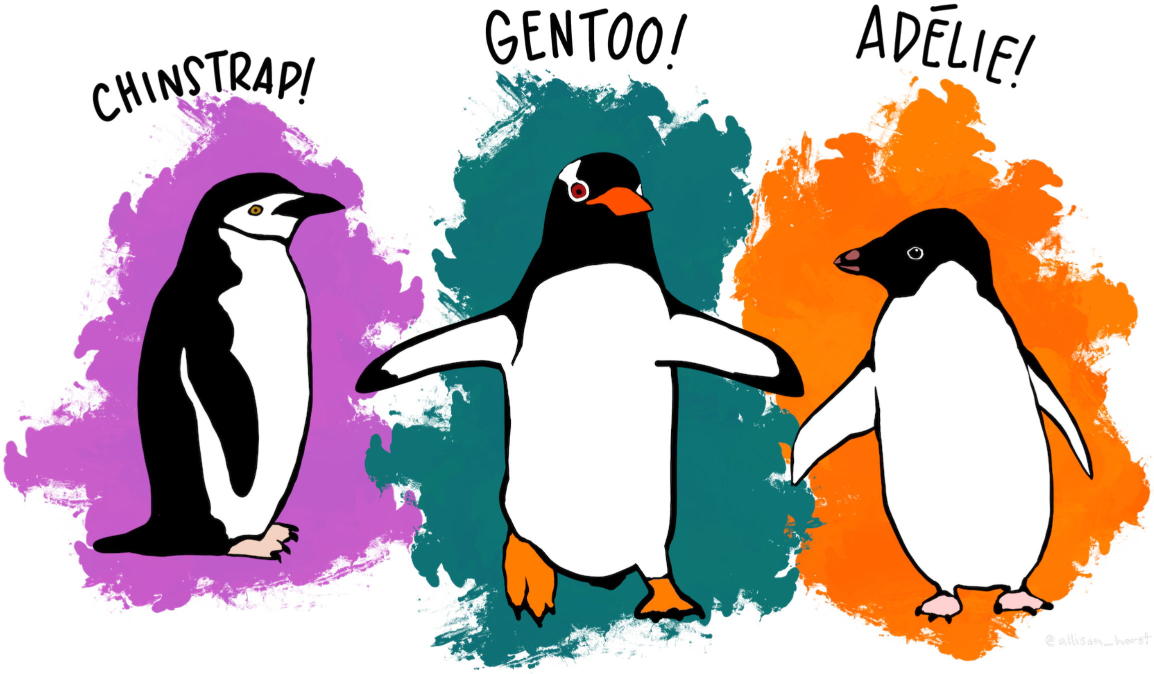
Practice for the Final

2023-11-28

Practicing for the Final

This is meant to help you prepare for the final "project" (really just an elongated assignment) which I will be handing out after the Module 4 quiz. The final will have about the same amount of guidance (or maybe even a bit more) that I've given you here. If you can work your way through these practice problems, you'll be in good shape!

I'm expecting you to be able to filter, summarize, and wrangle the data in ways you need, choose the appropriate visualization, choose the appropriate analysis, and correctly interpret the analysis for the question I've asked you.



Some important notes:

- For both this and the final project, we are going to use the `palmerpenguins` dataset, which we've used before! You can learn more about it here.
- I will not be giving you an answer key for your final project since part of your grade for the assignment is to choose the correct visualizations and analyses.
- Because we are working through this together in class, I will be grading this assignment on completion only! That said, make sure you actually answer ***every question*** that's in here for full credit… there are quite a few!

Structure & Guidelines

This practice/review is structured as 3 different problem sets. For each problem set, I am presenting you with a initial question to guide your thinking and analysis.

Data

Assume that nothing carries over between problem sets.

Each problem set is stand-alone, meaning that you should always start with the `penguins` data frame at the beginning of each problem set. If you should use a data frame that you created *within* the problem set, I explicitly state so.

For example, in Problem Set 1, you should use the `dream` data frame that you create for the entire problem set; at the start of Problem Set 2, start over with the `penguins` data frame.

Interpreting Statistical Results

When I ask you to interpret statistical results, you should roughly follow these guidelines.

- the cut-off for our p-values is always 0.05
- report the p-value that we are focused on
- if there are multiple p-values of interest, report all of them
- state whether the p-value indicates a significant difference/relationship
- if applicable, state whether we should or should not reject the null hypothesis

Plotting

All plots should be made using `ggplot2`.

