HW 3

Enter your name and EID here: Austine Do (ahd589)

You will submit this homework assignment as a pdf file on Gradescope.

For all questions, include the R commands/functions that you used to find your answer (show R chunk). Answers without supporting code will not receive credit. Write full sentences to describe your findings.

Part 1

In this first part, you will conduct a data analysis of a dataset about a community of penguins in the Antarctic. Install the package containing this dataset and look up the documentation:

```
# Install the package containing the dataset
install.packages("palmerpenguins")

# Read the documentation
?palmerpenguins::penguins
```

Then save the data in your environment:

```
# Save the object as a dataframe
penguins <- as.data.frame(palmerpenguins::penguins)
```

Question 1: (1 pt)

In the documentation, you should have learned that there are 3 different species of penguins. Use your favorite web browser and include an image representing the 3 species below:

How was the data obtained? Write a sentence to cite the source of the data. You will cite this source in the caption of each of your visualization in this part of the assignment.

These images were obtained from Wikipedia, Britannica, and eBird.
https://www.britannica.com/animal/chinstrap-penguin (https://www.britannica.com/animal/chinstrap-penguin)
https://gifts.worldwildlife.org/gift-center/gifts/Species-Adoptions/Gentoo-Penguin-Chick.aspx (https://gifts.worldwildlife.org/gift-center/gifts/Species-Adoptions/Gentoo-Penguin-Chick.aspx)
https://www.britannica.com/animal/Adelie-penguin (https://www.britannica.com/animal/Adelie-penguin)

Question 2: (1 pt)

In your assignment, you will compare 2 numeric variables for each species. Pick 2 numeric variables in the `penguins` dataset and write a question you would be able to answer with your data analysis:

Is there a relationship between bill length and flipper length across each species?

What do you expect the answer of that question would be? *Note: there is right or wrong answer here as it does not matter if your data analysis does or does not match with your expectations.*

I do expect there to be relationship between the 2 measurements across species but it may be a weak relationship

---

Question 3: (1 pt)

How many rows and columns are there in this dataset? What does each row represent? Quickly check if there are any weird values for the variables in this dataset.

```
# Looking at the structure of the dataset and values contained in the columns
glimpse(penguins)
```

```
## 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…
```

```
summary(penguins)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

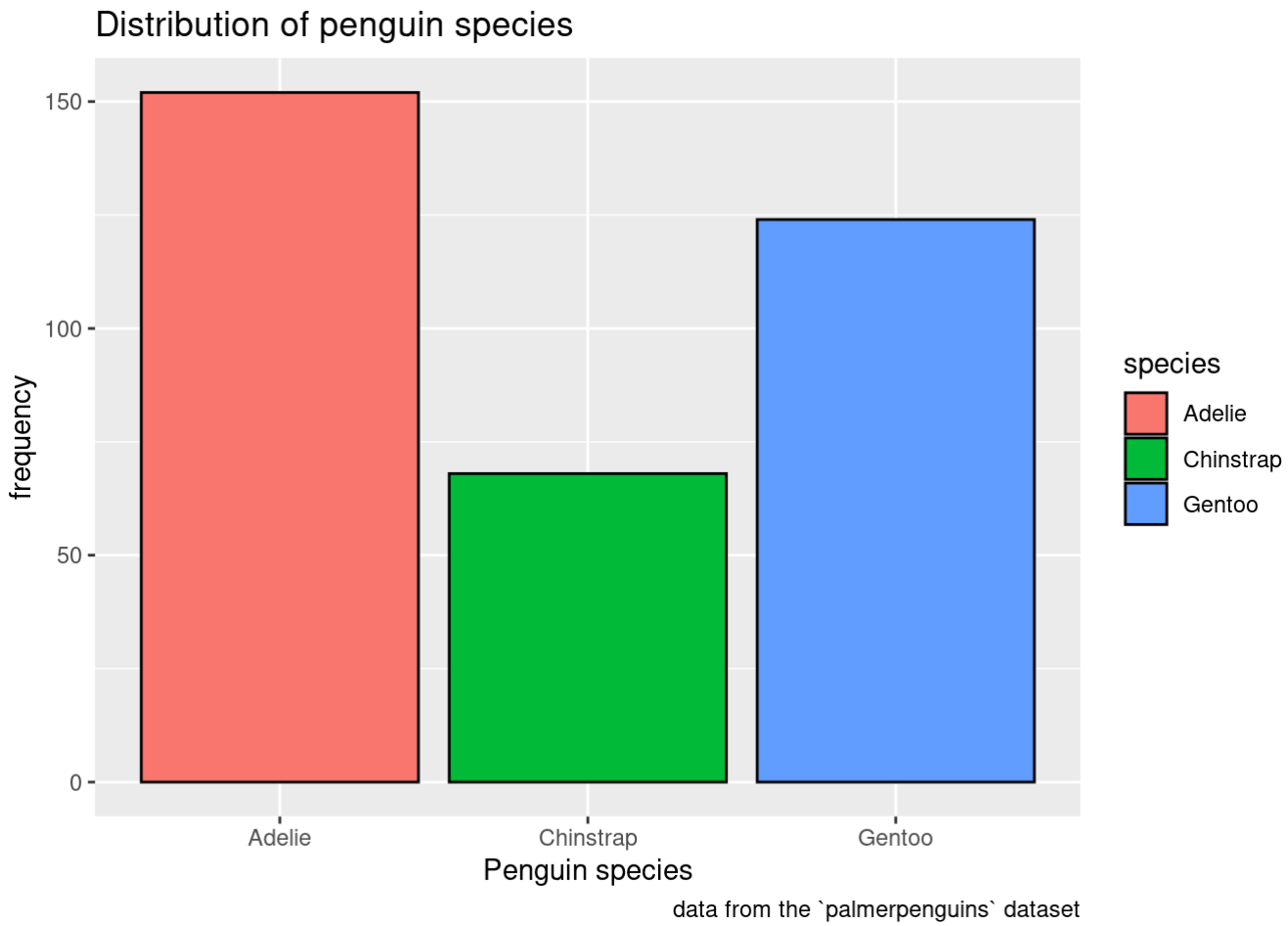
There are 344 rows in this dataset where each row represent an individual penguin. When checking the dataset there were a few NA values but I did not notice any glaring/weird values except for some outliers in certain variables like bill length, flipper length, and body mass.

Question 4: (2 pts)

Using an appropriate visualization, represent the distribution of `species`. Also find appropriate statistics. Write a sentence to interpret each visualization. *Note: make sure to add labels and a caption.*

```
# This visualizes the distribution of species of penguins on a bar plot and uses
# table and proportion table to describe the distribution

ggplot(data = penguins) +
    geom_bar(aes(x = species, fill = species), color = 'black') +
    ggtitle('Distribution of penguin species') +
    labs(x = 'Penguin species',
         y = 'frequency',
         caption = 'data from the `palmerpenguins` dataset')
```




