Lab 1: Barplots, Histograms, Boxplots

Work through the examples and complete the exercises below.

First read in the `penguins` data. I removed the `NA` values for you.

```
penguins <- read.csv("https://raw.githubusercontent.com/marciero/MAT150/main/class_data/pengu
```

This is the penguins data we have seen.

```
head(penguins)
```

```
  ...1 species    island year bill_length_mm bill_depth_mm flipper_length_mm
1    1  Adelie Torgersen 2007           39.1          18.7               181
2    2  Adelie Torgersen 2007           39.5          17.4               186
3    3  Adelie Torgersen 2007           40.3          18.0               195
4    5  Adelie Torgersen 2007           36.7          19.3               193
5    6  Adelie Torgersen 2007           39.3          20.6               190
6    7  Adelie Torgersen 2007           38.9          17.8               181
  body_mass_g above_average_weight    sex
1        3750                    0   male
2        3800                    0 female
3        3250                    0 female
4        3450                    0 female
5        3650                    0   male
6        3625                    0 female
```

We can list the different species. The `$` is how you select columns in the data frame.

```
unique(penguins$species)
```

```
[1] "Adelie"    "Gentoo"    "Chinstrap"
```

We will also load in tidyverse if you havent done that.

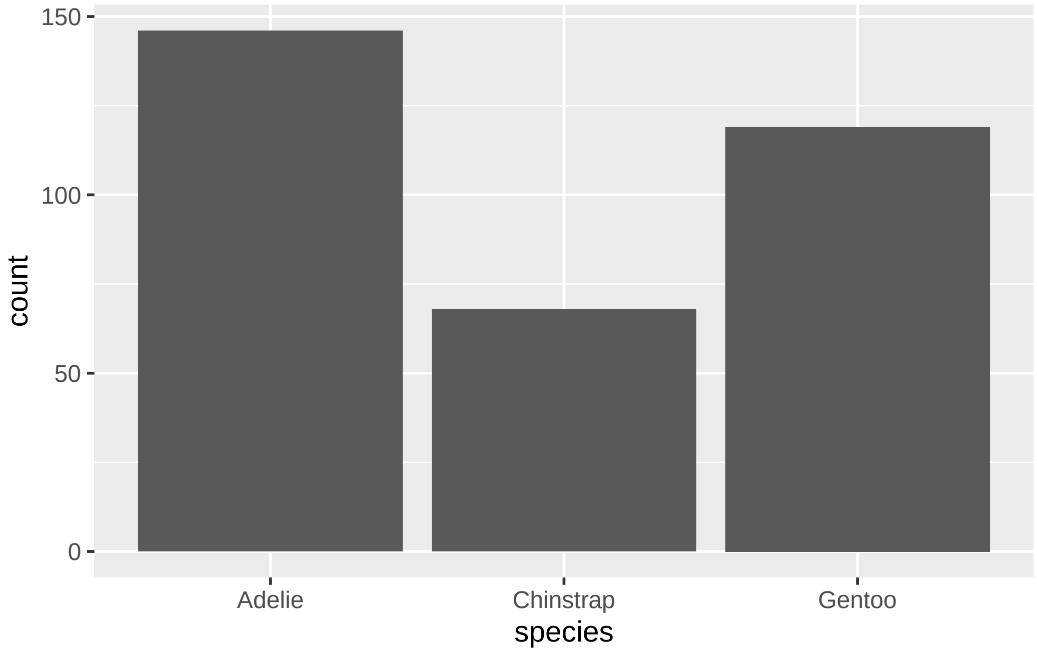
```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
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 becom
```

We will make extensive use of `ggplot` for visualizations. It is part of the `tidyverse` meta package. To create a bar plot us execute the code below.