Your options for plot types to choose from are:

- multiple histogram plots
  - use transparency (`alpha`)
  - use `position = "identity"` with multiple groups to see the full distributions
- multiple density plots
  - use transparency (`alpha`) with multiple groups to see the full distributions
- box-and-whisker plot
  - add points on top of the box plot to show the distribution of the points
- scatter plot
  - add the linear model to every scatter plot

Note: All plots should have modified axis labels and legend labels and a theme.

In many cases, this might mean capitalizing the axis label or legend label. In other cases, you might want to put units in parentheses after the words (e.g., Body Mass (g)).

Set-Up

Let's load our packages and get started! Load the `tidyverse` and `palmerpengiuns` packages.

```
library(tidyverse)
library(palmerpenguins)
```

We will be using the `penguins` data frame. It exists as part of the `palmerpenguins` package, but if you want it to show up in your environment, run the following code chunk.

```
penguins <- penguins
```

Problem Set 1

Objective: For this problem set, we are going to look at the difference in flipper lengths for each species on Dream island.

First things first, let's make a dataframe with only the penguins on Dream island. We will want to use this dataframe for the rest of this problem set; call it `dream`.

```
dream <- penguins %>%
  filter(island == "Dream")
```

Based on our question, which variable is independent and which is dependent? Which is continuous and/or which is categorical?

Answer:

First, calculate the range (minimum and maximum) and mean of the flipper lengths for each species

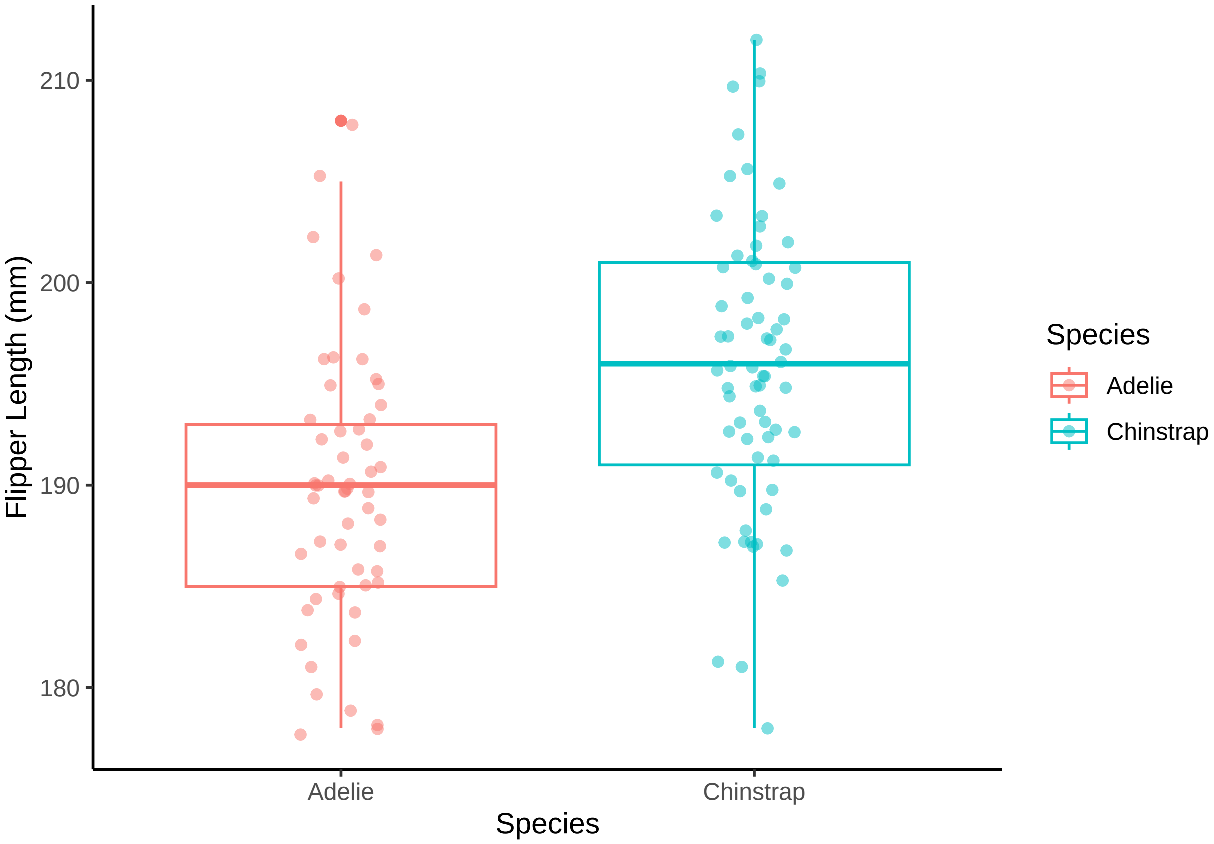
```
dream %>%
  group_by(species) %>%
  summarise(min_flipper = min(flipper_length_mm, na.rm = TRUE),
            max_flipper = max(flipper_length_mm, na.rm = TRUE),
            mean_flipper = mean(flipper_length_mm, na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   species   min_flipper max_flipper mean_flipper
##   <fct>           <int>       <int>        <dbl>
## 1 Adelie            178         208         190.
## 2 Chinstrap         178         212         196.
```