```
table(penguins$species)
```

```
##
##    Adelie Chinstrap    Gentoo
##       152        68       124
```

```
prop.table(table(penguins$species))
```

```
##
##    Adelie Chinstrap    Gentoo
## 0.4418605 0.1976744 0.3604651
```

Adelie and Gentoo penguins make up a majority of the observations in the dataset while Chinstrap penguins make up a small portion of the penguin observations. According to the proportion table Adelie, Gentoo, and Chinstrap penguins make up 44.2 %, 36.0 %, and 19.8% of the penguin species in the dataset respectively.

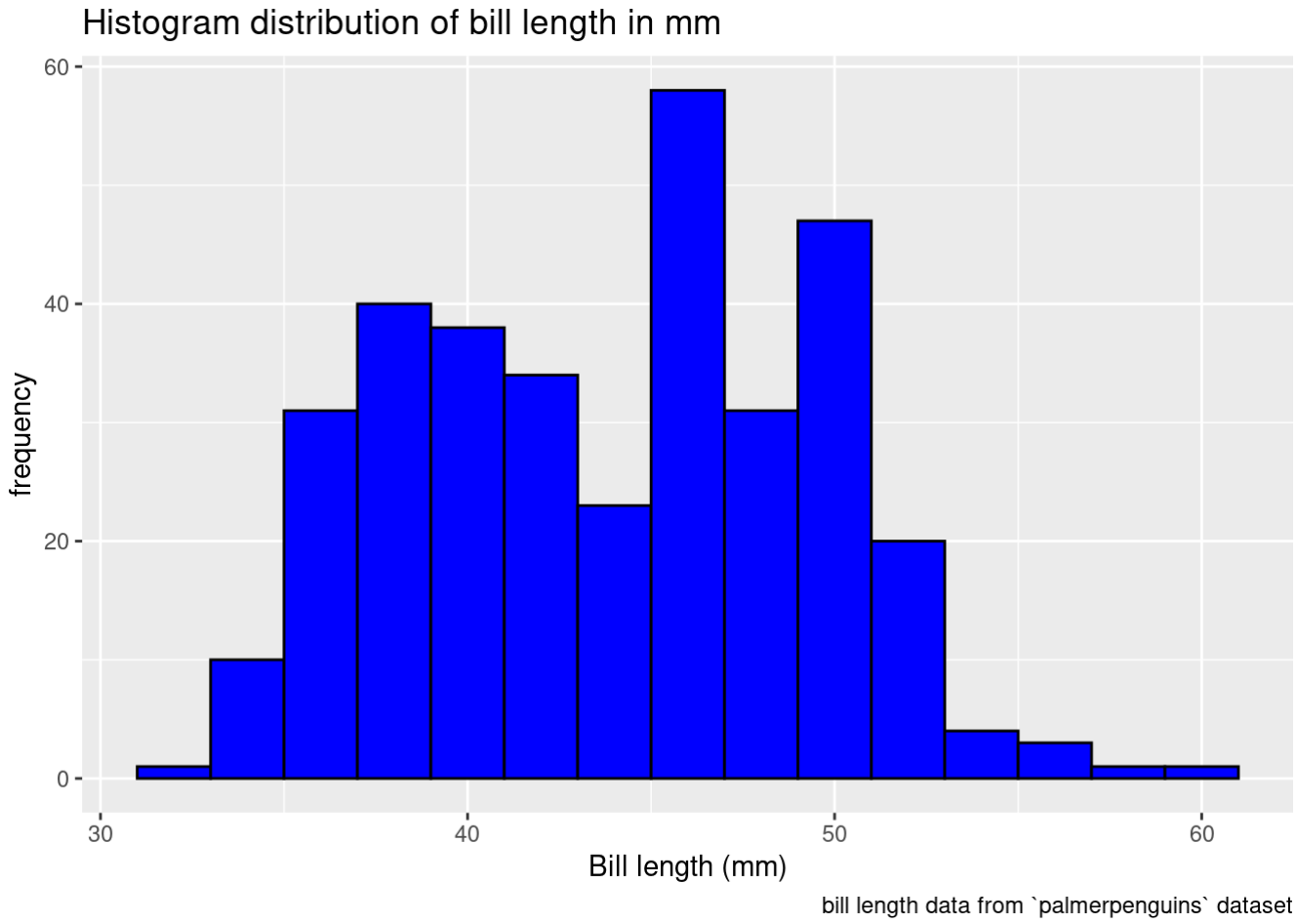
Question 5: (2 pts)

Using appropriate visualizations, represent the univariate distributions of the two numeric variables you picked, using two different geometric objects. Also find appropriate statistics. Write a sentence to interpret each visualization. *Note: make sure to add labels, a caption, and adjust some options to improve the visualization. For*

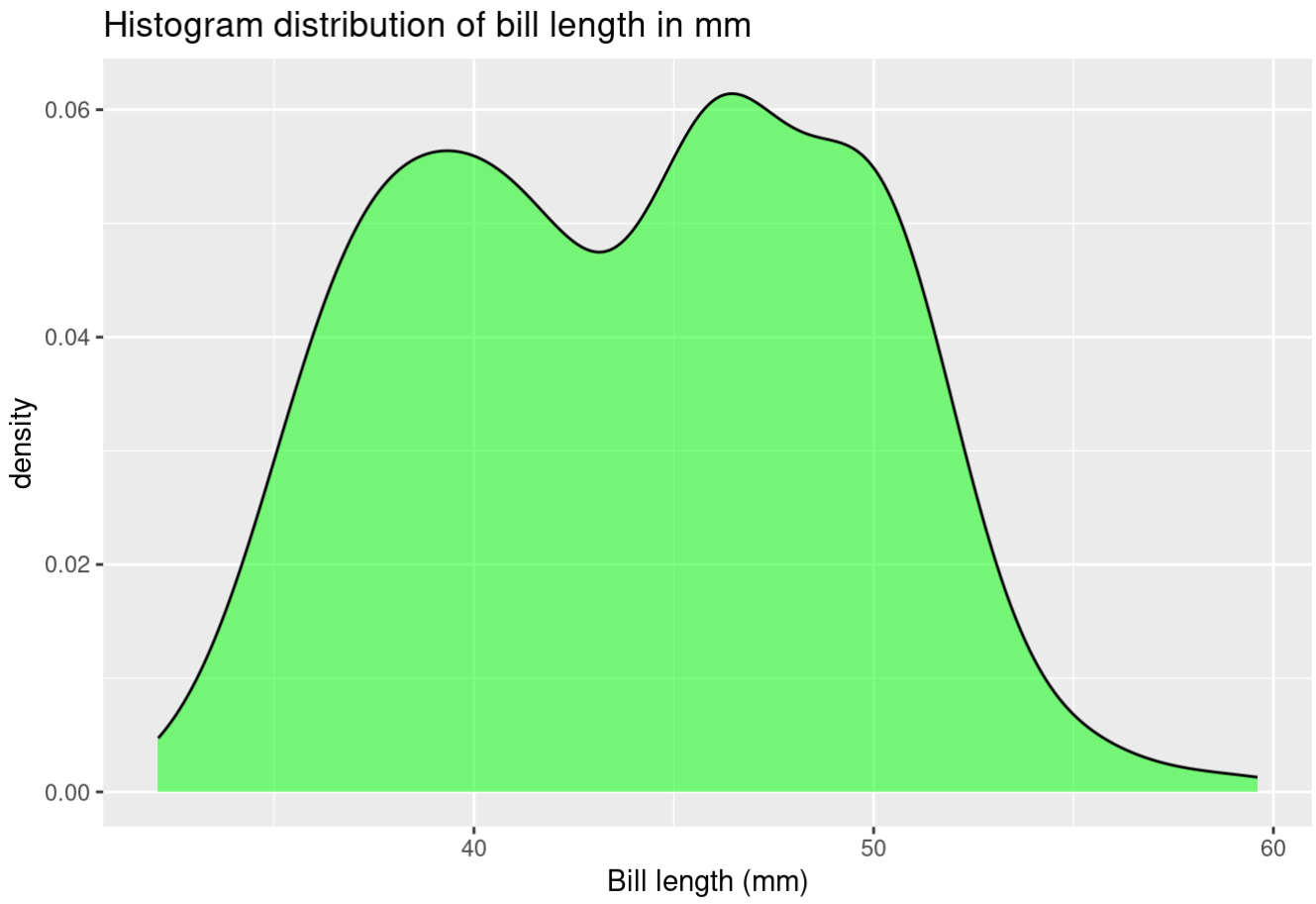
example: if using a histogram, adjust the `binwidth` and `center`. Address any warning message that might appear.

First numeric variable

```
# Histogram and Density plot of bill length variable of the penguins
ggplot(data = penguins) +
    geom_histogram(aes(x = bill_length_mm), binwidth = 2, na.rm = T, fill = 'blue', colo
r = 'black') +
    labs(title = 'Histogram distribution of bill length in mm',
         x = 'Bill length (mm)',
         y = 'frequency',
         caption = 'bill length data from `palmerpenguins` dataset')
```




