r4ds-ch1

2025-05-03

Ch 1 Data visualization

```
library(tidyverse)
```

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

```
library(palmerpenguins)
```

```
##
## Attaching package: ’palmerpenguins’
##
## The following objects are masked from ’package:datasets’:
##
##     penguins, penguins_raw
```

```
library(ggthemes)
```

1.2 First steps

Want to look at relationship between flipper length and body weight for penguins on Palmer islands

1.2.1 The penguins data frame

Tabular data is tidy if each value in its own cell, each variable in its own column, and each observation in its own row

```
penguins  # tibble
```

```
## # A tibble: 344 x 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>
```

```
glimpse(penguins)  # See all variables and first few observations
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

```
#View(penguins)  # Opens interactive tab in RStudio

?penguins
```

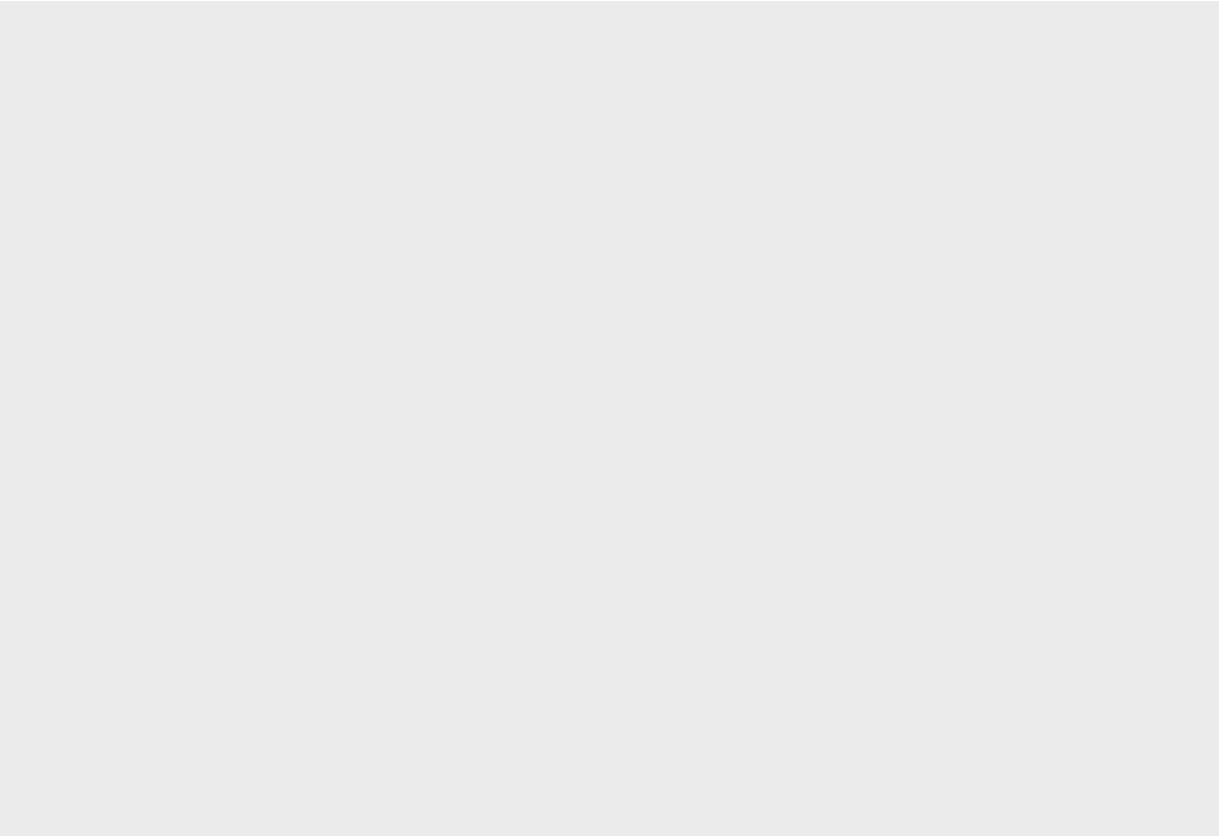
```
## Help on topic 'penguins' was found in the following packages:
##
##   Package                 Library
##   palmerpenguins          /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library
##   datasets                /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library
##
##
## Using the first match ...
```

1.2.2 Ultimate goal

Display relationship between flipper length and body weight, taking into consideration species of the penguin

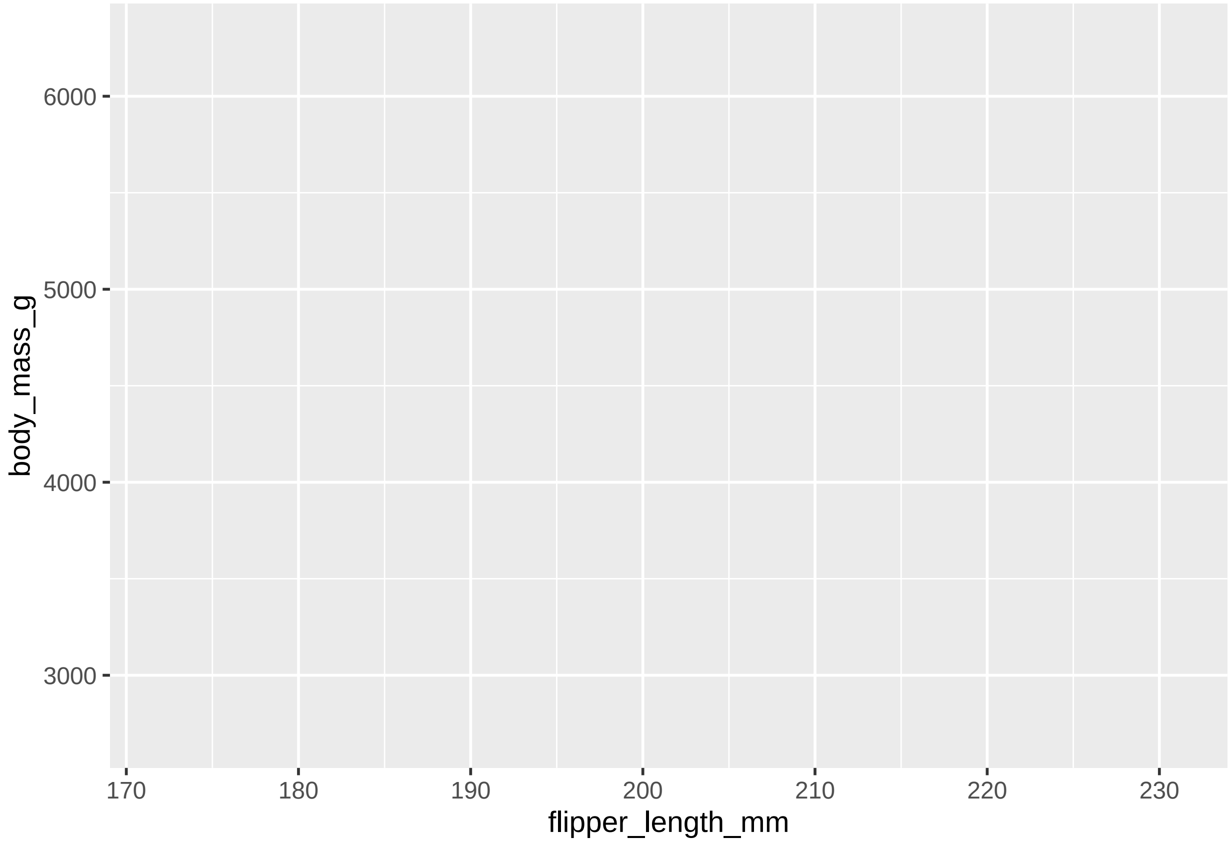
1.2.3 Creating a ggplot

```
ggplot(data = penguins)  # Defining plot object to add layers to
```



So far, empty canvas

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

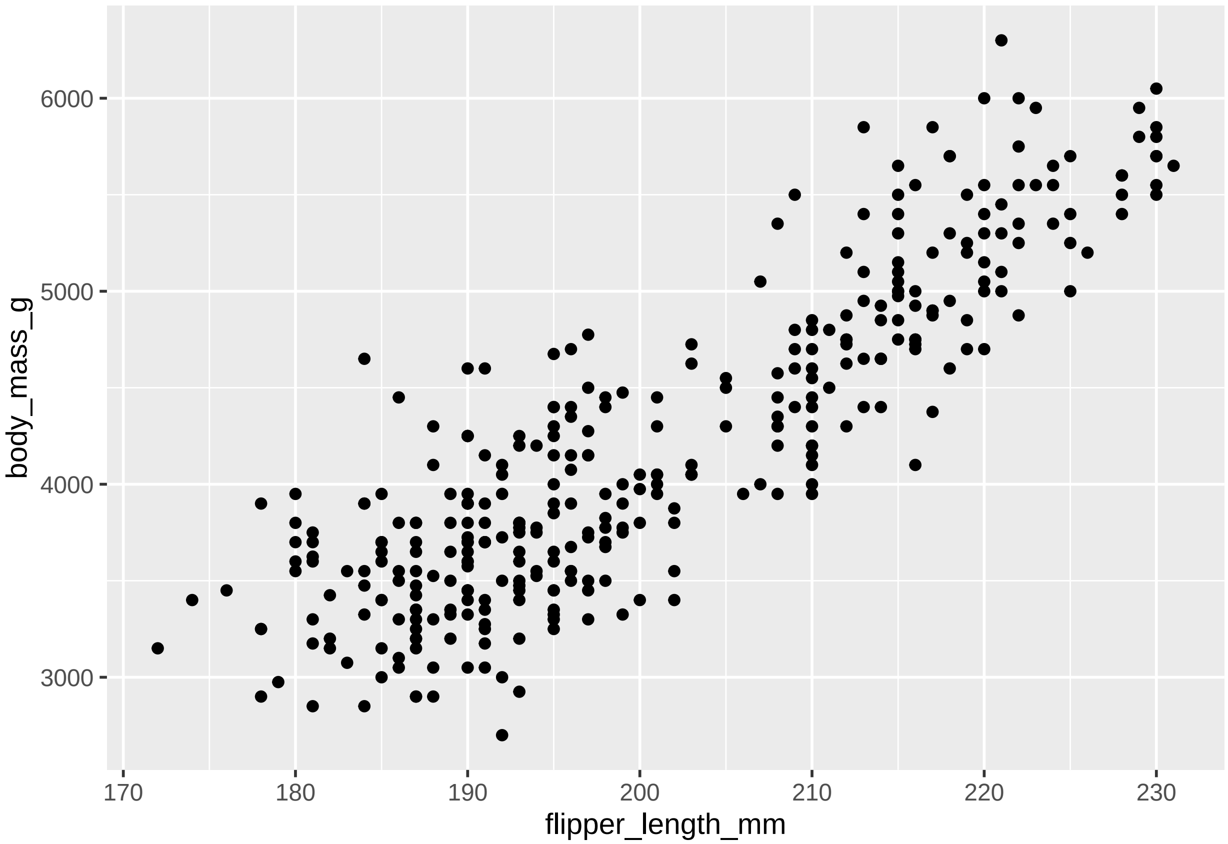


```
# Use aes to specify which variables to use for x and y axes
```

Geom is a geometrical object that a plot uses to represent data e.g., geom_boxplot() for boxplot and geom_point() for scatterplot

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) + geom_point() # adds a layer of points to plot, creating a scatterplot
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Warning message: ggplot2 has no way representing point on plot if there is a missing value

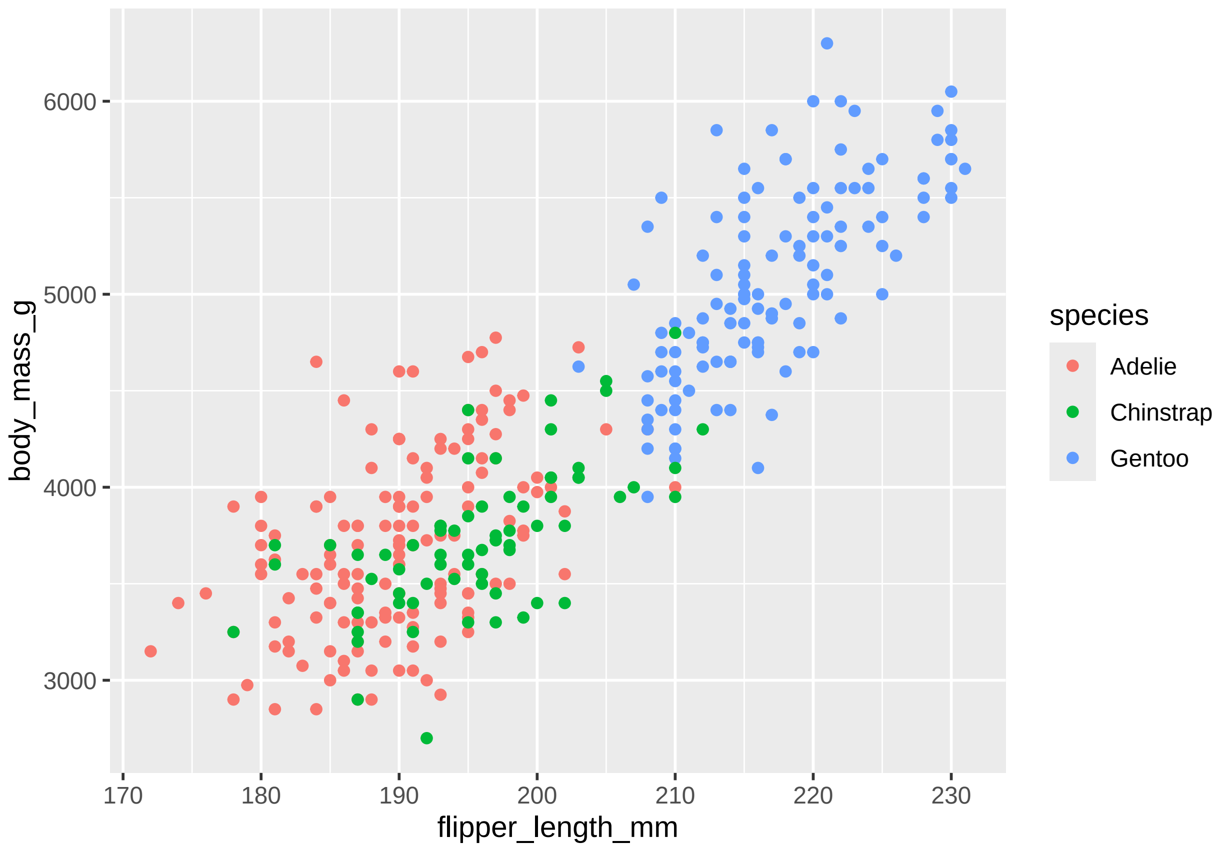
So far relationship appears positive, fairly linear, and moderately strong

1.2.4 Adding aesthetics and layers

Does relationship differ by species?

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Scaling: ggplot2 automatically assigns unique value to aesthetic (i.e. color) for each unique level of categorical variable (i.e. species)

Also adds a legend

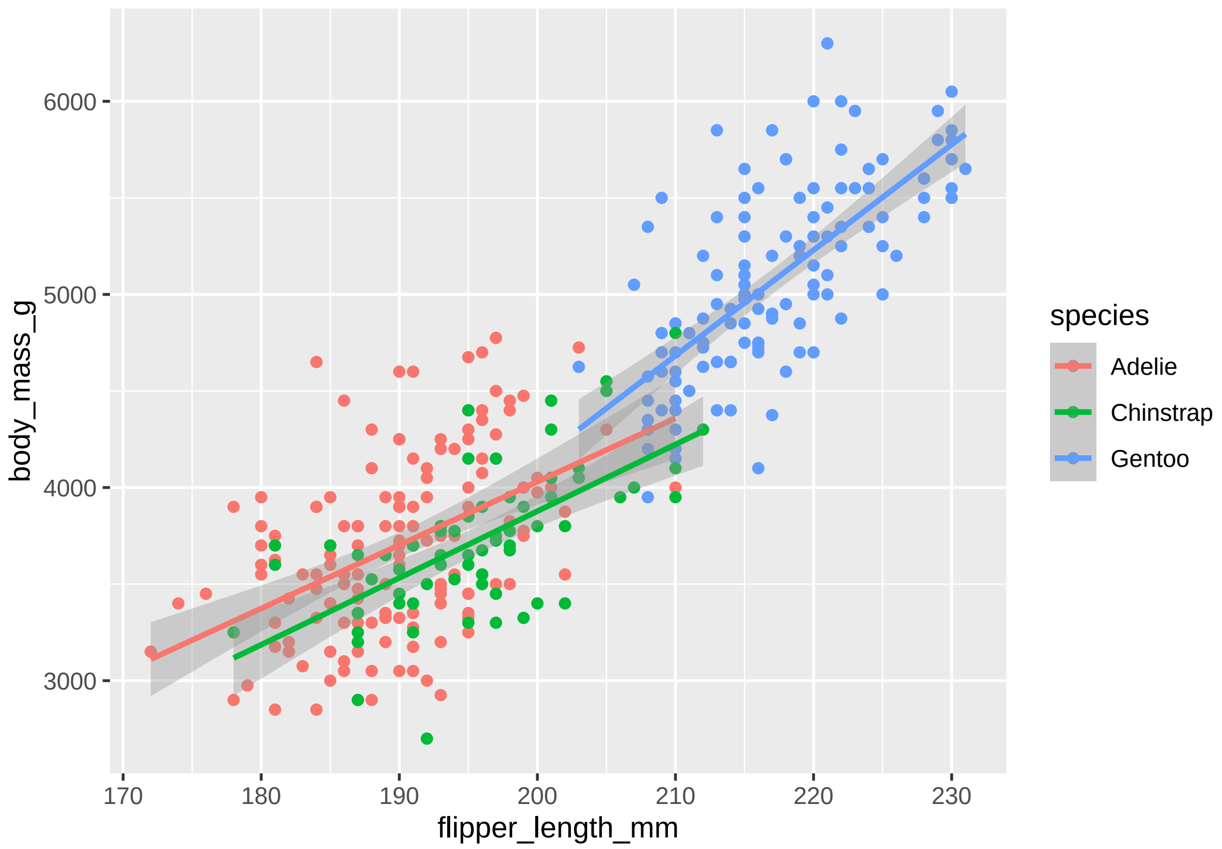
Now add another layer, display smooth curve of relationship between body mass and flipper length

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)
) +
  geom_point() +
  geom_smooth(method = lm)  # New geometric object, use linear model
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
# to draw a line of best fit
```

Will draw a line for each of the species, different from goal plot

Why? Aesthetic mappings at global level are passed down to each subsequent layer, but each geom function can also take a mapping argument This allows aesthetics at local level in addition to those inherited

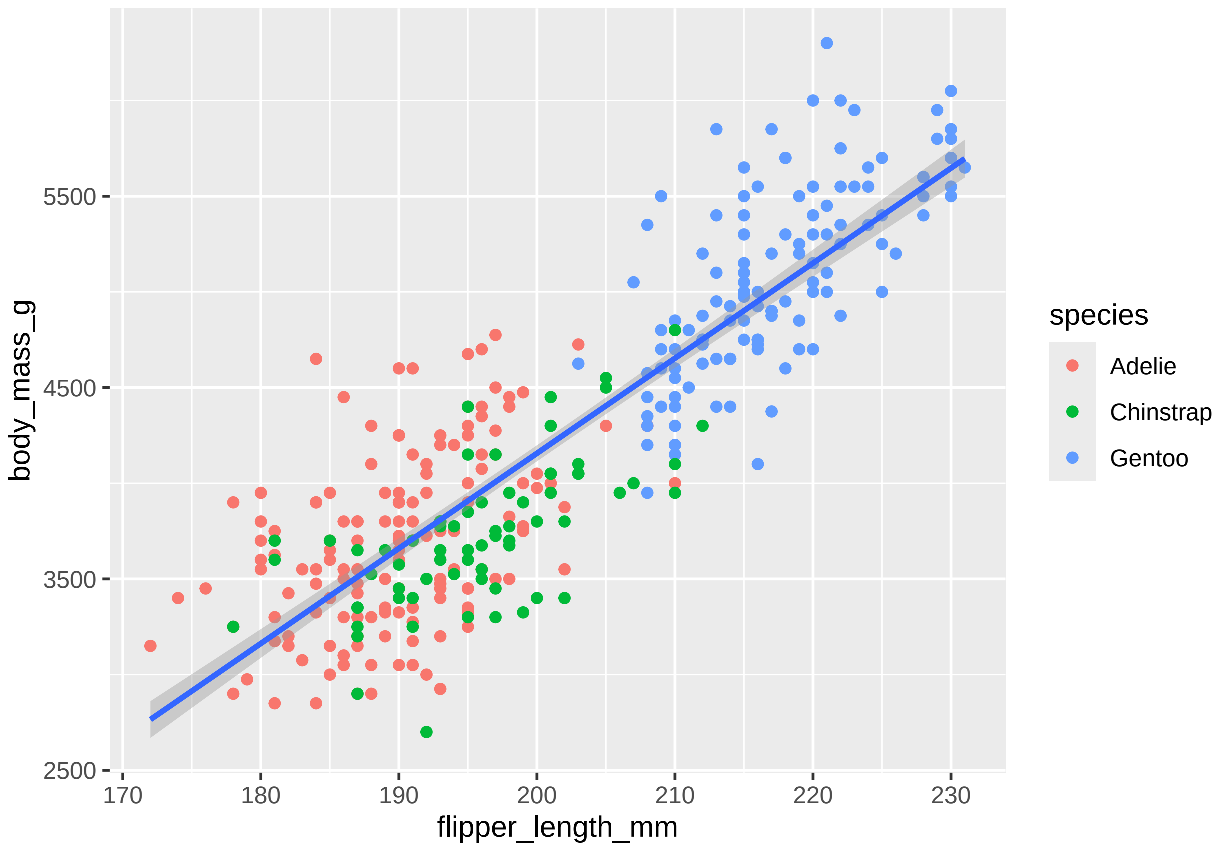
Since want points to be colored by species but not separated lines, specify color in geom_point() only!

