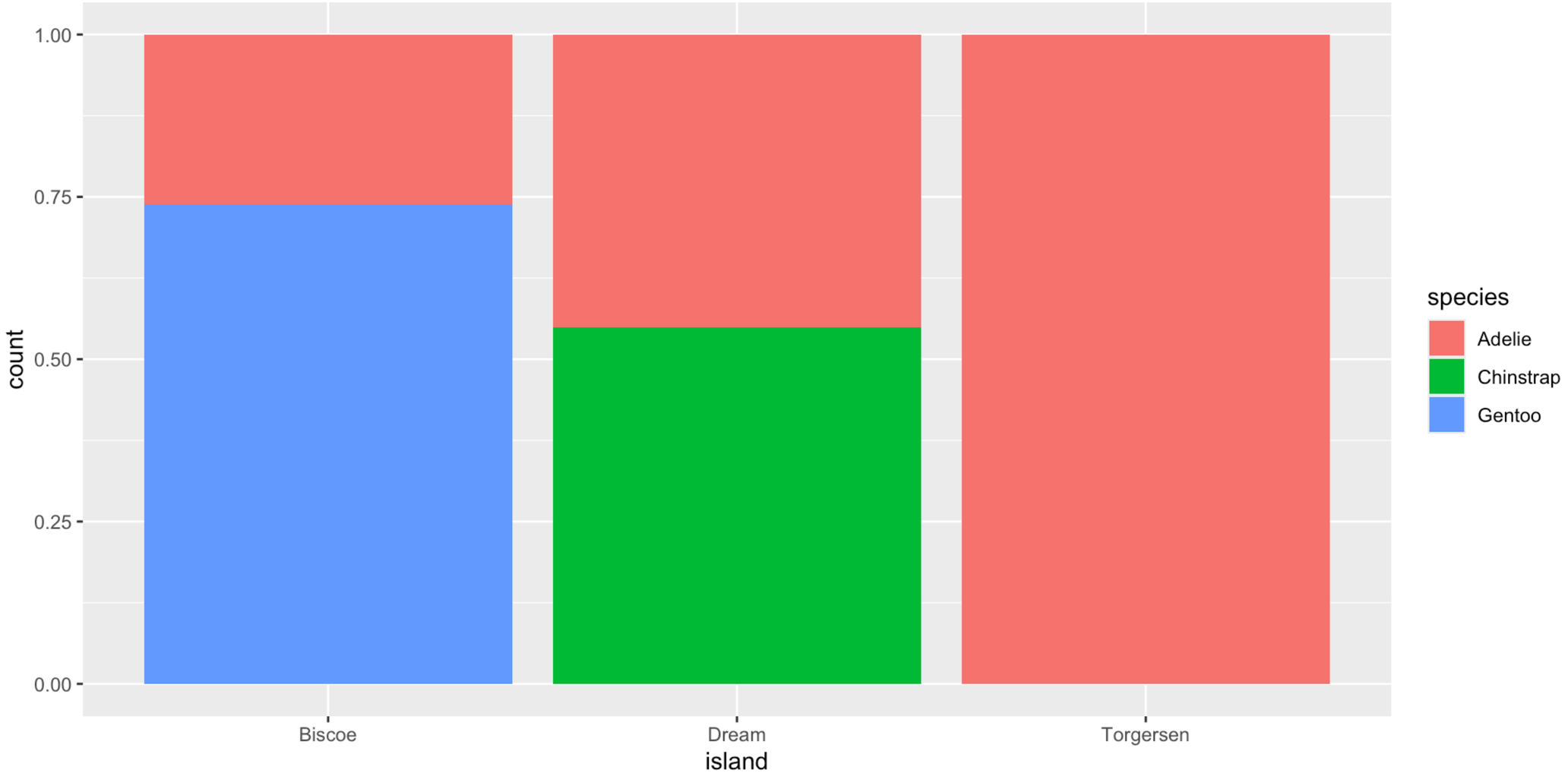
We can use stacked bar plots to visualize the relationship between two categorical variables.

```
1 ggplot(penguins, aes(x = island, fill = species)) +
2   geom_bar()
```



A relative frequency plot created by setting position = “fill” in the geom, is more useful for comparing species distributions across islands since it’s not affected by the unequal numbers of penguins across the islands.