```
ggplot(data = penguins) +
    geom_density(aes(x = bill_length_mm), fill = 'green', alpha = 0.5, na.rm = T) +
    labs(title = 'Histogram distribution of bill length in mm',
         x = 'Bill length (mm)',
         y = 'density',
         caption = 'bill length data from `palmerpenguins` dataset')
```



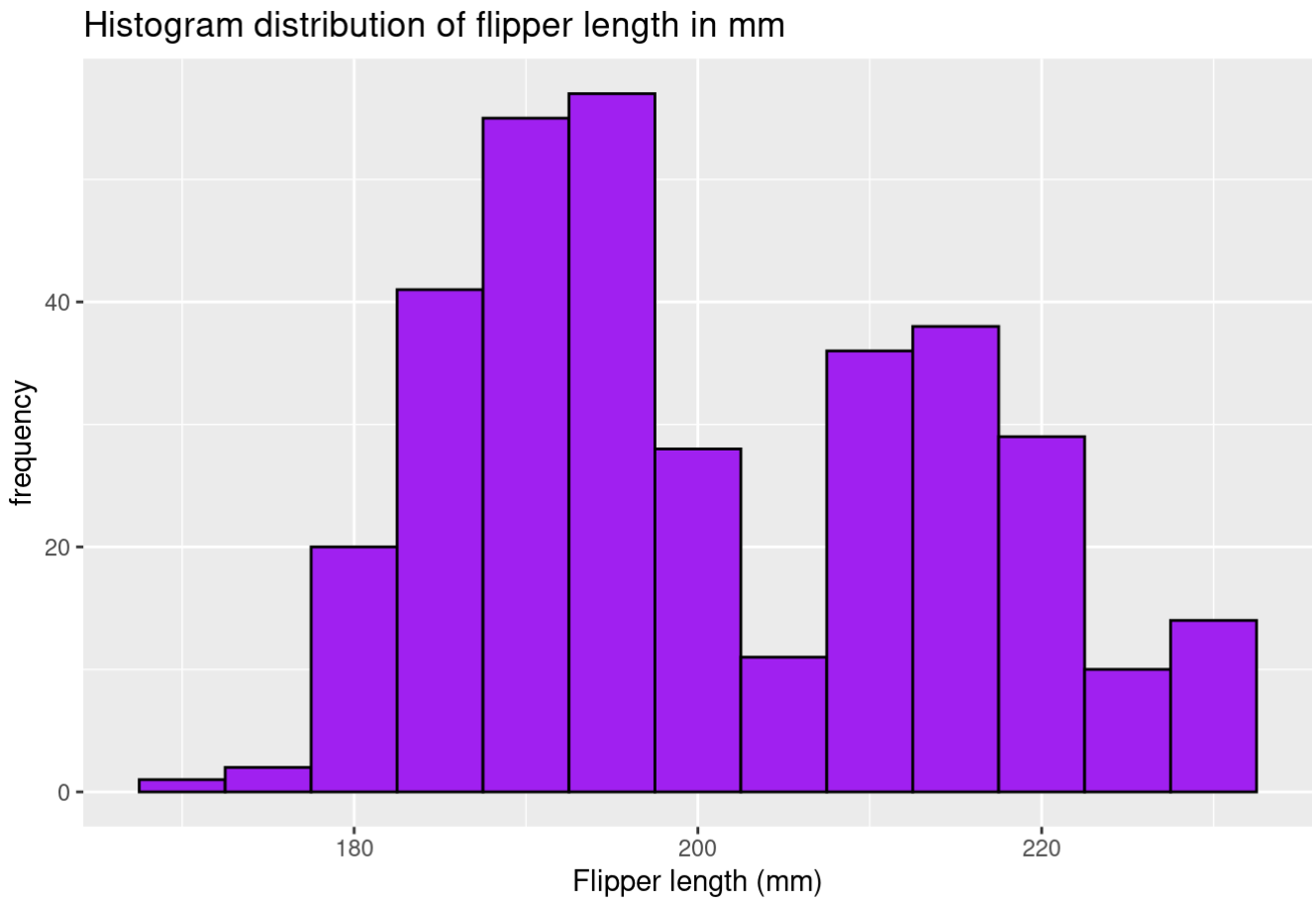

```
summary(penguins$bill_length_mm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   32.10   39.23   44.45   43.92   48.50   59.60       2
```

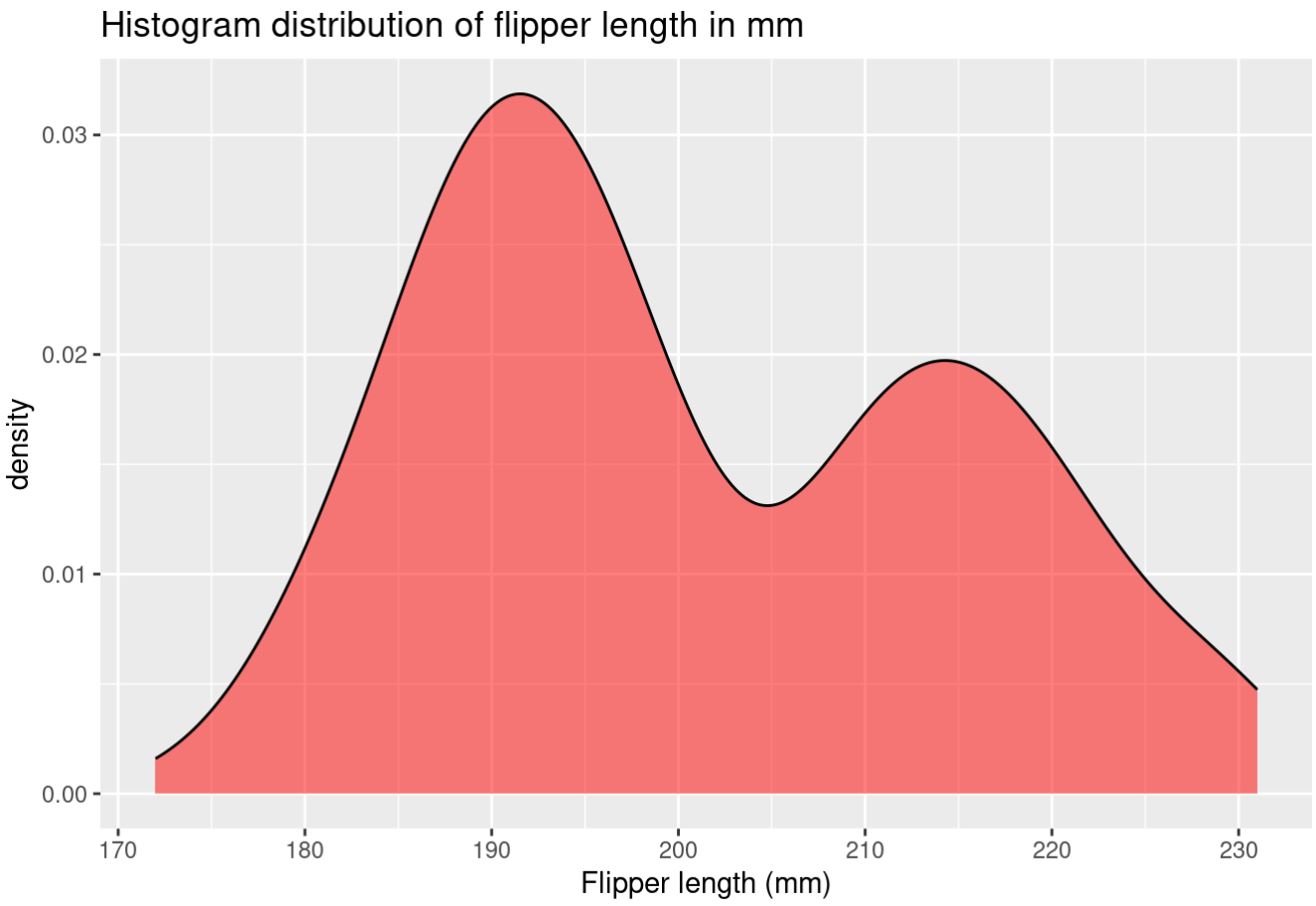
The histogram plot shows us that there is some positive skew as a few penguins have very long bill lengths compared to the rest of the penguins and that the distribution is bimodal. The density plot also shows that there is some slight positive skew and this plot shows us more clearly that the distribution is bimodal, most likely from a male and female difference.

Second numeric variable

```
# your code goes below (replace this comment with something meaningful)
ggplot(data = penguins) +
    geom_histogram(aes(x = flipper_length_mm), binwidth = 5, na.rm = T, fill = 'purple',
color = 'black') +
    labs(title = 'Histogram distribution of flipper length in mm',
         x = 'Flipper length (mm)',
         y = 'frequency',
         caption = 'Flipper length data from `palmerpenguins` dataset')
```




