

Lab Exercises Week 04

Alison Lawyer

2024-10-02

```
# keep this chunk in all your RMarkdown scripts
```

```
knitr::opts_chunk$set(echo = TRUE)
```

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

```
# List required packages
```

```
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     3.5.1      v tibble     3.2.1
```

```
## v lubridate  1.9.3      v tidyr      1.3.1
```

```
## v purrr       1.0.2
```

```
## -- 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(ggplot2)
```

```
library(forcats)
```

For this lab, you will create various data visualizations of the palmerpenguins dataset to practice your understanding of ggplot and the grammar of graphics and apply visualization best practices.

```
# load the palmerpenguins data this time, you can use the
```

```
# cleaned dataset called 'penguins'
```

```
library(palmerpenguins)
```

```
data(penguins)
```

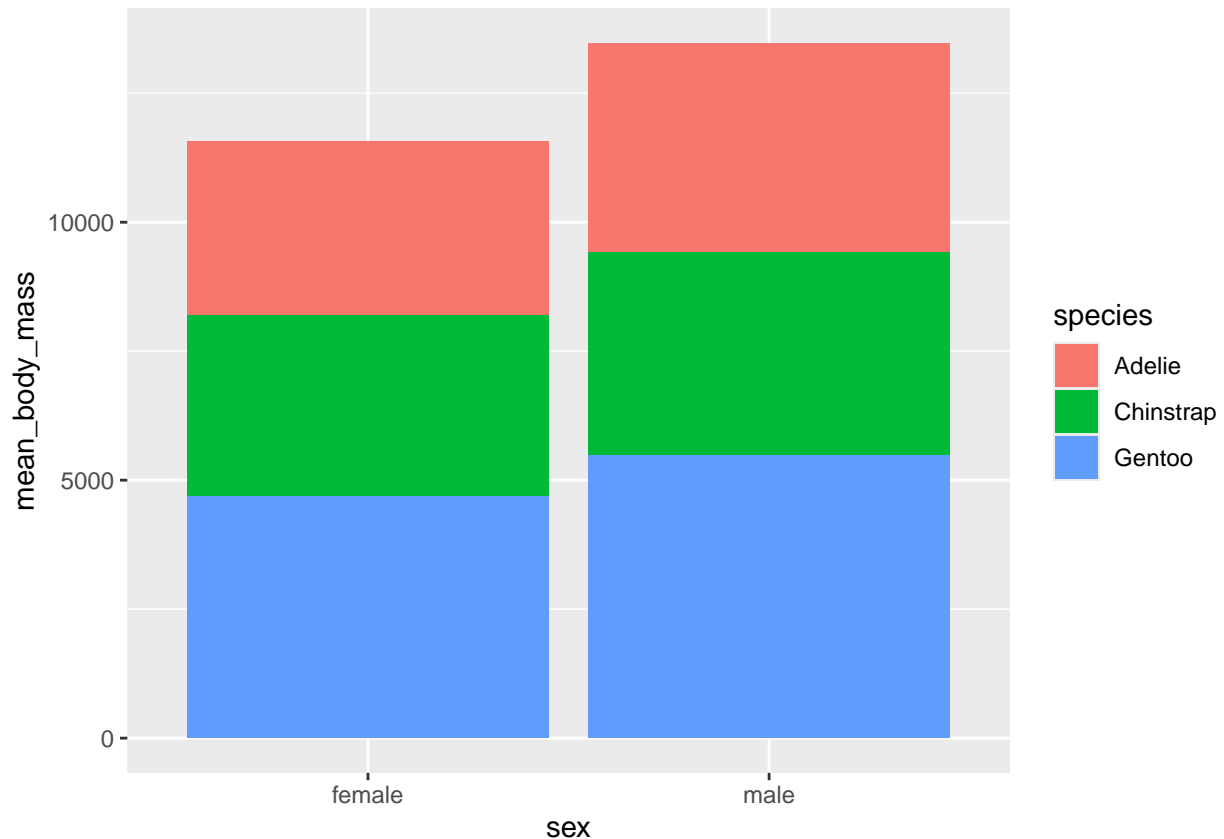
Exercise 1

Last week you examined the sexual dimorphism in body mass using simple bar charts. Now, let's get a bit more advanced and create a grouped bar chart that summarizes this information in one figure. The figure will plot the mean body mass of males and females, with different bars colored by species.

```
# Create a ggplot bar chart with sex on the x-axis, mean  
# body mass on the y-axis, and bar fill color for species.  
# Be sure to exclude rows with NAs for body mass before  
# plotting. Note the default placement of the grouped bars
```