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

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



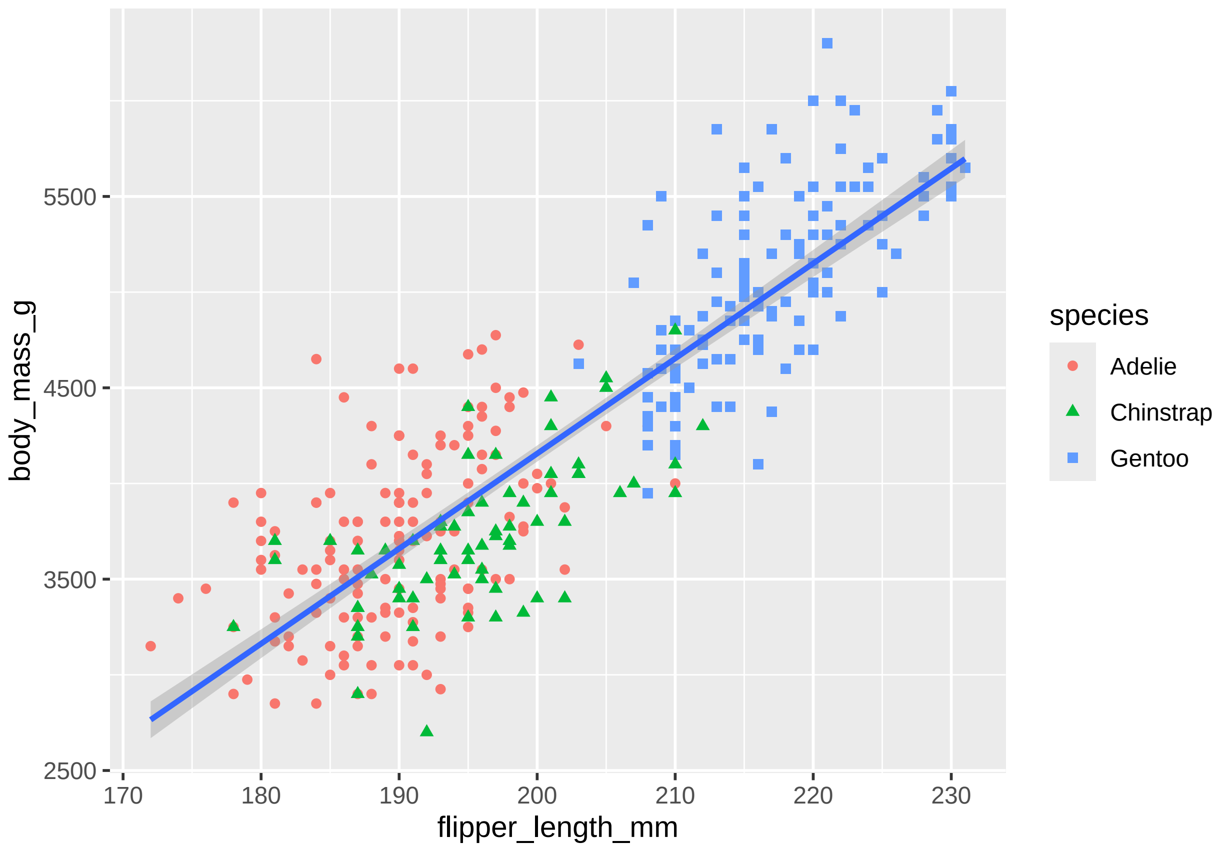
Also differentiate species by shape, more accessible

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

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



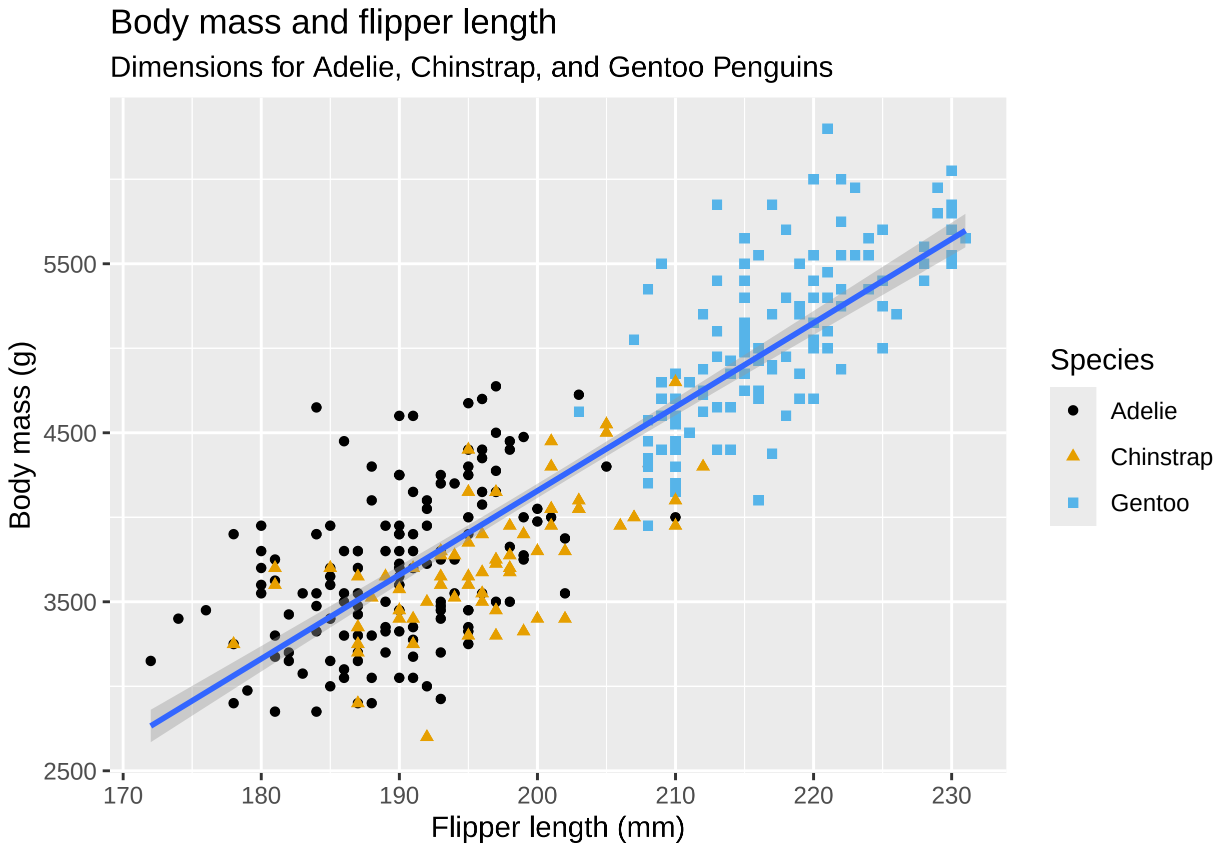
Now add/specify labels in a new layer

```
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") +
  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" # Defines label for legend
    ) +
  scale_color_colorblind()  # Improve color palette to be colorblind safe
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Now matches goal plot!

1.2.5 Exercises

1. How many rows in penguins? How many columns

```
dim(penguins)
```

```
## [1] 344   8
```

344 rows, 8 columns

2. What does bill_depth_mm describe?

```
?penguins
```

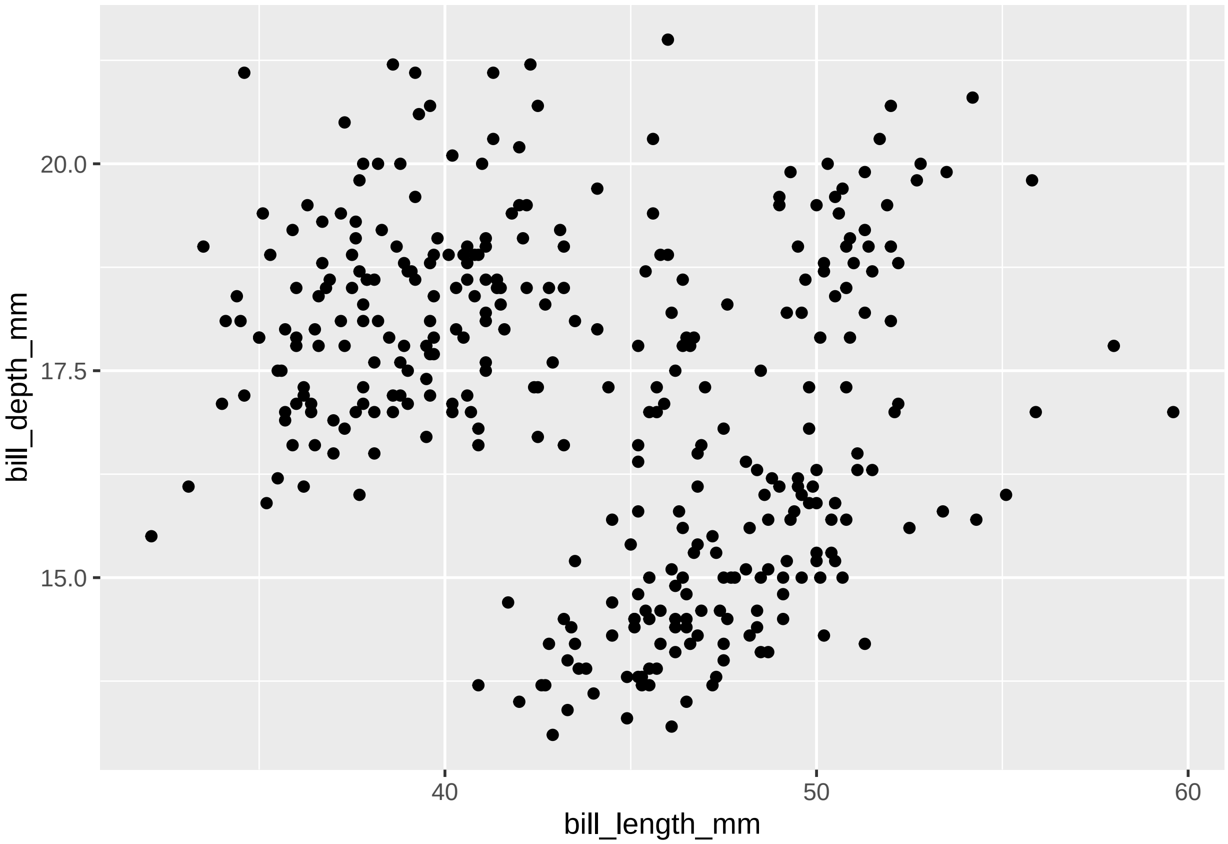
```
## Help on topic 'penguins' was found in the following packages:
##
##   Package                 Library
##   palmerpenguins          /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library
##   datasets                /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library
##
##
## Using the first match ...
```

The bill depth in millimeters of a penguin

3. Make a scatterplot of bill_depth_mm (on y-axis) vs bill_length_mm (x-axis)

```
ggplot(
  data = penguins,
  mapping = aes(x = bill_length_mm, y = bill_depth_mm)
) +
  geom_point()
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

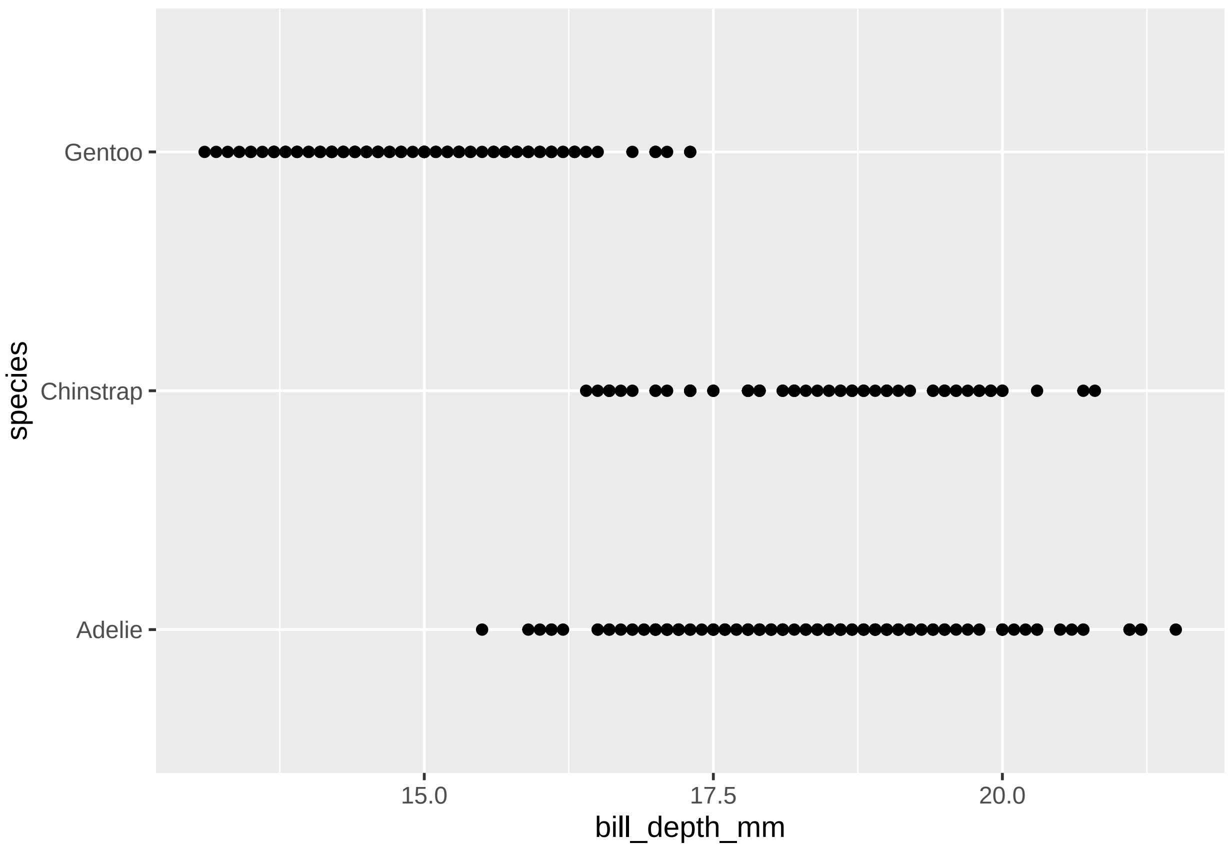


Relationship between two variables is non-linear, very weak, and negative

4. Try a scatterplot of species vs bill_depth_mm

```
ggplot(
  data = penguins,
  mapping = aes(x = bill_depth_mm, y = species)
) +
  geom_point()
```

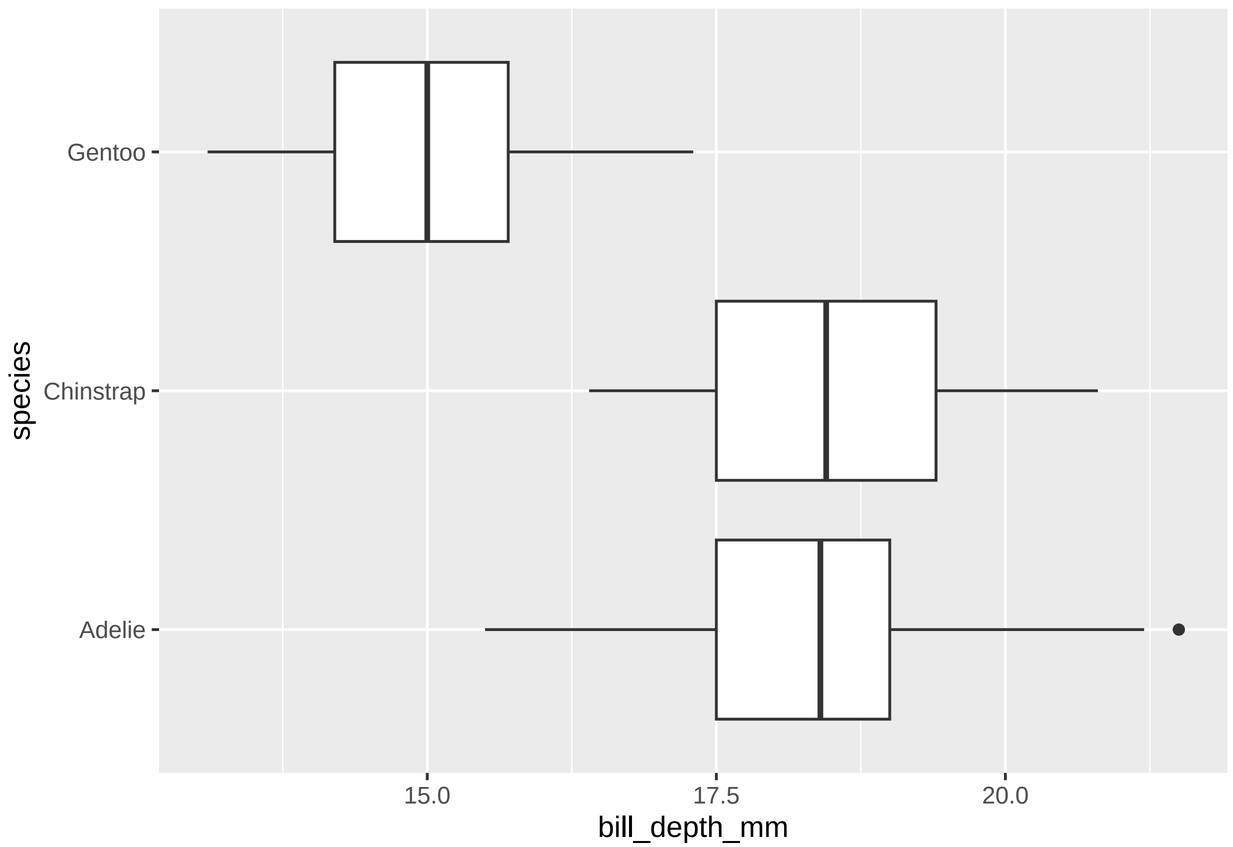
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



A better choice to compare bill depth between species might be a boxplot

```
ggplot(
  data = penguins,
  mapping = aes(x = bill_depth_mm, y = species)
) +
  geom_boxplot()
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



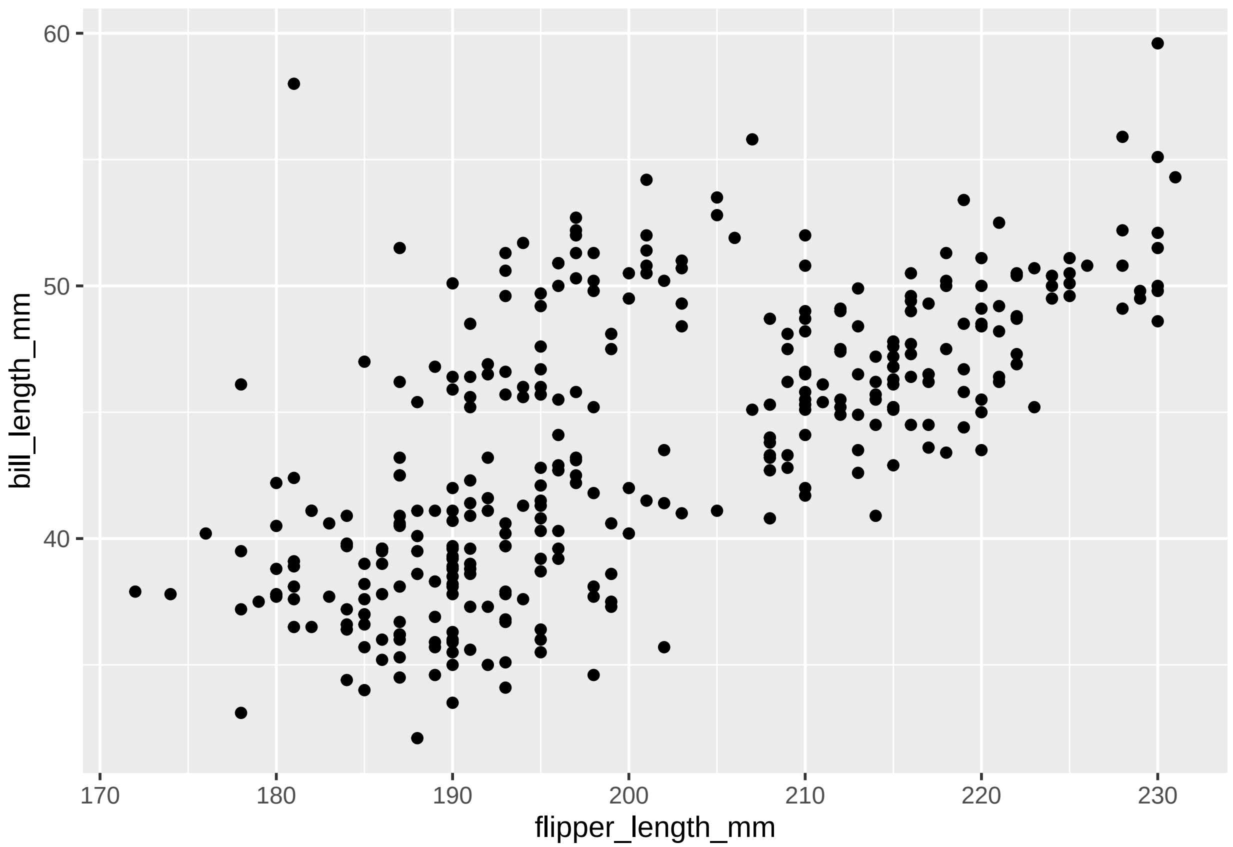
5. Why does the following give an error and how to fix?

```
#ggplot(data = penguins) +
  #geom_point()
```

There is an error because the mapping (axes) have not been specified, to fix, add some axes to create a scatterplot

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

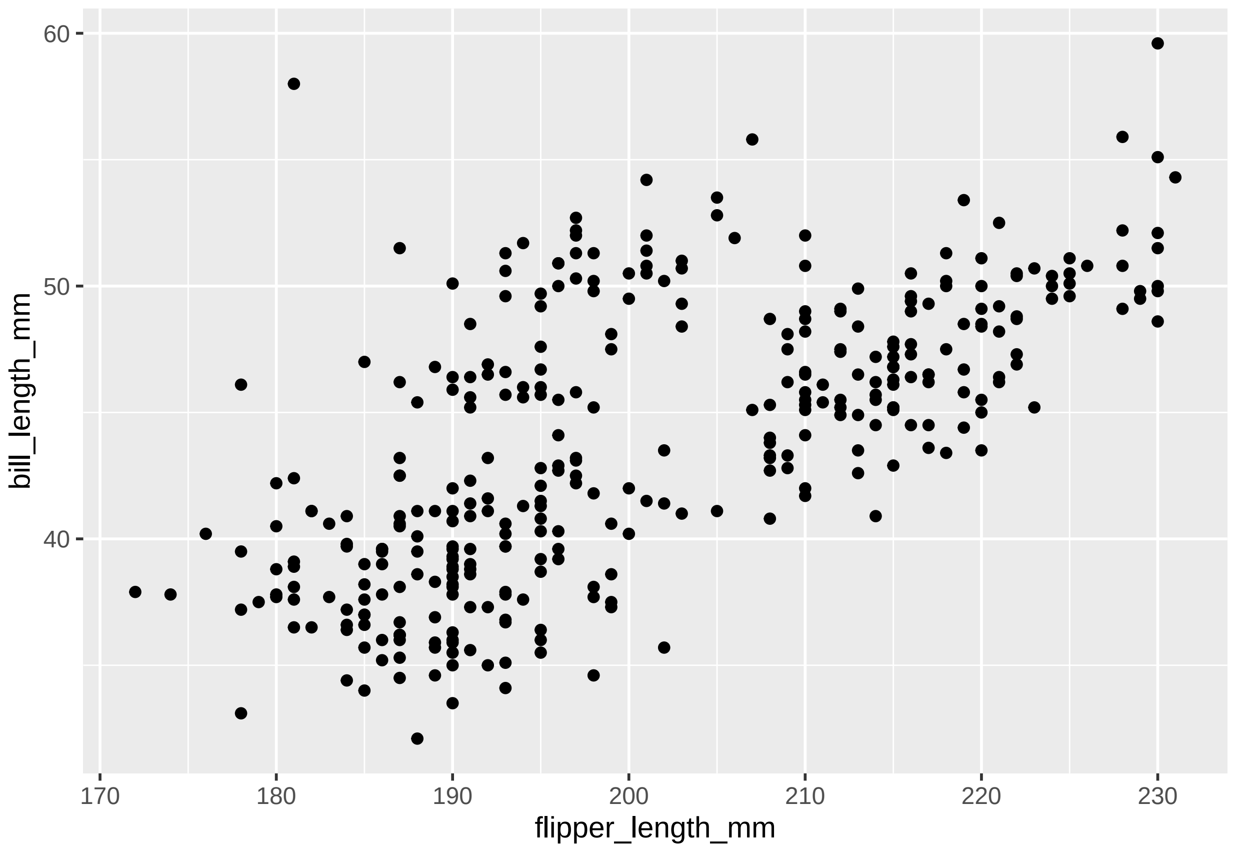


6. What does the argument na.rm do in geom_point()?

```
?geom_point()
```

If TRUE missing values are silently removed, the default is FALSE, which means missing values are removed with a warning

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE)
```

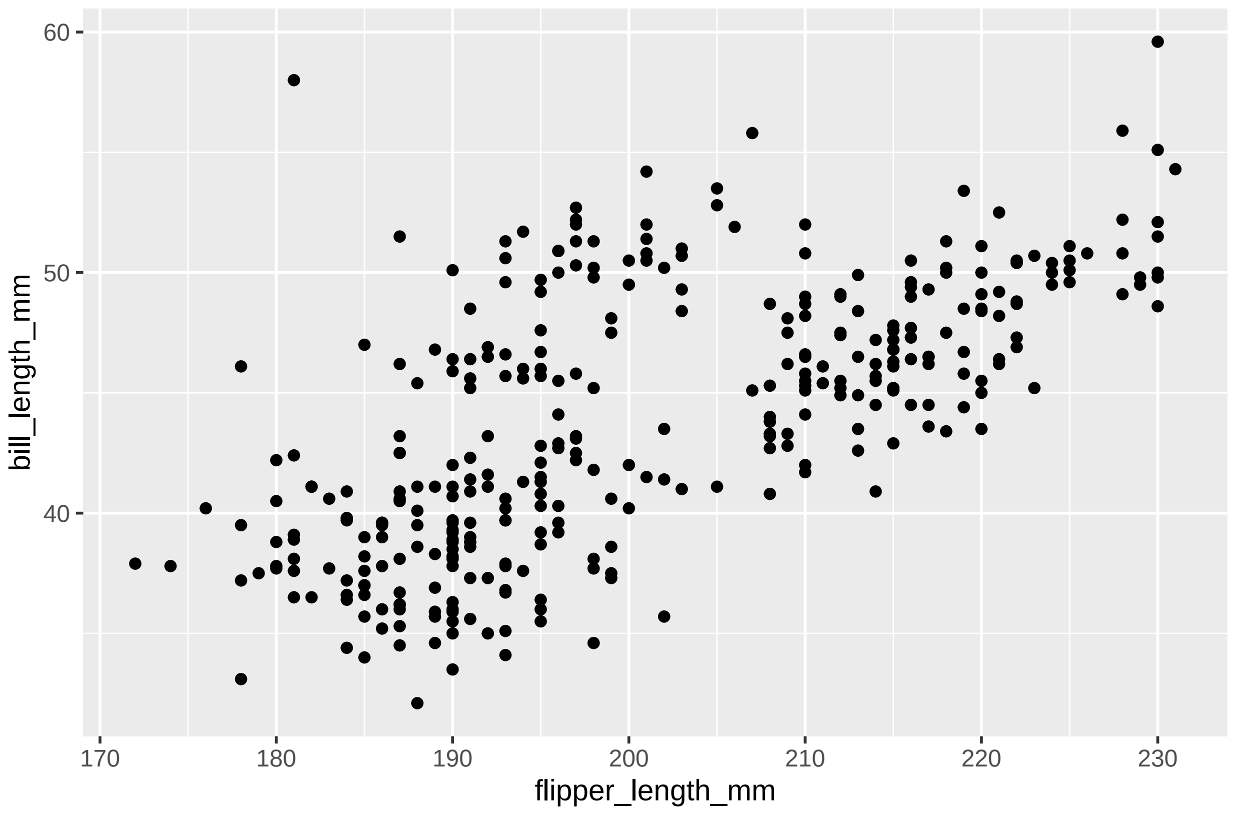


Now, no more warning at the top!