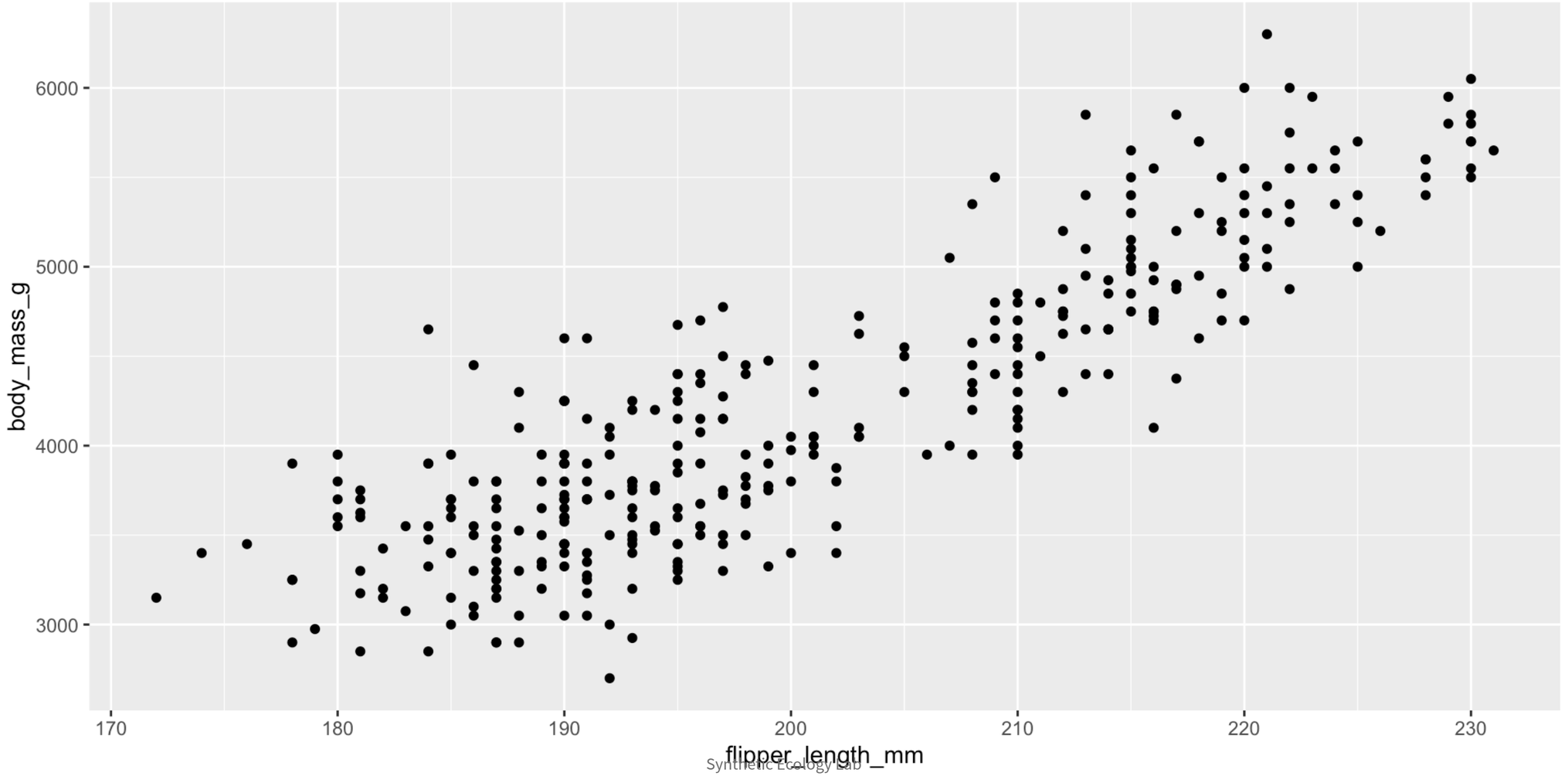
```
ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar(position = "fill")
```



Two numerical variables

A scatterplot is probably the most commonly used plot for visualizing the relationship between two numerical variables.