```
ggplot(data = penguins) +
    geom_density(aes(x = flipper_length_mm), fill = 'red', alpha = 0.5, na.rm = T) +
    labs(title = 'Histogram distribution of flipper length in mm',
         x = 'Flipper length (mm)',
         y = 'density',
         caption = 'Flipper length data from `palmerpenguins` dataset')
```



Flipper length data from `palmerpenguins` dataset

```
summary(penguins$flipper_length_mm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   172.0   190.0   197.0   200.9   213.0   231.0       2
```

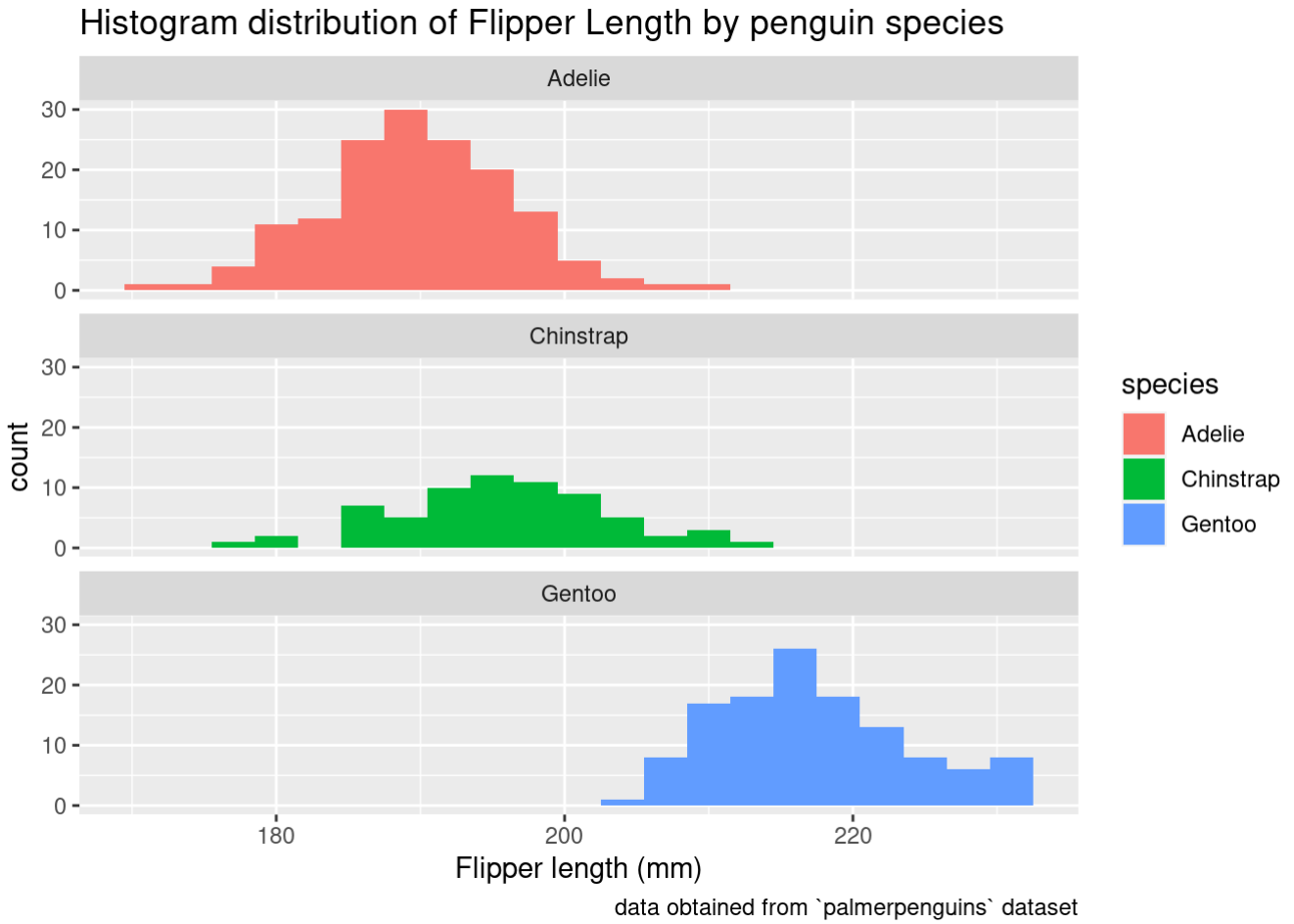
Question 6: (2 pts)

Using appropriate visualizations, represent the distributions of each of the two numeric variables you picked across species, using two different geometric objects. Write a sentence to interpret each visualization. *Note: make sure to add labels, a caption, and adjust some options to improve the visualization. Address any warning message that might appear.*

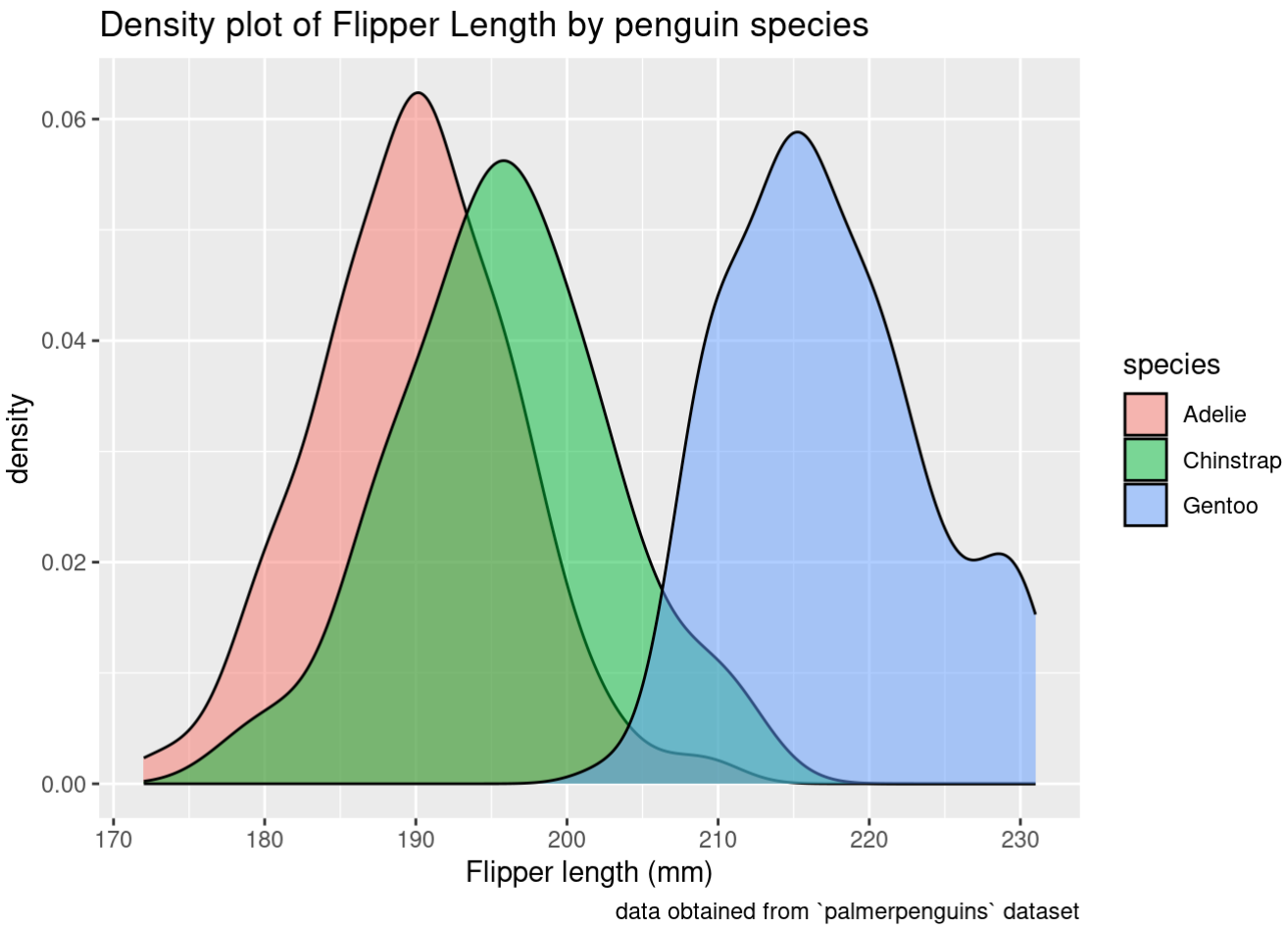
> First numeric variable and species