7. Add following caption to plot from previous exercise: "Data comes from the palmerpenguins package."

```r
?labs()  # Check documentation

ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE) +
  labs(caption = "Data comes from the palmerpenguins package.")
```

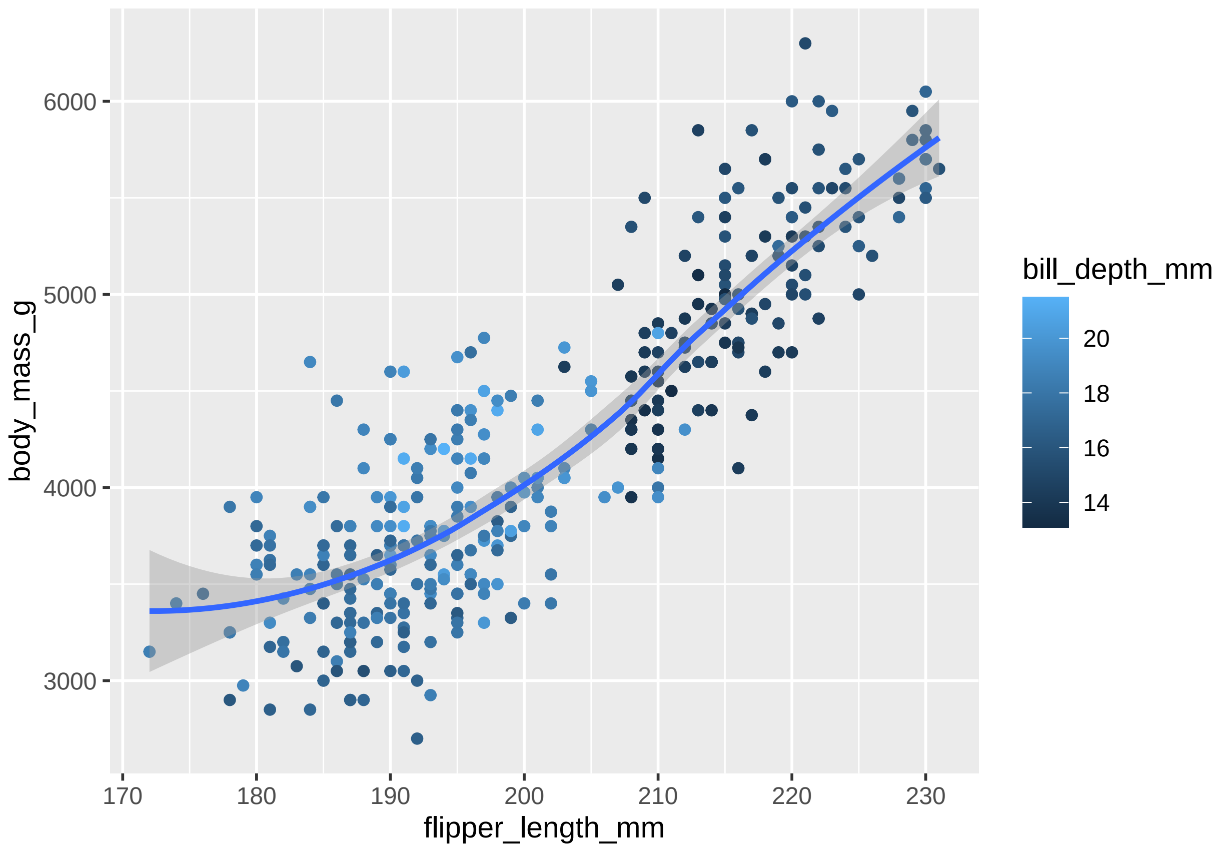


Data comes from the palmerpenguins package.

8. Recreate visualization. What aesthetic to map bill_depth_mm to, and at global or geom level?

```
# Since one line of best fit for all bill depths, map color to geom level
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point(
    mapping = aes(color = bill_depth_mm),
    na.rm = TRUE
  ) +
  geom_smooth(na.rm = TRUE)  # Non-linear model
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



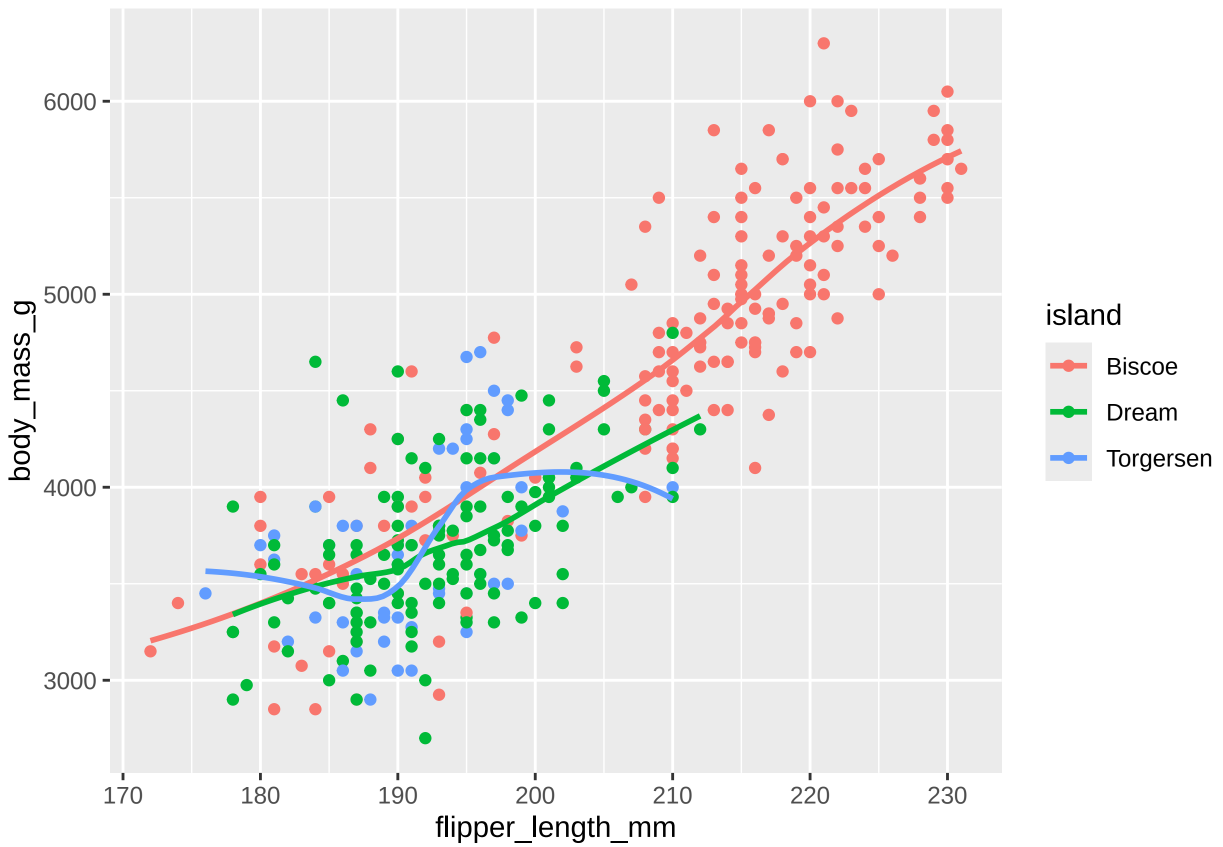
9. Run code in head, predict output, then check with R

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = island)
) +
  geom_point() +
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Well first I thought you need to specify method but it automatically decided to use "loess" Output: scatterplot of relationship between body mass and flipper length, with different colors for each island, and also three different lines of best fit with confidence intervals not displayed because color aesthetic is specified at global level and thus inherited by geom_smooth()

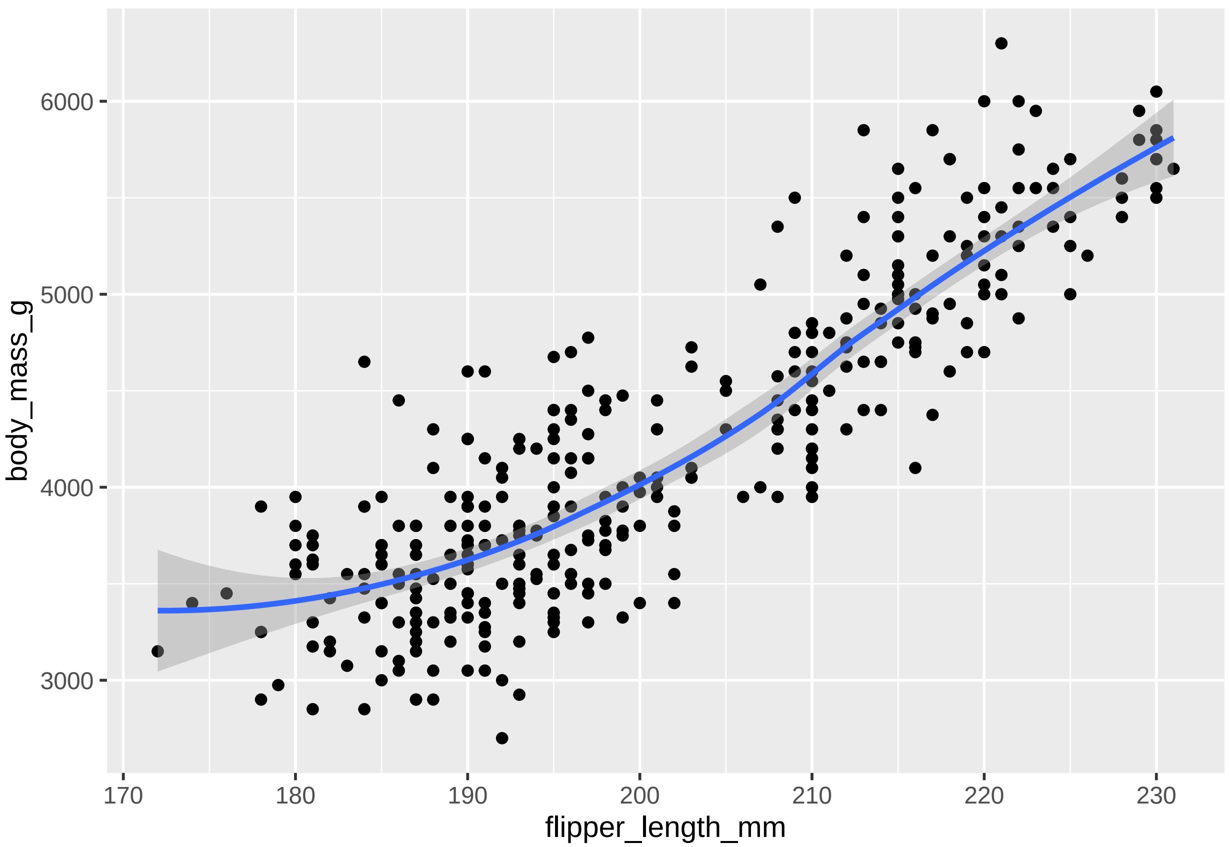
10. Will these two graphs look different? Why/why not?

```
ggplot(  # Plot 1
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point() +
  geom_smooth()
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula = ’y ~ x’
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

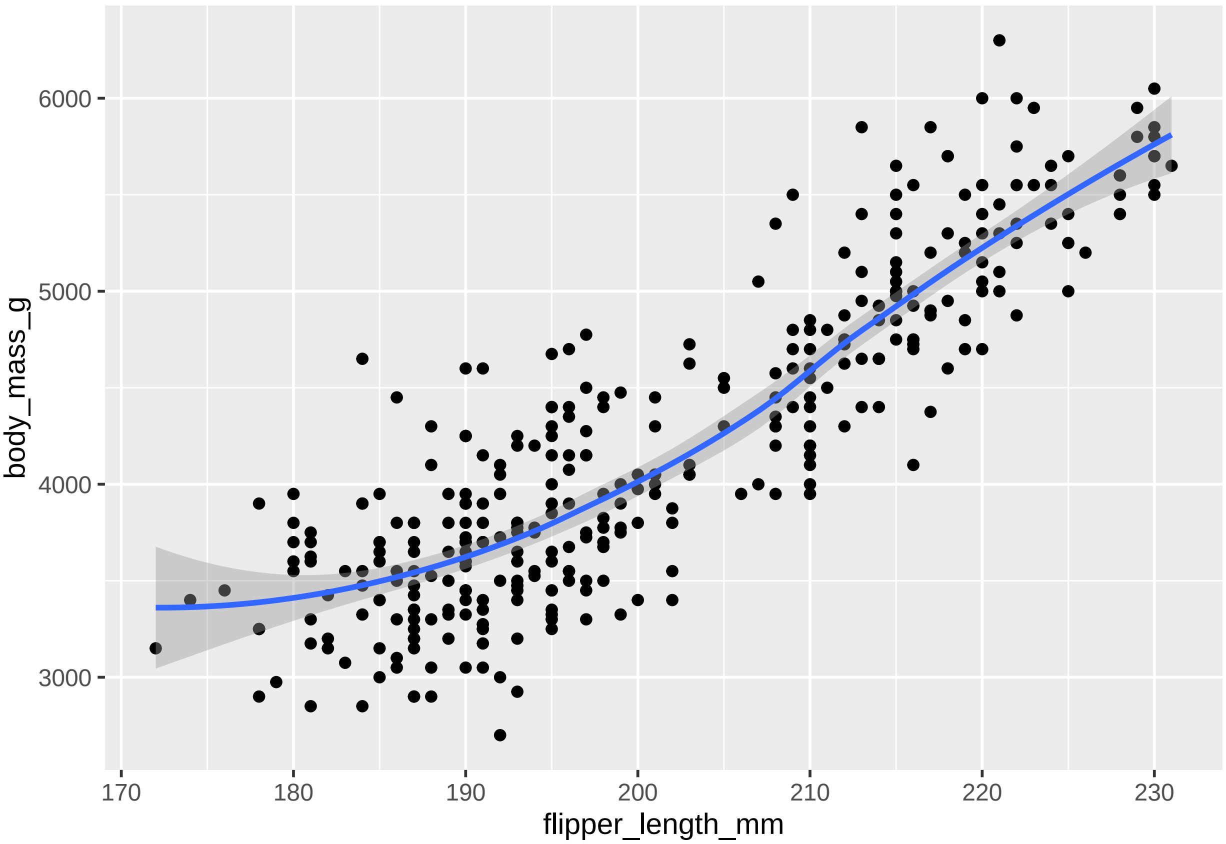
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
ggplot() +  # Plot 2
  geom_point(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  ) +
  geom_smooth(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  )
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range (`stat_smooth()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



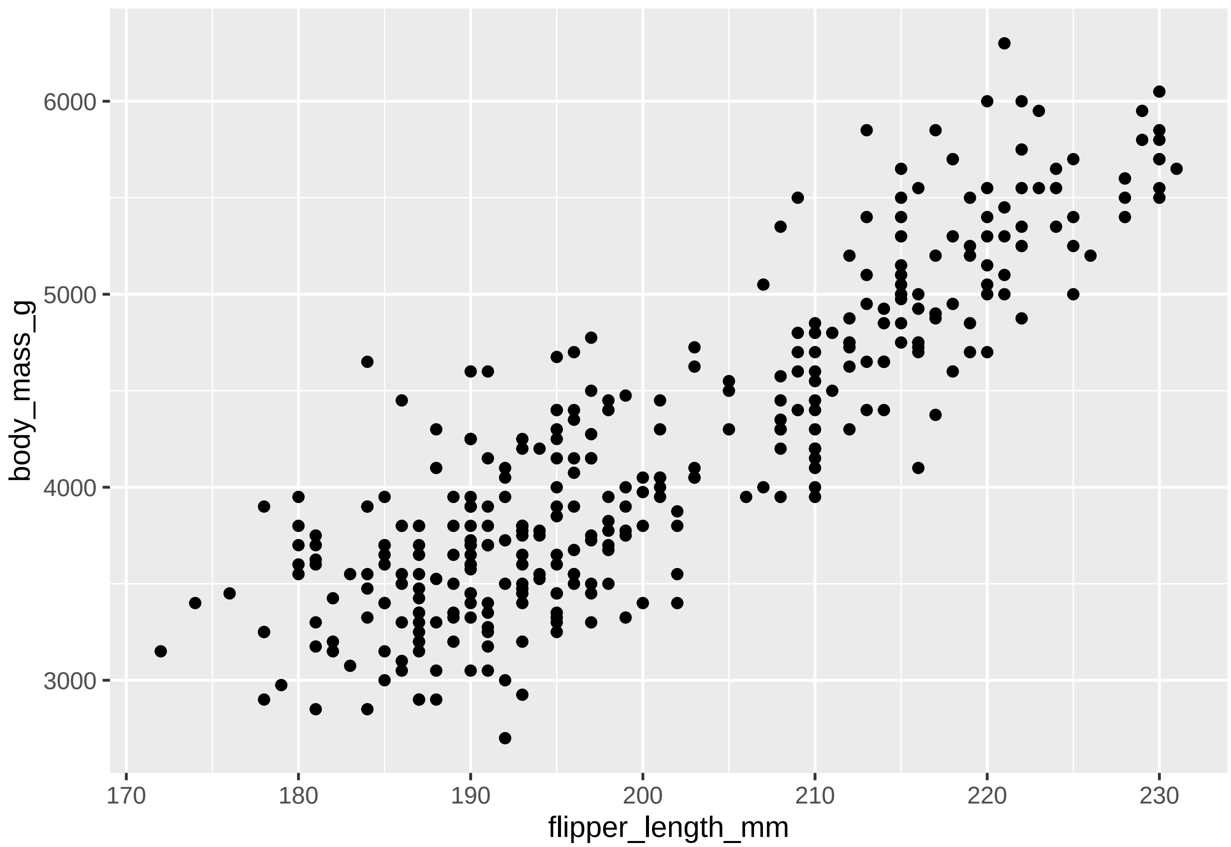
These graphs will look the same, because in plot 1 the two geoms will inherit the data and aesthetic mappings, whereas in plot 2 the two geoms specify their own local data and mappings Since the two (inherited in plot 1, local in plot 2) are the same, the graphs will look the same

1.3 ggplot2 calls

So far, have been writing very explicit ggplot2 code

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

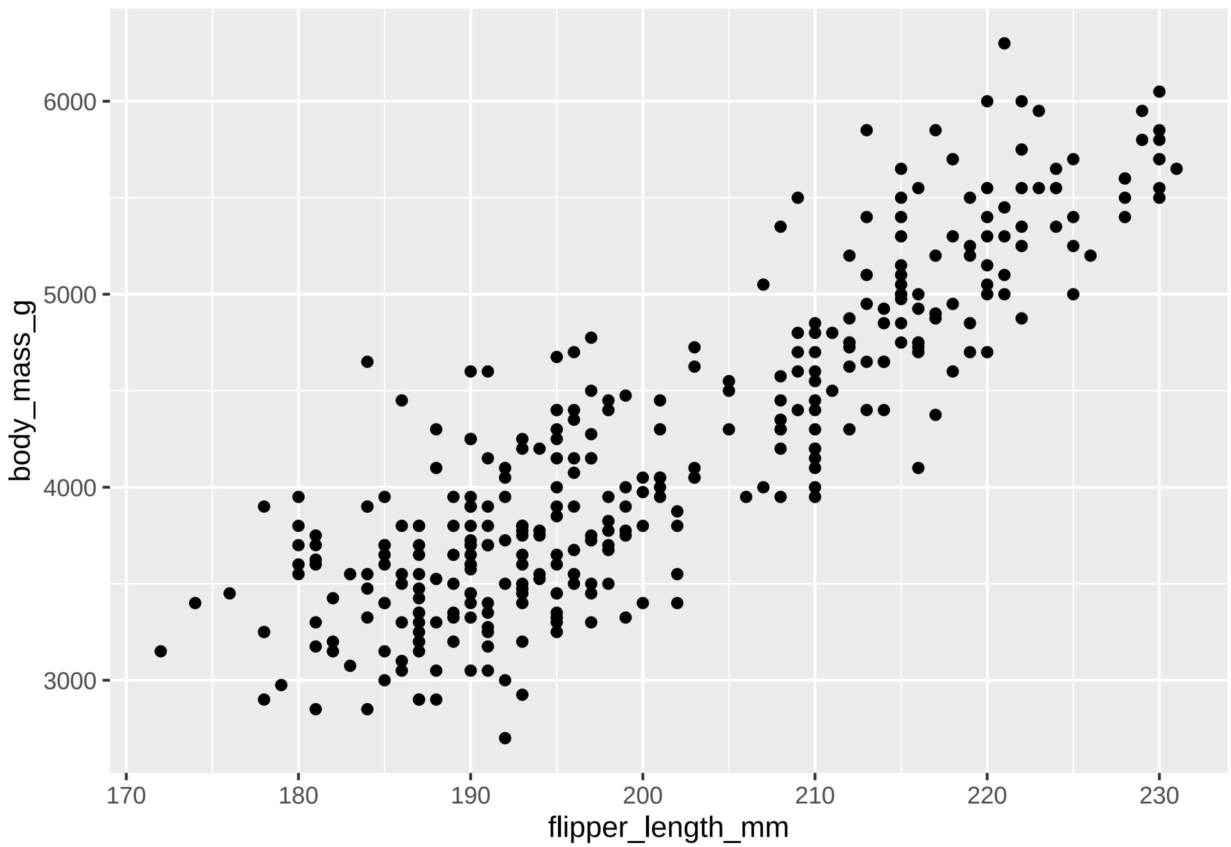
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



By not naming data and mapping, can write more concise code

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

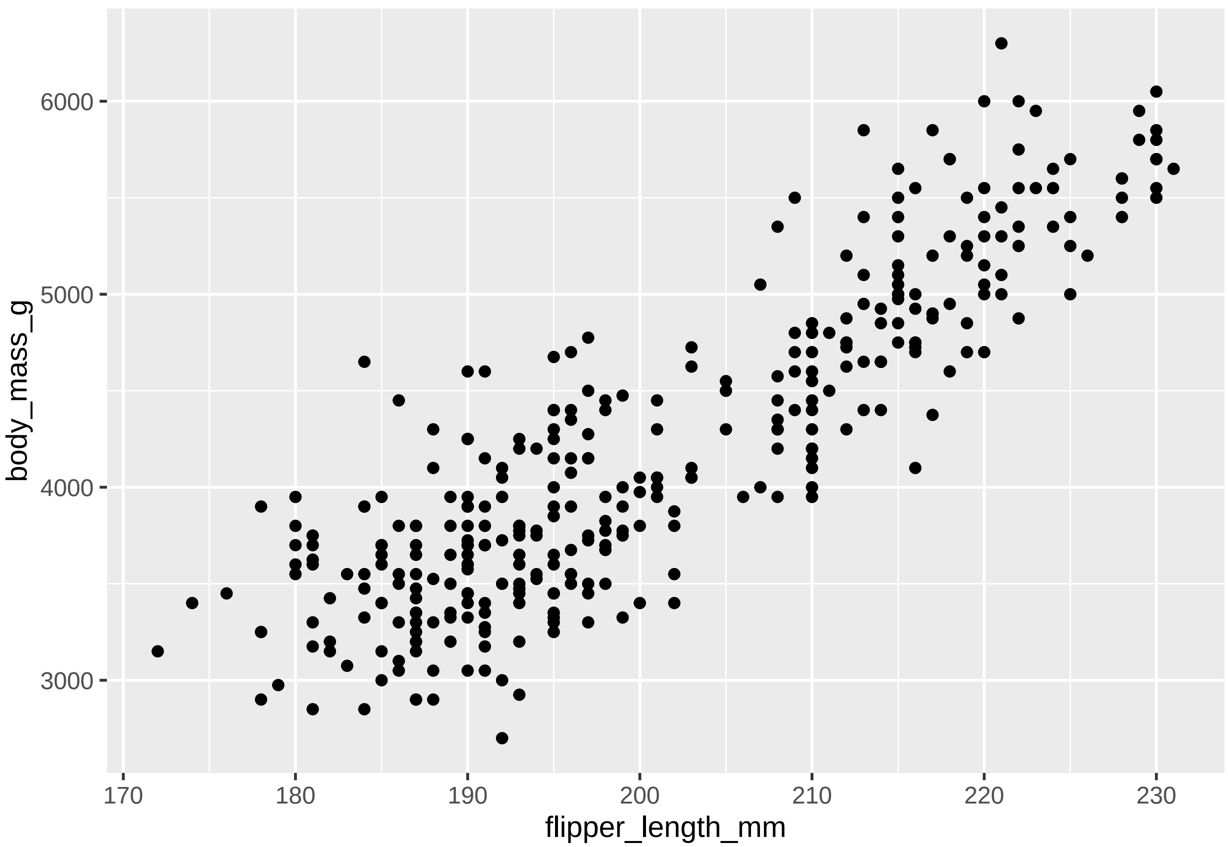
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