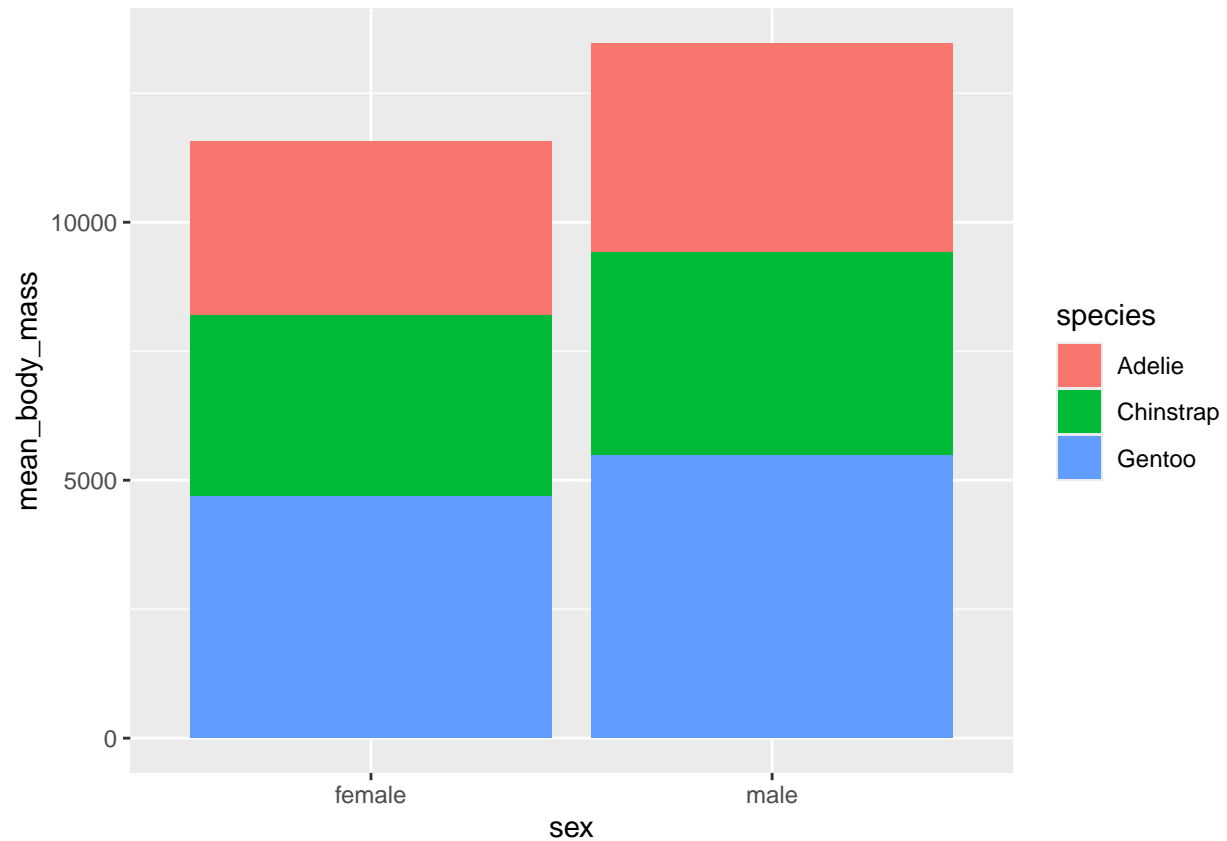
```
# on the plot
na.omit(penguins) %>%
  group_by(sex, species) %>%
  summarize(mean_body_mass = mean(body_mass_g)) %>%
  ggplot(aes(x = sex, y = mean_body_mass, fill = species)) +
  geom_col()
```

'summarise()' has grouped output by 'sex'. You can override using the '.groups' argument.