Choose an effective visualization method for this data. Use `ggplot2`.

```
# boxplot, multiple histogram, or multiple density plot are all acceptable

ggplot(dream, aes(x = species, y = flipper_length_mm, color = species)) +
  geom_boxplot() +
  geom_jitter(width = 0.1, alpha = 0.5) +
  labs(x = "Species",
       y = "Flipper Length (mm)",
       color = "Species") +
  theme_classic()
```



Write out the null and alternative hypotheses.

Null:

Alternative:

Run the appropriate statistical test.

```
t.test(data = dream, flipper_length_mm ~ species)
```

```
##
##  Welch Two Sample t-test
##
## data:  flipper_length_mm by species
## t = -4.937, df = 120.37, p-value = 2.581e-06
## alternative hypothesis: true difference in means between group Adelie and group Chinstrap is not equa
## 95 percent confidence interval:
##  -8.534213 -3.648561
## sample estimates:
##    mean in group Adelie mean in group Chinstrap
##                189.7321                195.8235
```

Interpret the results of your statistical test:

- What is the p-value?
- Is the p-value above or below 0.05?
- What does your answer to the question above mean?
- Should we reject or fail to reject the null hypothesis?

Answer:

Should we run pairwise comparisons? If yes, do so below and interpret:

```
# no need, only 1 pair compared
```

Problem Set 2

For this problem set, we want to know if there is a relationship between flipper length and bill length amongst all penguins (we aren't going to worry about species right now).

Are our variables of interest continuous and/or categorical?

Take note that in this example, there is no dependent or independent variable per say. We don't have any reason to think that flipper length influences bill length or vice versa. We just want to determine if there is a relationship or not.

That said, treat flipper length as the *independent* variable (x-axis) and bill length as the *dependent* variable (y-axis).

First, calculate the mean and standard deviation for both bill length and flipper length.

```
bill_flipper <- penguins %>%
  summarise(mean_bill = mean(bill_length_mm, na.rm = TRUE),
            sd_bill = sd(bill_length_mm, na.rm = TRUE),
            mean_flipper = mean(flipper_length_mm, na.rm = TRUE),
            sd_flipper = sd(flipper_length_mm, na.rm = TRUE))
bill_flipper
```

```
## # A tibble: 1 x 4
##   mean_bill sd_bill mean_flipper sd_flipper
##       <dbl>   <dbl>        <dbl>      <dbl>
## 1      43.9    5.46         201.       14.1
```