In the future will learn about the pipe |>, which allows the following code

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



1.4 Visualizing distributions

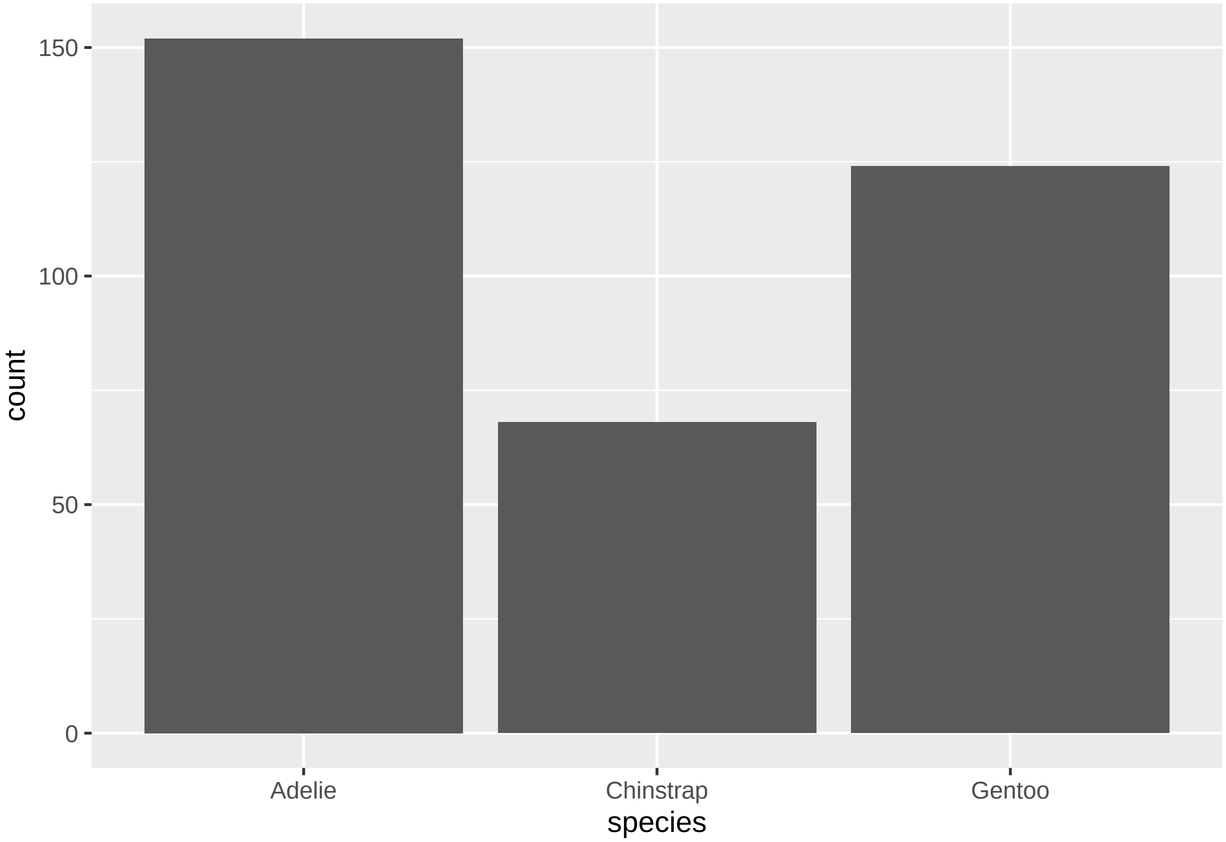
How you visualize depends on whether variable is categorical or numerical

1.4.1 A categorical variable

Categorical variable takes one of a small set of values

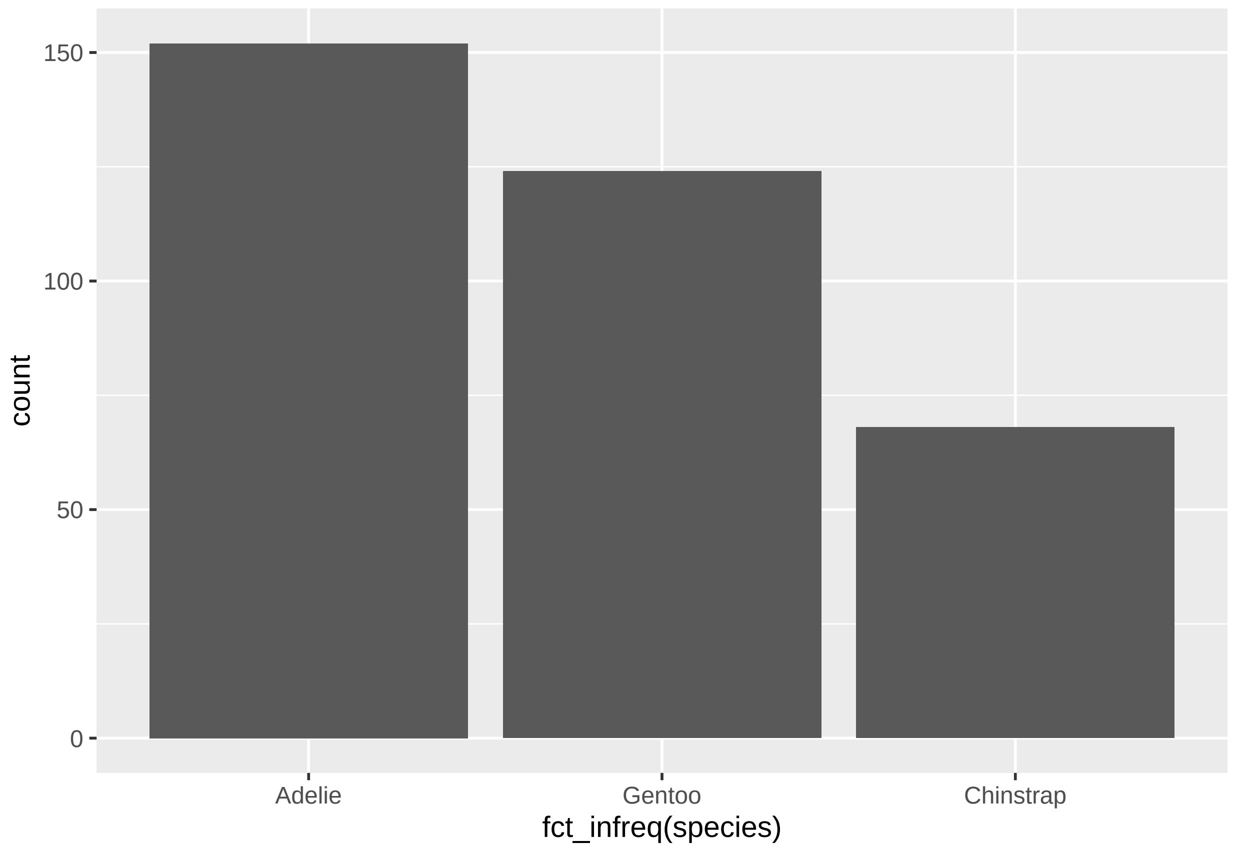
To examine distribution, can use a bar chart

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



When levels are non-ordered, often preferable to reorder bars based on frequency, which requires transforming variable to a factor (how R handles categorical data) then reordering the levels

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



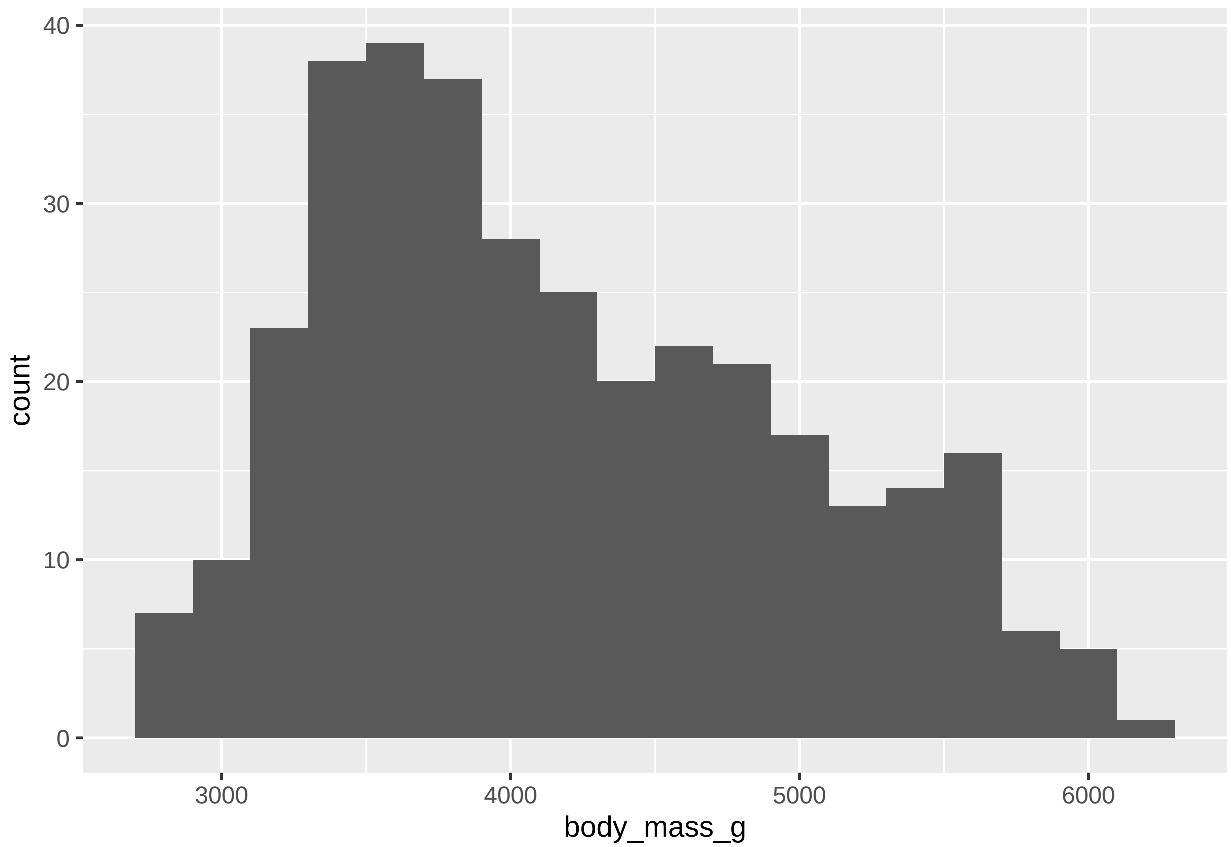
1.4.2 A numerical variable

Quantitative variable can take wide range of numeric values and it makes sense to add/subtract/average those values, it can also can be continuous or discrete

One common visualiziation for continuous is histogram

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

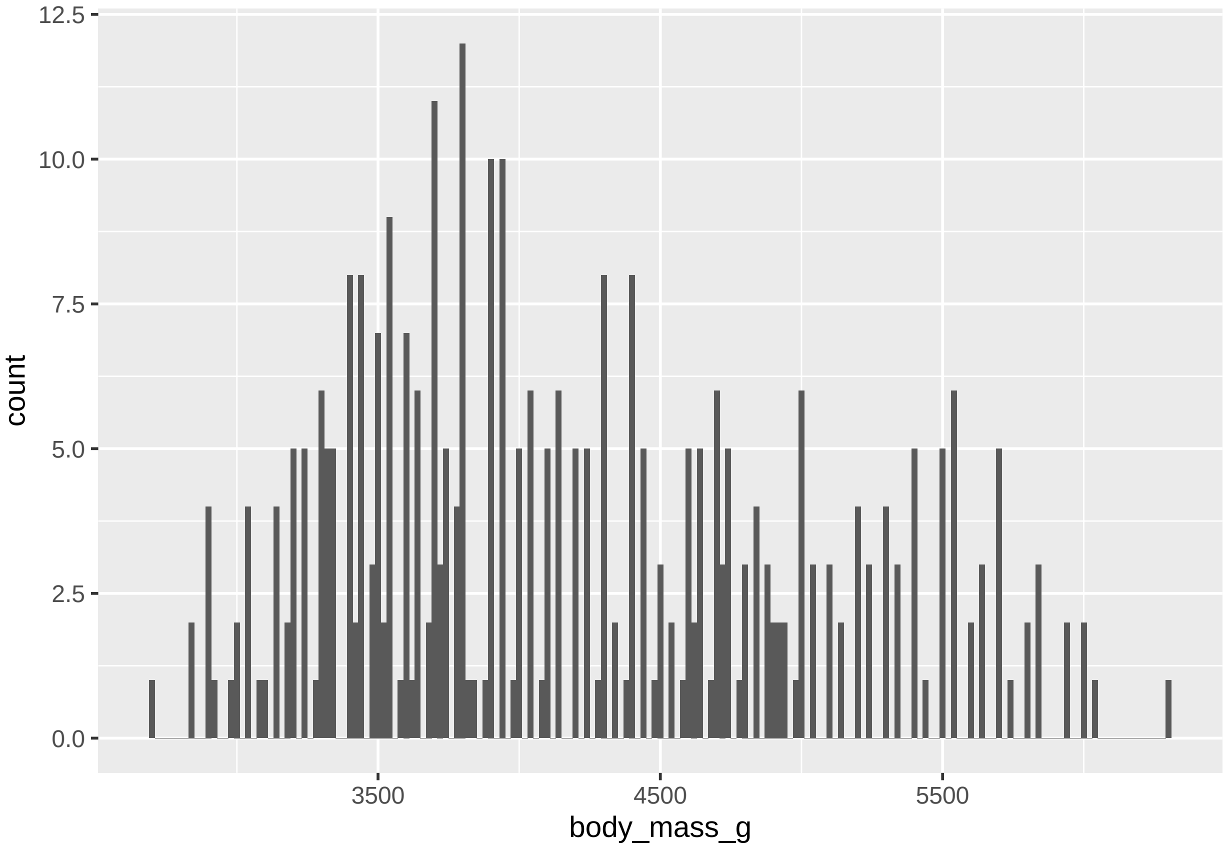
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



Need to find balance in binwidth

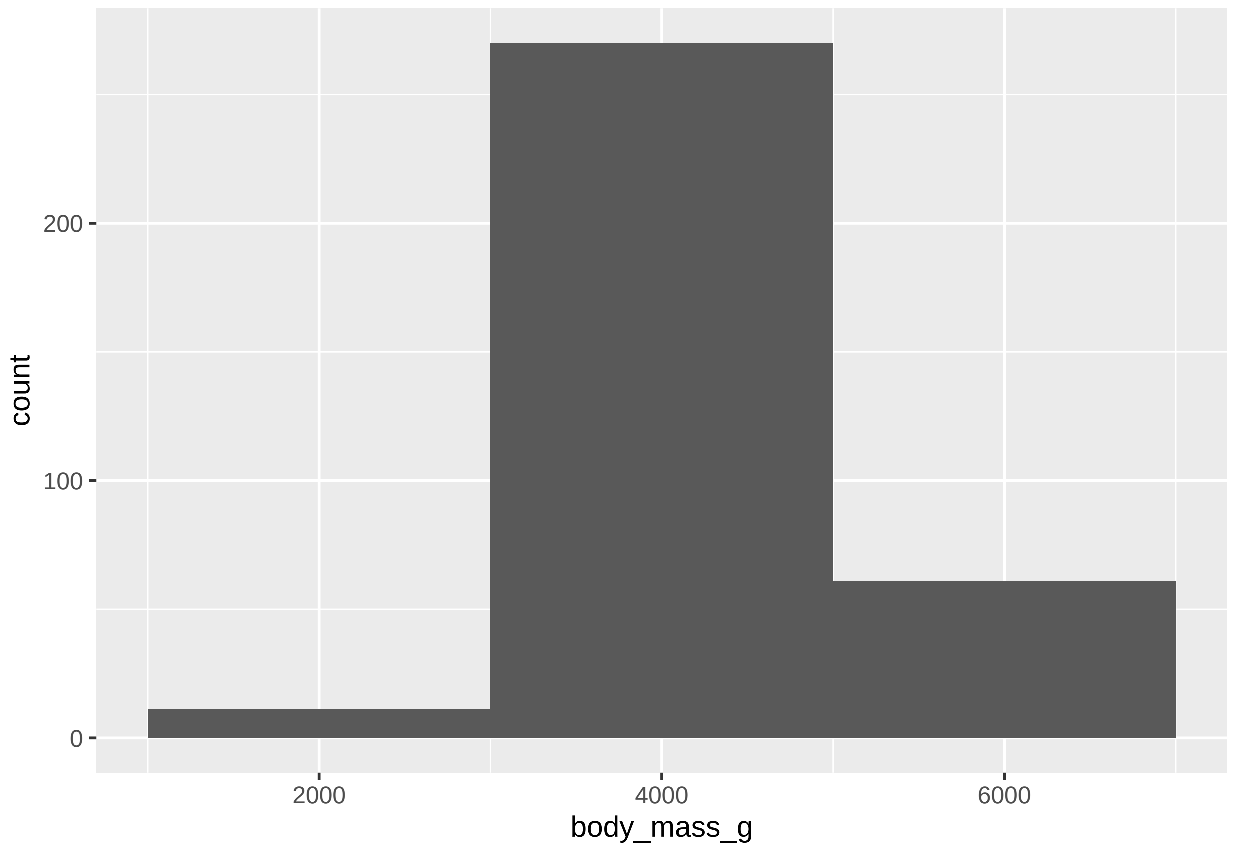
```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 20)
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



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

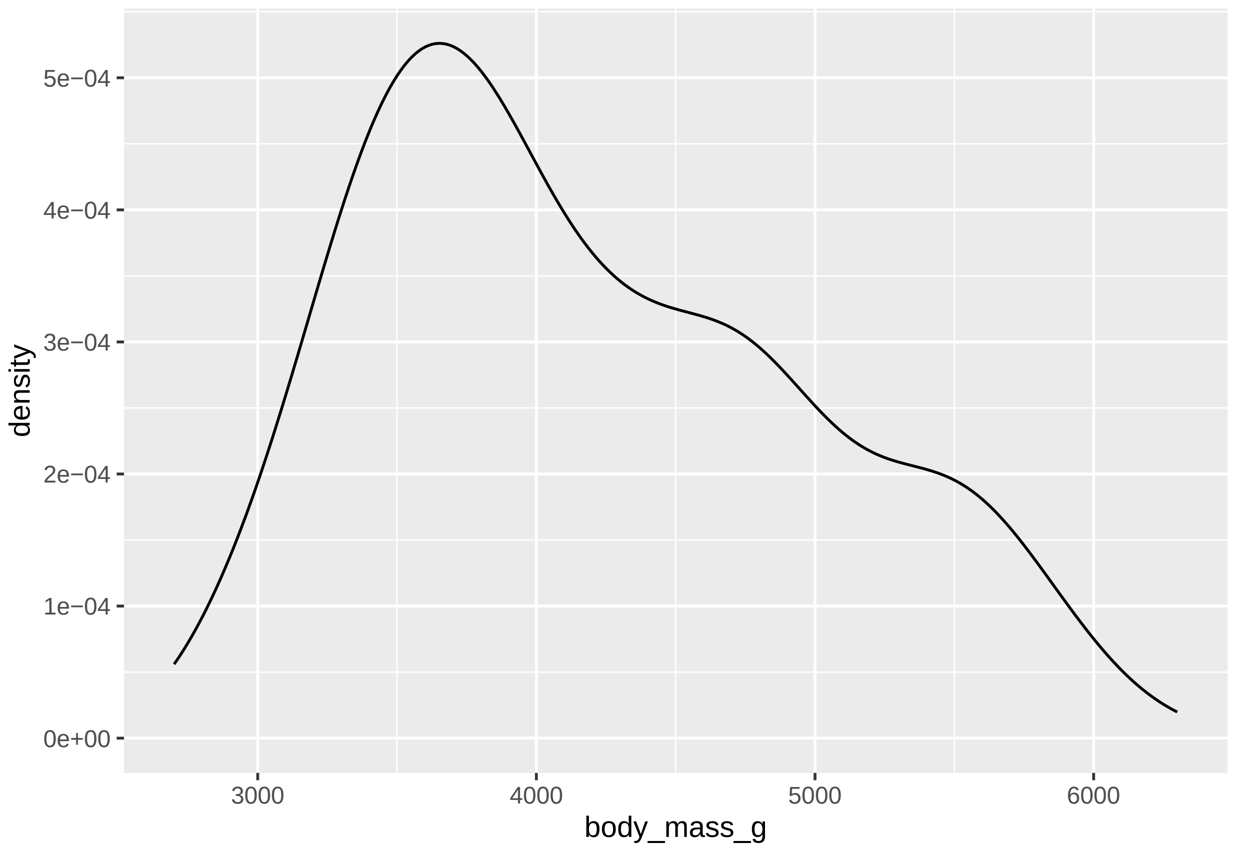
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



An alternative is a density plot, smoothed-out version of histogram

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

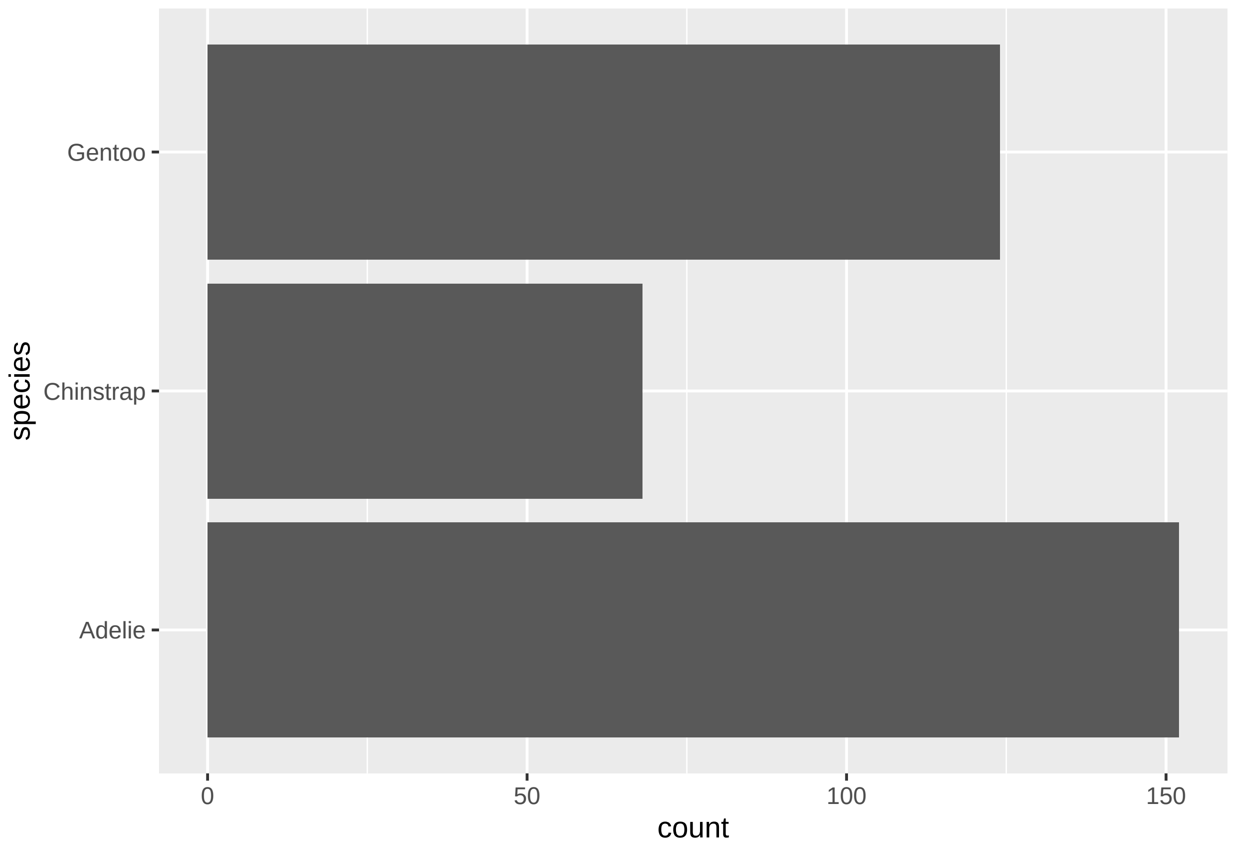
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



1.4.3 Exercises

1. Make a bar plot of species of penguin with species assigned to y aesthetic

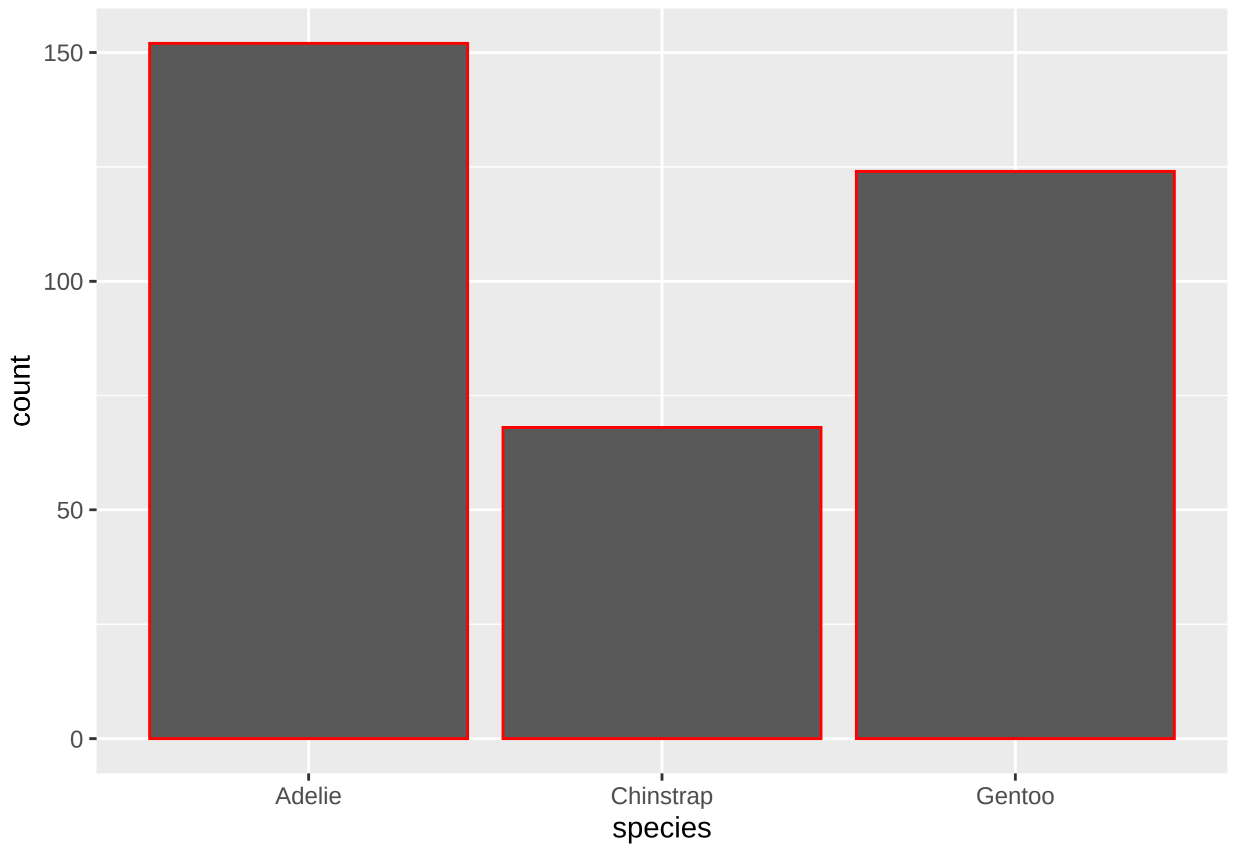
```
ggplot(penguins, aes(y = species)) +
  geom_bar()
```



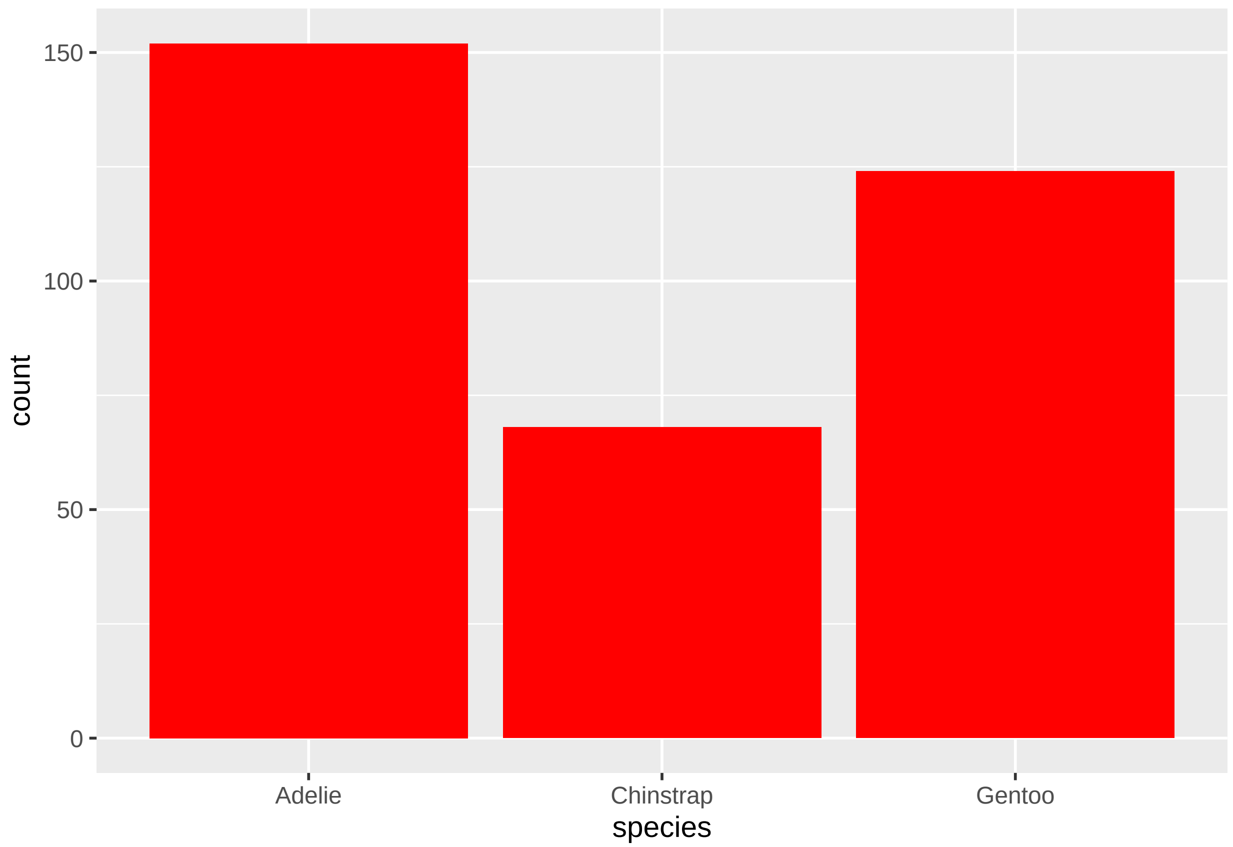
The bar plot is now horizontal instead of vertical

2. How are the two plots different? Which aesthetic more useful to change the color of the bars?