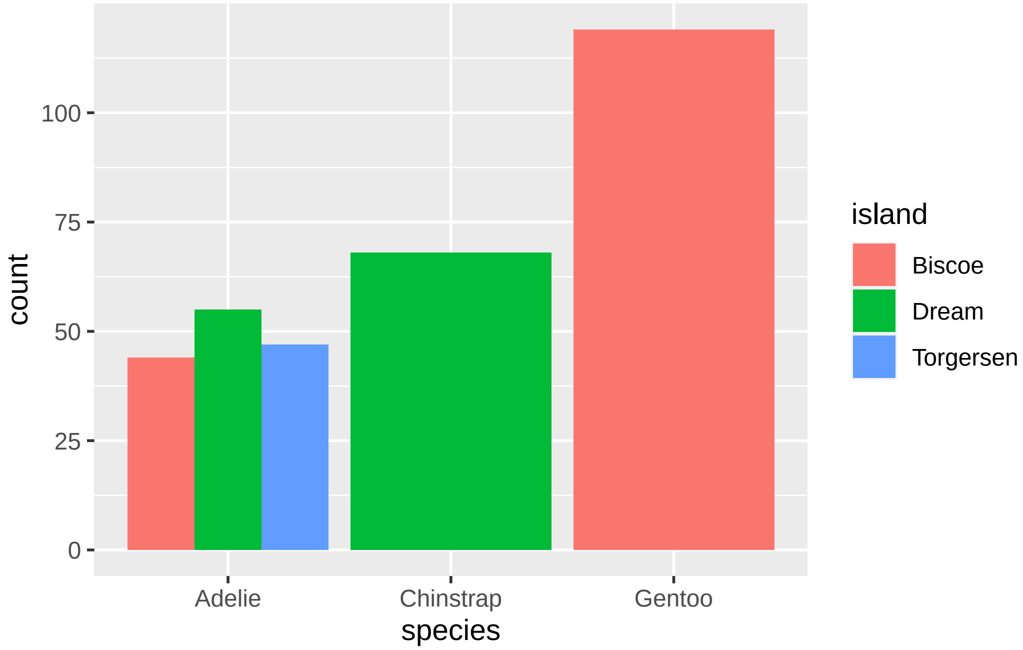
Here is how the commands work: `ggplot()` creates the plot. `aes` is for "aesthetic mapping". This is where you tell R what the x and y are, if any. Then you tell R what the data set is. Then, you add layers to the plot with the "+" sign. Every type of plot has its own "geom". Bar charts are geom_bar. We can add more arguments/options inside geom_bar(), as we will see soon.

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



Nice. If we want seperate bars, we can use the `position = "dodge"` argument. Note that it is not an aesthetic, so it sits outside the `aes()`

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



The mpg data is a built-in data set with tidyverse- we dont have to load it. Take a look at the data set using head(). You can type that right in the console rather than in your script file. Which data are categorical and which are numerical?

```r
head(mpg)
```

```
# A tibble: 6 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(l5)   f        18    29 p     compa~
2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

Exercise: For the mpg data 1. Make a bar plot of the class data. 2. Make a bar plot that also displays drv data to your bar plot using `fill = drv` (drv is the type of drive train the vehicle has) 3. Try the above with position = "dodge" 4. Try creating the graph the other way around - with heights for drv and fill indicating the class.

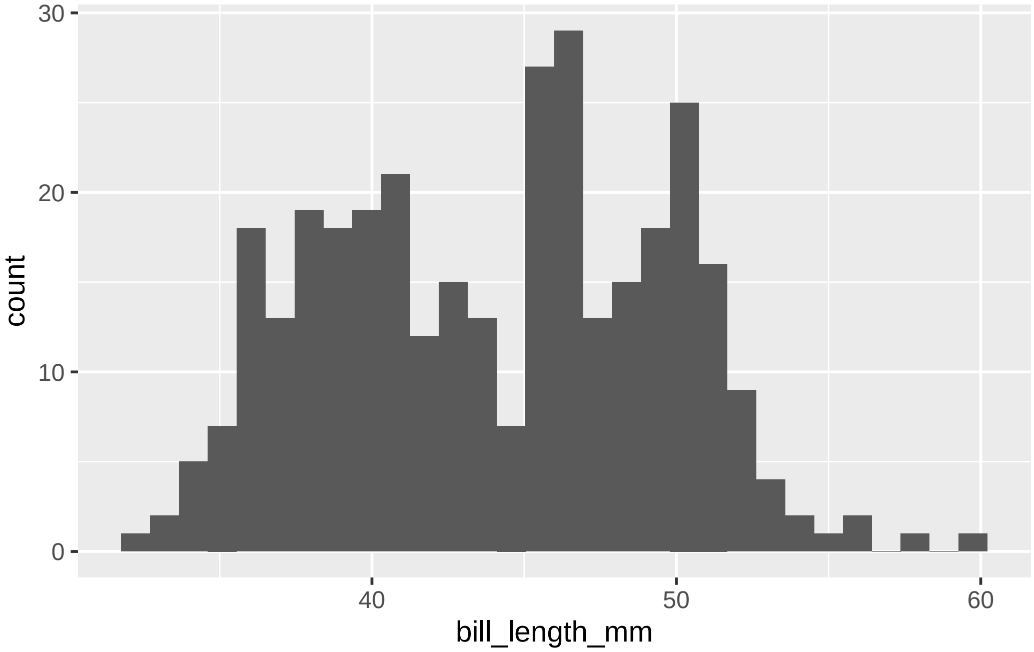
Remark: If our data is tabulated with count data, or if we simply want to plot the actual y values rather than counts, we will use `geom_col()`. As an example, we might do

Histograms

We make a "histogram" of bill lengths. This is a way to visualize the distribution of this numerical variable.