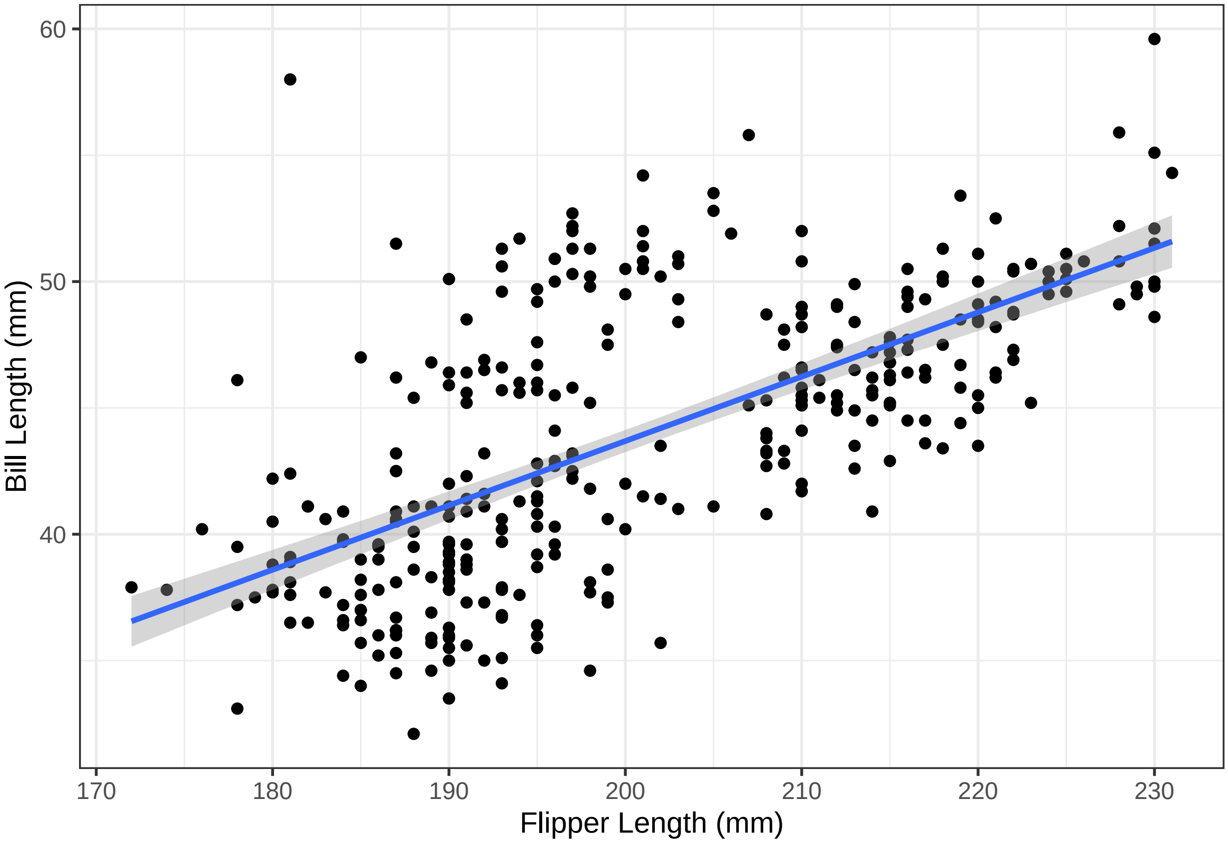
Choose the best way to visualize the relationship between these two variables

```
ggplot(penguins, aes(flipper_length_mm, bill_length_mm)) +
  geom_point() +
  geom_smooth(method = "lm") +
    labs(x = "Flipper Length (mm)",
         y = "Bill Length (mm)") +
  theme_bw()
```

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

```
## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



To do the rest of this problem set, the easiest way is to create a new data frame with no NA values. Run the code chunk below and use that new data frame for the rest of the problem set.

```
penguins_noNA <- penguins %>%
  filter(!is.na(bill_length_mm),
         !is.na(flipper_length_mm))
```

Calculate the correlation coefficient and the `r^2` value.

- According to the correlation coefficient, is the relationship positive, negative or is there no relationship?

*Answer:

- What does the `r^2` value tell us? Remember, we usually multiply the `r^2` by 100 to represent this as a percentage.

*Answer:

```
r <- cor(x = penguins_noNA$flipper_length_mm, y = penguins_noNA$bill_length_mm)
r
```

```
## [1] 0.6561813
```

```r
r^2 * 100
```

```
## [1] 43.0574
```

Run the appropriate statistical test for this data.

```r
lm_model <- lm(data = penguins_noNA, bill_length_mm ~ flipper_length_mm)
summary(lm_model)
```

```
##
## Call:
## lm(formula = bill_length_mm ~ flipper_length_mm, data = penguins_noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5792 -2.6715 -0.5721  2.0148 19.1518
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -7.26487    3.20016   -2.27   0.0238 *
## flipper_length_mm  0.25477    0.01589   16.03   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.126 on 340 degrees of freedom
## Multiple R-squared:  0.4306, Adjusted R-squared:  0.4289
## F-statistic: 257.1 on 1 and 340 DF,  p-value: < 2.2e-16
```

Using variables and numbers from the summary above, write out the equation of the line of best fit.

Answer: bill_length_mm = 0.25*flipper_length_mm - 7.26

Interpret the results of your statistical test:

- What is the p-value?
- Is the p-value above or below 0.05?
- What does your answer to the question above mean?

*Answer:

Should we run pairwise comparisons? If yes, do so below and interpret:

```r
# no pairs to compare
```

Adding a Variable

Let’s add in the species variable into our analysis! Keep using the `penguins_noNA` data frame.

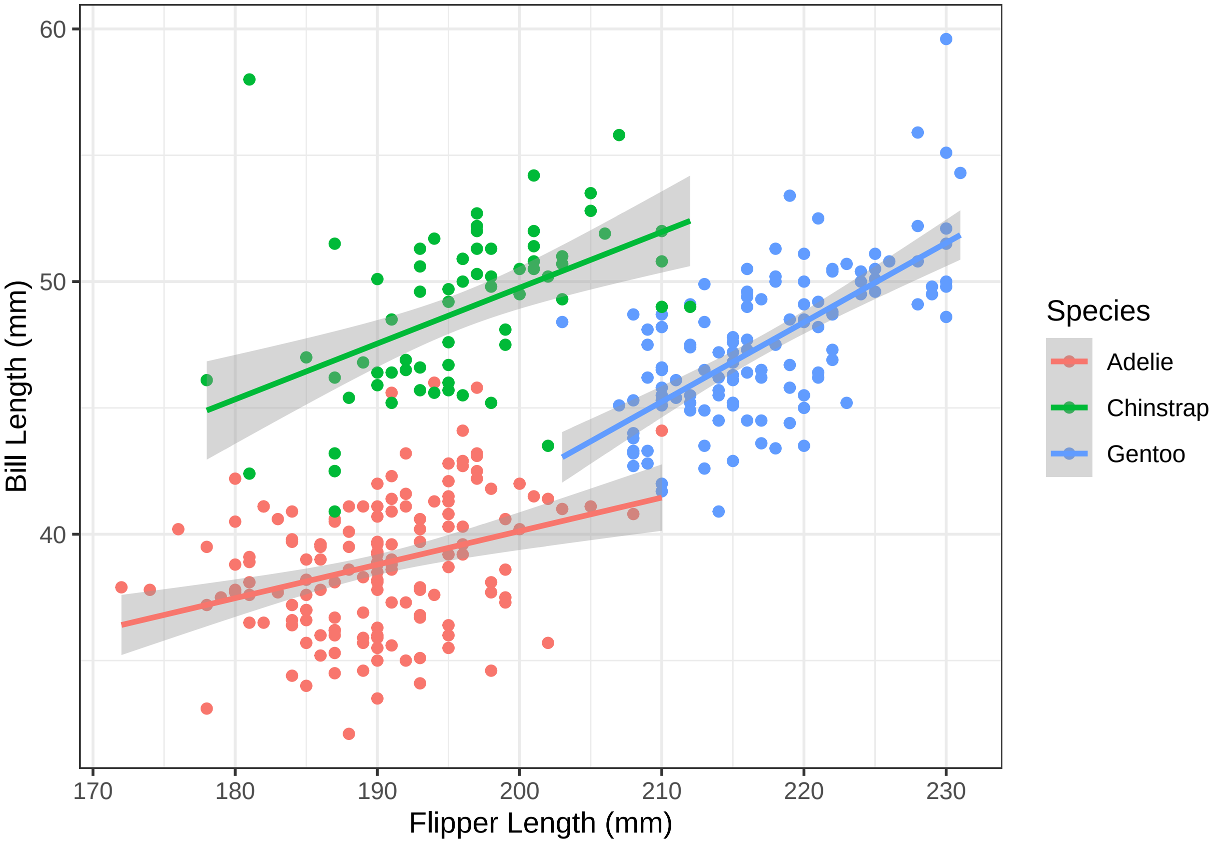
First, plot the data, this time including species as a variable in the plot.