```
# Visualizes Flipper length distribution using a histogram and density plot

ggplot(data = penguins) +
    geom_histogram(aes(x = flipper_length_mm, fill = species), binwidth = 3, na.rm = T)
+
    facet_wrap(~species, ncol = 1) +
    labs(title = 'Histogram distribution of Flipper Length by penguin species',
         x = 'Flipper length (mm)',
         caption = 'data obtained from `palmerpenguins` dataset')
```



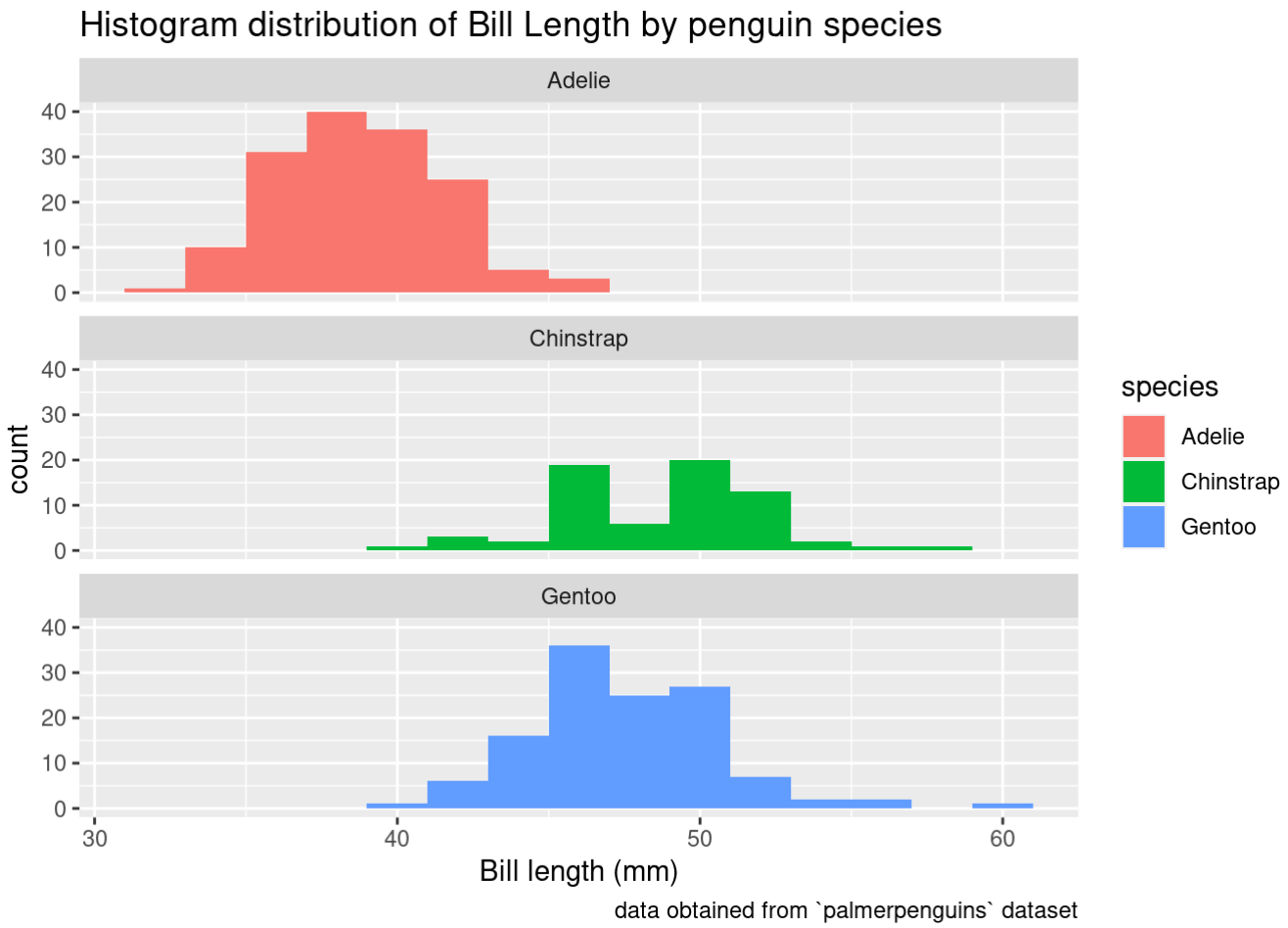

```
ggplot(data = penguins) +
    geom_density(aes(x = flipper_length_mm, fill = species), alpha = 0.5, na.rm = T) +
    labs(title = 'Density plot of Flipper Length by penguin species',
         x = 'Flipper length (mm)',
         caption = 'data obtained from `palmerpenguins` dataset')
```



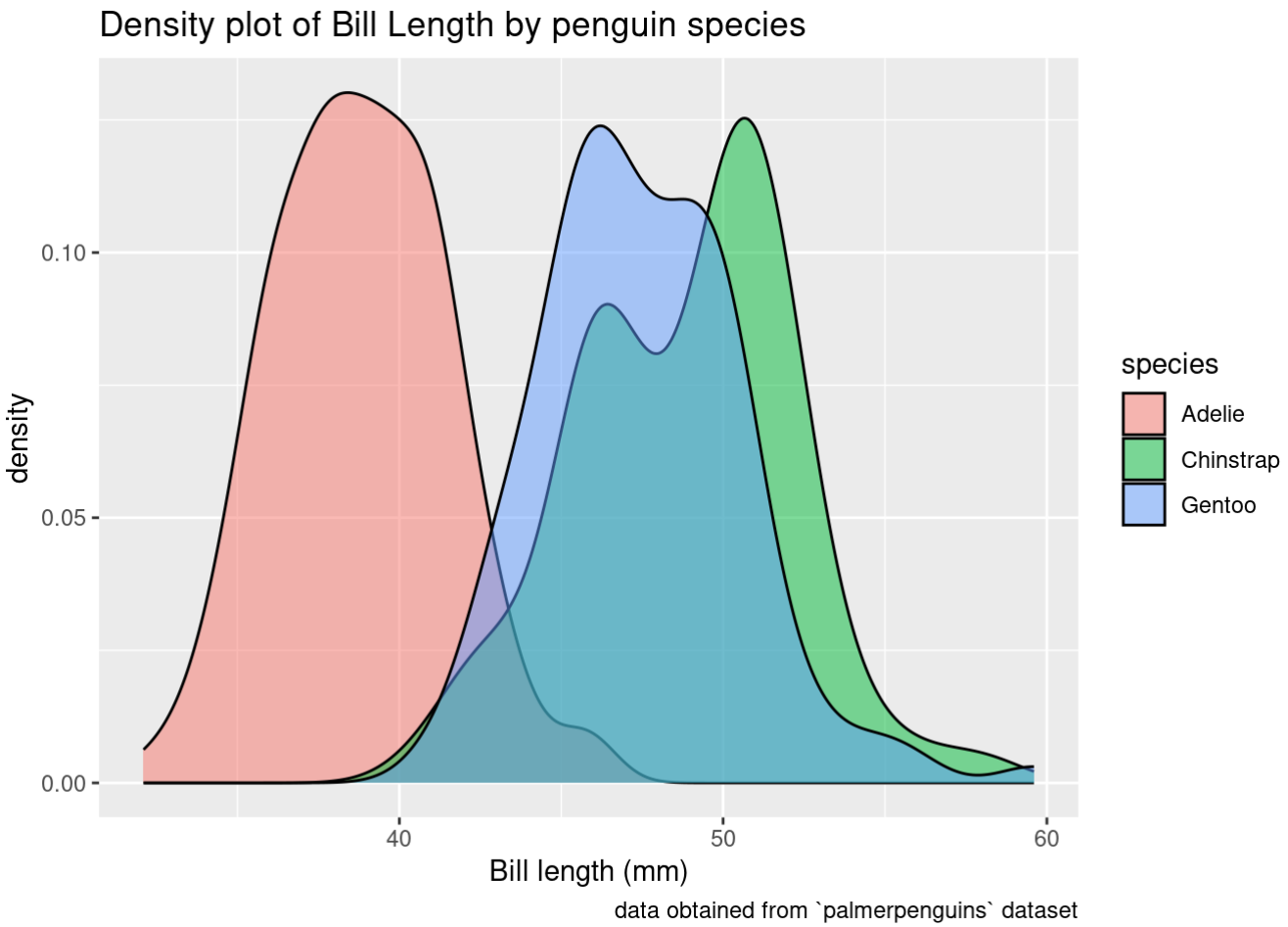
In the histogram, we can clearly see the differences between the distributions of flipper lengths across the 3 species of penguins in the dataset, and that becomes even more clear when using the density plot to overlay the distributions of the flipper lengths of the different species. We can see that the Gentoo penguins generally have the longest flipper lengths while the Adelie penguins have the shorter flipper lengths and the Chinstrap penguins have a flipper lengths that lies between the 2 other penguins

Second numeric variable and species