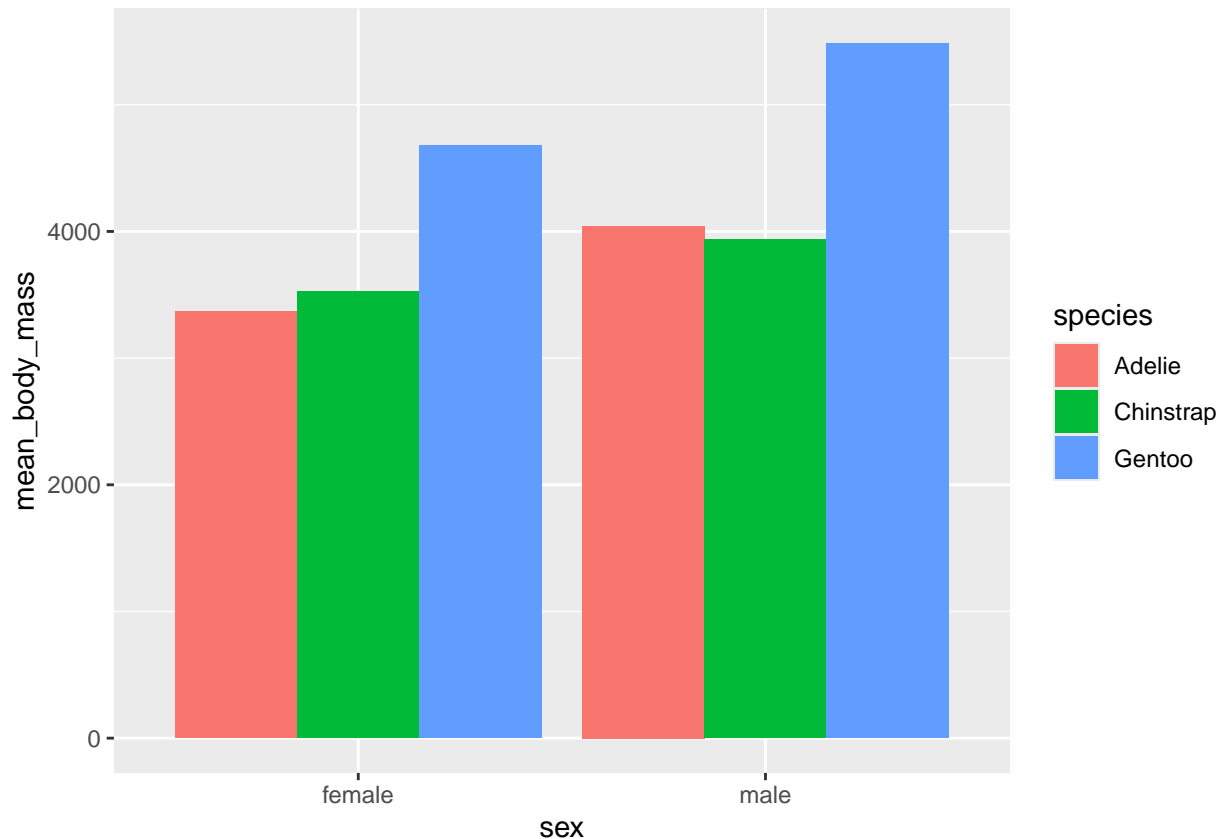
```
# Explore the different effect of putting species within
# the aes() function vs. outside
na.omit(penguins) %>%
  group_by(sex, species) %>%
  summarize(mean_body_mass = mean(body_mass_g)) %>%
  ggplot(aes(x = sex, y = mean_body_mass)) + geom_bar(stat = "identity",
  aes(fill = species))
```

'summarise()' has grouped output by 'sex'. You can override using the '.groups' argument.