```r
ggplot(penguins, aes(flipper_length_mm, bill_length_mm, color = species)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "Flipper Length (mm)",
       y = "Bill Length (mm)",
       color = "Species") +
  theme_bw()
```

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

```
## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



Run the model again, this time including species and the species *interaction* with "independent" variable.

```r
mlm_model <- lm(data = penguins_noNA, bill_length_mm ~ flipper_length_mm * species)
summary(mlm_model)
```

```
##
## Call:
## lm(formula = bill_length_mm ~ flipper_length_mm * species, data = penguins_noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6977 -1.7046  0.0596  1.5571 12.4394
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        13.58714    6.05061   2.246 0.025380 *
## flipper_length_mm                   0.13269    0.03183   4.168 3.91e-05 ***
## speciesChinstrap                   -7.99376   10.48117  -0.763 0.446190
## speciesGentoo                     -34.32335    9.81983  -3.495 0.000537 ***
## flipper_length_mm:speciesChinstrap   0.08813    0.05405   1.631 0.103915
## flipper_length_mm:speciesGentoo      0.18152    0.04775   3.801 0.000171 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.55 on 336 degrees of freedom
## Multiple R-squared:  0.7851, Adjusted R-squared:  0.7819
## F-statistic: 245.5 on 5 and 336 DF,  p-value: < 2.2e-16
```

Interpret the results from the model above. Focus on the p-values for the independent variables and/or interaction terms, not the overall model.

*Answer:

Problem Set 3

For this problem set, we want to know if there is a difference in the body mass of the penguin species.

Based on our question, which variable is independent and which is dependent? Which is continuous and/or which is categorical?

First, calculate the minimum, maximum, and mean of the body mass for each species. Call this dataframe `body_mass`.

```
body_mass <- penguins %>%
  group_by(species) %>%
  summarise(min_mass = min(body_mass_g, na.rm = TRUE),
            max_mass = max(body_mass_g, na.rm = TRUE),
            mean_mass = mean(body_mass_g, na.rm = TRUE))
body_mass
```

```
## # A tibble: 3 x 4
##   species   min_mass max_mass mean_mass
##   <fct>        <int>    <int>     <dbl>
## 1 Adelie        2850     4775     3701.
## 2 Chinstrap     2700     4800     3733.
## 3 Gentoo        3950     6300     5076.
```

What if we wanted the values in the newly created `body_mass` column to be in kilograms instead of grams?

Let’s do this in 2 parts.

- First, we will create a function that converts a number from grams to kilograms.
- Second, we will use that function that we just created to modify each column (hint: we will want to use `mutate`).

Our first step is to write a function that converts a number (not a column, mind you) from grams to kilograms.

```
g_to_kg <- function(g){
  kg <- g / 1000
  print(kg)
}
```

Convert all the columns in the `body_mass` data frame from grams to kilograms using the function you wrote above. You don't need to save this dataframe as anything, but if you do, call it `body_mass_kg`.

```
body_mass %>%
  mutate(min_mass = g_to_kg(min_mass),
         max_mass = g_to_kg(max_mass),
         mean_mass = g_to_kg(mean_mass))
```

```
## [1] 2.85 2.70 3.95
## [1] 4.775 4.800 6.300
## [1] 3.700662 3.733088 5.076016
```

```
## # A tibble: 3 x 4
##   species   min_mass max_mass mean_mass
##   <fct>        <dbl>    <dbl>     <dbl>
## 1 Adelie        2.85     4.78      3.70
## 2 Chinstrap     2.7      4.8       3.73
## 3 Gentoo        3.95     6.3       5.08
```

```
# alternative option
# copy code from above
# add a mutate function before the summarise function and modify summarise function columns
penguins %>%
  group_by(species) %>%
  mutate(body_mass_kg = g_to_kg(body_mass_g)) %>%
  summarise(min_mass = min(body_mass_kg, na.rm = TRUE), # body_mass_kg instead of body_mass_g
            max_mass = max(body_mass_kg, na.rm = TRUE),
            mean_mass = mean(body_mass_kg, na.rm = TRUE))
```