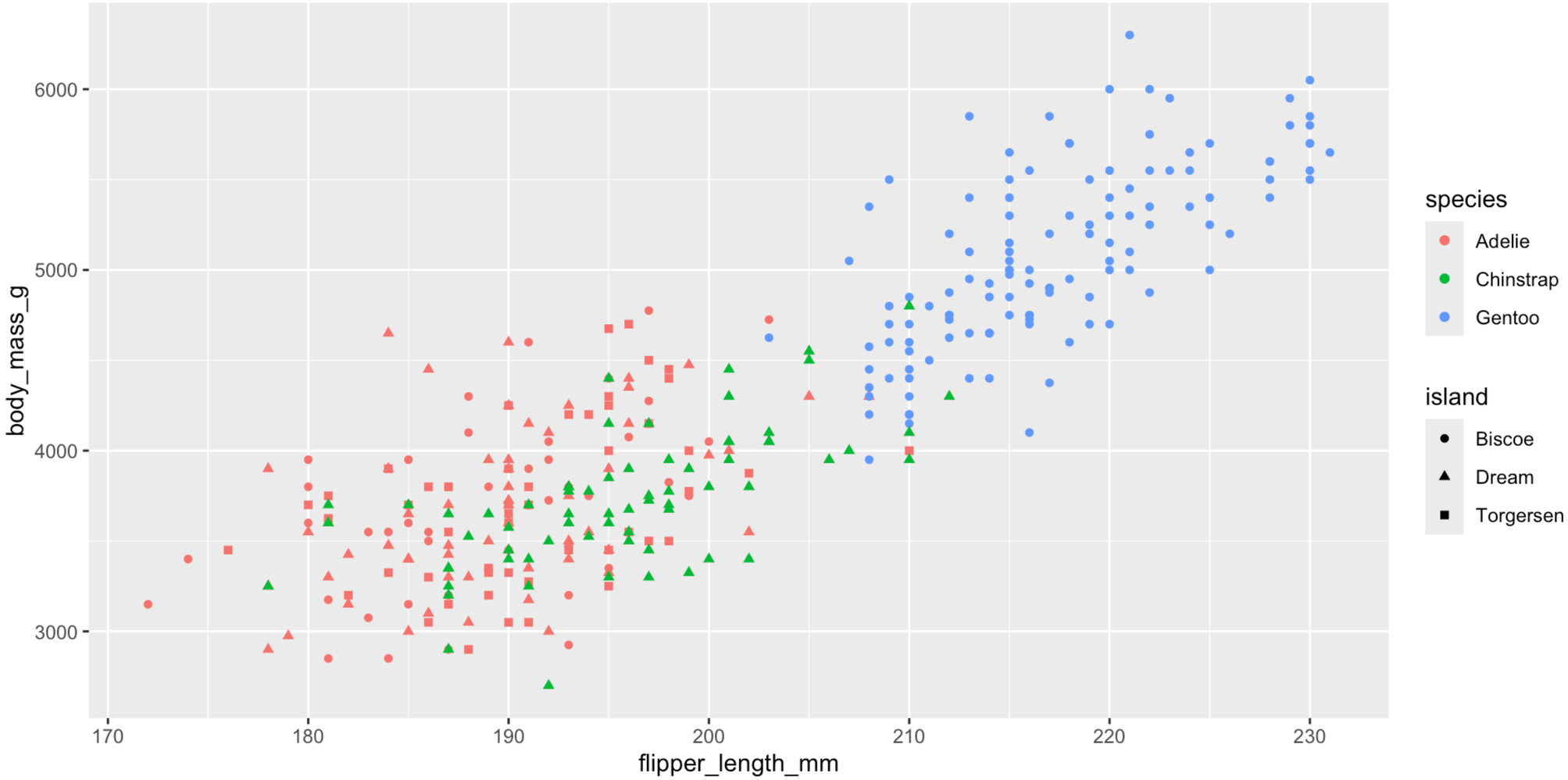
```
1 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
2   geom_point()
```