```
# Visualizes Bill length distribution using a histogram and density plot
ggplot(data = penguins) +
    geom_histogram(aes(x = bill_length_mm, fill = species),binwidth = 2, na.rm = T) +
    facet_wrap(~species, ncol = 1) +
    labs(title = 'Histogram distribution of Bill Length by penguin species',
         x = 'Bill length (mm)',
         caption = 'data obtained from `palmerpenguins` dataset')
```




```
ggplot(data = penguins) +
    geom_density(aes(x = bill_length_mm, fill = species), alpha = 0.5, na.rm = T) +
    labs(title = 'Density plot of Bill Length by penguin species',
         x = 'Bill length (mm)',
         caption = 'data obtained from `palmerpenguins` dataset')
```

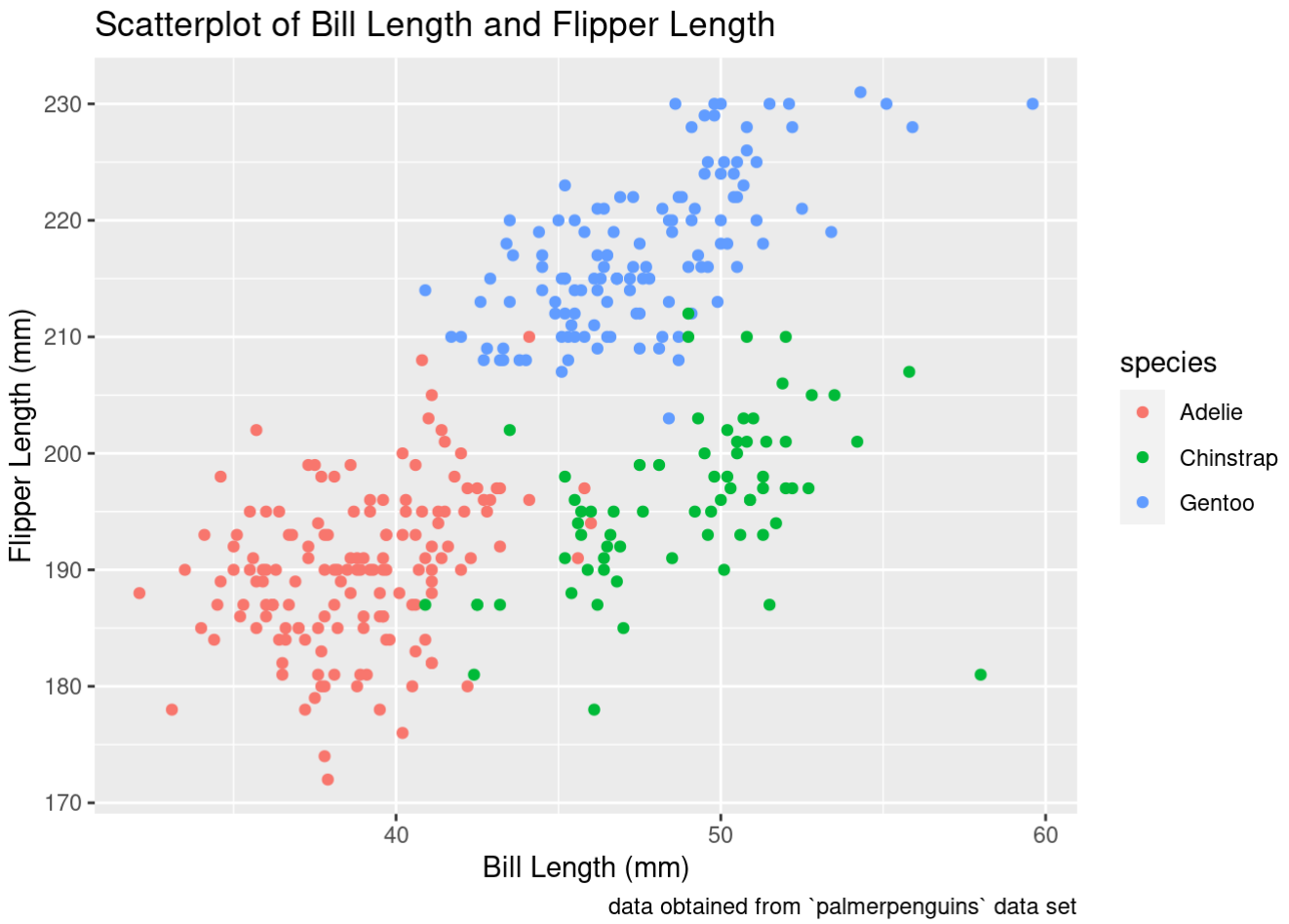


In the histogram distribution of bill length across the 3 species of penguins, we can see that Adelie penguins generally have shorter bills than the Chinstrap and Gentoo penguins and the Gentoo and Chinstrap penguins have very similar bill length distribution. The density plot reiterates the observation that the distribution of the bill lengths of the Chinstrap and Gentoo penguins are very similar and the Adelie penguins' bill length distribution is much lower than the other 2 species.

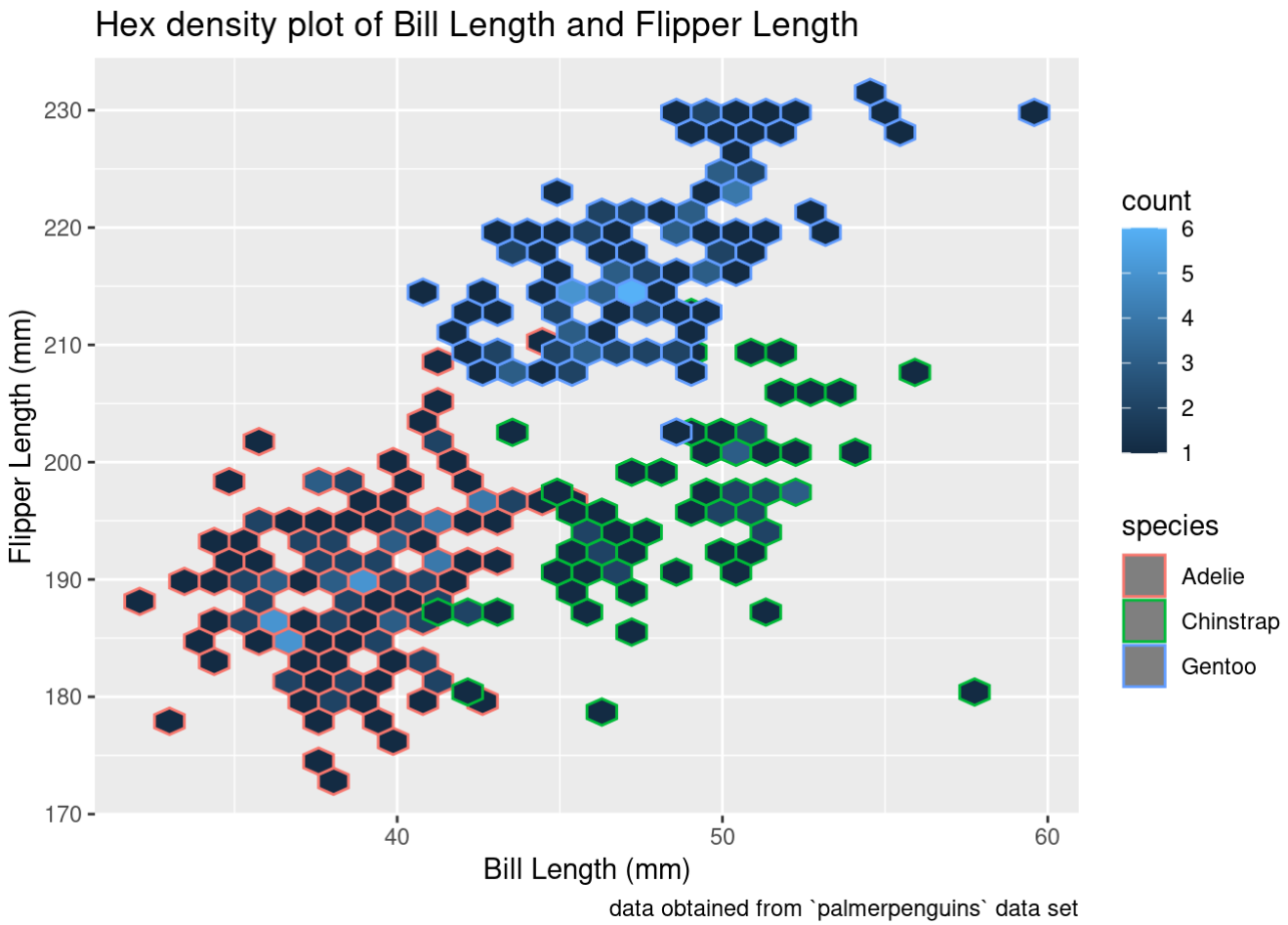
Question 7: (2 pts)

Using an appropriate visualization, represent the relationship between the three variables (`species` and the two numeric variables you picked). Write a sentence to interpret this visualization. Also discuss if your data analysis met your expectations. *Note: make sure to add labels, a caption, and adjust some options to improve the visualization. Address any warning message that might appear.*