```r
ggplot(penguins, aes(x = species)) +
  geom_bar(color = "red")  # Plot 1
```



```
ggplot(penguins, aes(x = species)) +
  geom_bar(fill = "red")  # Plot 2
```



Fill is more useful, color just changes the bar edge color

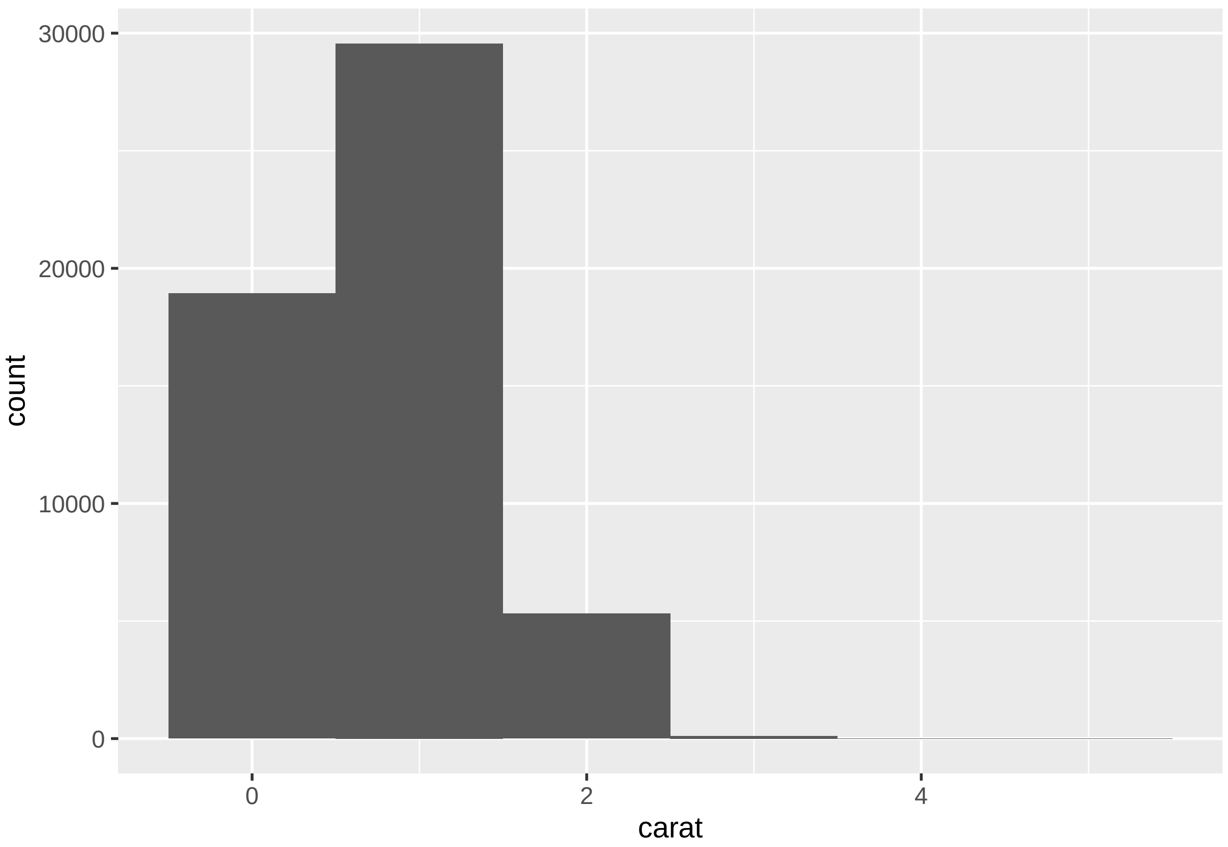
3. What does the bins argument in geom_histogram() do?

```
?geom_histogram()
```

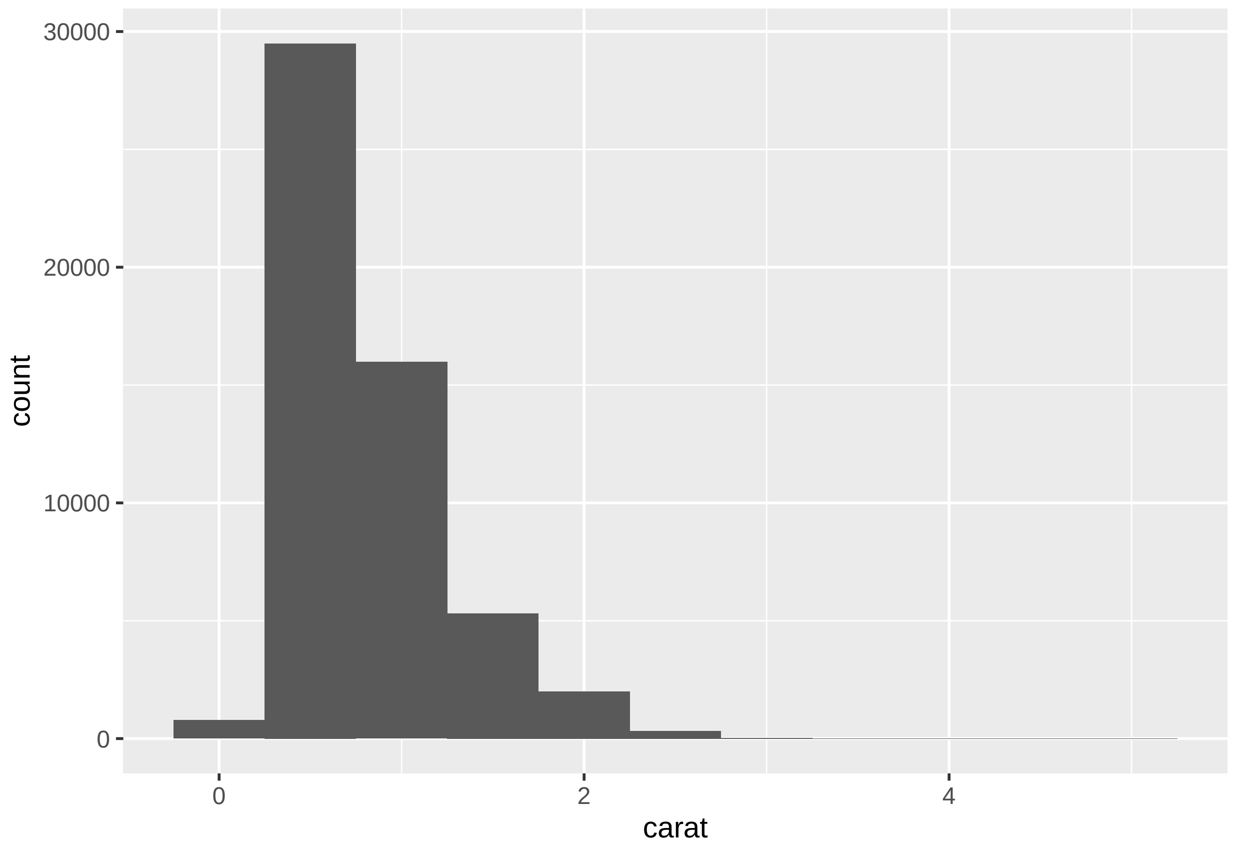
It sets the number of bins in the histogram, it is overridden by binwidth

4. Make a histogram of carat variable in diamonds dataset in tidyverse, experiment with different binwidths

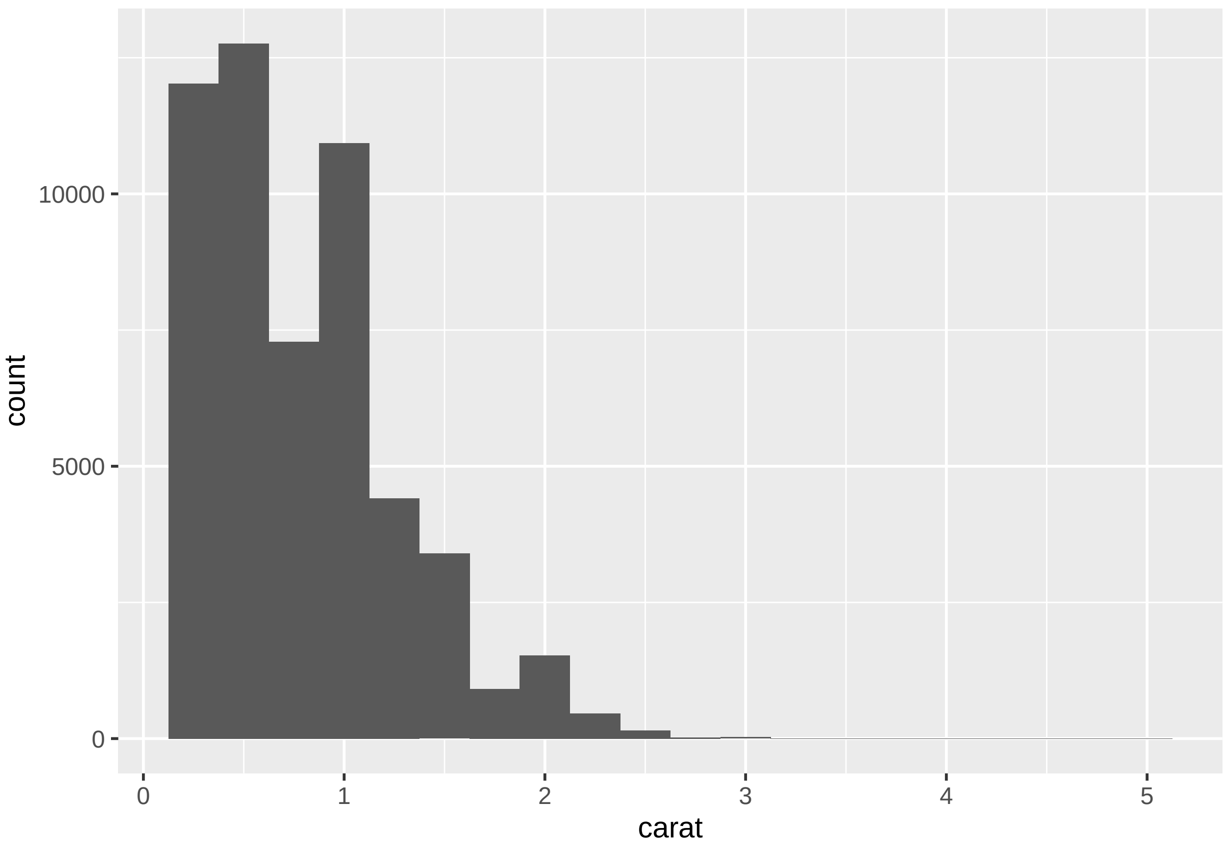
```
ggplot(diamonds, aes(x = carat)) +
  geom_histogram(binwidth = 1)
```



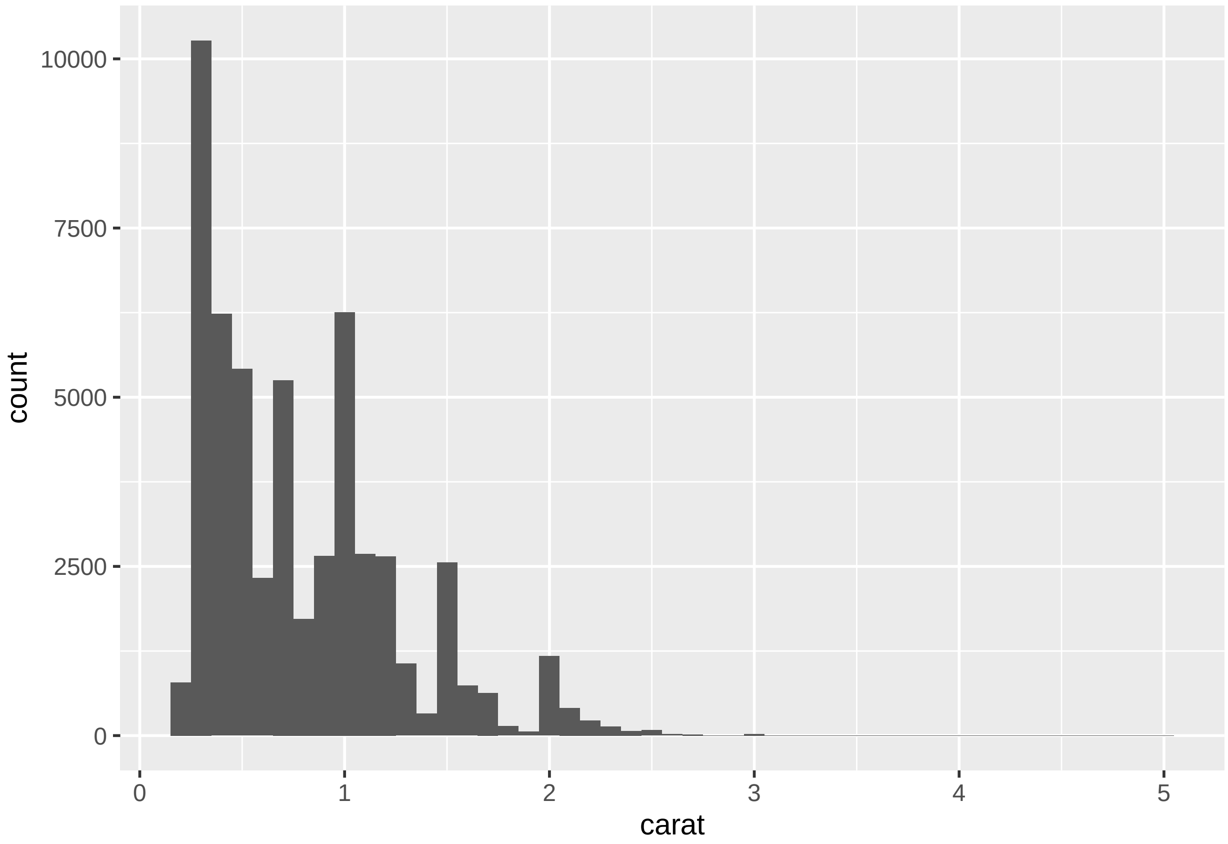
```
ggplot(diamonds, aes(x = carat)) +
  geom_histogram(binwidth = 0.5)
```