```r
ggplot(aes(x = bill_length_mm), data = penguins) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

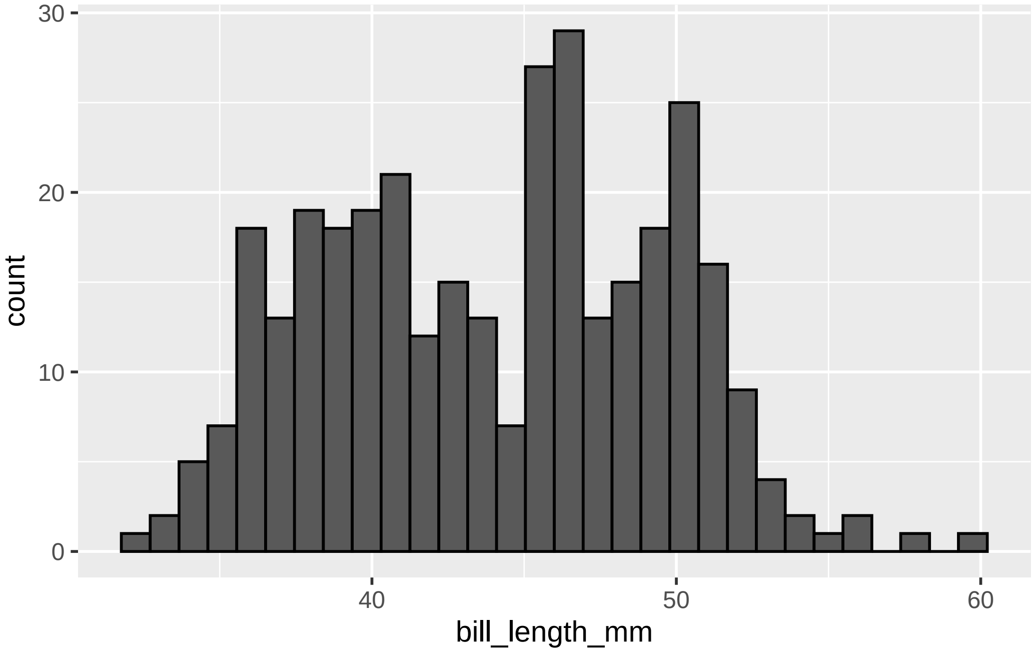


Note the the command has the same format as for bar plots. This time with geom_histogram. We can specify the binwidth with `binwidth = 30` for example. The idea is to use a binwidth that gives you a sense of the shape of the data; that is the "distribution". If we use too small a binwidth, many data points will get their own bin. The other extreme would be with a very large width, you might get just one bin. Try it!

To make our histogram look a little nicer we can use the following to outline each bin. Note that color is not an aesthetic- it does not appear inside `aes()`. That is because it is not mapping variables in the data to colors. It is simply making part to the graph a different color.

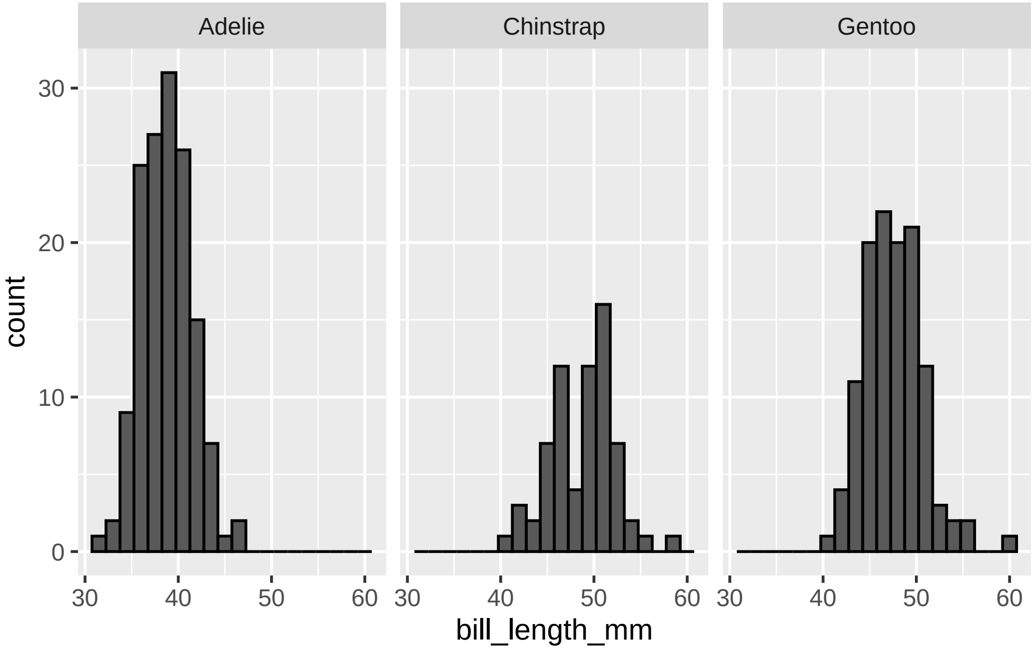
```
ggplot(aes(x = bill_length_mm), data = penguins) +
  geom_histogram(color = "black")    ## "white" works too!
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



You might notice that the distribution is somewhat "bimodal". What do you think causes that? We can actually create separate histograms using facet_wrap. Notice we add a new layer