```
# Using hex density plot and scatter plot to visualize the relationship across species,
bill length, and flipper length
ggplot(data = penguins) +
    geom_point(aes(x = bill_length_mm, y = flipper_length_mm, color = species), na.rm =
T) +
    labs(title = 'Scatterplot of Bill Length and Flipper Length',
         x = 'Bill Length (mm)',
         y = 'Flipper Length (mm)',
         caption = 'data obtained from `palmerpenguins` data set')
```




```
ggplot(data = penguins) +
    geom_hex(aes(x = bill_length_mm, y = flipper_length_mm, color = species), na.rm = T)
+
    labs(title = 'Hex density plot of Bill Length and Flipper Length',
         x = 'Bill Length (mm)',
         y = 'Flipper Length (mm)',
         caption = 'data obtained from `palmerpenguins` data set')
```



From both the scatter plot and hex density plot, we can see that across bill length, flipper length, and species we can see clustering where the 3 species in the data set occur in a specific bill and flipper length. This is something I expected to see in the visualization because generally species will have similar measurements, and while it is true that the measurements can overlap and we do see that in the visualization for variables like bill length the fact remains that we can still see a pattern across the 3 variables in this case.

Part 2

In this part, you will interact with a database that lives on the **edupod server** so make sure to log into https://edupod.cns.utexas.edu/ (https://edupod.cns.utexas.edu/)

Databases are commonly used when you want to perform operations on multiple, large datasets without reading them into memory on your local computer. Indeed, datasets can be many gigabytes and often cannot be "physically" imported!

We will interact with a database called `filmrev.db`, consisting of over 10 million user ratings for over 10,000 movies, and some metadata about each film (title, genre, etc.). The package `dbplyr` let us write code that can get translated to a `SQL` query based on `dplyr` commands, which are sent to the database to perform operations. That's why you should only use `dplyr` commands in this part.

Note: Since we are sending queries to a database, it is normal that running your code might take a little longer than usual. So make sure to submit code when you are pretty sure it's going to work!

```
# Upload package
library(dbplyr)

# Make a connection with the database
connection <- DBI::dbConnect(RSQLite::SQLite(),
                             "/stor/work/SDS322E_LG_Fall2023/filmrev.db")
```

We can take a look at the tables contained in the database:

```
# Content of our connection to the database
src_dbi(connection)
```

```
## src:  sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
## tbls: movies, ratings, tags
```

We will work with the `ratings` and `movies` datasets so let's save them as objects in our environment.

```
# Content of our connection to the database
ratings <- tbl(connection, "ratings")
movies <- tbl(connection, "movies")
tags <- tbl(connection, "tags")

# They do not appear as data frames in our environment but we can still take a look at t
heir content with head():
head(ratings)
```

```
## # Source:   SQL [6 x 4]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   userId movieId rating timestamp
##    <int>   <int>  <dbl>     <int>
## 1      1     122      5 838985046
## 2      1     185      5 838983525
## 3      1     231      5 838983392
## 4      1     292      5 838983421
## 5      1     316      5 838983392
## 6      1     329      5 838983392
```

```
head(movies)
```

```
## # Source:   SQL [6 x 3]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   movieId title                              genres
##   <chr>   <chr>                              <chr>
## 1 1       Toy Story (1995)                   Adventure|Animation|Children|Comed…
## 2 2       Jumanji (1995)                     Adventure|Children|Fantasy
## 3 3       Grumpier Old Men (1995)            Comedy|Romance
## 4 4       Waiting to Exhale (1995)           Comedy|Drama|Romance
## 5 5       Father of the Bride Part II (1995) Comedy
## 6 6       Heat (1995)                        Action|Crime|Thriller
```

```
head(tags)
```

```
## # Source:   SQL [6 x 4]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   userId movieId tag             timestamp
##   <chr>  <chr>   <chr>           <chr>
## 1 15     4973    excellent!      1215184630
## 2 20     1747    politics        1188263867
## 3 20     1747    satire          1188263867
## 4 20     2424    chick flick 212 1188263835
## 5 20     2424    hanks           1188263835
## 6 20     2424    ryan            1188263835
```

Question 8: (1 pt)

Identify the key variable(s) we would need to join 1) `ratings` and `movies`, 2) `ratings` and `tags`, 3) `movies` and `tags`.

The key variable we would need to join these data frames is 'movieId'. The key variables to join ratings and tags is 'movieId', 'userId', and 'timestamp'. The key varible to join movies and tags is 'movieId'

Question 9: (2 pts)

Let's focus on the `ratings` and `movies` datasets. Using `dplyr` core functions, find how many distinct movies there are in each dataset. Do they contain the same number of movies?

```
# Using the summarize function to find all the distinct movie IDs in both the `ratings` and `movies` dataset
ratings |>
  summarize(n_distinct(movieId))
```

```
## # Source:   SQL [1 x 1]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   `n_distinct(movieId)`
##                   <int>
## 1                 10677
```