```r
ggplot(diamonds, aes(x = carat)) +
  geom_histogram(binwidth = 0.25)
```



```
ggplot(diamonds, aes(x = carat)) +
  geom_histogram(binwidth = 0.1)
```



Using a smaller binwidth (like 0.1) reveals how higher carat values peak at carat values that are divisible by 0.5 and then taper off after

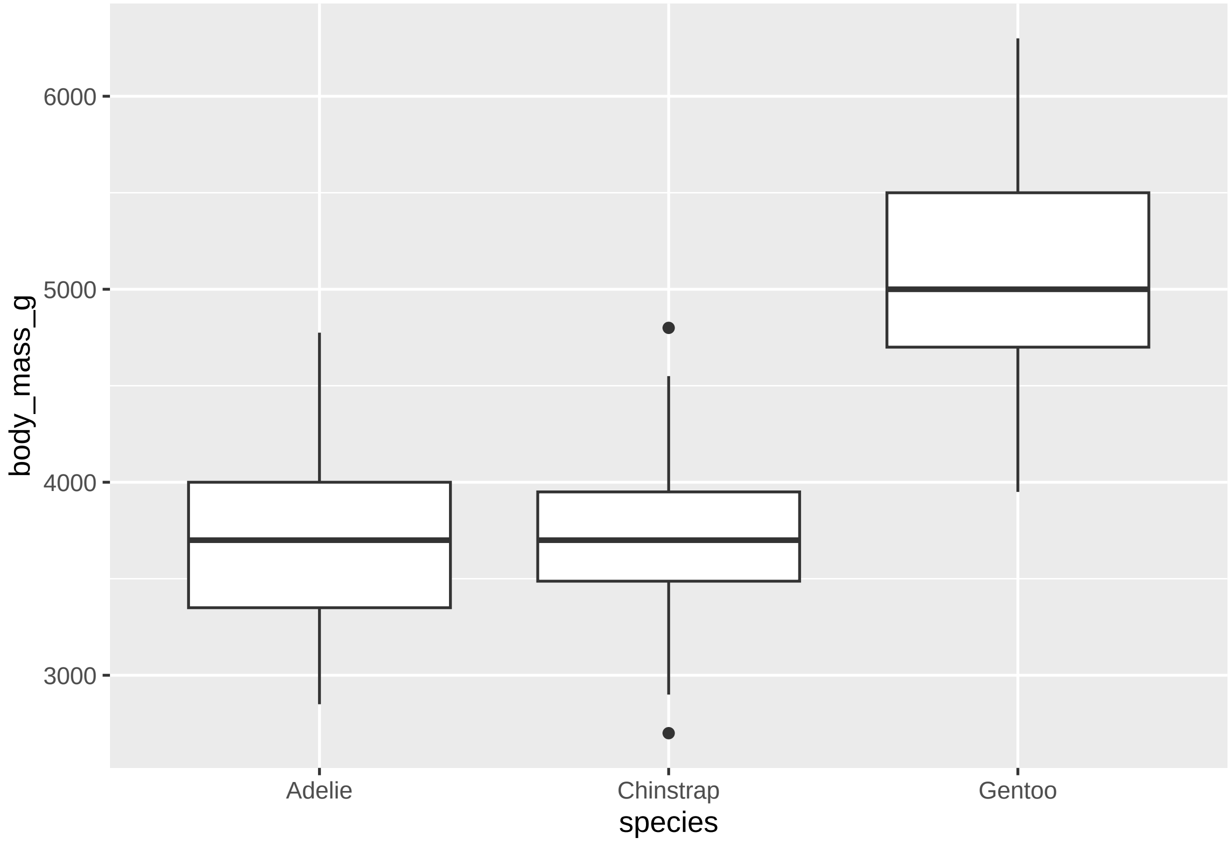
1.5 Visualizing relationships

1.5.1 A numerical and a categorical variable

Can use side-by-side box plot

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

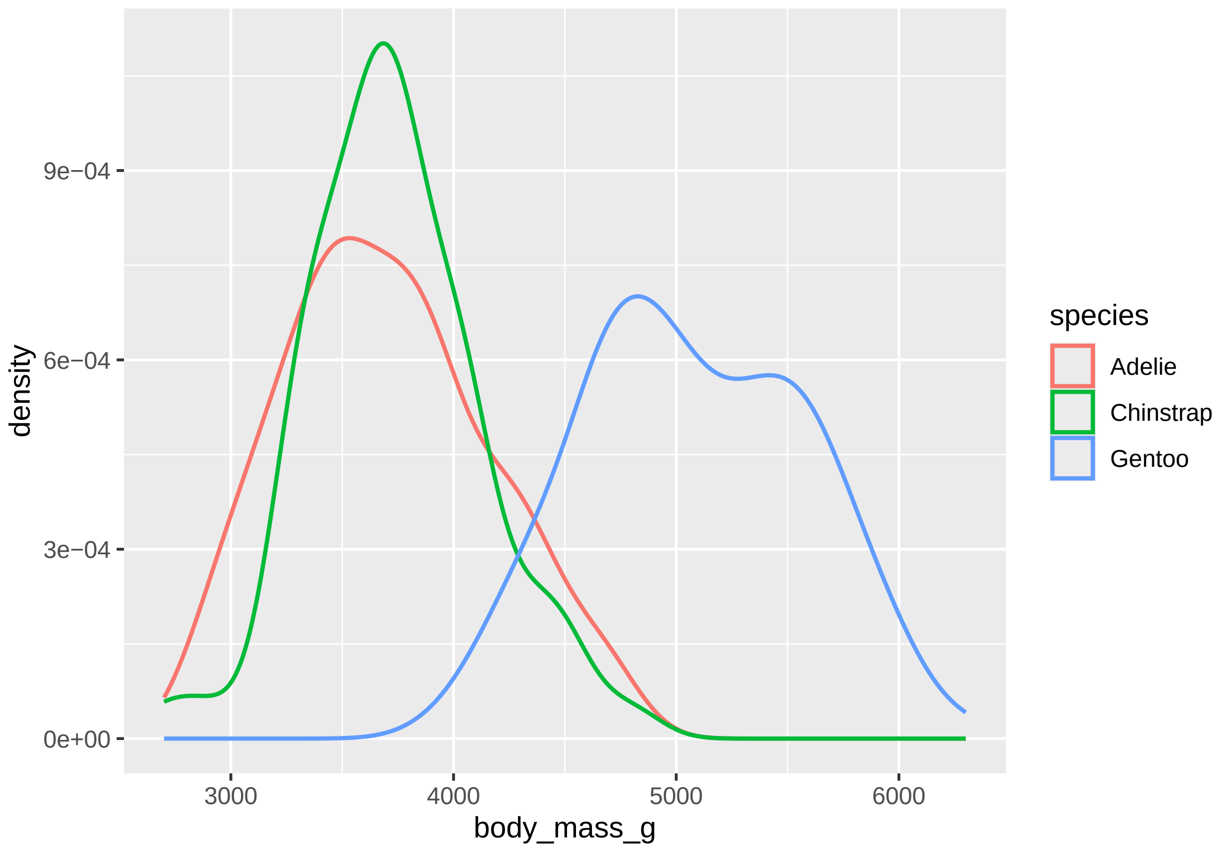
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



Alternatively can make density plots

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

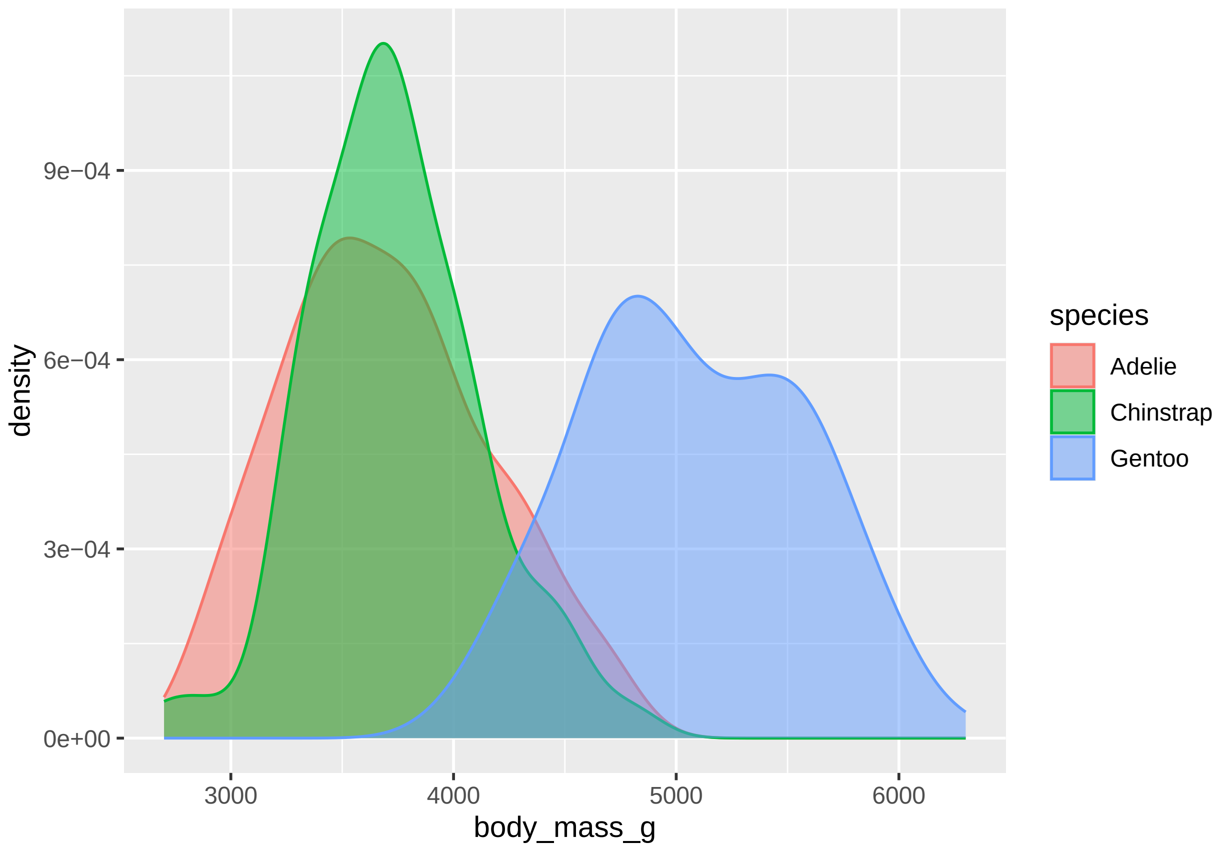
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



Alpha aesthetic adds transparency to filled curves, with 0 being completely transparent and 1 being completely opaque

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

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```

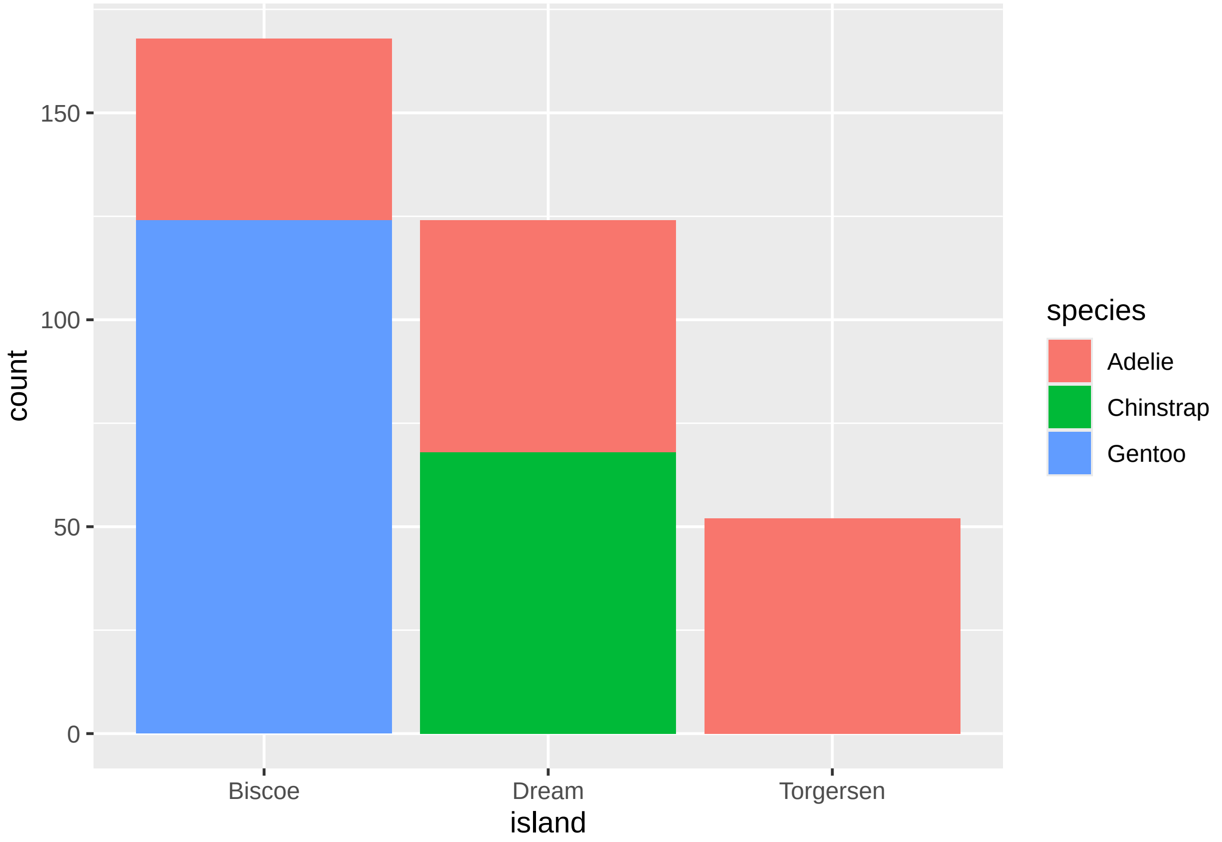


Map variables to aesthetics if want visual attribute to vary based on values of that variable (e.g. color & fill), otherwise set value of an aesthetic (e.g. alpha)

1.5.2 Two categorical variables

Can use stacked bar plots, for example to visualize distribution of species within each island

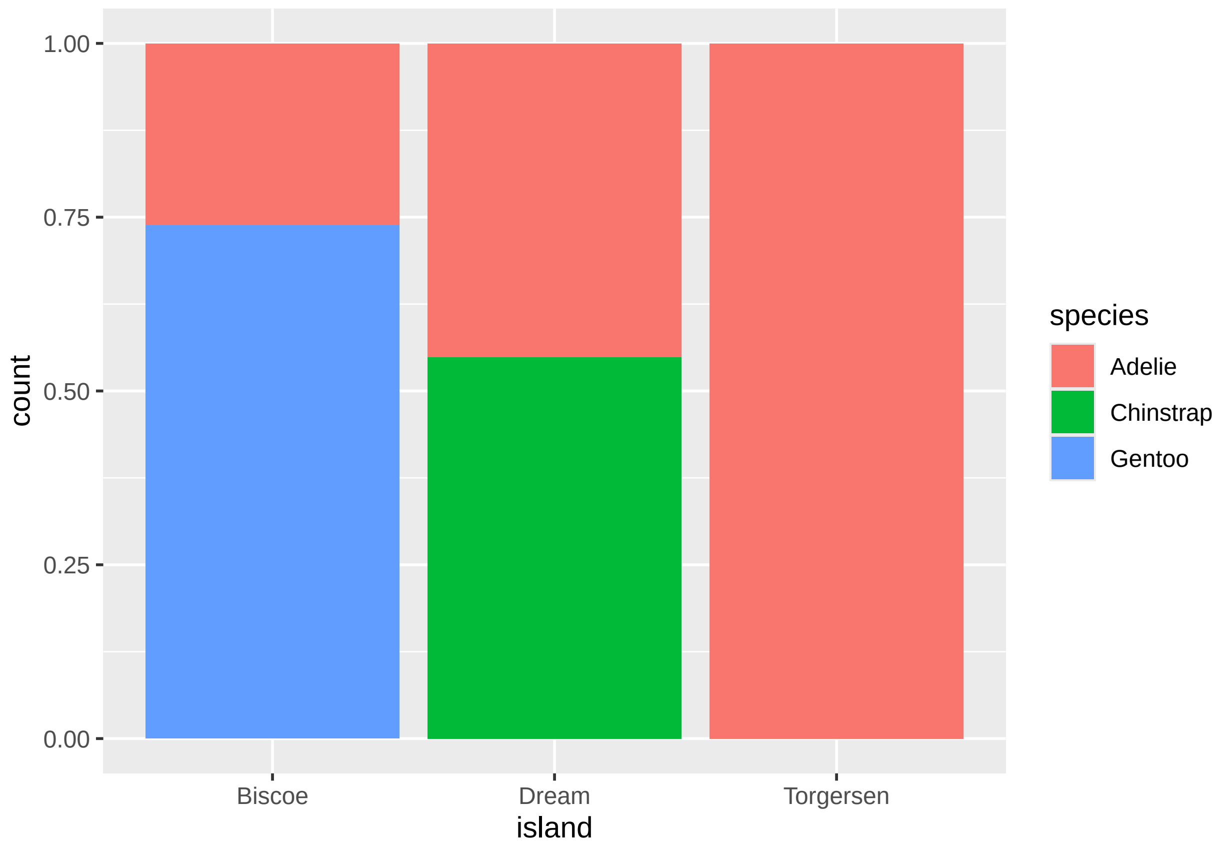
```r
ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar()
```



Frequencies shows roughly equal number of Adelie on each island, but not sure about percentages of species on each island

By setting position to "fill" in the geom, can compare species distributions across islands

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



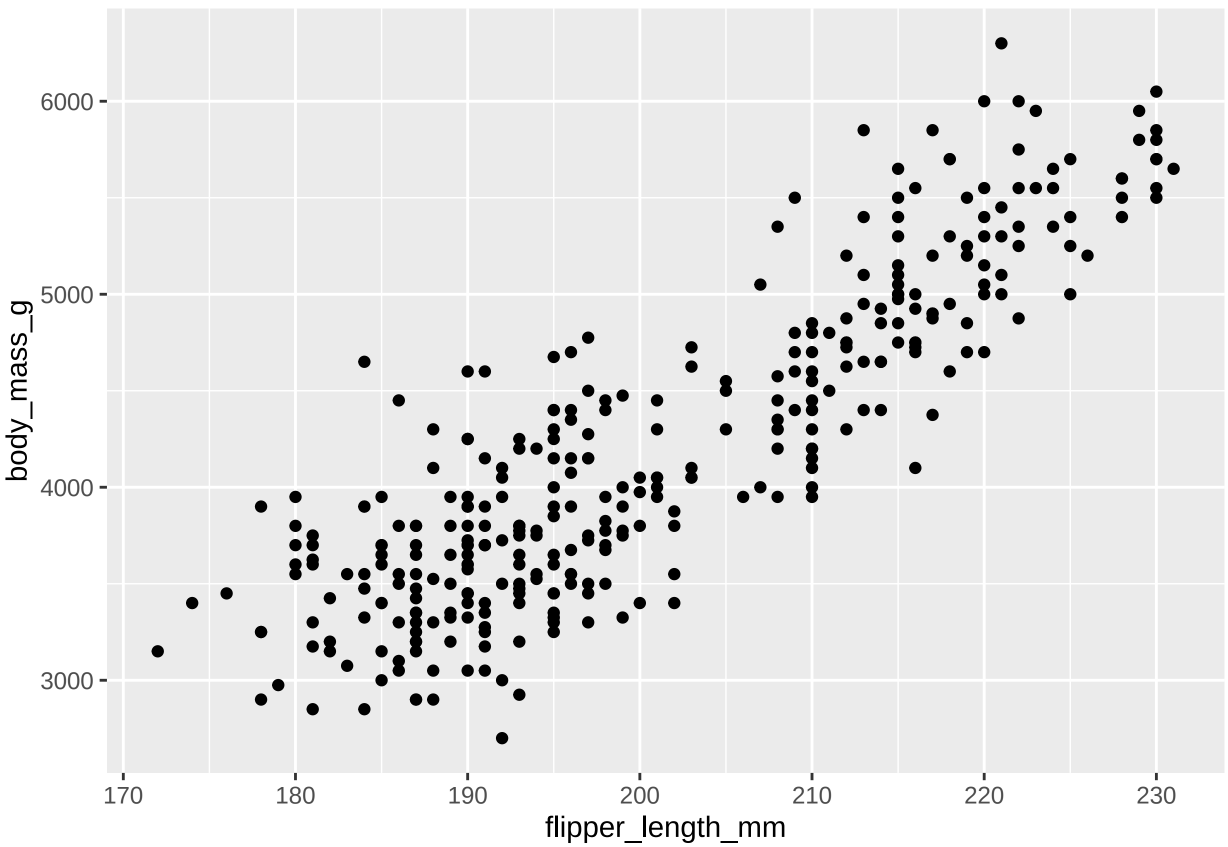
Map the variable separated into bars to the x aesthetic, and the variable that will change the colors of the bars to the fill aesthetic

1.5.3 Two numerical variables

Scatterplot probably most common

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

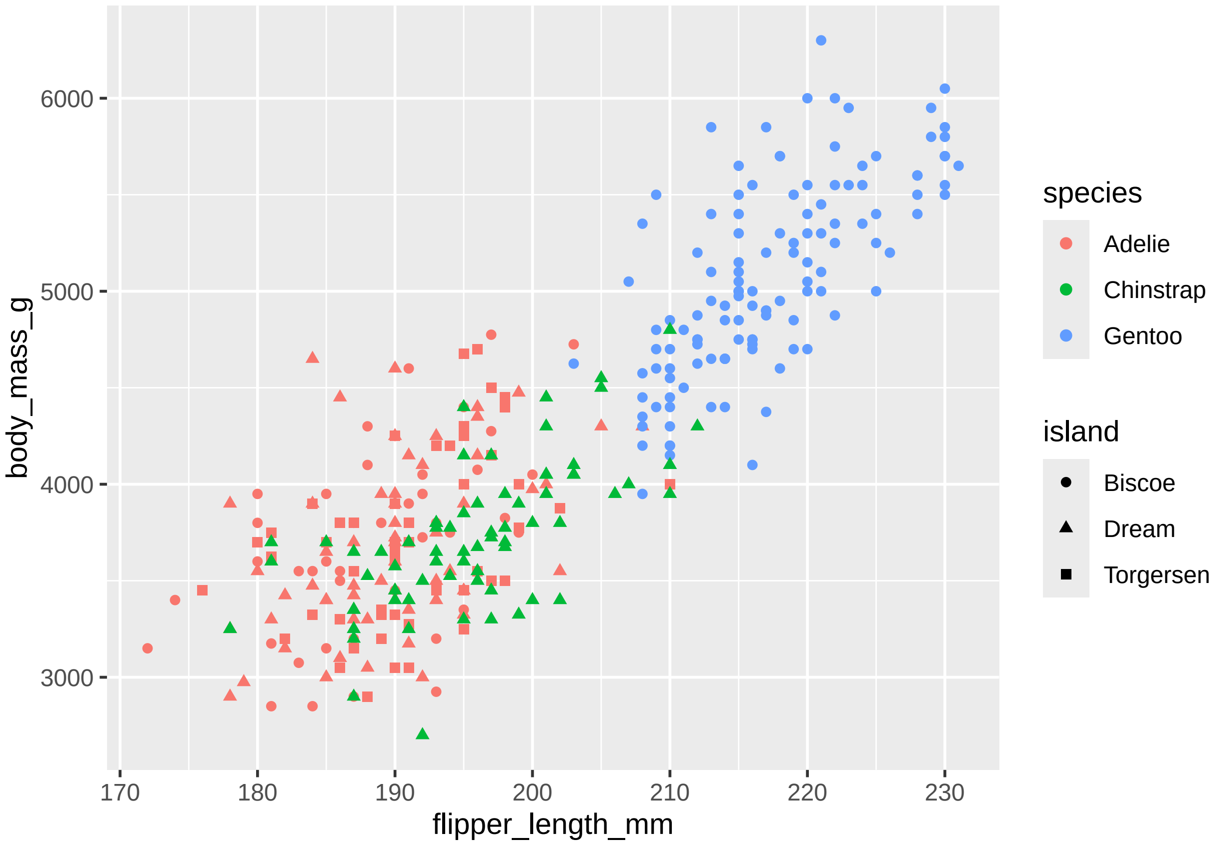


1.5.4 Three or more variables

Can incorporate variables by mapping them to additional aesthetics

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

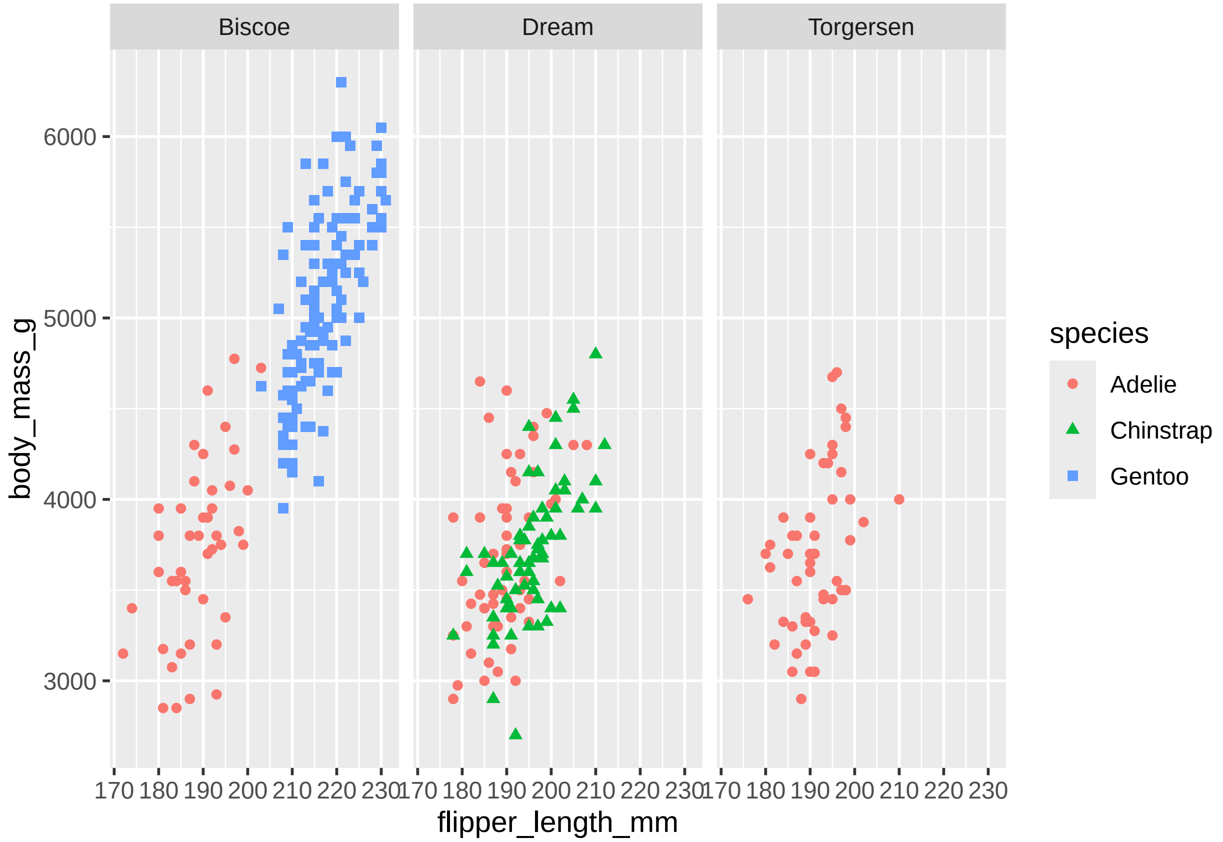


Adding too many aesthetics to one plot can make it difficult to read Another way, primarily for categorical variables, is to split plot into facets or subplots displaying a subset of the data

Use facet_wrap() which takes a formula (~cat_variable) as a first argument

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



1.5.5 Exercises

1. Which variables in mpg are categorical? Which are numerical?

