Module 3: Reporting, Data Wrangling and Graphing (II)

Yaqi Shi

07/12/2024

Outline

Last module we reviewed how to tidy and plot data.

In this module, we will continue our discussion on

- Graphing (ggplot2) with a real-world dataset
- Git + Github

ggplot

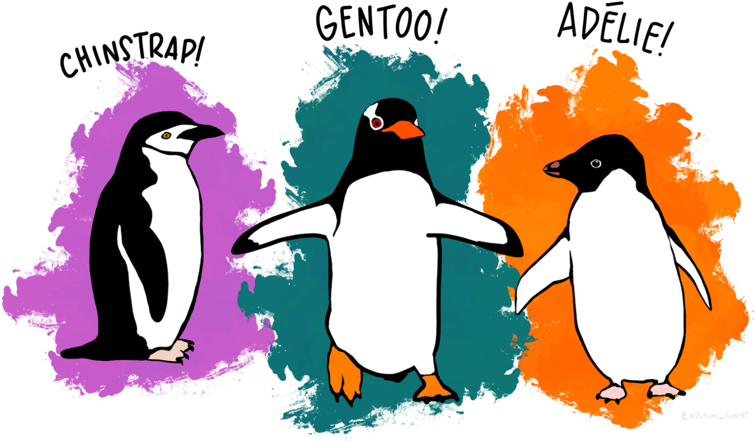
- `ggplot` is the graphing package that goes with the `tidyverse` in R
- Very powerful to make a wide range of graphics
- Same pattern as `tidyverse`"‘, but using"+" to connect.

How to write?

- Specify the data using
  `ggplot(data = diamonds)`
- Specify the x-/y-axis,
  `ggplot(data = diamonds, mapping = aes(x = cut))`
- Specify the types of plots with `geom`, e.g.
  `+ geom_bar()`

Data used - palmerpenguins

The palmerpenguins is a R package with data from the Long Term Ecological Research Network. It contains two dataset for 344 penguins and 3 species of penguins from 3 islands in the Palmer Archipelago, Antarctica.



Install and load package

From what we have learned so far, how to install package? What packages are we going to use?

Install and load package

From what we have learned so far, how to install package? What packages are we going to use?

```
install.packages("palmerpenguins")
```

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

Skim the data

How many observations? How many variables? What type?

```
head(penguins)
```

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_le
##   <fct>   <fct>              <dbl>         <dbl>
## 1 Adelie  Torgersen           39.1          18.7
## 2 Adelie  Torgersen           39.5          17.4
## 3 Adelie  Torgersen           40.3          18
## 4 Adelie  Torgersen           NA            NA
## 5 Adelie  Torgersen           36.7          19.3
## 6 Adelie  Torgersen           39.3          20.6
## # i 2 more variables: sex <fct>, year <int>
```

Quick summary of the data

Here is a summary of the data and one specific column

```
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
```

```
summary(penguins$species)
```

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

Scatter plot

Consider a scatter plot of flipper length and body mass for `species = "Adelie"`

First let's prepare the data to plot (Hint: filter).

Scatter plot

Consider a scatter plot of flipper length and body mass for `species = "Adelie"`

First let's prepare the data to plot (Hint: filter).

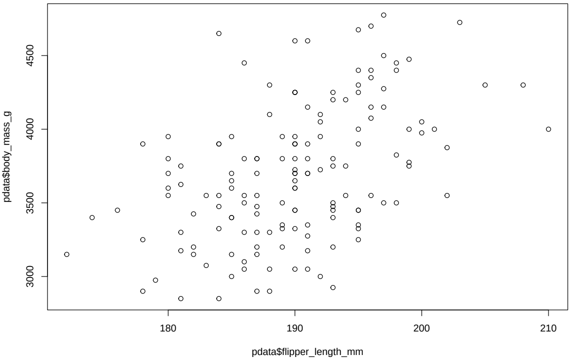
```r
# First way
pdata <- penguins %>% filter(species == "Adelie")

# Second way
pdata <- penguins[penguins$species == "Adelie", ]
```

Using basic R package

We can first try using basic way to make the plot

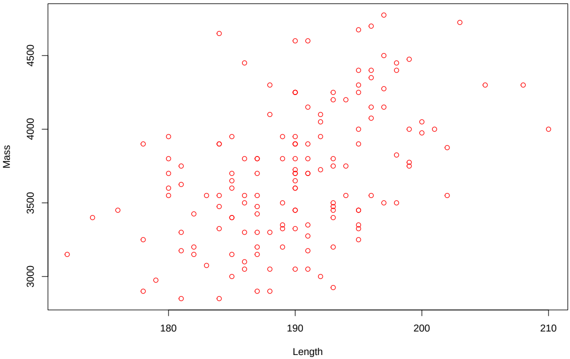
```
# Quick plot using basic R
plot(x = pdata$flipper_length_mm, y = pdata$body_mass_g)
```



Imporove it!

We can change the arguments inside to improve the plot

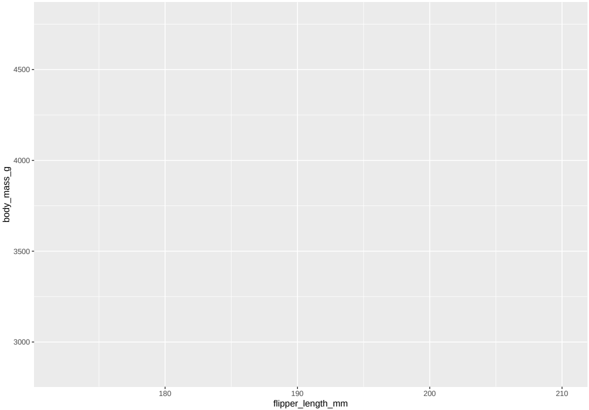
```
# change x-axis and y-axis labels
plot(x = pdata$flipper_length_mm, y = pdata$body_mass_g, xlab
```



A blank canvas

aes stands for aesthetic and tells ggplot the main characteristics of your plot (x, y, and if the color or fill vary by group)

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



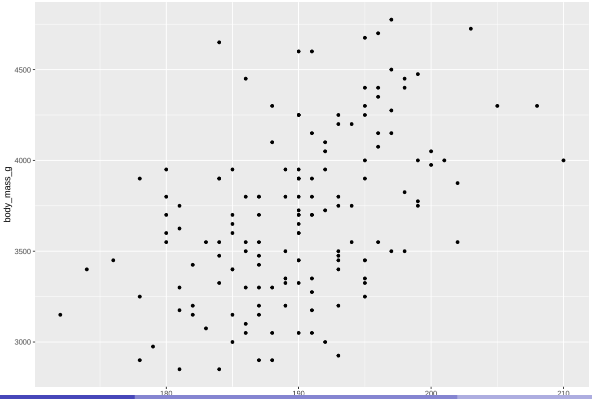
Another way of using the pipeline

```
# Using ggplot instead
penguins %>%
  filter(species == "Adelie") %>%
  ggplot()
```

Add the points in the blank plot

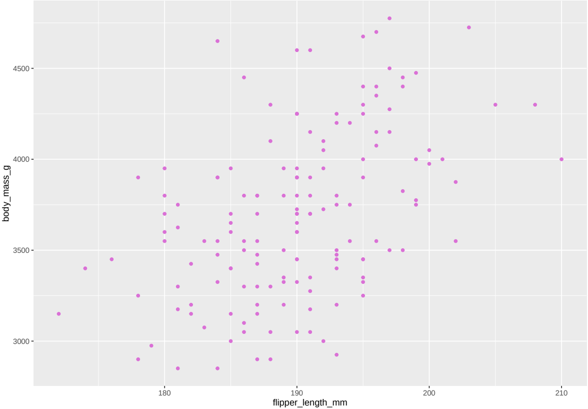
Add layers with ggplot using the +

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x=flipper_length_mm, y = body_mass_g))
```



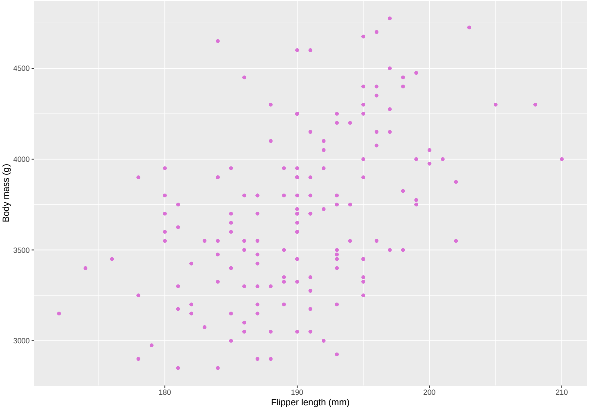
Change the color of the points

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x=flipper_length_mm, y = body_mass_g), color = "orchid")
```



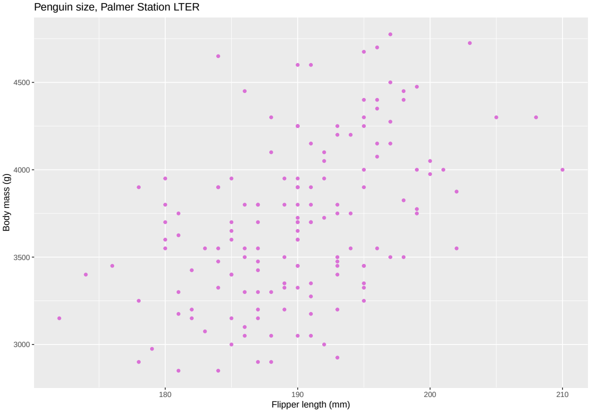
Change the label of the axis

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "orchid") +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)")
```



Add the title

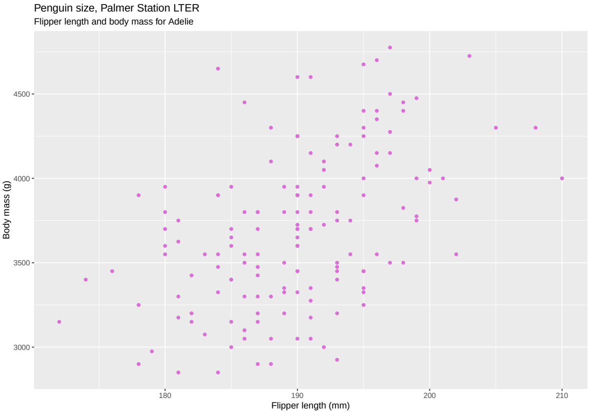
```r
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "orchid") +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  labs(title = "Penguin size, Palmer Station LTER")
```



```r
  theme(plot.title = element_text(hjust = 0.5))
```

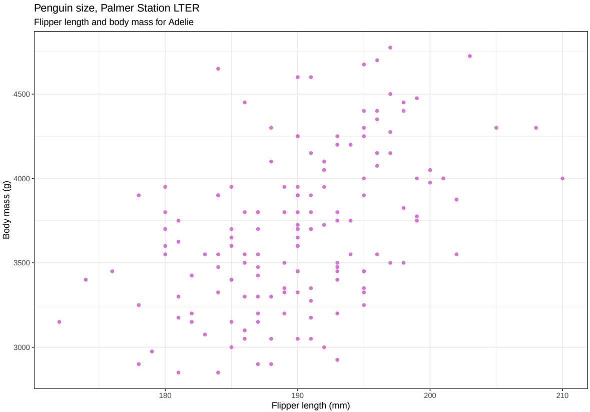
Add the subtitle

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "orchid") +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  ggtitle("Penguin size of Adelie") +
  labs(title = "Penguin size, Palmer Station LTER",
       subtitle = "Flipper length and body mass for Adelie")
```



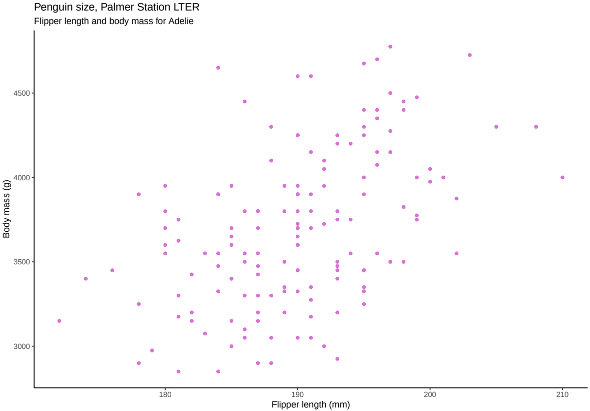
Change the theme of the plot

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "orchid") +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  ggtitle("Penguin size of Adelie") +
  labs(title = "Penguin size, Palmer Station LTER",
       subtitle = "Flipper length and body mass for Adelie")+
  theme_bw()
```



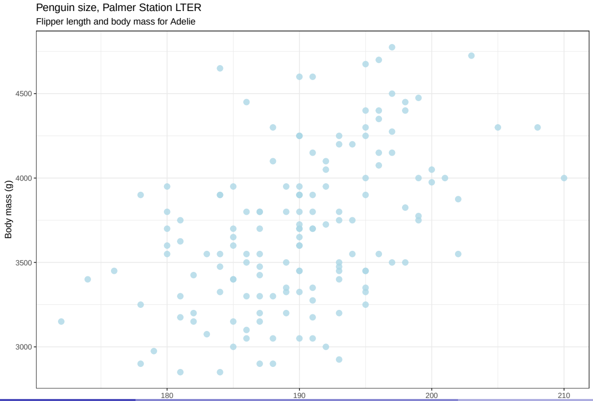
Change the theme of the plot

```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "orchid") +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  ggtitle("Penguin size of Adelie") +
  labs(title = "Penguin size, Palmer Station LTER",
       subtitle = "Flipper length and body mass for Adelie")+
  theme_classic()
```



Change the size of the points

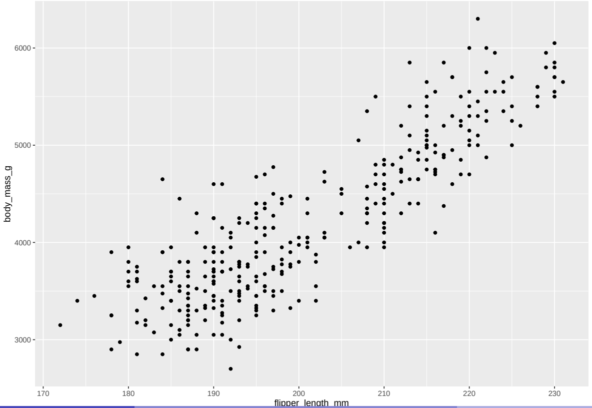
```
penguins %>%
  filter(species == "Adelie") %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g), color = "lightblue", size = 3,
             alpha = 0.8) +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  ggtitle("Penguin size of Adelie") +
  labs(title = "Penguin size, Palmer Station LTER",
       subtitle = "Flipper length and body mass for Adelie")+
  theme_bw()
```



Scatter plot of flipper length and body mass for ALL species

We start from a basic scatter plot

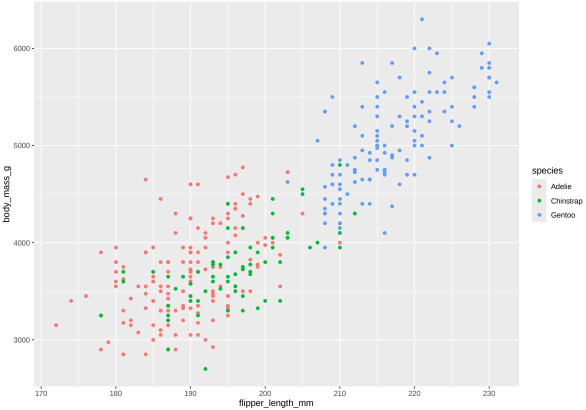
```
# Basic
# No need filter
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g))
```



How to identify different species

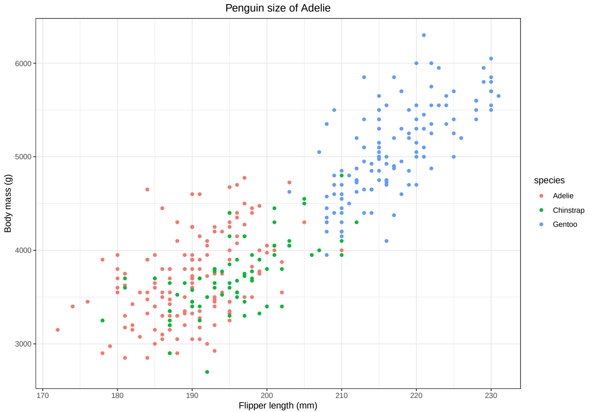
We can use different color for different species.

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



Improve the plot

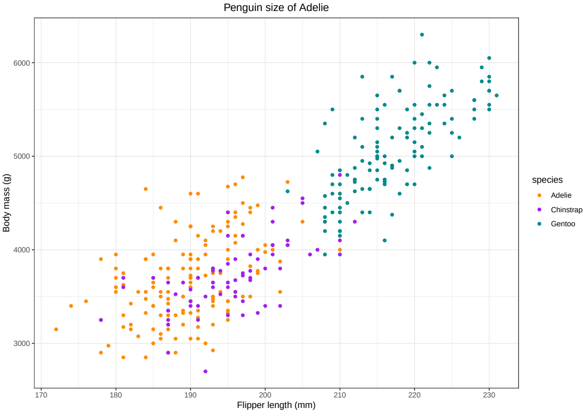
```r
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = species))  +
  xlab("Flipper length (mm)") +
  ylab("Body mass (g)") +
  ggtitle("Penguin size of Adelie") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Choose different colors:

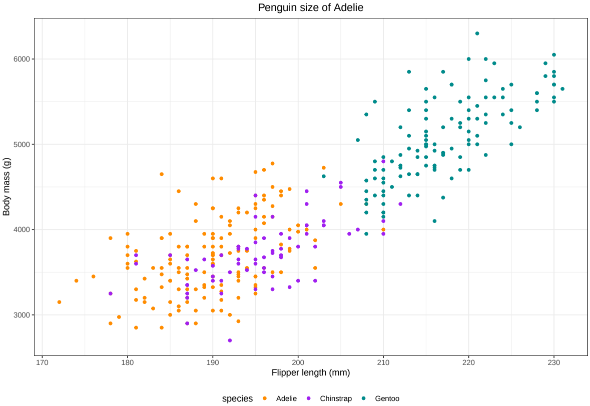
You can manually change the color rather than using pre-defined colors.

```
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = species))  +
  xlab("Flipper length (mm)") + ylab("Body mass (g)") + ggtitle("Penguin size of Adelie") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(values = c("darkorange","purple","cyan4"))
```



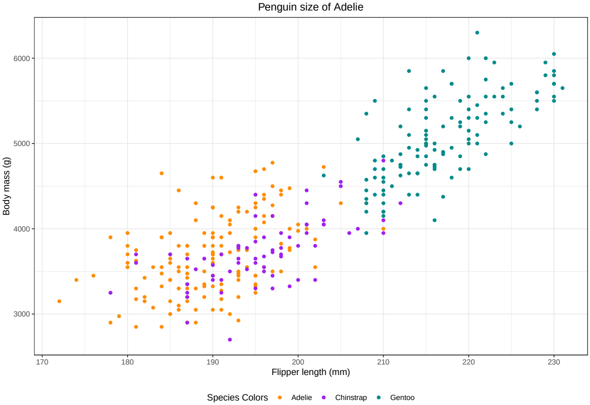
Modify the legends

```
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = species))  +
  xlab("Flipper length (mm)") + ylab("Body mass (g)") + ggtitle("Penguin size of Adelie") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(values = c("darkorange","purple","cyan4")) +
  theme(legend.position = "bottom")
```



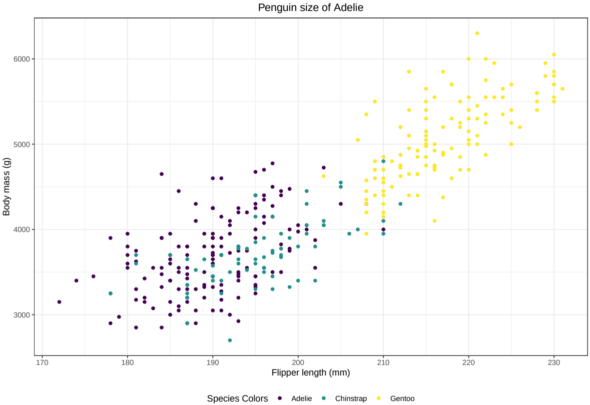
Change the title of the legend

```r
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = species))  +
  xlab("Flipper length (mm)") + ylab("Body mass (g)") + ggtitle("Penguin size of Adelie") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(values = c("darkorange","purple","cyan4")) +
  theme(legend.position = "bottom") +
  labs(color = "Species Colors")
```



Change color scheme

```
penguins %>%
  ggplot() +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = species))  +
  xlab("Flipper length (mm)") + ylab("Body mass (g)") + ggtitle("Penguin size of Adelie") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5)) +
  theme(legend.position = "bottom") +
  labs(color = "Species Colors")+
  scale_color_viridis_d()
```



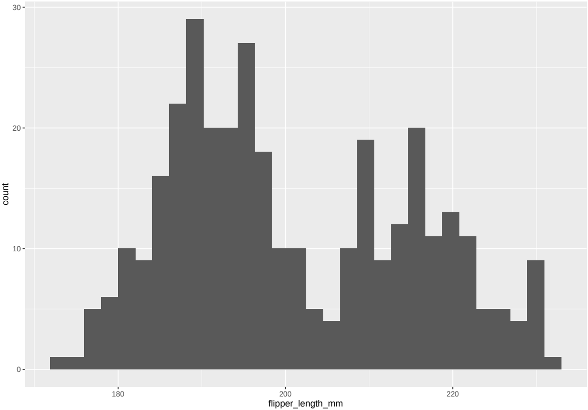
Other plot types?

Other types are also available, e.g. histograms, bar charts, box plots, line graphs and scatter plots.

Histogram: plot a histgoram of flipper length

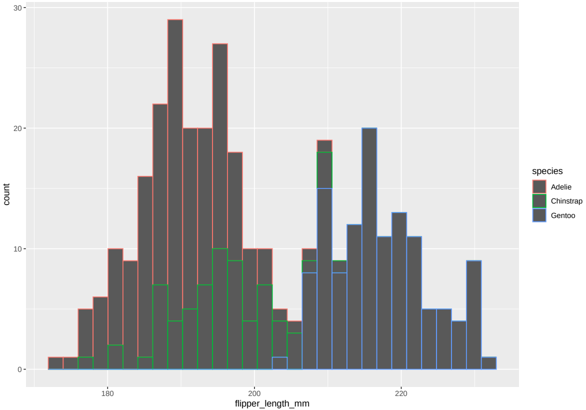
Here is a basic histogram:

```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm))
```



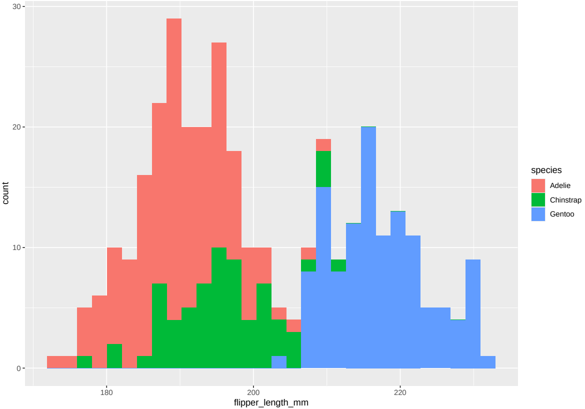
Identify different species

```
penguins %>%
  ggplot() +
  geom_histogram(aes(x = flipper_length_mm, color = species))
```



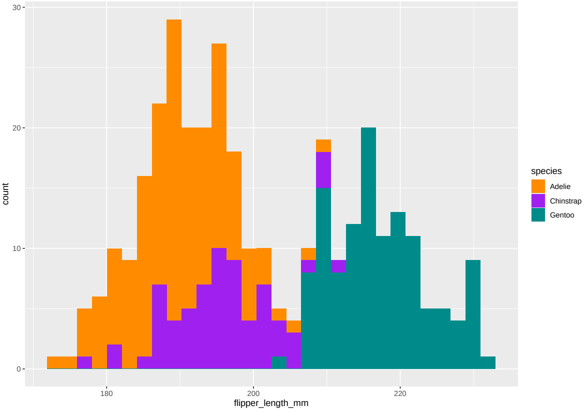
Update with a better way of coloring

```
# This looks not good
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species))
```



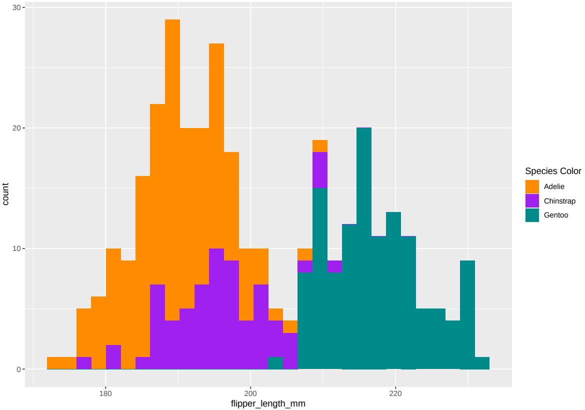
Change color manually

```r
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species)) +
   scale_fill_manual(values = c("darkorange","purple","cyan4"))
```



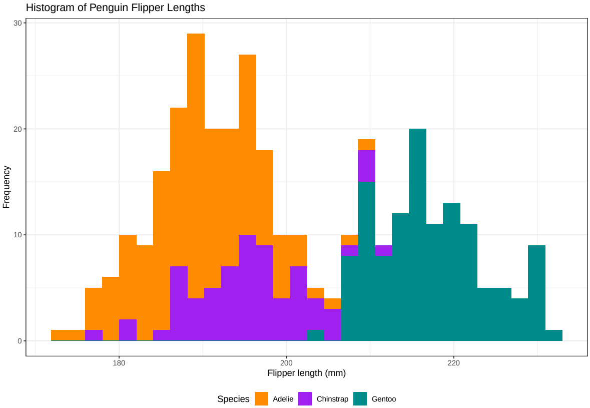
Change the title of legend

```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species)) +
   scale_fill_manual(values = c("darkorange","purple","cyan4")) +
   labs(fill = "Species Color")
```



Improve the plot!

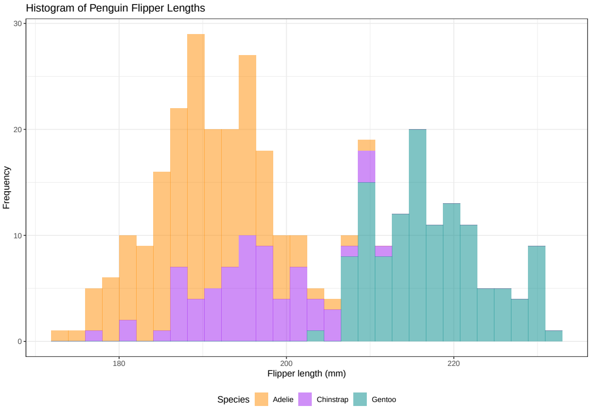
```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species)) +
   scale_fill_manual(values = c("darkorange","purple","cyan4")) +
   labs(fill = "Species") + theme_bw() +
   theme(legend.position = "bottom")  + xlab("Flipper length (mm)") +
   ylab("Frequency") + ggtitle("Histogram of Penguin Flipper Lengths")
```



Overlapping plots

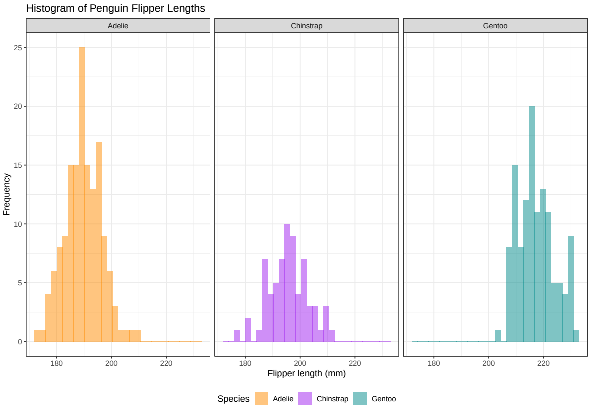
We may consider change the transparency of the plots

```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species), alpha = 0.5) +
   scale_fill_manual(values = c("darkorange","purple","cyan4")) +
   labs(fill = "Species") + theme_bw() + theme(legend.position = "bottom")  +
   xlab("Flipper length (mm)") + ylab("Frequency") +
   ggtitle("Histogram of Penguin Flipper Lengths")
```



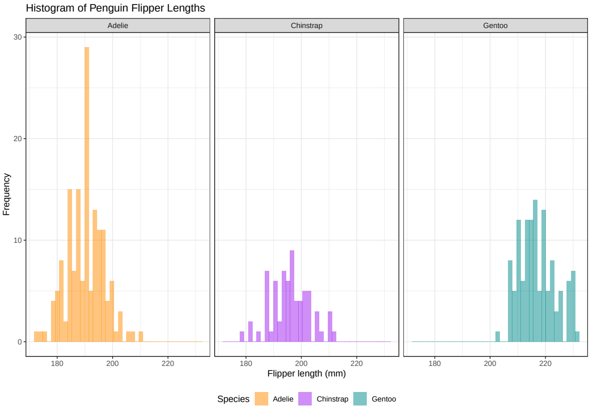
Consider facet plots

```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species), alpha = 0.5) +
   scale_fill_manual(values = c("darkorange","purple","cyan4")) +
   labs(fill = "Species") + theme_bw() + theme(legend.position = "bottom")  +
   xlab("Flipper length (mm)") + ylab("Frequency") +
   ggtitle("Histogram of Penguin Flipper Lengths") +
   facet_wrap(. ~ species)
```



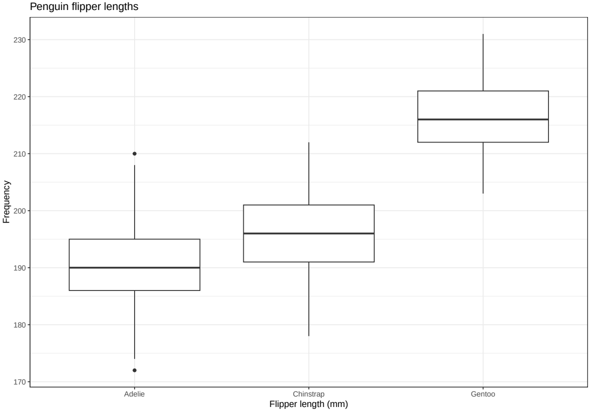
Change bins for histogram

```
penguins %>%
   ggplot() +
   geom_histogram(aes(x = flipper_length_mm, fill = species), alpha = 0.5, bins = 40) +
   scale_fill_manual(values = c("darkorange","purple","cyan4")) +
   labs(fill = "Species") + theme_bw() +
   theme(legend.position = "bottom")  + xlab("Flipper length (mm)") +
   ylab("Frequency") + ggtitle("Histogram of Penguin Flipper Lengths") +
   facet_wrap(. ~ species)
```



Boxplots

```
ggplot(data = penguins, aes(x = species, y = flipper_length_mm)) +
  geom_boxplot(show.legend = FALSE) +
  xlab("Flipper length (mm)") +
  ylab("Frequency") +
  labs(title = "Penguin flipper lengths") +
  theme_bw()
```



Barcharts

First, try to summarize the penguin data by species and returns the proportion of each penguin types.

Barcharts

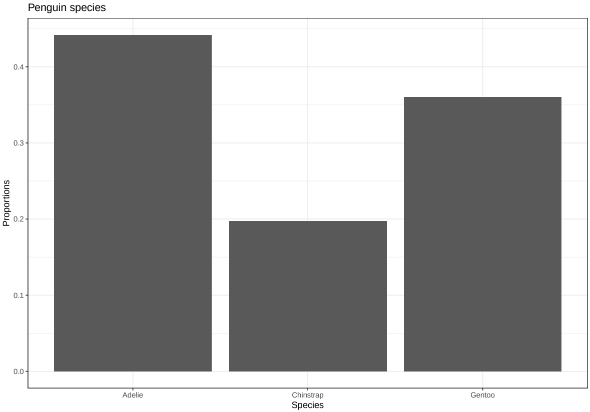
First, try to summarize the penguin data by species and returns the proportion of each penguin types.

```
gplot <- penguins %>%
  group_by(species) %>%
  tally() %>%
  mutate(prop = n / sum(n))
gplot
```

```
## # A tibble: 3 x 3
##   species       n  prop
##   <fct>     <int> <dbl>
## 1 Adelie      152 0.442
## 2 Chinstrap    68 0.198
## 3 Gentoo      124 0.360
```

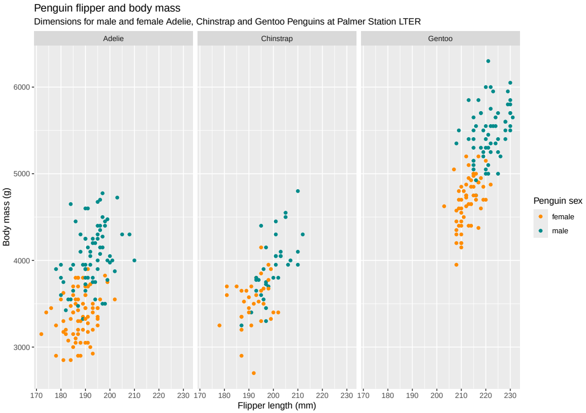
Barcharts

```
ggplot(data = gplot, aes(x = species, y = prop))  +
  geom_bar(stat = "identity") +
  xlab("Species") +
  ylab("Proportions") +
  labs(title = "Penguin species") +
  theme_bw()
```



Faceting

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = sex)) +
  scale_color_manual(values = c("darkorange","cyan4"), na.translate = FALSE) +
  labs(title = "Penguin flipper and body mass",
       subtitle = "Dimensions for male and female Adelie, Chinstrap and Gentoo Penguins at Palmer Station LTER"
       x = "Flipper length (mm)",
       y = "Body mass (g)",
       color = "Penguin sex") +
  facet_wrap(~species)
```



Git + Github

What is Git?

- A control system to manage projects
- Good for tracking history

Git + Github

What is Git?

- A control system to manage projects
- Good for tracking history

What about Github?

- Cloud-based service for managing Git repositories
- Useful for teamwork
- Just like “Dropbox”

Why Git + Github?

- You can undo anything
- You won't need to keep undo-ing things (merge/load difference)
- You can identify exactly when and where changes were made
- Teamwork

Some github terminology

- User: A Github account for you (e.g., jules32).
- Organization: The Github account for one or more user (e.g., datacarpentry).
- Repository: A folder within the organization that includes files dedicated to a project.
- Local Github: Copies of Github files located your computer.
- Remote Github: Github files located on the https://github.com website.

Basic Git commands and workflow

When you are working on a your local machine you typically get started by:

- `git clone`: Cloning a remote repository to work on locally. This is a way to work with an ongoing project or edit someone else's project that is available remotely (aka on GitHub).

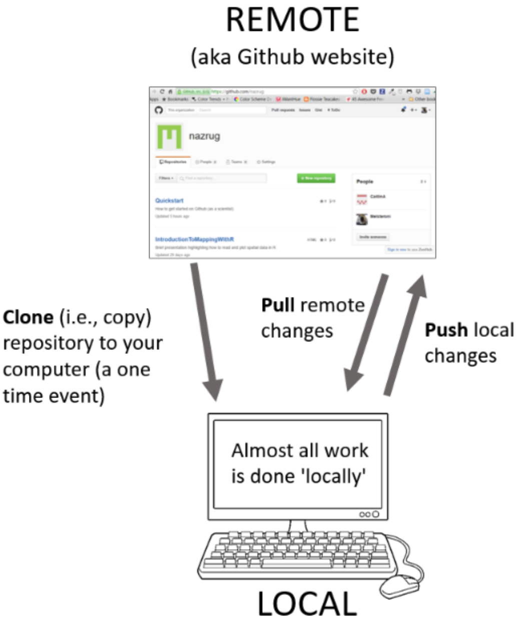
From there, the typical workflow involves:

- `git add`: Adding files to your repo
- `git commit`: Commiting changes you have made
- `git push`: Pushing changes to a remote repository (aka GitHub)

For a collaborative project, or work between desktop and personal laptop, you would use the following first before `git add`

- `git pull`: Pulling changes from a remote repository

Illustration diagram



REMOTE

(aka Github website)

Clone (i.e., copy) repository to your computer (a one time event)

Pull remote changes

Push local changes

Almost all work is done 'locally'

LOCAL

Let's Git

Download Github and set up your github profile.

- Github Desktop, a GUI for using Github [link]
- (Optional) Learn how to use command line for Github management
  - Command line tutorial [link]

Resources

This tutorial is based on

- Monica Alexander's ggplot tutorial [link]
- Jesse Gronsbell's Github tutorial [link]

Other resources:
https://kbroman.org/github_tutorial/pages/resources.html