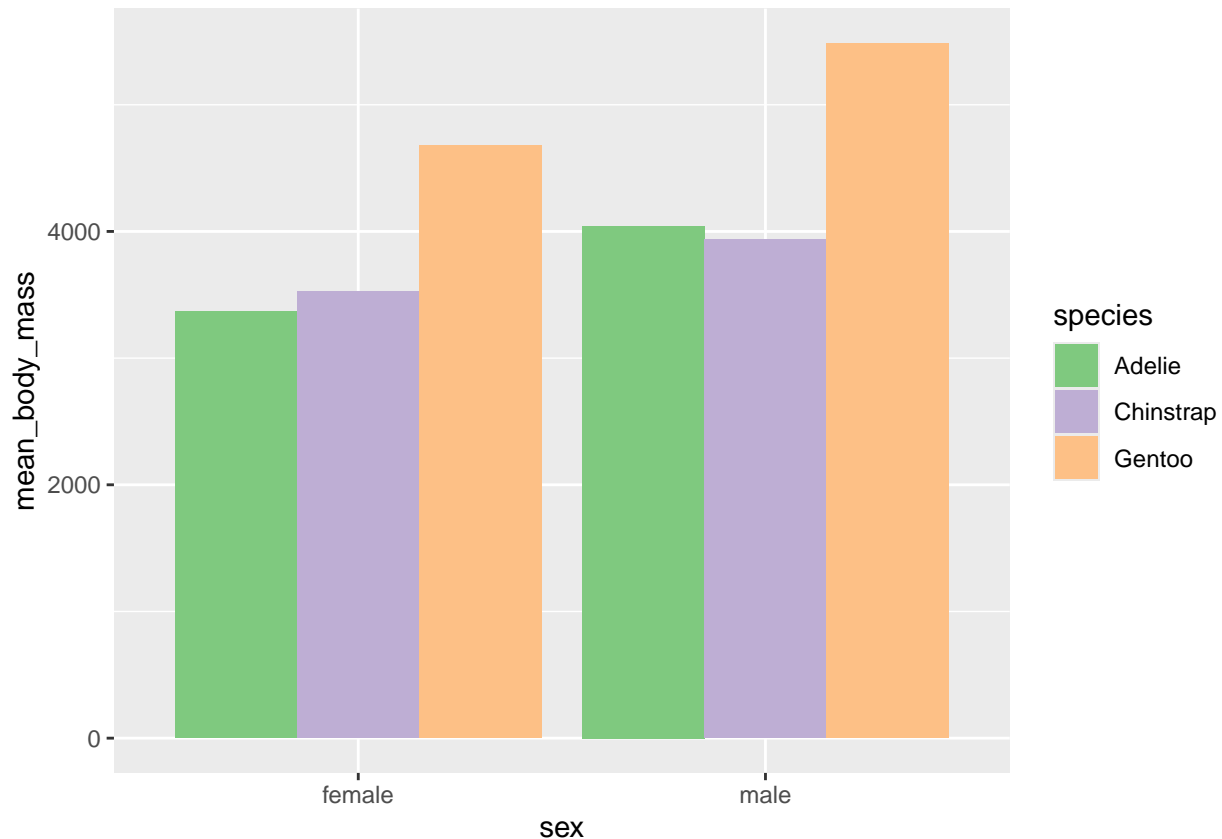
```
# Modify your code from above to have the bars for each
# species appear side-by-side ('dodged' instead of on top
# of each other ('stacked'))
na.omit(penguins) %>%
  group_by(sex, species) %>%
  summarize(mean_body_mass = mean(body_mass_g)) %>%
  ggplot(aes(x = sex, y = mean_body_mass, fill = species)) +
  geom_bar(stat = "identity", position = "dodge")
```

```
## 'summarise()' has grouped output by 'sex'. You can override using the '.groups'
## argument.
```



```
# Change the fill colors by applying a custom palette. You
# can do this in many ways. One easy way is to add a
# function scale_fill_brewer() to your ggplot command to
# adjust the color scale. You can look up the help for this
# function to see available palette names for different
# kinds of scales (diverging, qualitative, sequential). Add
# this function and use palette = 'Accent' or any of the
# other palettes.
na.omit(penguins) %>%
  group_by(sex, species) %>%
  summarize(mean_body_mass = mean(body_mass_g)) %>%
  ggplot(aes(x = sex, y = mean_body_mass, fill = species)) +
  geom_bar(stat = "identity", position = "dodge") + scale_fill_brewer(palette = "Accent")
```

```
## 'summarise()' has grouped output by 'sex'. You can override using the '.groups'
## argument.
```



Question: Summarize the species differences you observe Answer: The Gentoo species is the largest of the penguins, the Chinstrap species appears to have the least difference between the sexes.