```
?mpg
```

Categorical: manufacturer, model name transmission type, type of drive train, fuel type, and class of car
Numerical: displacement, year, cylinders, city miles per gallon, highway miles per gallon

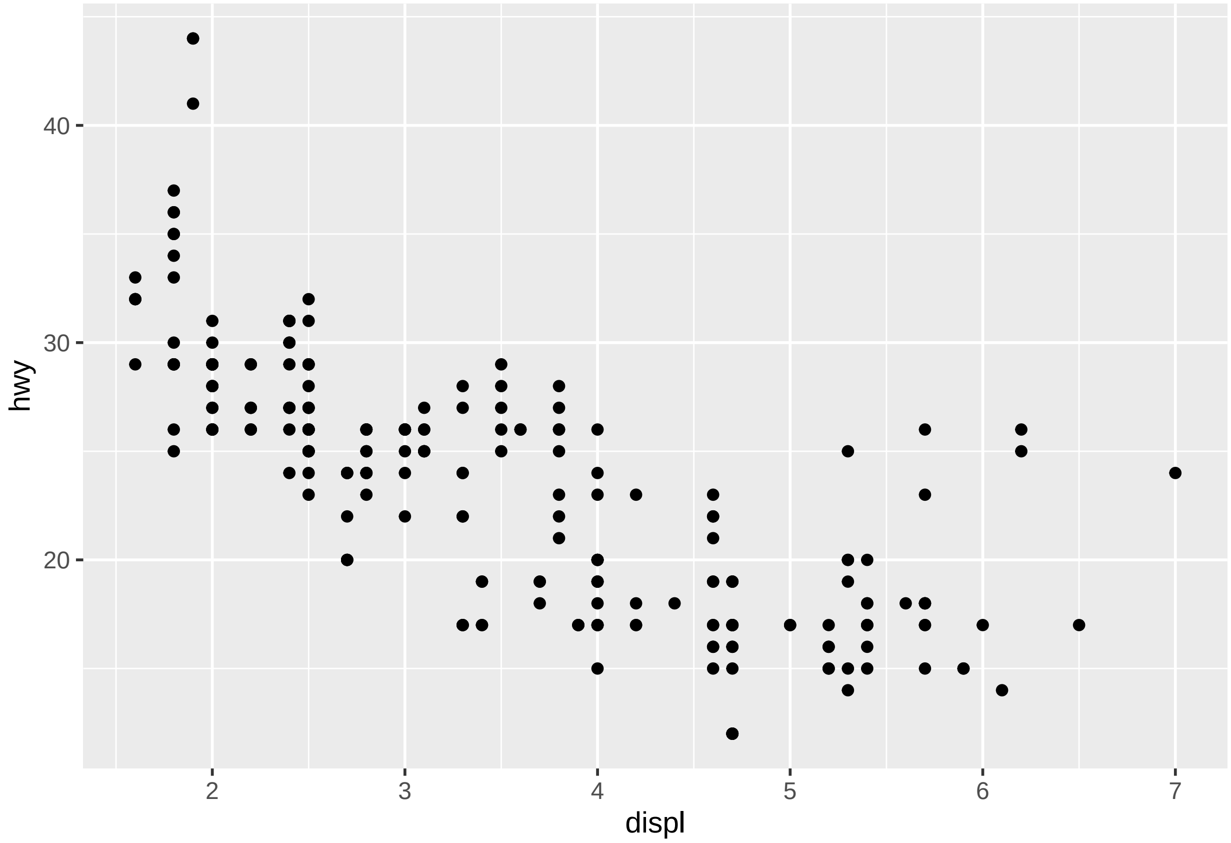
```
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4           1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4           2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4           2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4           2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4           2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4           3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p     comp~
## # i 224 more rows
```

You can see this information when you run mpg by looking at the type of the data in the column, such as a character vector for manufacturer and model (categorical variables) versus double for displacement and integer for cty and hwy (numerical variables)

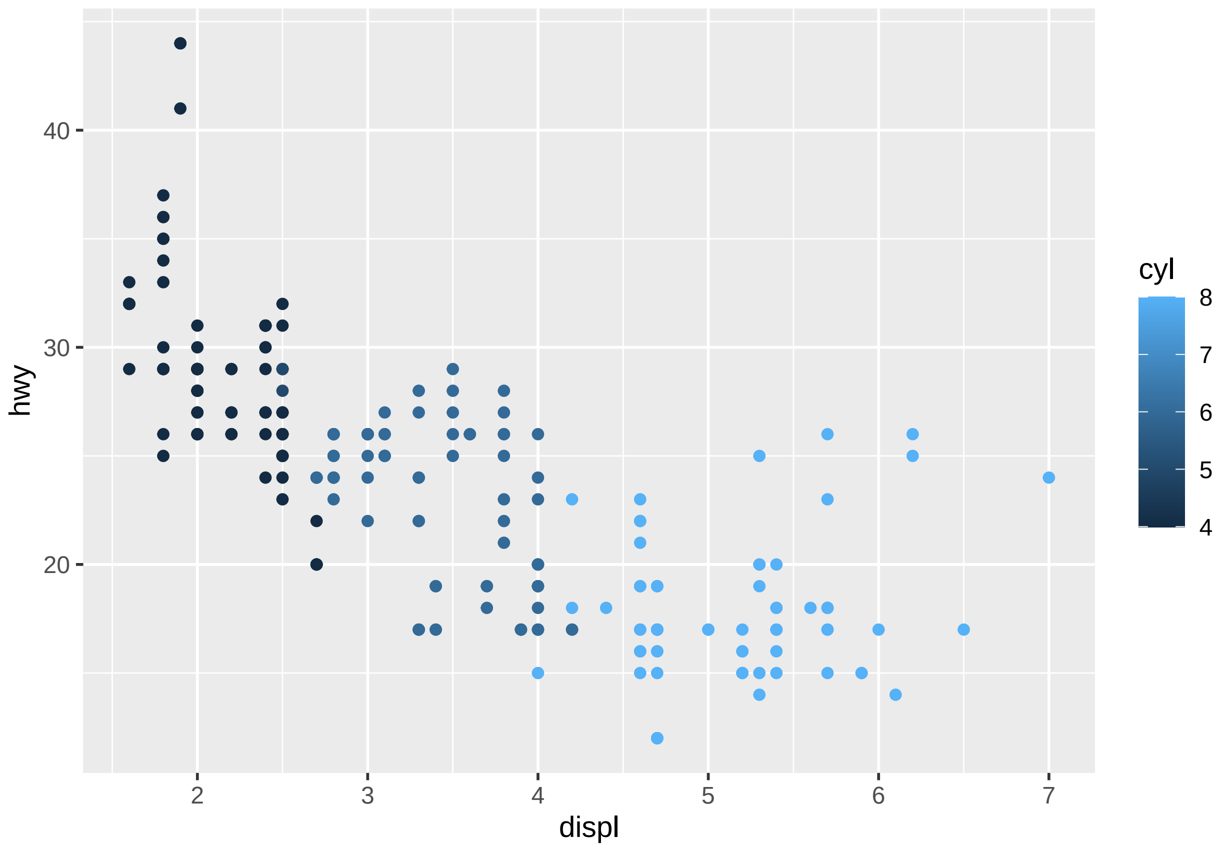
2. Make a scatterplot of hwy vs displ

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
```

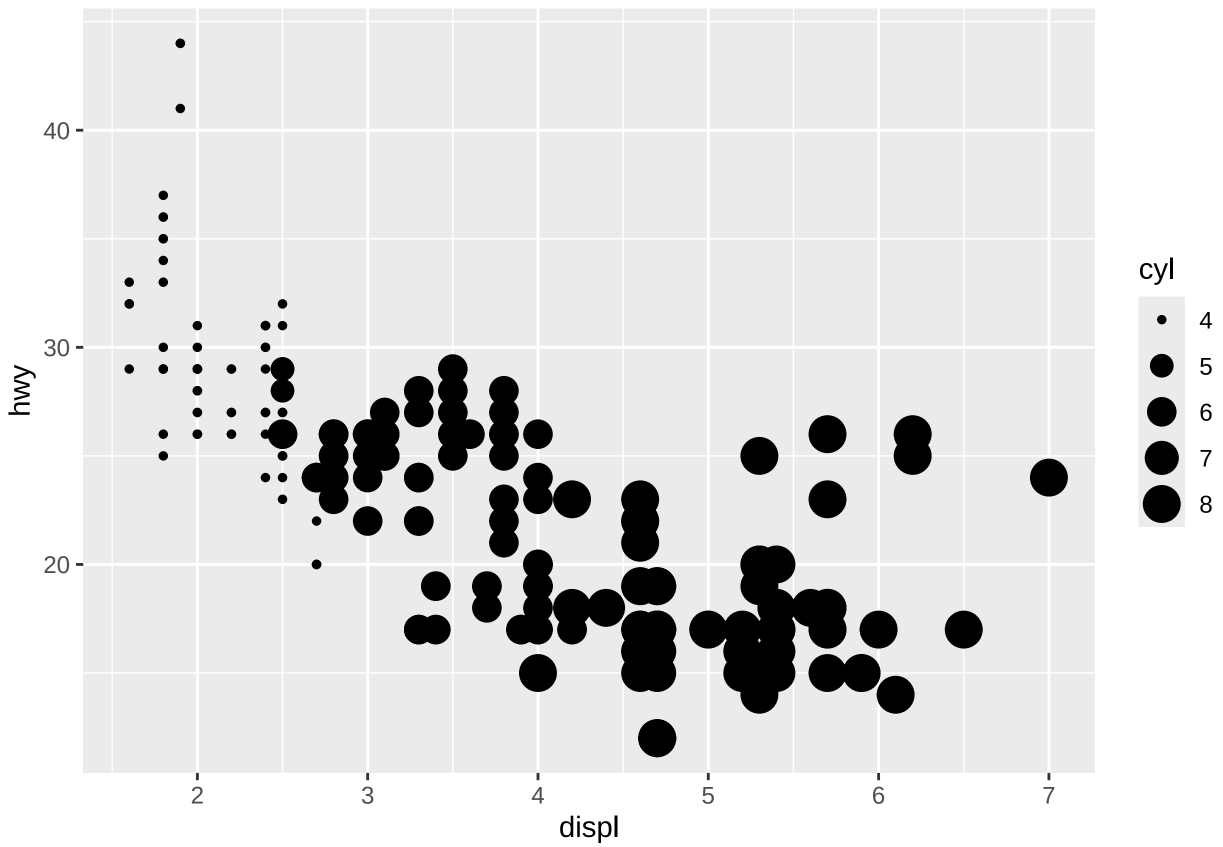


Next, map a third, numerical variable to color, then size, then color and size, then shape

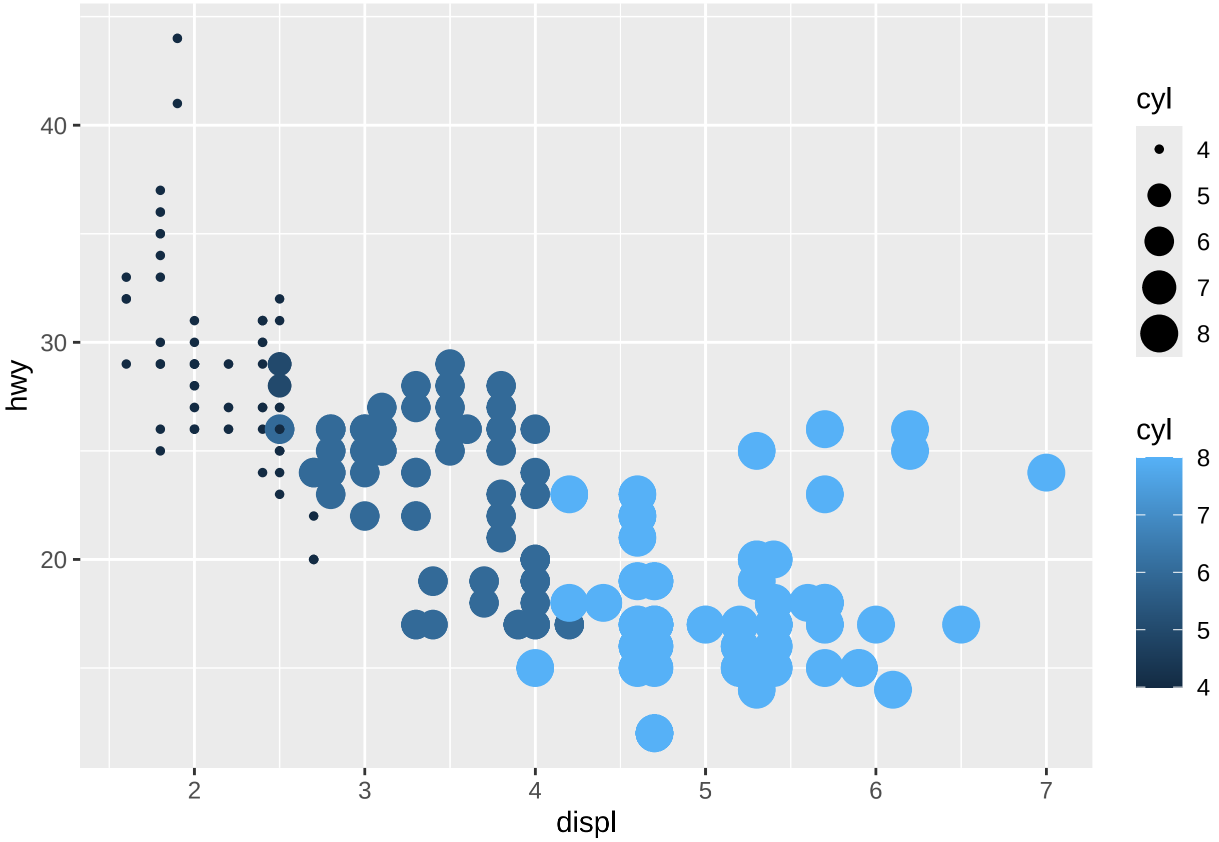
```
ggplot(mpg, aes(x = displ, y = hwy, color = cyl)) +
  geom_point()
```



```
ggplot(mpg, aes(x = displ, y = hwy, size = cyl)) +
  geom_point()
```



```r
ggplot(mpg, aes(x = displ, y = hwy, color = cyl, size = cyl)) +
  geom_point()
```

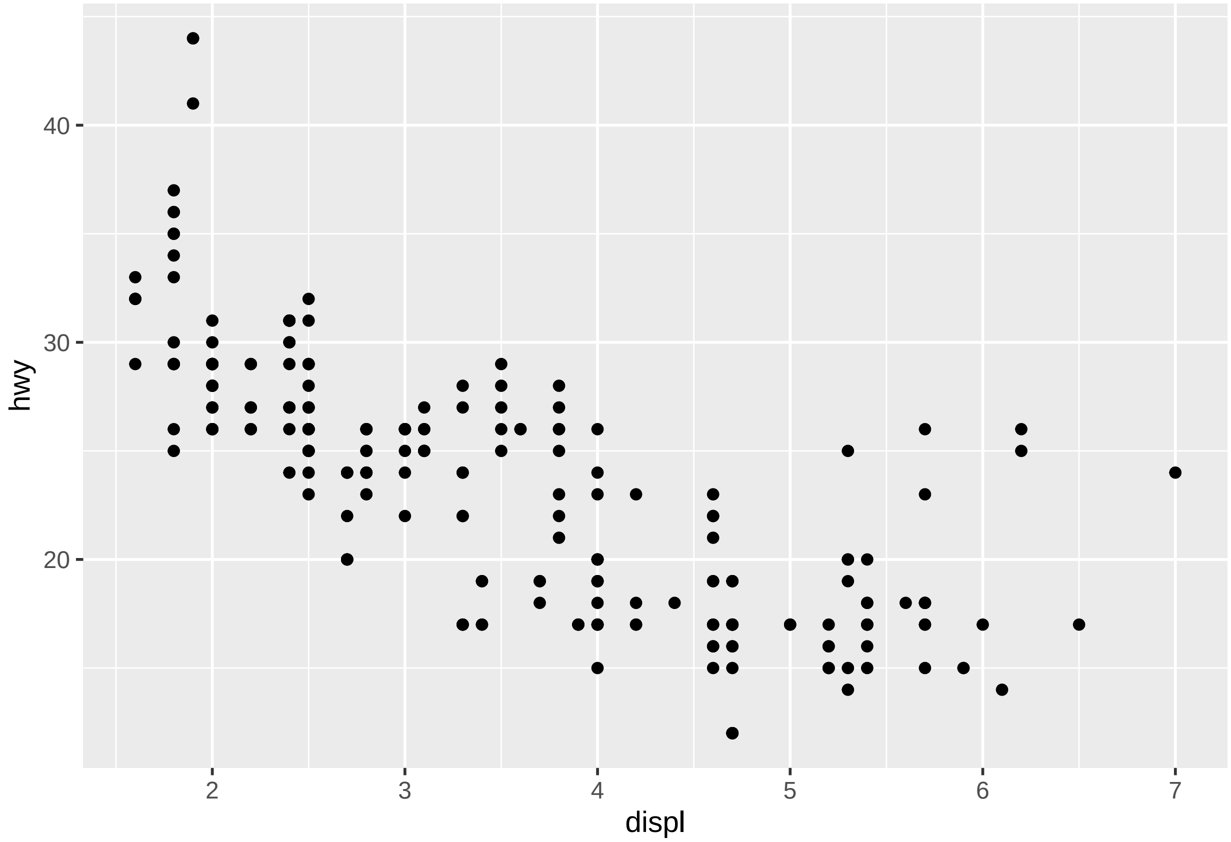


```
#ggplot(mpg, aes(x = displ, y = hwy, shape = cyl)) +
  #geom_point()
```

These aesthetics behave similarily between categorical and numerical variables except for shape which does not allow for a numerical variable For numerical variables, the values are split up into different buckets which behave the same as different values in a categorical variable

3. What happens if you map a third variable to linewidth?

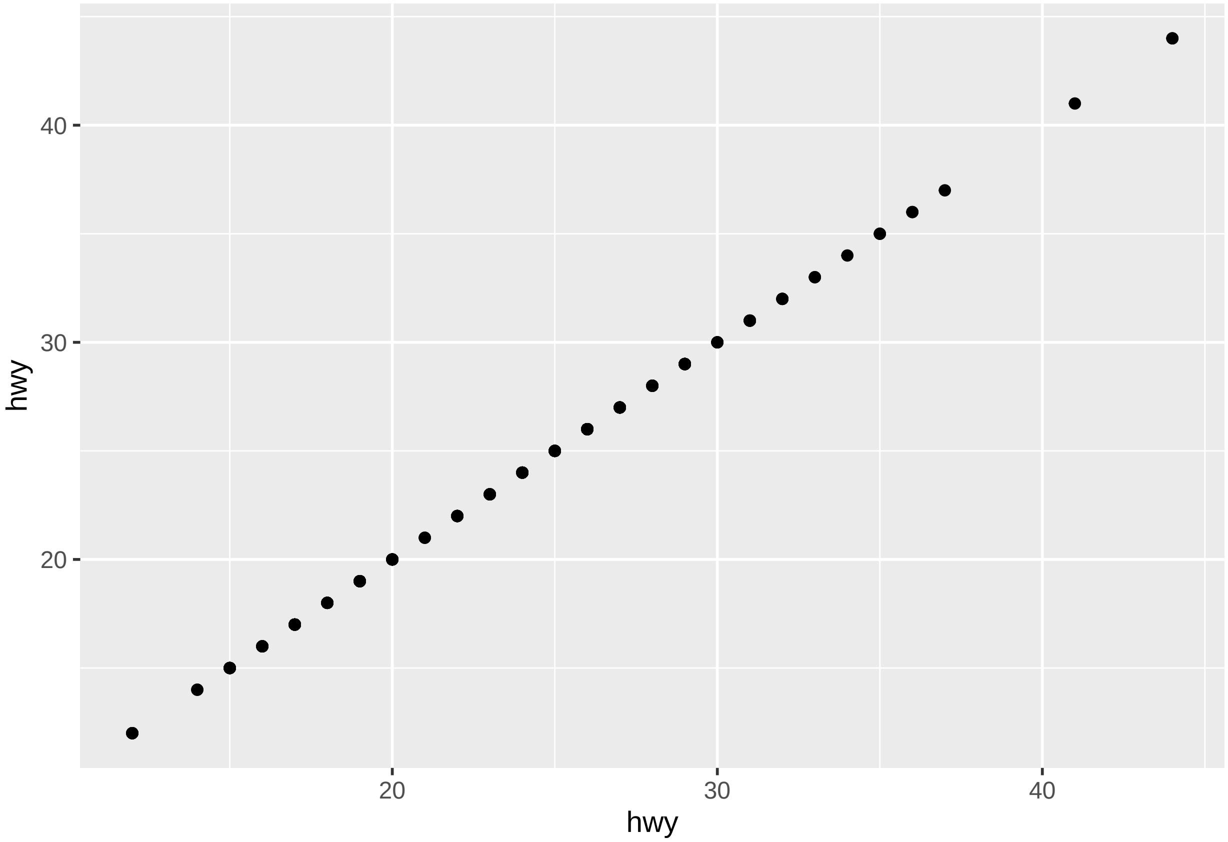
```
ggplot(mpg, aes(x = displ, y = hwy, linewidth = cyl)) +
  geom_point()
```



Nothing, because there is no line being drawn in a scatterplot

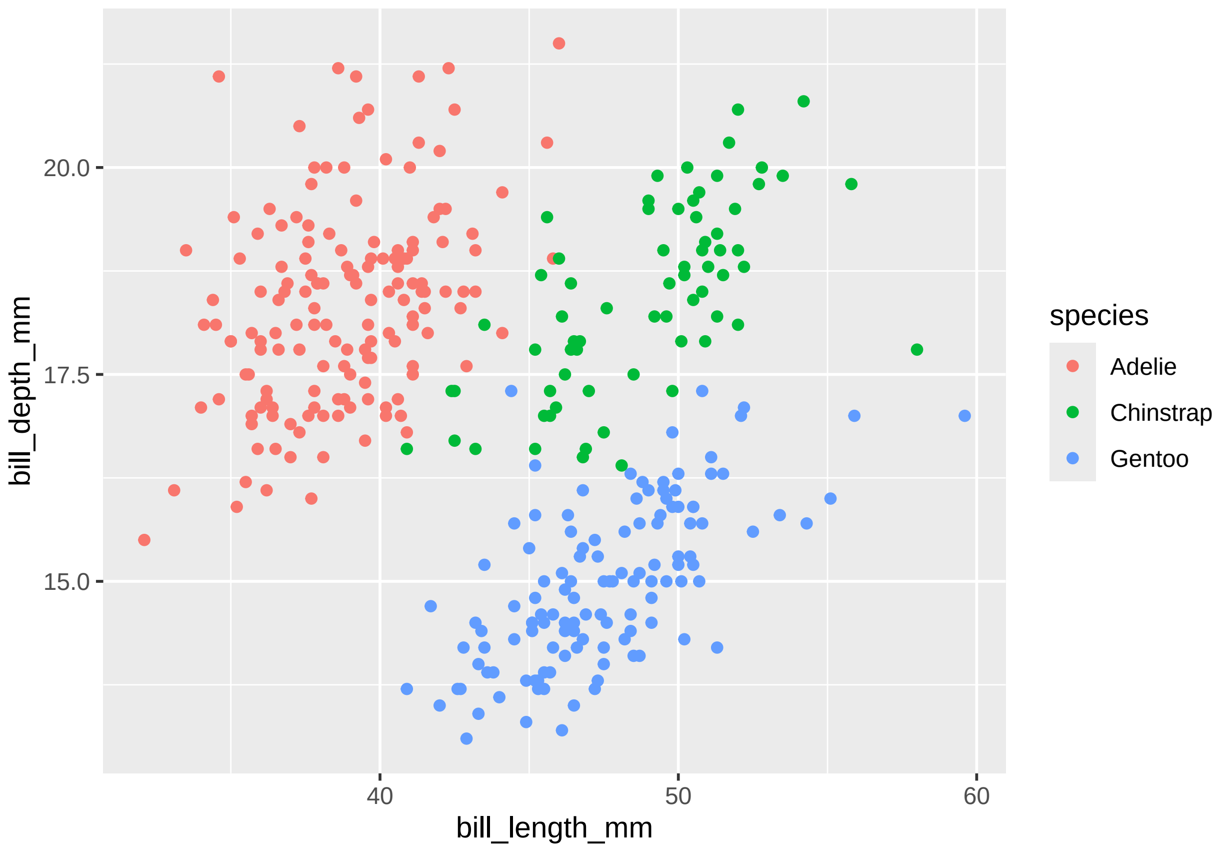
4. What happens if you map the same variable to multiple aesthetics? As seen in the 3rd plot in Exercise #2, variations in the value of that variable are represented aesthetically in two different ways If both x and y are mapped to the same numerical variable in a scatterplot, the plot would be a straight line

```
ggplot(mpg, aes(x = hwy, y = hwy)) +
  geom_point()
```



5. Make a scatterplot of bill_depth_mm vs bill_length_mm and color the points by species

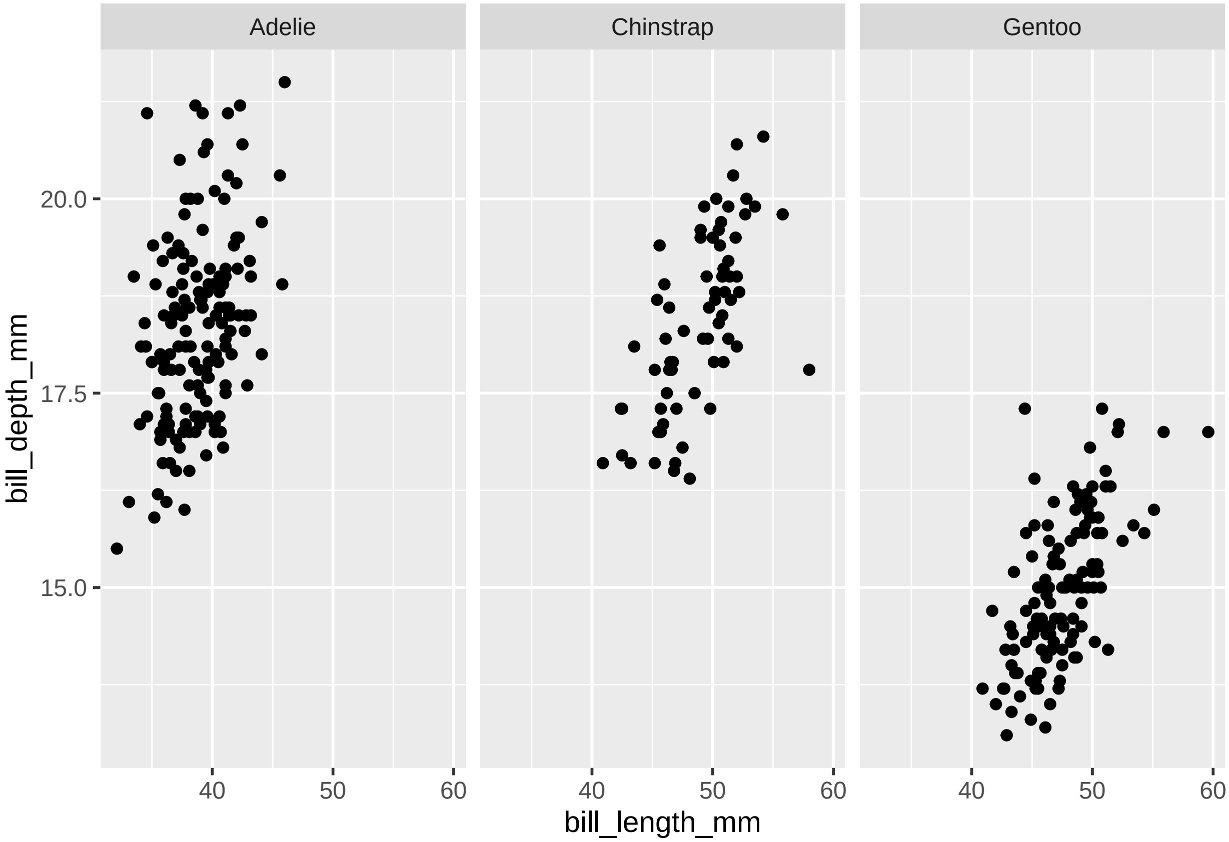
```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point(na.rm = TRUE)
```



What does adding color reveal about the relationship between the two variables? It shows that within each species the relationship is more linear than if you looked at the data for all the species as a whole, you also can see that the range of bill length and depth values for each species is different by how the different colors are divided into different parts of the plot

What about faceting the species?

```r
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point(na.rm = TRUE) +
  facet_wrap(~species)
```