```
movies |>
  summarize(n_distinct(movieId))
```

```
## # Source:   SQL [1 x 1]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   `n_distinct(movieId)`
##                   <int>
## 1                 10681
```

They do not contain the same number of movies in each of the dataset, they differ by 4 distinct movies.

If we wanted to look if some movies that are included in `movies` but not in `ratings`, what joining `dplyr` functions would we use? *You do not need to run this command as it will take a long time to run. Just write which function you would use.*

We would use anti_join(movies, ratings, by = 'movieId')

Question 10: (2 pts)

Let's summarize the `ratings` dataset per movie. Using `dplyr` core functions, find the mean rating (call it `mean_rating`) and also the number of ratings (call it `num_ratings`) **for each movie**. Save the resulting dataset in your environment as `ratings_per_movie`.

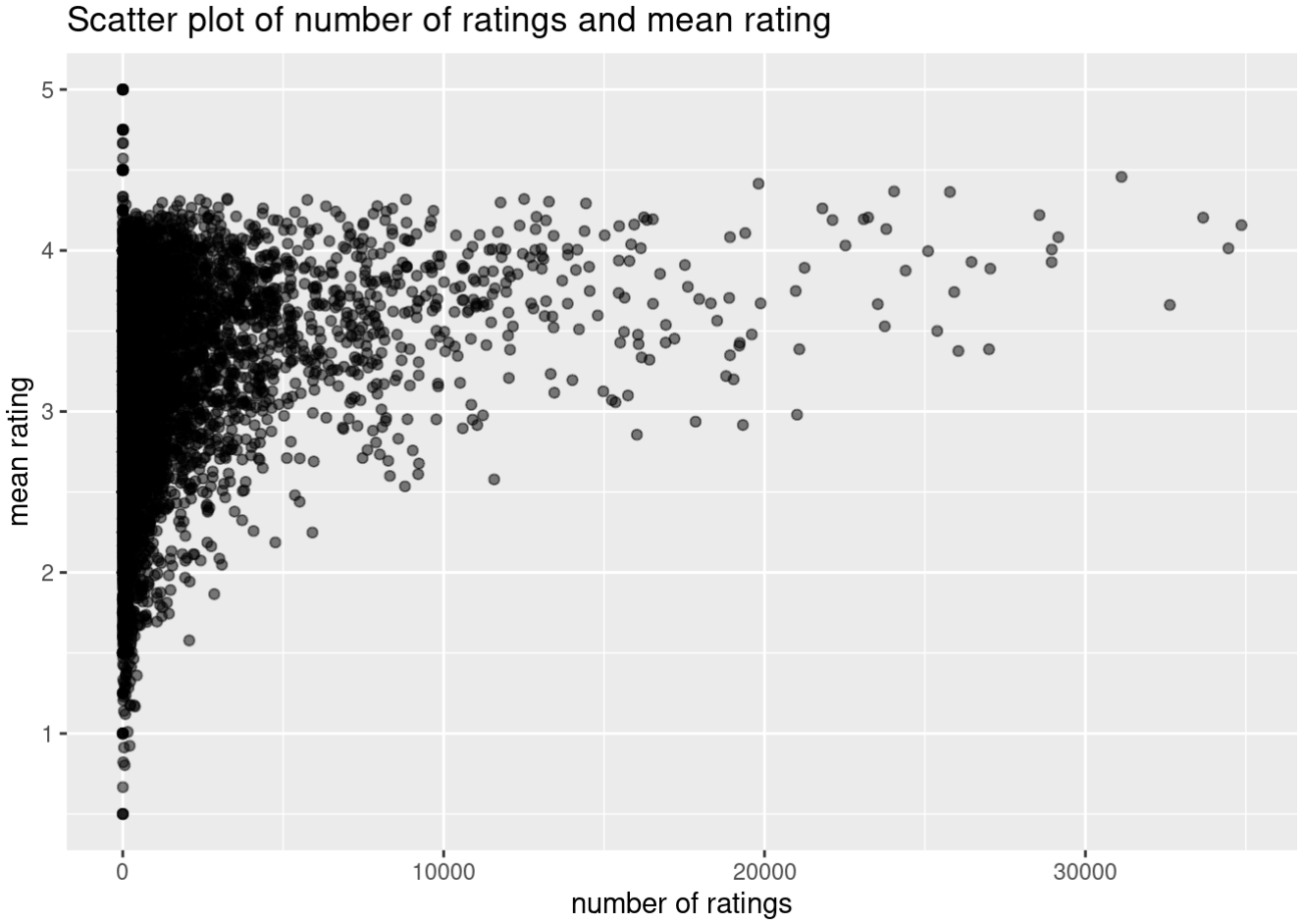
```
# Using the ratings dataset, group the ratings by movie and find the mean rating and num
ber of ratings per movie
ratings_per_movie <- ratings |>
                        group_by(movieId) |>
                        summarize(mean_rating = mean(rating, na.rm = T),
                                  num_ratings = n())

head(ratings_per_movie)
```

```
## # Source:   SQL [6 x 3]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   movieId mean_rating num_ratings
##     <int>       <dbl>       <int>
## 1       1        3.93       26449
## 2       2        3.21       12032
## 3       3        3.15        7790
## 4       4        2.86        1764
## 5       5        3.08        7135
## 6       6        3.81       13696
```

Make a quick scatterplot to investigate the relationship between `num_ratings` and `mean_rating`. Write a sentence to interpret this visualization.

```
# Scatter plot of number of ratings and mean rating from the ratings_per_movie dataset w
e created
ggplot(data = ratings_per_movie) +
  geom_point(aes(x = num_ratings, y = mean_rating), alpha = 0.5) +
  labs(title = 'Scatter plot of number of ratings and mean rating',
       x = 'number of ratings',
       y = 'mean rating')
```



This plot shows us that there were a lot of movies in the `ratings` dataset that had a low number of ratings. For the movies with low number of ratings, the mean rating varied a lot while for the movies with a higher number of ratings generally did not vary too much and in fact as the number of ratings increased the spread of the mean ratings decreased.

Question 11: (2 pts)

Consider the `ratings_per_movie` dataset created previously. Now, let's find which movie has the highest mean rating IF it has also received a decent amount of ratings. Only keep the movies with a decent amount of ratings (pick something you think makes sense!) then keep the top 5 movies with the maximum average rating by using `slice_max(mean_rating, n = 5)`. *Note: slice() and top_n() functions do not translate with SQL queries so we cannot use those.* To find which movie corresponds to the `movieId`, we need to have that information from the `movies` dataset. Pipe `left_join()` to join `movies`. Which movie has the highest average rating for a large number of ratings?

```
# Filtering the ratings_per_movie dataset for an adequate amount of ratings and finding
the top 5 highest average rating movie
ratings_per_movie |>
  left_join(movies, by = 'movieId') |>
  filter(num_ratings >= 5000) |>
  slice_max(mean_rating, n = 5)
```

```
## # Source:   SQL [5 x 5]
## # Database: sqlite 3.41.2 [/stor/work/SDS322E_LG_Fall2023/filmrev.db]
##   movieId mean_rating num_ratings title                          genres
##     <int>       <dbl>       <int> <chr>                          <chr>
## 1     318        4.46       31126 Shawshank Redemption, The (1994) Drama
## 2     858        4.42       19814 Godfather, The (1972)            Crime|Drama
## 3      50        4.37       24037 Usual Suspects, The (1995)       Crime|Myster…
## 4     527        4.36       25777 Schindler's List (1993)          Drama|War
## 5     912        4.32       12507 Casablanca (1942)                Drama|Romance
```

The movie that has the highest average rating and with an sufficiently high amount of ratings is Shawshank Redemption, The (1994).

Question 12: (1 pt)

You can convert your `dplyr` code into `SQL` queries (indeed, this is what happens behind the scenes thanks to `dbplyr`)! `SQL` stands for *structured query language* and is commonly used to communicate with databases. Let's translate your `dplyr` query from the previous question into the equivalent `SQL` query. Simply pipe your code into `show_query()`. How does the `SQL` query compared to the `dplyr` query?

```
# your code goes below (replace this comment with something meaningful)
ratings_per_movie |>
  left_join(movies, by = 'movieId') |>
  filter(num_ratings >= 5000) |>
  slice_max(mean_rating, n = 5) |>
  show_query()
```

```
## <SQL>
## SELECT `movieId`, `mean_rating`, `num_ratings`, `title`, `genres`
## FROM (
##   SELECT *, RANK() OVER (ORDER BY `mean_rating` DESC) AS `q01`
##   FROM (
##     SELECT `LHS`.*, `title`, `genres`
##     FROM (
##       SELECT
##         `movieId`,
##         AVG(`rating`) AS `mean_rating`,
##         COUNT(*) AS `num_ratings`
##       FROM `ratings`
##       GROUP BY `movieId`
##     ) AS `LHS`
##     LEFT JOIN `movies`
##       ON (`LHS`.`movieId` = `movies`.`movieId`)
##   )
##   WHERE (`num_ratings` >= 5000.0)
## )
## WHERE (`q01` <= 5)
```

The code look similar since it uses the same key words such as 'GROUP BY' and 'LEFT JOIN' but it looks like it requires more work to think and type out the correct code for the assigned task

Finally, make sure to disconnect from the database to free up memory on the server. Run the following code when you are finished (you can always reconnect again later by running the code at the beginning).

```
# Disconnect
DBI::dbDisconnect(connection)
```

Formatting: (1 pt)

Knit your file! You can knit into html and once it knits in html, click on `Open in Browser` at the top left of the window that pops out. **Print** your html file into pdf from your browser.

Is it working? If not, try to decipher the error message: look up the error message, consult websites such as stackoverflow (https://stackoverflow.com/) or crossvalidated (https://stats.stackexchange.com/).

Finally, remember to select pages for each question when submitting your pdf to Gradescope.