Exercise 2

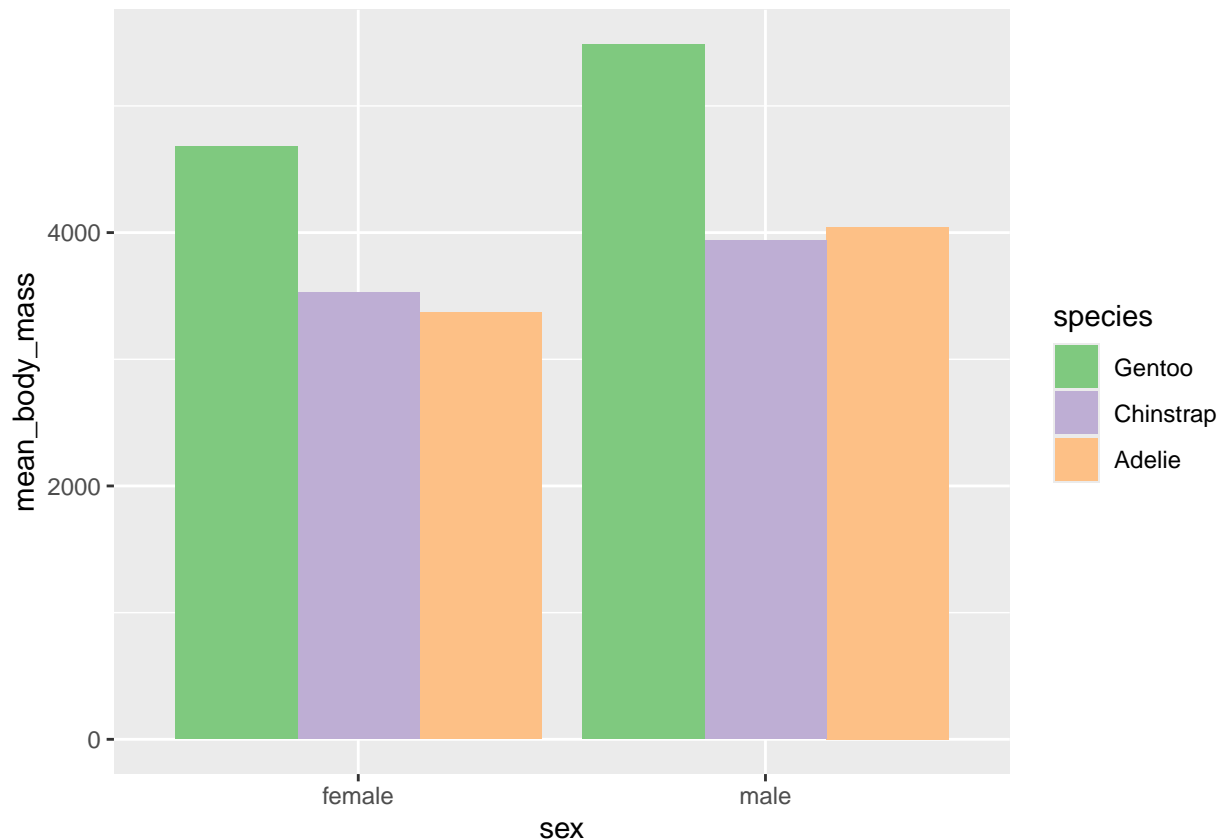
Now let's change the order in which the bars appear. This is often helpful for clearer presentation of results, for example by ordering bars by their relative values from small to large or vice versa.

```
# By default, the order in which species are plotted within
# each sex is based on their alphabetical order given the
# species names. Now let's change this order and plot
# species by their mean body mass, in descending order,
# within each sex. This can be achieved by reordering the
# 'species' factor variable by the mean body mass for each
# species using the function fct_reorder from the forcats
# package. Install the package and make sure you load it
# into your environment before proceeding. See if you can
# figure out what goes into the fct_reorder function by
# using the R help for the function. Tip: you can use
# desc() to reverse the order from ascending to descending.
```

```
# Note: The fct_reorder function sorts by overall mean
# across both sexes, which is usually what we want, as
# changing the order of species for each category on the
# x-axis would get confusing.
```

```
penguins %>%
  drop_na() %>%
  group_by(sex, species) %>%
  summarize(mean_body_mass = mean(body_mass_g, na.rm = TRUE),
            sd_body_mass = sd(body_mass_g, na.rm = TRUE)) %>%
  mutate(species = fct_reorder(species, desc(mean_body_mass))) %>%
  ggplot(aes(x = sex, y = mean_body_mass, fill = species)) +
  geom_col(position = "dodge") + scale_fill_brewer(palette = "Accent")
```

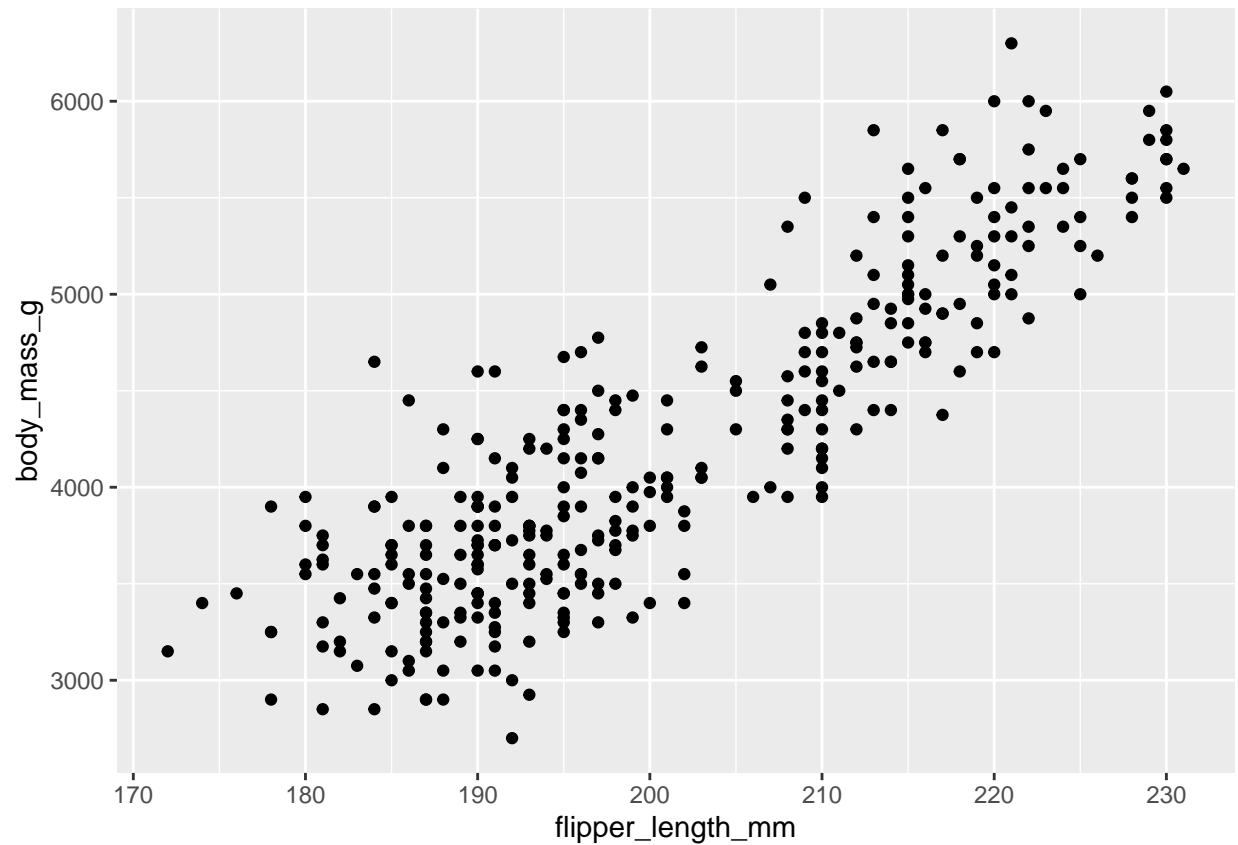
'summarise()' has grouped output by 'sex'. You can override using the '.groups' argument.



Exercise 3

Now explore the relationship between flipper length and body mass using a scatter plot.

```
# Create a scatterplot of flipper length by body mass for
# all samples
na.omit(penguins) %>%
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) + geom_point()
```



Exercise 4

As scatterplots are often used to visualize or examine associations in the data, let's help that process by adding a simple regression line through the scatter of points. You can do this in ggplot without running a formal regression analysis.

```
na.omit(penguins) %>%  
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) + geom_point() +  
  geom_smooth(method = "lm")
```

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