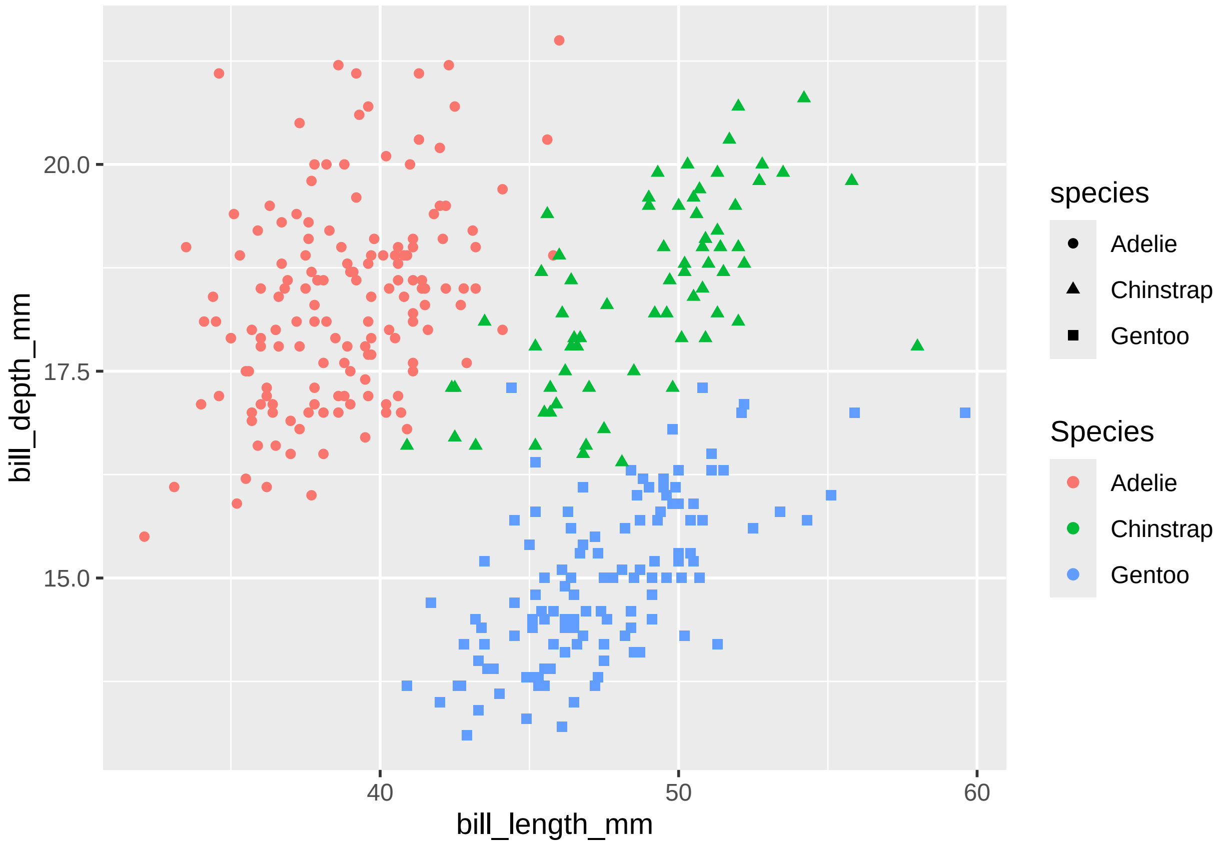


Faceting the species again shows how the relationship is more linear than if you just looked at all the data without differentiating the species in which it would be a messy nonlinear plot Here it is also very clear that Gentoo penguins have less bill depth and that Adelie penguins have lower bill length

6. Why does the following yield two separate legends? How would you fix it to combine the two legends?

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_length_mm, y = bill_depth_mm,
    color = species, shape = species
  )
) +
  geom_point() +
  labs(color = "Species")
```

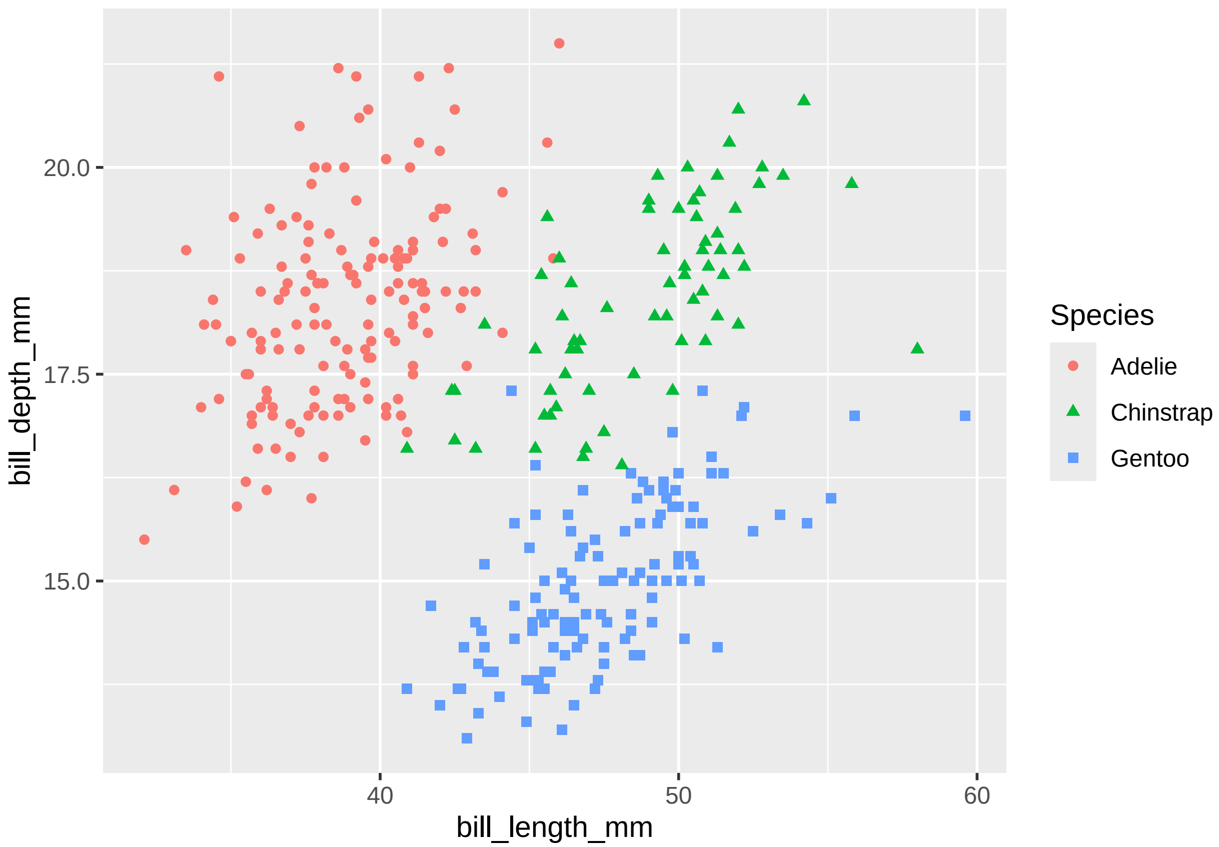
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



If you want the legend labeled Species, you have to also rename the shape label to Species otherwise it will make one called Species with the color and one called species with the shape

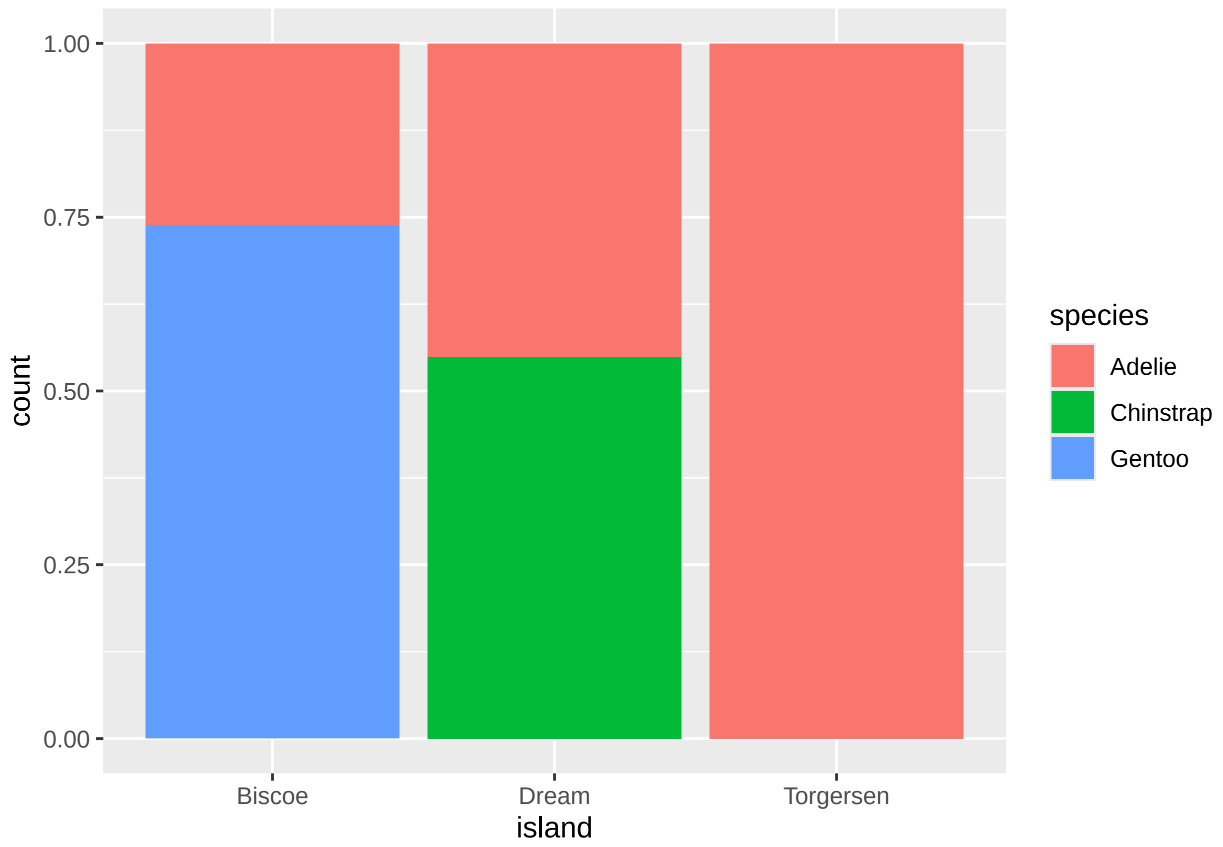
```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_length_mm, y = bill_depth_mm,
    color = species, shape = species
  )
) +
  geom_point() +
  labs(color = "Species", shape = "Species")
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

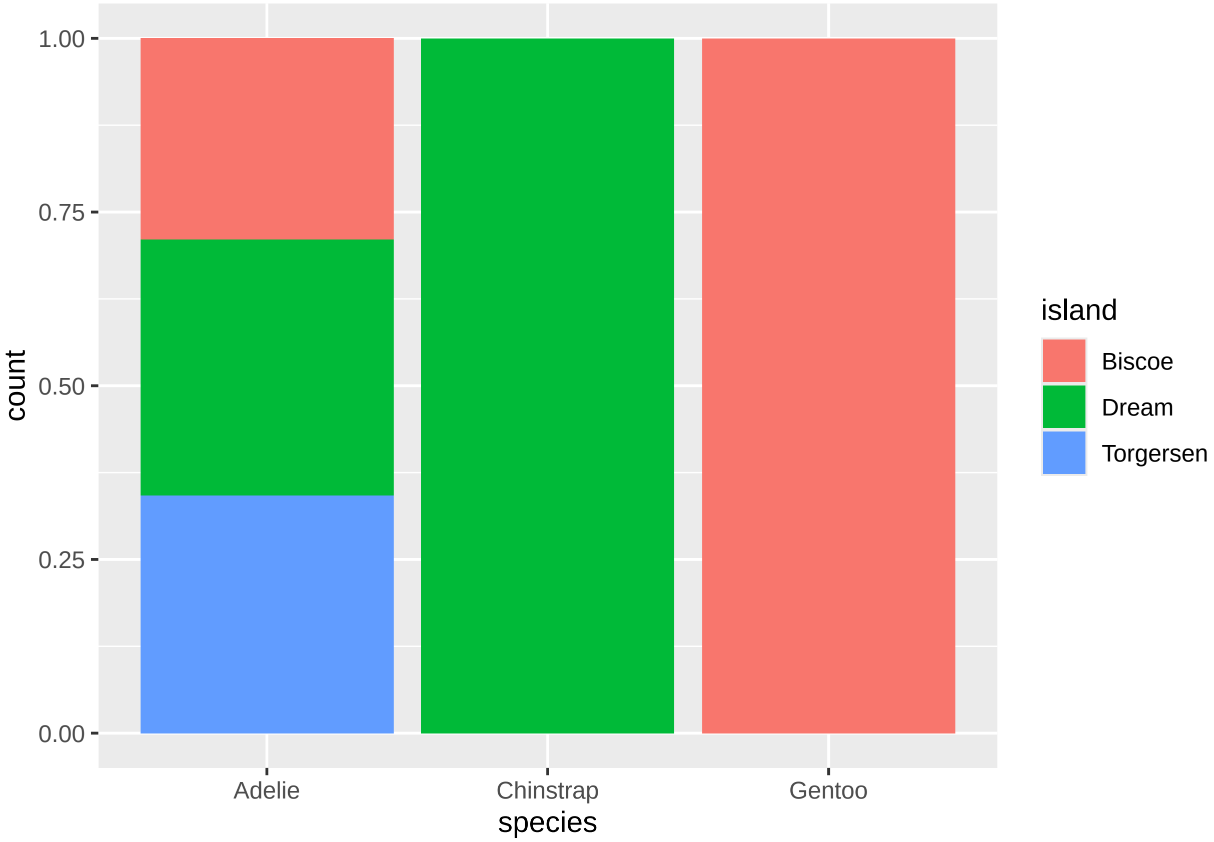


7. Create the two following stacked bar plots, which question can you answer with the first one and which with the second?

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



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



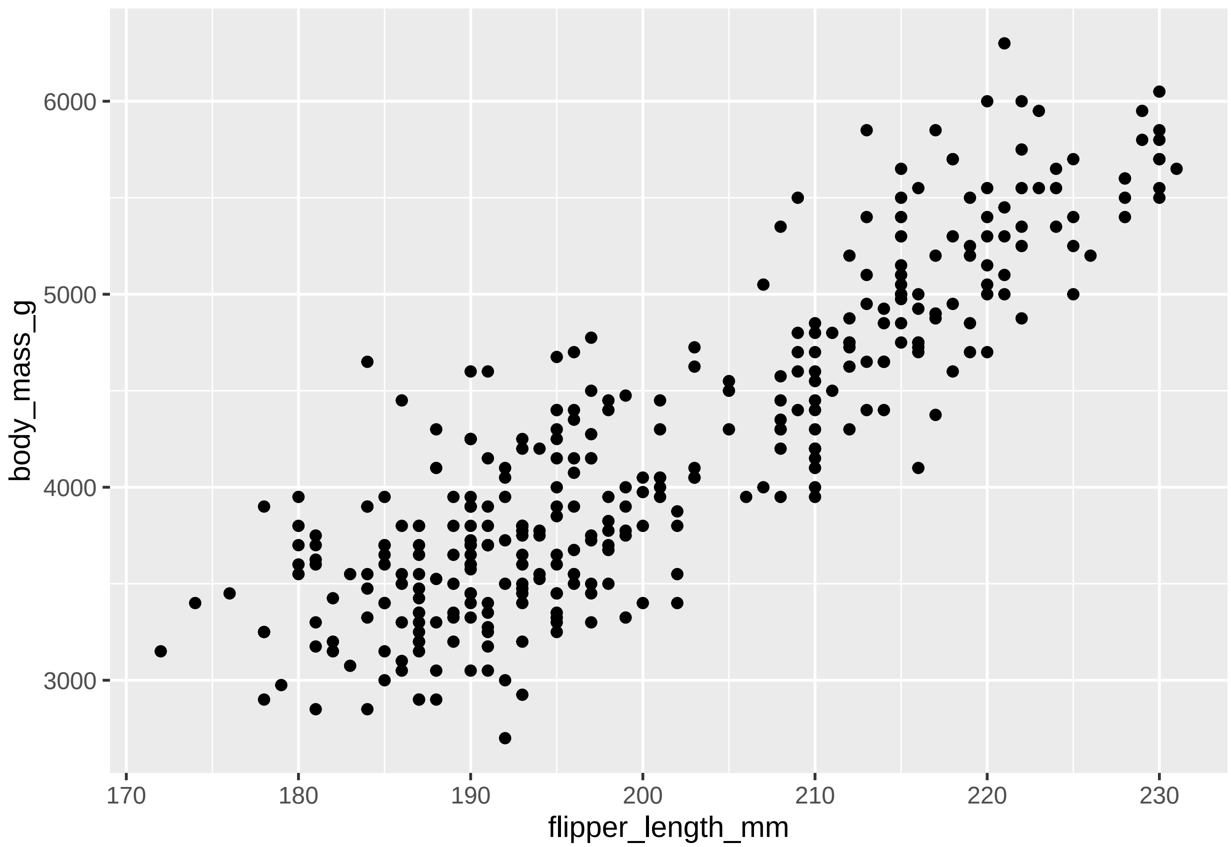
The first plot can answer the question what is the species makeup of penguins on each island The second plot can answer the question of on which islands do each species of penguin reside and what proportion of the species resides on each island (for the Adelie)

1.6 Saving your plots

ggsave() will save the most recently created plot to the disk

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
ggsave(filename = "penguin-plot.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

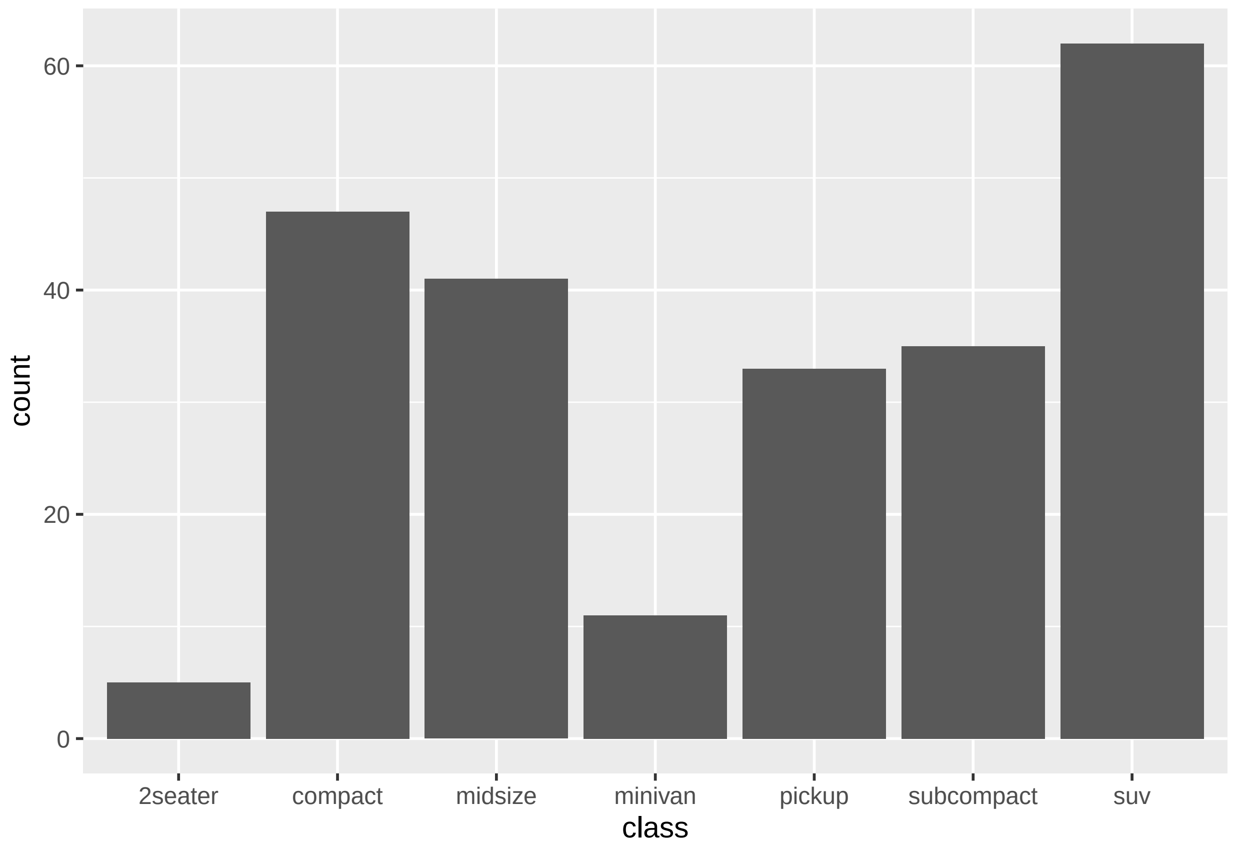
Saves plot to your working directory

Will take height and width dimension from current plotting device, for reproducible code will want to specify them

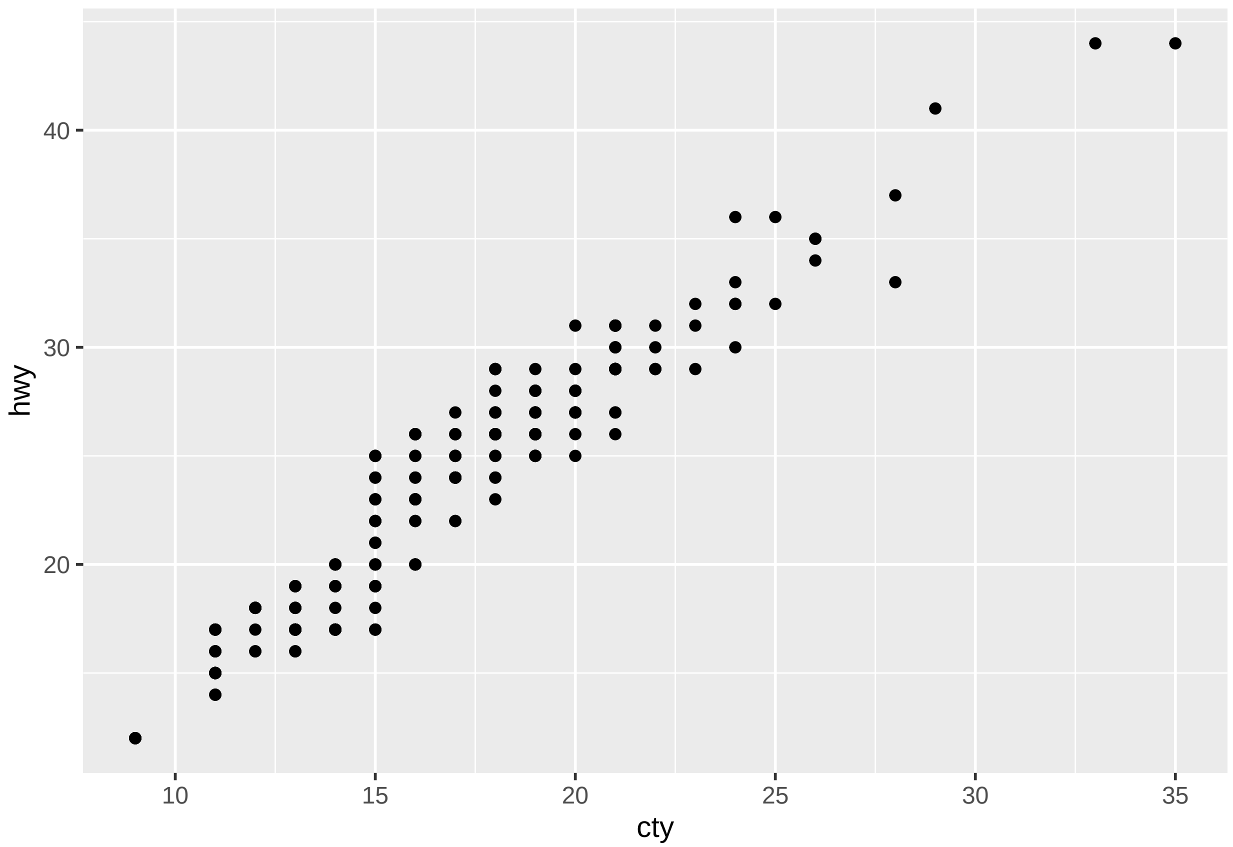
1.6.1 Exercises

1. Run the following lines of code, which of the two plots is saved as mpg-plot.png and why?

```
ggplot(mpg, aes(x = class)) +
  geom_bar()
```



```r
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
```



```
ggsave("mpg-plot.png")
```

```
## Saving 6.5 x 4.5 in image
```

The scatterplot is saved since ggsave saves the most recently created plot, and the scatterplot code is executed after the bar plot code and thus is more recent

2. What do you need to change in the above code to save the plot as a PDF instead of a PNG? How could you find out what type of images would work in ggsave()?

```
?ggsave
ggsave("mpg-plot.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

The device (png, pdf) is inferred in the above exercises by the filepath, so changing the extension to pdf will save the plot as a pdf You can find out what type of images work by checking the documentation using the help operator

1.7 Common problems

When using ggplot, make sure to put + in right place

```
#ggplot(data = mpg)
#+ geom_point(mapping = aes(x = displ, y = hwy))
```

The above code will not work, + has to come at the end of the line