Three or more variables

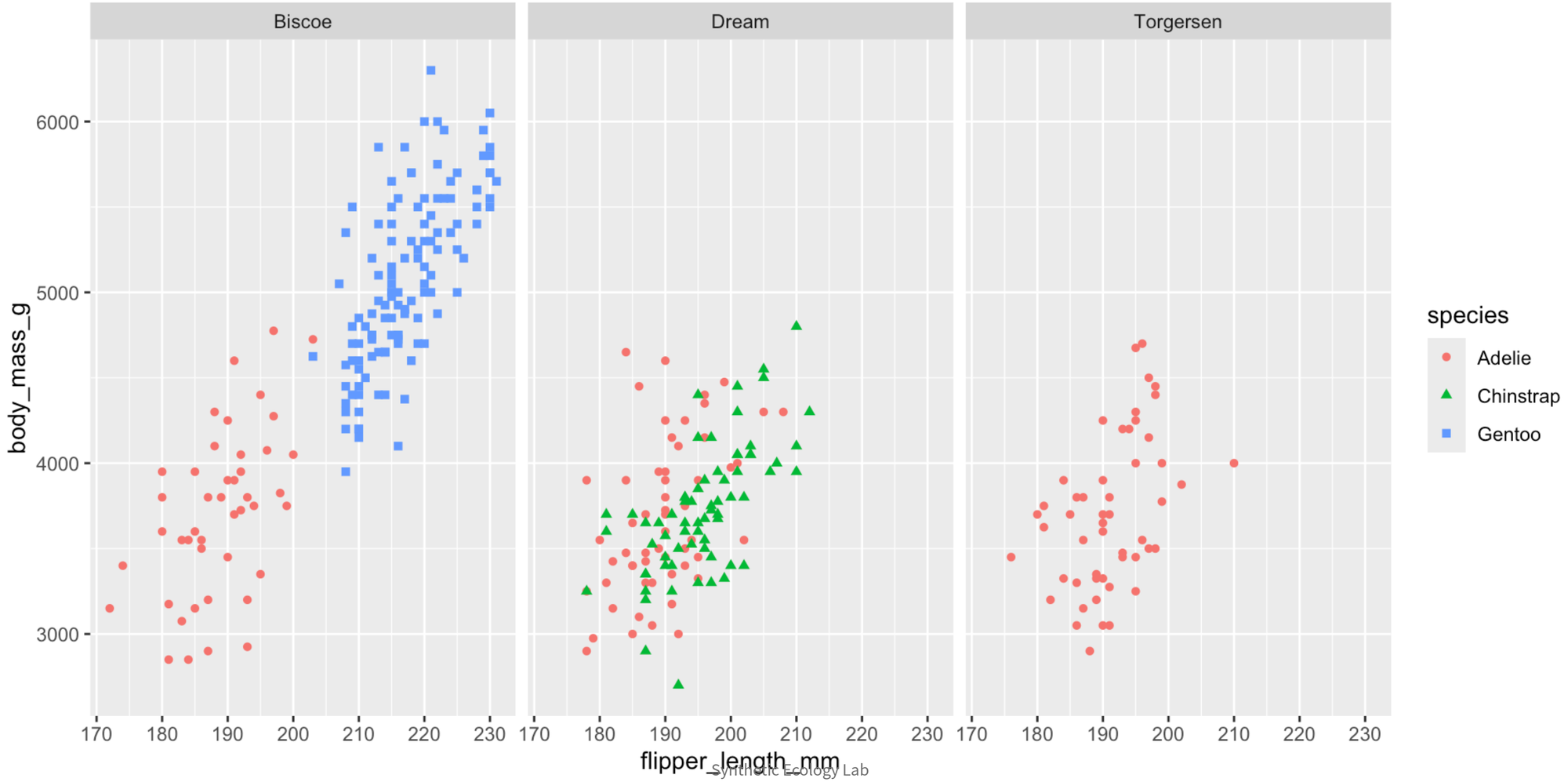
We can incorporate more variables into a plot by mapping them to additional aesthetics.

```
1 ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
2   geom_point(aes(color = species, shape = island))
```



However adding too many aesthetic mappings to a plot makes it cluttered and difficult to make sense of. Another way, which is particularly useful for categorical variables, is to split your plot into facets, subplots that each display one subset of the data.

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = species, shape = species)) +
  facet_wrap(~island)
```



Summary

In this course, you've learned the basics of data visualization with ggplot2.

1. We started with the basic idea that underpins ggplot2: a visualization is a mapping from variables in your data to aesthetic properties like position, color, size and shape.
2. You then learned about increasing the complexity and improving the presentation of your plots layer-by-layer.
3. You also learned about commonly used plots for visualizing the distribution of a single variable as well as for visualizing relationships between two or more variables, by leveraging additional aesthetic mappings and/or splitting your plot into small multiples using faceting.