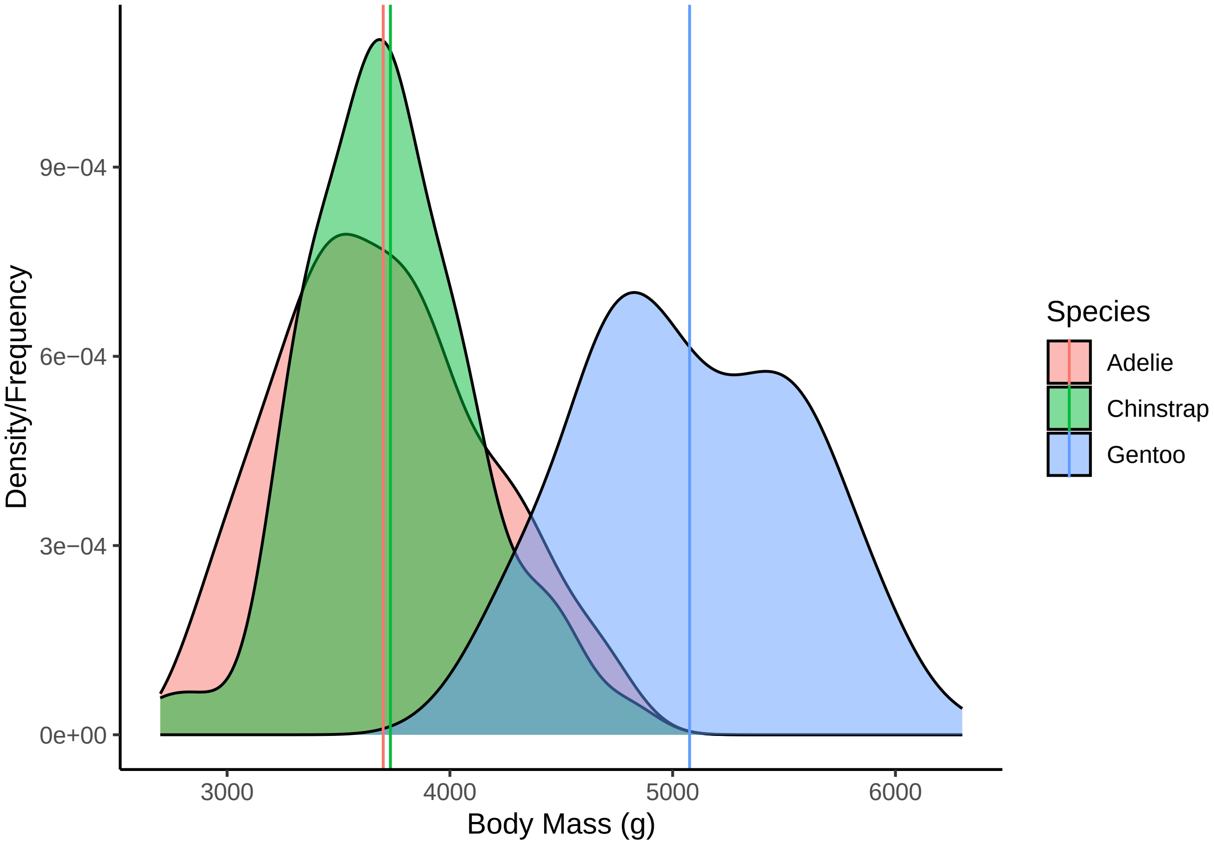
```
##   [1] 3.750 3.800 3.250    NA 3.450 3.650 3.625 4.675 3.475 4.250 3.300 3.700
##  [13] 3.200 3.800 4.400 3.700 3.450 4.500 3.325 4.200 3.400 3.600 3.800 3.950
##  [25] 3.800 3.800 3.550 3.200 3.150 3.950 3.250 3.900 3.300 3.900 3.325 4.150
##  [37] 3.950 3.550 3.300 4.650 3.150 3.900 3.100 4.400 3.000 4.600 3.425 2.975
##  [49] 3.450 4.150 3.500 4.300 3.450 4.050 2.900 3.700 3.550 3.800 2.850 3.750
##  [61] 3.150 4.400 3.600 4.050 2.850 3.950 3.350 4.100 3.050 4.450 3.600 3.900
##  [73] 3.550 4.150 3.700 4.250 3.700 3.900 3.550 4.000 3.200 4.700 3.800 4.200
##  [85] 3.350 3.550 3.800 3.500 3.950 3.600 3.550 4.300 3.400 4.450 3.300 4.300
##  [97] 3.700 4.350 2.900 4.100 3.725 4.725 3.075 4.250 2.925 3.550 3.750 3.900
## [109] 3.175 4.775 3.825 4.600 3.200 4.275 3.900 4.075 2.900 3.775 3.350 3.325
## [121] 3.150 3.500 3.450 3.875 3.050 4.000 3.275 4.300 3.050 4.000 3.325 3.500
## [133] 3.500 4.475 3.425 3.900 3.175 3.975 3.400 4.250 3.400 3.475 3.050 3.725
## [145] 3.000 3.650 4.250 3.475 3.450 3.750 3.700 4.000
##  [1] 3.500 3.900 3.650 3.525 3.725 3.950 3.250 3.750 4.150 3.700 3.800 3.775
## [13] 3.700 4.050 3.575 4.050 3.300 3.700 3.450 4.400 3.600 3.400 2.900 3.800
## [25] 3.300 4.150 3.400 3.800 3.700 4.550 3.200 4.300 3.350 4.100 3.600 3.900
## [37] 3.850 4.800 2.700 4.500 3.950 3.650 3.550 3.500 3.675 4.450 3.400 4.300
## [49] 3.250 3.675 3.325 3.950 3.600 4.050 3.350 3.450 3.250 4.050 3.800 3.525
## [61] 3.950 3.650 3.650 4.000 3.400 3.775 4.100 3.775
##   [1] 4.500 5.700 4.450 5.700 5.400 4.550 4.800 5.200 4.400 5.150 4.650 5.550
##  [13] 4.650 5.850 4.200 5.850 4.150 6.300 4.800 5.350 5.700 5.000 4.400 5.050
##  [25] 5.000 5.100 4.100 5.650 4.600 5.550 5.250 4.700 5.050 6.050 5.150 5.400
##  [37] 4.950 5.250 4.350 5.350 3.950 5.700 4.300 4.750 5.550 4.900 4.200 5.400
##  [49] 5.100 5.300 4.850 5.300 4.400 5.000 4.900 5.050 4.300 5.000 4.450 5.550
##  [61] 4.200 5.300 4.400 5.650 4.700 5.700 4.650 5.800 4.700 5.550 4.750 5.000
##  [73] 5.100 5.200 4.700 5.800 4.600 6.000 4.750 5.950 4.625 5.450 4.725 5.350
##  [85] 4.750 5.600 4.600 5.300 4.875 5.550 4.950 5.400 4.750 5.650 4.850 5.200
##  [97] 4.925 4.875 4.625 5.250 4.850 5.600 4.975 5.500 4.725 5.500 4.700 5.500
## [109] 4.575 5.500 5.000 5.950 4.650 5.500 4.375 5.850 4.875 6.000 4.925    NA
## [121] 4.850 5.750 5.200 5.400
```

```
## # A tibble: 3 x 4
##   species   min_mass max_mass mean_mass
##   <fct>        <dbl>    <dbl>     <dbl>
## 1 Adelie        2.85     4.78      3.70
## 2 Chinstrap     2.7      4.8       3.73
## 3 Gentoo        3.95     6.3       5.08
```