```
ggplot(aes(x = bill_length_mm), data = penguins) +
  geom_histogram(binwidth = 1.5, color = "black")  +
  facet_wrap(~ species)
```



The `diamonds` data set comes with tidyverse, so again we dont have to load it. It is a data set of diamond prices. and other variables. Inspect the data by executing `glimpse(diamonds)`in the console. Also try `View(diamonds)` to see the data in a spreadsheet format. Now try adding an R code chunk below with glimpse, so that it will render in your finished document. (You can use the green C button with the plus sign in the top right of the editor window to add a new chunk.)

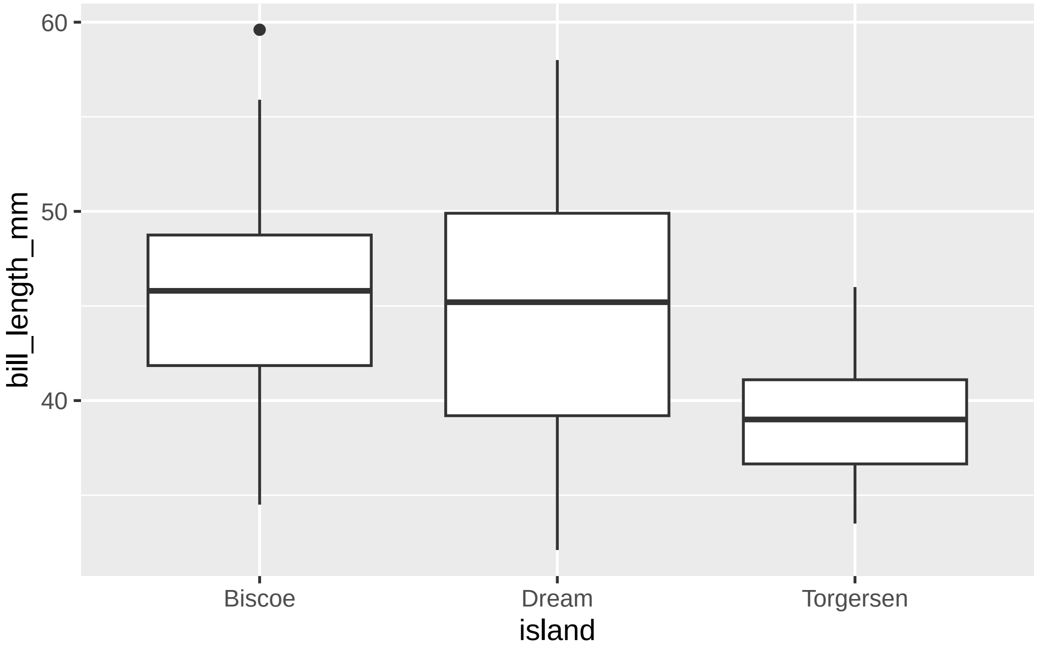
Exercise: For the diamonds data, insert code chunks to do the following

- Plot a histogram of price. Find a reasonable binwidth that illustrates the shape of the distribution
- Try `geom_density()` instead of `geom_histogram` and see what you get. (Dont use a binwidth-it does not make sense for this plot.)
- What other "categorical" variables may influence price? Try facet_wrap on one of them.
- Now try `aes(fill = )` on one of the categorical variables. (With no facet_wrap)

Boxplots.

These are created using the `geom_boxplot()`. We can specify a categorical variable as the x-variable to create side-by-side boxplots

```r
penguins %>%
  ggplot(aes(x = island, y = bill_length_mm)) +
  geom_boxplot()
```



Exercise: For the mpg data, create a boxplot of hwy mileage by class.