Back to the question at hand…choose an effective visualization method for this data (you will want to use the original **penguins** dataframe—and grams are fine). Practice adding lines that represent to mean values in the plot. Thinking critically about which columns to use!

```
ggplot(penguins, aes(x = body_mass_g, fill = species)) +
  geom_density(alpha = 0.5) +
  geom_vline(data = body_mass, aes(xintercept = mean_mass, color = species)) +
  labs(x = "Body Mass (g)",
       y = "Density/Frequency",
       fill = "Species",
       color = "Species") +
  theme_classic()
```

```
## Warning: Removed 2 rows containing non-finite values (‘stat_density()‘).
```



Write out the null and alternative hypotheses.

Null:

Alternative:

Run the appropriate statistical test.

```
aov_mod <- aov(data = penguins, body_mass_g ~ species)
summary(aov_mod)
```

```
##              Df    Sum Sq  Mean Sq F value Pr(>F)
## species       2 146864214 73432107   343.6 <2e-16 ***
## Residuals   339  72443483   213698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness
```

Interpret the results of your statistical test:

- What is the p-value?
- Is the p-value above or below 0.05?
- What does your answer to the question above mean?
- Should we reject or fail to reject the null hypothesis?

*Answers:

Should we run pairwise comparisons? If yes, do so below and interpret:

```r
TukeyHSD(aov_mod)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = body_mass_g ~ species, data = penguins)
##
## $species
##                        diff       lwr       upr     p adj
## Chinstrap-Adelie   32.42598 -126.5002  191.3522 0.8806666
## Gentoo-Adelie    1375.35401 1243.1786 1507.5294 0.0000000
## Gentoo-Chinstrap 1342.92802 1178.4810 1507.3750 0.0000000
```

*Answer: