Lab 2. Fundamental Chart Types

PUBH 6199: Visualizing Data with R, Summer 2025

Xindi (Cindy) Hu, ScD

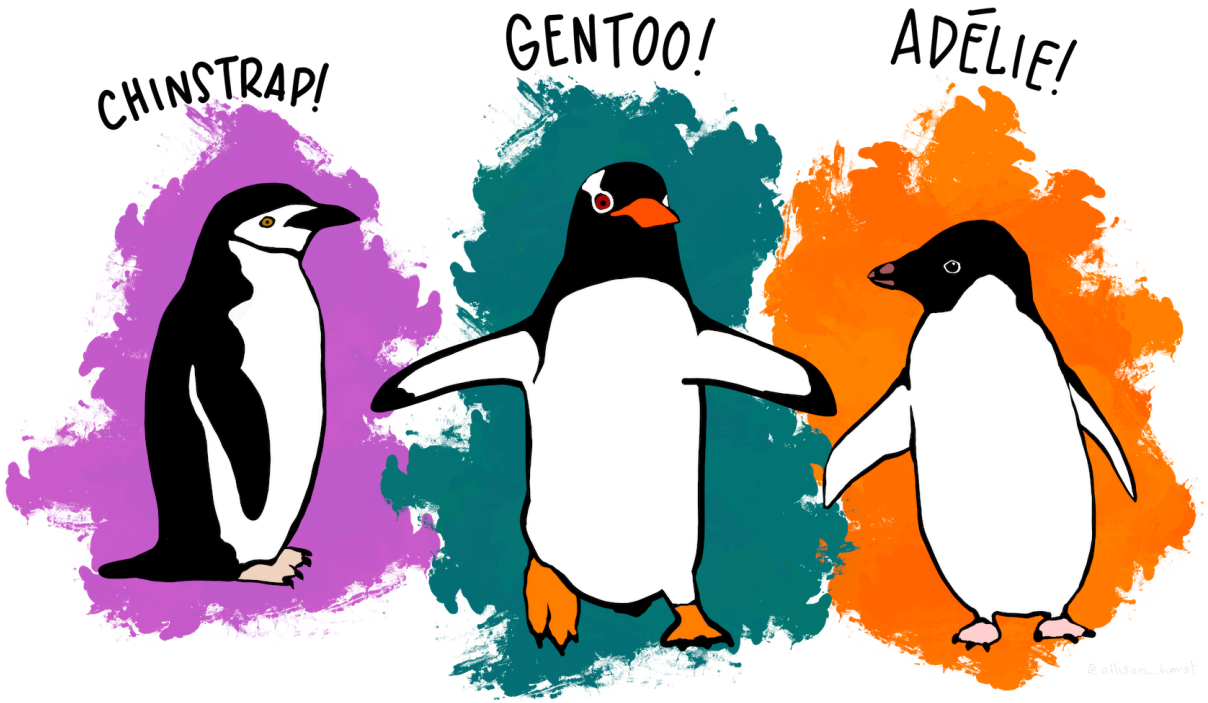
2025-05-29

Outline for today

- **Introducing the datasets**
- Review chart types to visualize basic quantitative information
- Review chart types to visualize trends over time
- Review chart types to visualize distribution
- Review chart types to visualize relationships

Introducing the `penguins` dataset from `{palmerpenguins}`

Contains body measurements for 344 penguins on three islands in the Palmer Archipelago.



Artwork by @allison_horst

Introducing the `penguins` dataset from `{palmerpenguins}`

```
Rows: 342
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, 36.7, 39.3, 38.9, 39.2, 34.1, 42.0…
$ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, 19.3, 20.6, 17.8, 19.6, 18.1, 20.2…
$ flipper_length_mm <int> 181, 186, 195, 193, 190, 181, 195, 193, 190, 186, 18…
$ body_mass_g       <int> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3475, 4250…
$ sex               <fct> male, female, female, female, male, female, male, NA…
$ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007…
```

Two penguins dropped out because missing data on important variables `species`, `flipper_length_mm`, and `body_mass_g`.

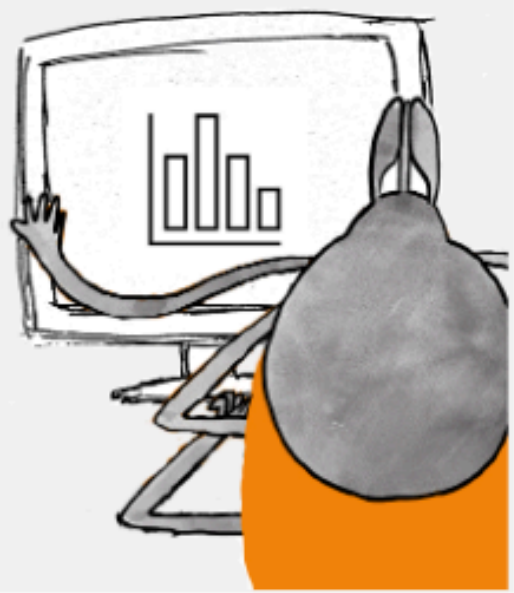
Introducing the lyme disease surveillance data

> Lyme disease has been a nationally notifiable condition in the U.S. sinde 1991. Local and state health departments collect reports of Lyme disease and shared the anonymized data with the Centers for Disease Control and Prevention (CDC). The CDC developed public use data sets to facilitate public health and reserach acccess to the data.

Lyme Disease Data Dashboard

In 2022, a new case definition for Lyme disease was implemented resulting in approximately 63,000 cases reported to CDC by state health departments and the District of Columbia. Use this dashboard to explore the Lyme disease case data that has been reported to CDC over the years.

Data Explained | Tables and Charts | Maps



Download state and local data on Lyme disease **case counts** and **incidence (cases/100k people)** over time here.

Introducing the lyme disease surveillance data

```
Rows: 714
Columns: 4
$ state          <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", …
$ year           <dbl> 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2…
$ cases          <dbl> 2, 24, 25, 24, 64, 25, 38, 41, 36, 66, 14, 51, 32, 36, …
$ rates_per_100k <dbl> 0.0, 0.5, 0.5, 0.5, 1.3, 0.5, 0.8, 0.8, 0.7, 1.3, 0.3, …
```

The dataset contains the number of Lyme disease cases and incidence rates (cases per 100,000 people) for 49 states (missing Hawaii) and DC in the U.S. from 2010 to 2023.

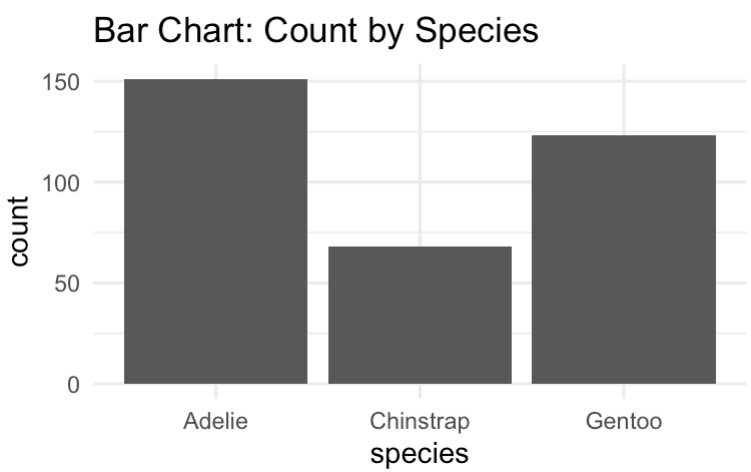
Outline for today

- Introducing the datasets
- **Review chart types to visualize basic quantitative information**
- Review chart types to visualize trends over time
- Review chart types to visualize distribution
- Review chart types to visualize relationships

Let’s start with a simple example, how many penguins are there in each species?

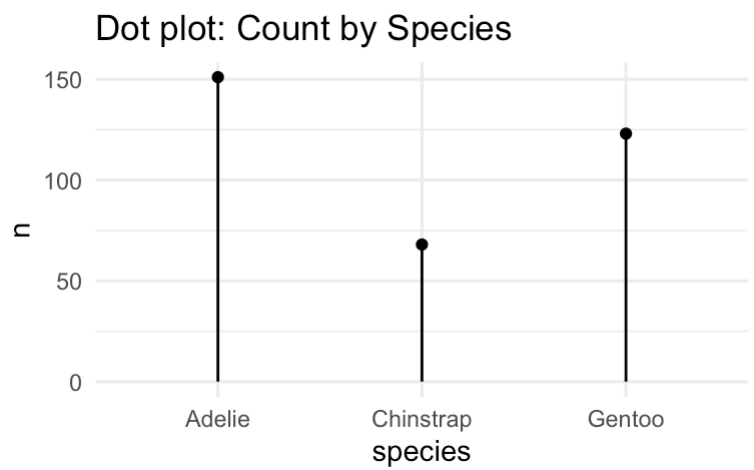
Bar chart is a good choice to visualize counts.

```
ggplot(penguins_clean, aes(x = species)) +
  geom_bar() +
  labs(title = "Bar Chart: Count by Species") +
  theme_minimal()
```



Dot plot improves upon it due to higher data-ink ratio.

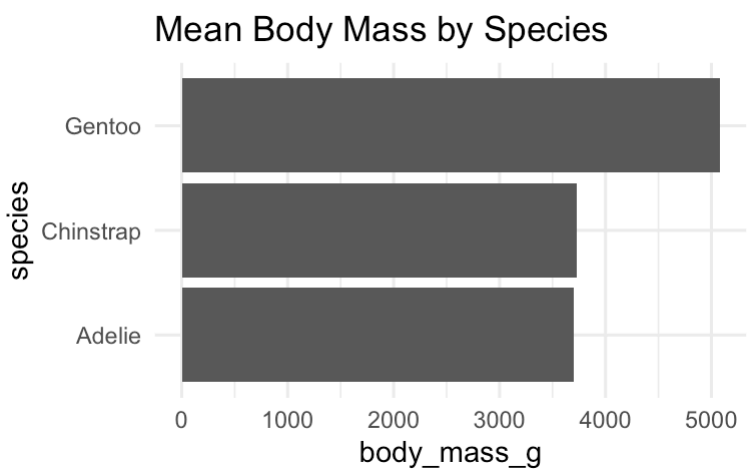
```
penguins_clean %>%
  count(species) %>%
  ggplot(aes(x = species, y = n)) +
  geom_point() +
  geom_linerange(aes(ymin = 0, ymax = n)) +
  labs(title = "Dot plot: Count by Species") +
  theme_minimal()
```



How to visualize the average body mass of each species?

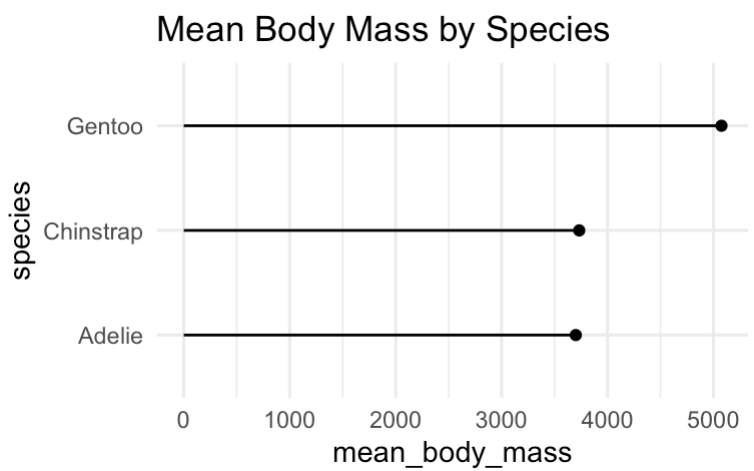
The default of `geom_bar()` is column plot, use `coord_flip()` to turn it horizontal.

```
ggplot(penguins_clean, aes(x = species, y = body_mass_g)) +
  stat_summary(fun = mean, geom = "bar") +
  labs(title = "Mean Body Mass by Species") +
  coord_flip() +
  theme_minimal()
```



Similarly, `coord_flip()` can be applied to dot plot

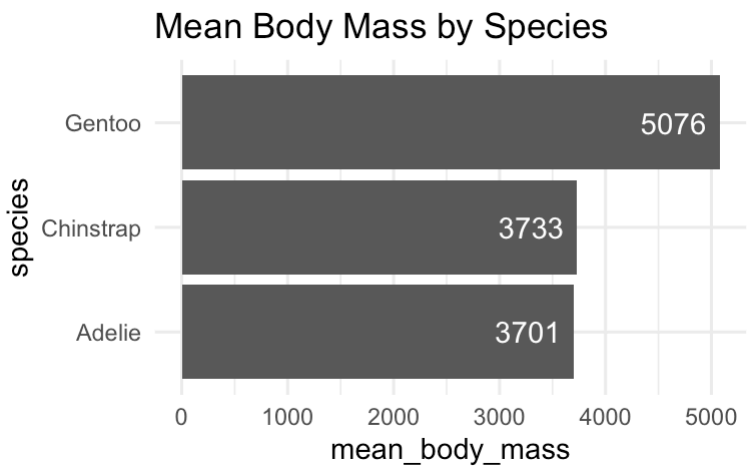
```
penguins_clean %>%
  group_by(species) %>%
  summarise(mean_body_mass = mean(body_mass_g, na.rm = TRUE)) %>%
  ggplot(aes(x = species, y = mean_body_mass)) +
  geom_point() +
  geom_linerange(aes(ymin = 0, ymax = mean_body_mass)) +
  labs(title = "Mean Body Mass by Species") +
  coord_flip() +
  theme_minimal()
```



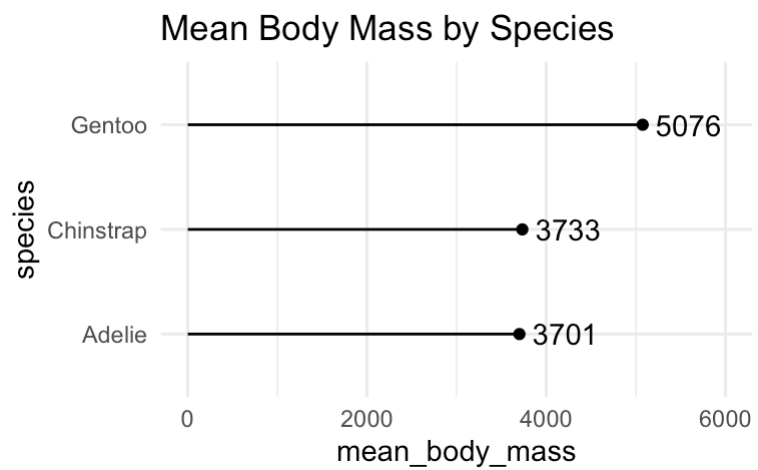
Add labels if the exact values are important

Use `geom_text()` to add labels to the bar chart and dot plot.

```
penguins_clean %>%
  group_by(species) %>%
  summarise(mean_body_mass = mean(body_mass_g, na.rm = TRUE)) %>%
  ggplot(aes(x = species, y = mean_body_mass)) +
  geom_col() +
  geom_text(aes(label = round(mean_body_mass)), hjust = 1.2, colo
  labs(title = "Mean Body Mass by Species") +
  coord_flip() +
  theme_minimal()
```



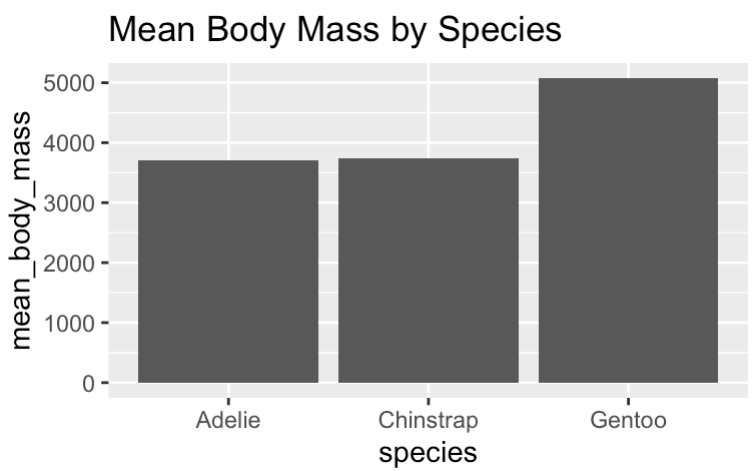

```
penguins_clean %>%
  group_by(species) %>%
  summarise(mean_body_mass = mean(body_mass_g, na.rm = TRUE)) %>%
  ggplot(aes(x = species, y = mean_body_mass)) +
  geom_point() +
  geom_linerange(aes(ymin = 0, ymax = mean_body_mass)) +
  geom_text(aes(label = round(mean_body_mass)), hjust = -0.2, col
  scale_y_continuous(limits = c(0, 6000)) + # set y-axis limits t
  labs(title = "Mean Body Mass by Species") +
  coord_flip() +
  theme_minimal()
```



When to use `geom_col()` vs `geom_bar()`?

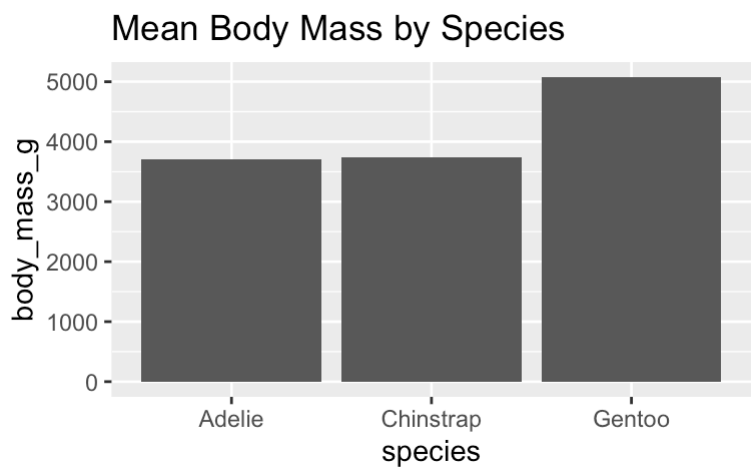
Use `geom_col()` when data is already summarized, e.g., mean body mass by species.

```
penguins_clean %>%
  group_by(species) %>%
  summarise(mean_body_mass = mean(body_mass_g, na.rm = TRU
  ggplot(aes(x = species, y = mean_body_mass)) +
  geom_col() +
  labs(title = "Mean Body Mass by Species")
```



Use `geom_bar()` when data is not summarized, e.g., count of species.

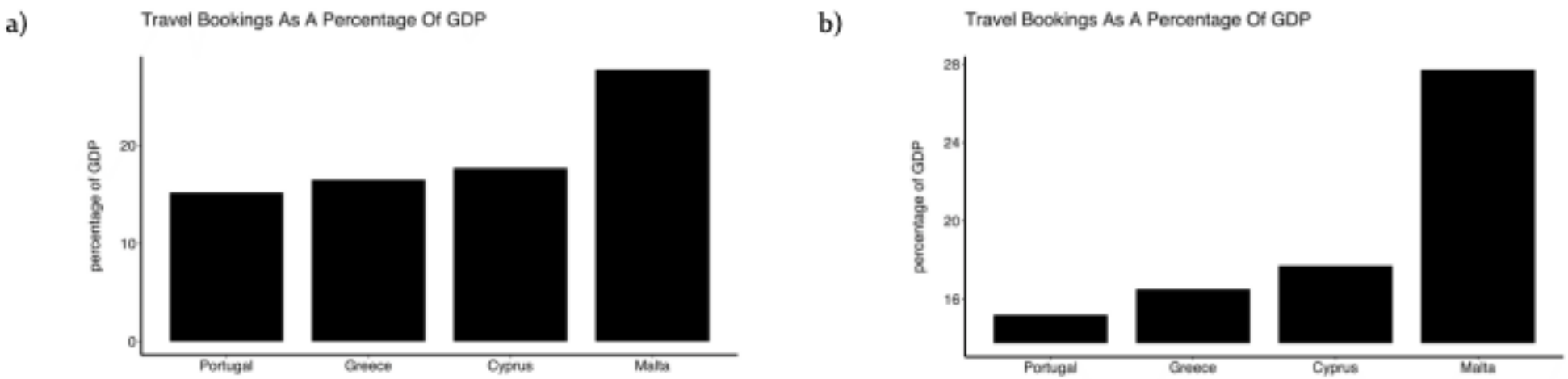
```
ggplot(penguins_clean, aes(x = species, y = body_mass_g))
  geom_bar(stat = "summary", fun = mean) +
  labs(title = "Mean Body Mass by Species")
```



IMPORTANT: Bar plot axis must start at zero

> Yang et al (2021) study provided empirical evidence that y-axis truncation leads to viewers to perceive illustrated differences as larger (i.e., a truncation effect).

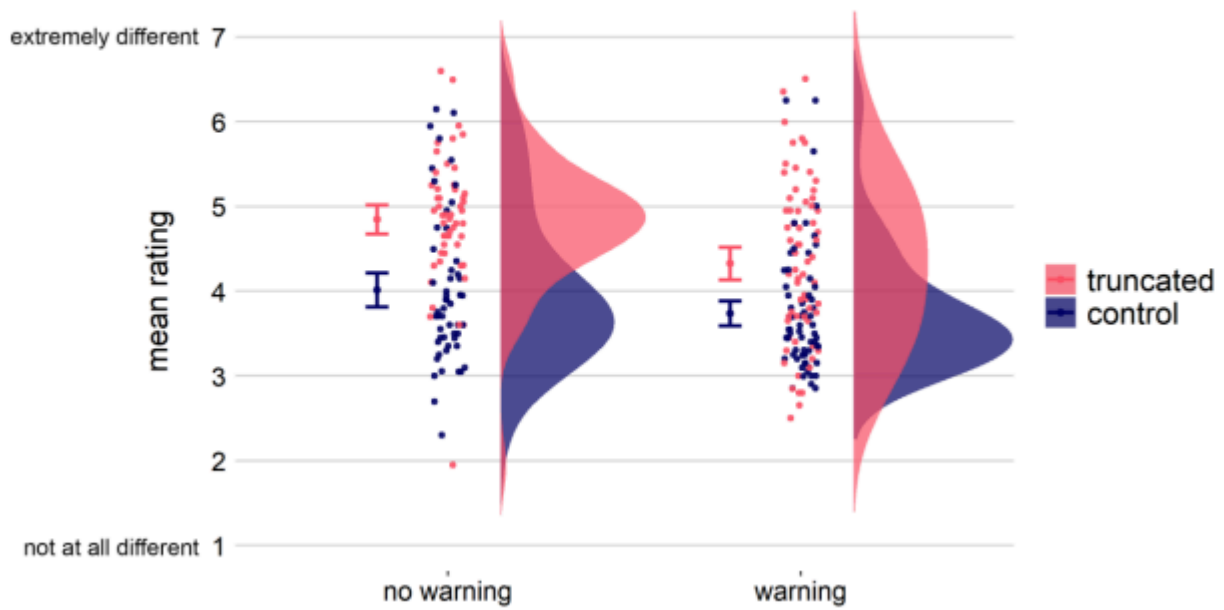
Figure 2.
(a) Sample Trial of a Control Graph (b) Sample Trial of a Truncated Graph



IMPORTANT: Bar plot axis must start at zero

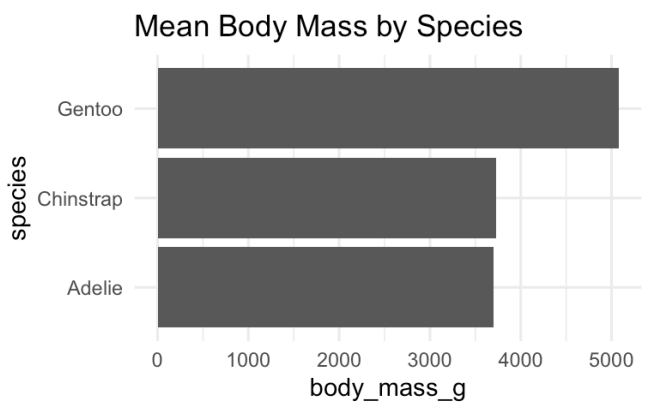
> Yang et al (2021) study provided empirical evidence that y-axis truncation leads to viewers to perceive illustrated differences as larger (i.e., a truncation effect).

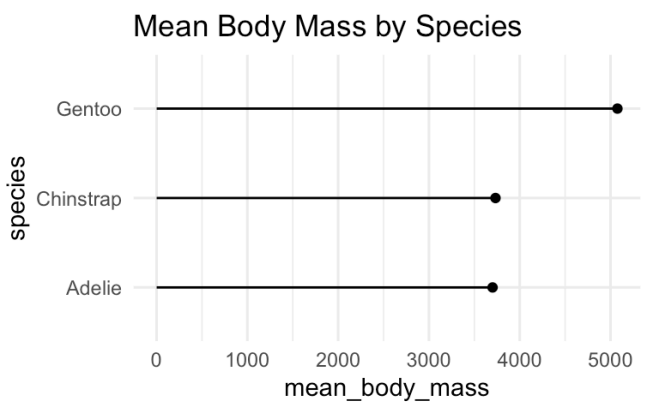
Figure 4.
Raincloud Plot for Study 3

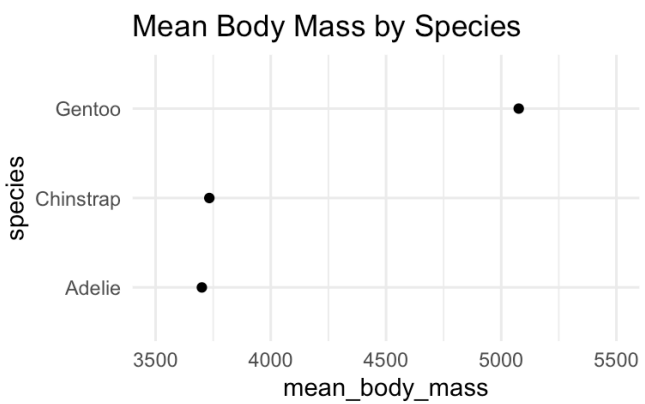


But you can cut the y-axis of dot plots!

A dot plot has has less ink and draw the eye to the end point rather than the middle of the bars. Cutting the y-axis allows easily differentiating differences in the values.







Bar plots/dot plots shine when comparing counts, but you should be careful when using them to summarize your data. Why?

Be careful when using bar plots/dot plots to summarize continuous data

Art by Allison Horst

- Bar plots hide the distribution of the data
- Bar plot makes readers infer that data are normally distributed with no outliers

Outline for today

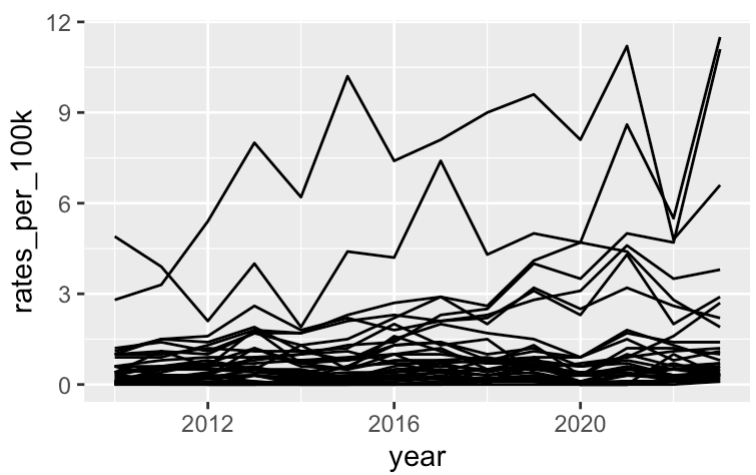
- Introducing the datasets
- Review chart types to visualize basic quantitative information
- **Review chart types to visualize trends over time**
- Review chart types to visualize distribution
- Review chart types to visualize relationships

Line plots show trends over time

> Going back to the lyme disease dataset, use line plot to show changes in lyme disease incidence rates (case per 100,000 people) by state, from 2010 to 2023.

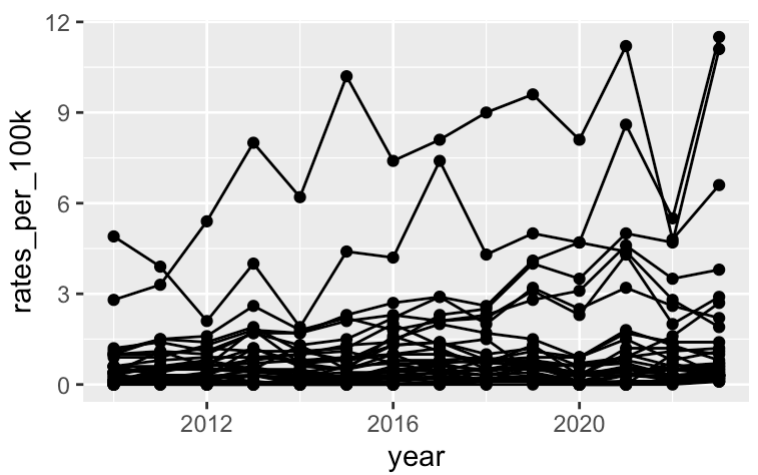
Basic line graph uses `geom_line()`

```
lyme %>%
  ggplot(aes(x = year, y = rates_per_100k, gro
  geom_line()
```



A line + point graph uses `geom_point()` to add points to the line graph.

```
lyme %>%
  ggplot(aes(x = year, y = rates_per_100k, gro
  geom_line()+
  geom_point()
```

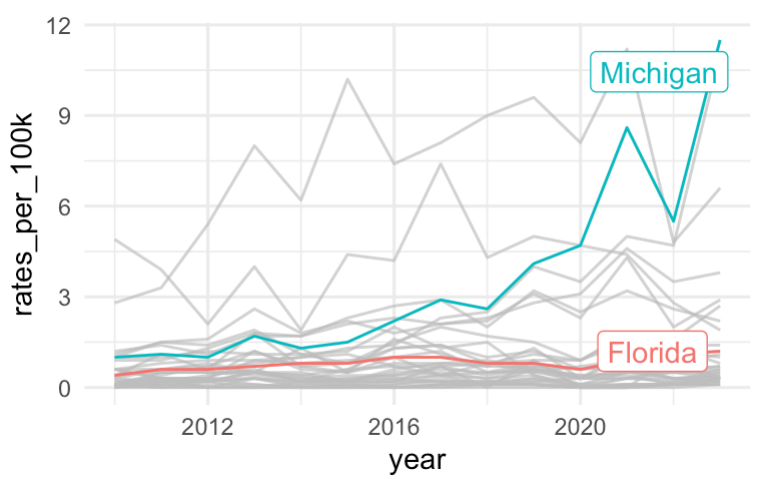


Improve upon “Spaghetti plots”

> A spaghetti plot is a line graph with many lines, which makes it hard to read. We can use `{gghighlight}` to draw attention to the lines of interest.

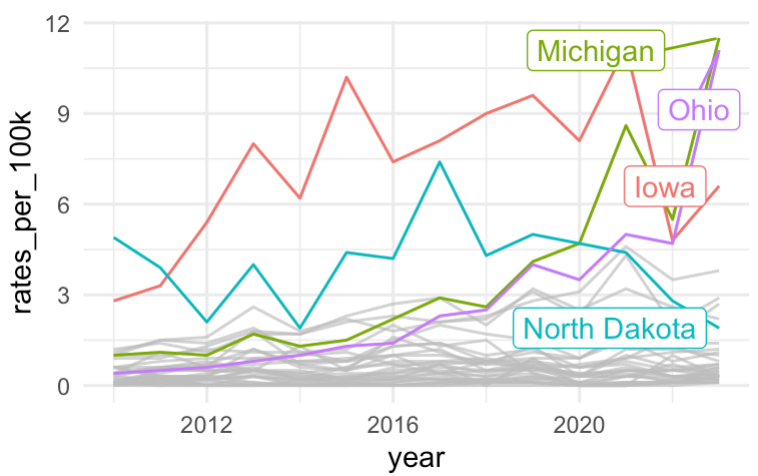
Highlight a specific state or a group of states

```
lyme %>%
  ggplot(aes(x = year, y = rates_per_100k, gro
  geom_line() +
  gghighlight::gghighlight(state %in% c("Michi
  theme_minimal()
```



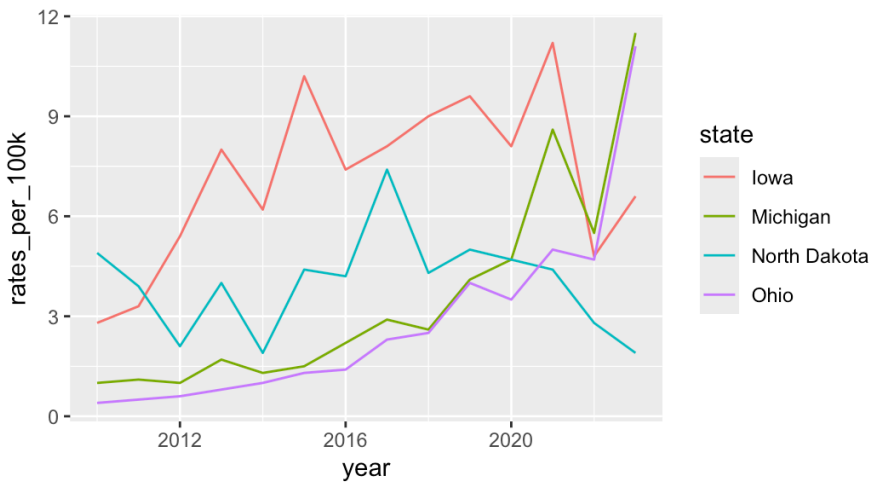
Highlight states based on the values of the incidence rate, for example, maximum rate exceeds 5 per 100k.

```
lyme %>%
  ggplot(aes(x = year, y = rates_per_100k, gro
  geom_line() +
  gghighlight::gghighlight(max(rates_per_100k)
  theme_minimal()
```

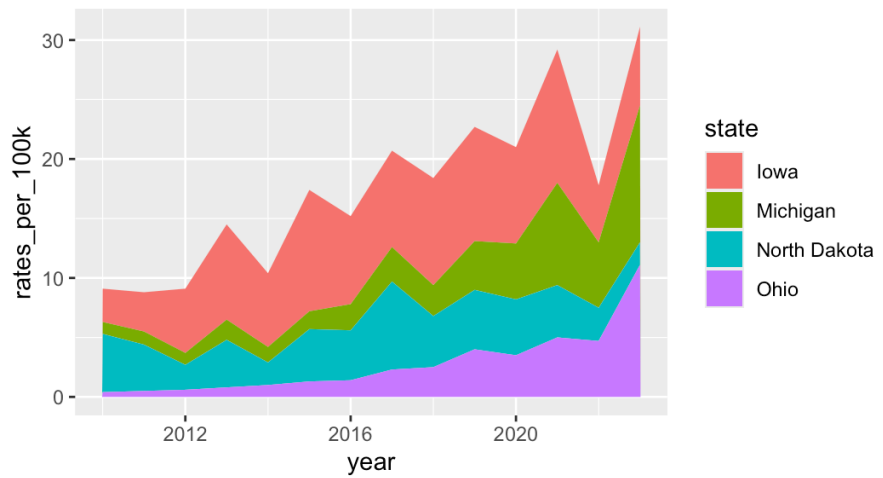


Area chart is similar to line graph, just filled in and stacked

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North D
  ggplot(aes(x = year, y = rates_per_100k, group = state,
  geom_line()
```




```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North D
  ggplot(aes(x = year, y = rates_per_100k, group = state,
  geom_area()
```



Stacked area charts are useful for showing the evolution of a whole and the relative proportions of each group that make up the whole. But it has a few drawbacks: low data-ink ratio, using area rather than position to encode data.

A variant of area charts: proportional stacked area charts

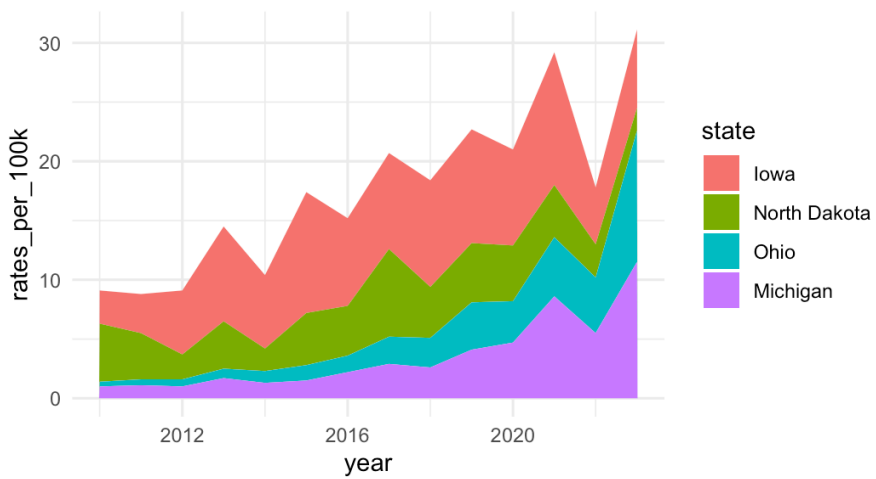
```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North Dakota")) |>
  ggplot(aes(x = year, y = rates_per_100k, group = state, fill = state)) +
  geom_area(position = "fill") + # this creates the proportional stacked area chart
  scale_y_continuous(labels = scales::percent_format(scale = 100))
```


Which group to put on the bottom?

It is important to consider which group you want to put it on the bottom of the area chart because it is the only group where your user can easily read the values off the chart.

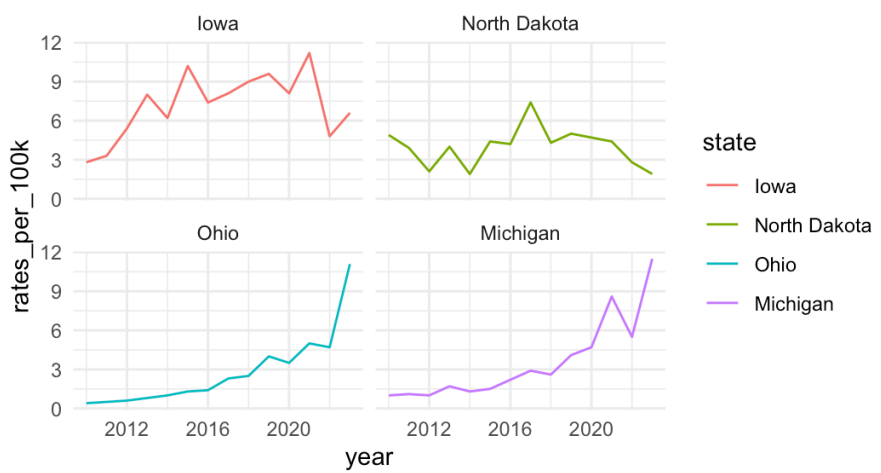
If you want to draw attention to “Michigan”, put it on the bottom.

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North D
  mutate(state = fct_relevel(state, "Michigan", after = In
  ggplot(aes(x = year, y = rates_per_100k, group = state,
  geom_area() +
  theme_minimal()
```



If all groups are equally important and you are not as interested in showing the whole, use a faceted line plot instead!

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North D
  mutate(state = fct_relevel(state, "Michigan", after = In
  ggplot(aes(x = year, y = rates_per_100k, group = state,
  geom_line() +
  facet_wrap(~ state) +
  theme_minimal()
```



🎯 Practice makes perfect!

~ Head over to lab2 notebook! ~

Outline for today

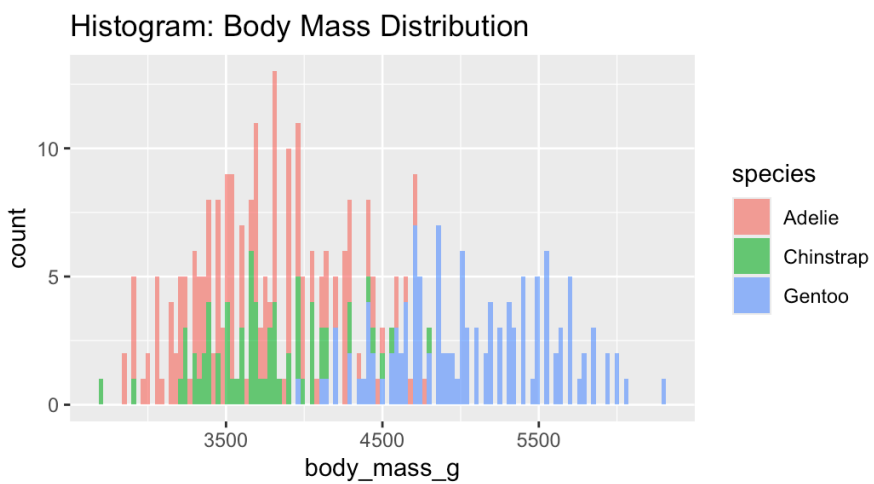
- Introducing the datasets
- Review chart types to visualize basic quantitative information
- Review chart types to visualize trends over time
- **Review chart types to visualize distribution**
- Review chart types to visualize relationships

Histogram

Histogram cuts a numeric variable into bins and counts the number of observations in each bin.

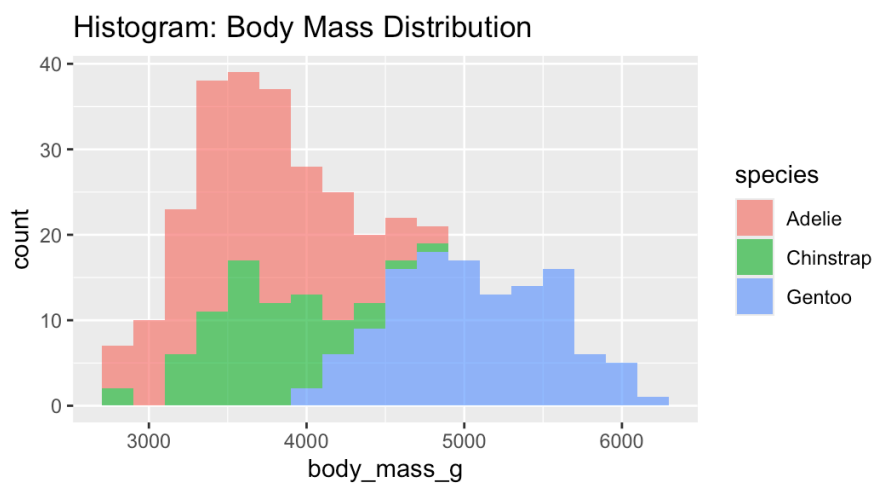
Too many bins!

```
1 ggplot(penguins_clean, aes(x = body_mass_g, fi
2   geom_histogram(binwidth = 30, alpha = 0.7) +
3   labs(title = "Histogram: Body Mass Distribut
```



A better `binwidth` parameter. The default is to divide data into 30 bins

```
1 ggplot(penguins_clean, aes(x = body_mass_g, fi
2   geom_histogram(binwidth = 200, alpha = 0.7)
3   labs(title = "Histogram: Body Mass Distribut
```

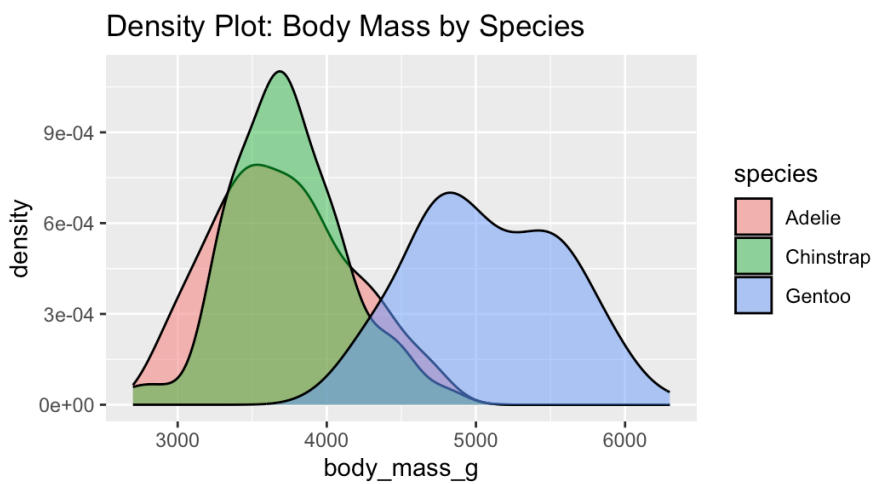


Density plot

Density plot uses the kernel density estimate to show the probability density function of a variable. Area under each density curve sums to 1.

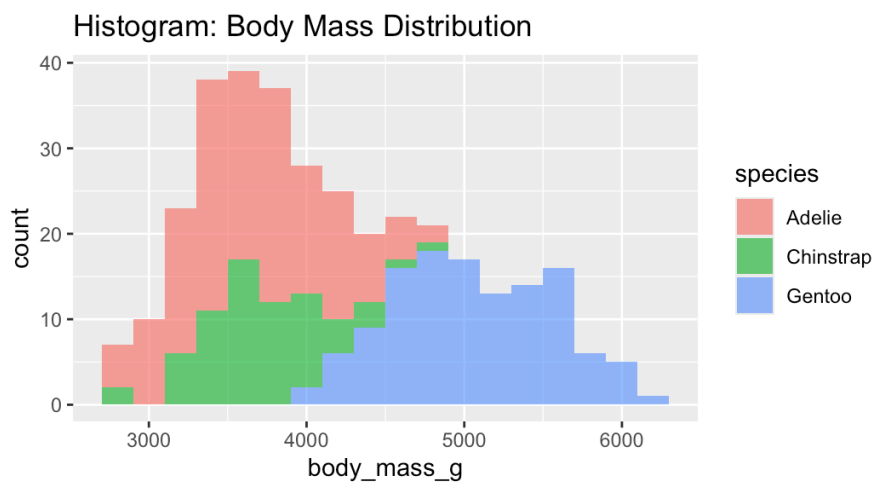
A smoothed version of a histogram

```
1 ggplot(penguins_clean, aes(x = body_mass_g, fi
2   geom_density(alpha = 0.5) +
3   labs(title = "Density Plot: Body Mass by Spe
```



Why is the histogram not the same?

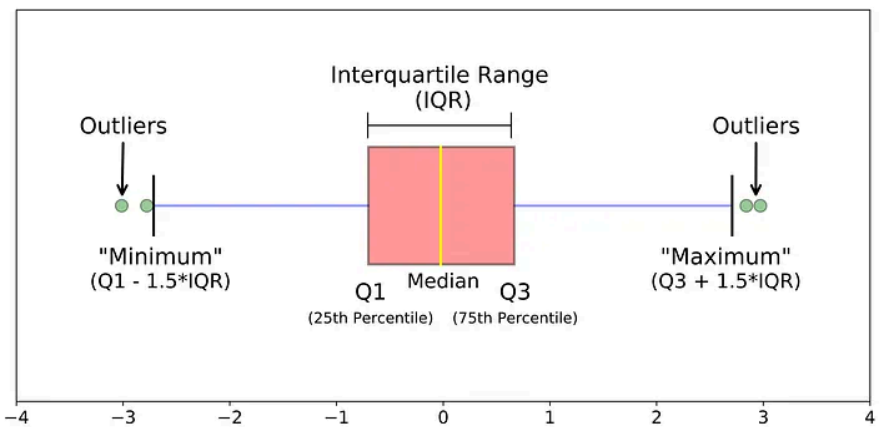
```
1 ggplot(penguins_clean, aes(x = body_mass_g, fi
2   geom_histogram(binwidth = 200, alpha = 0.7)
3   labs(title = "Histogram: Body Mass Distribut
```



Density plot does not indicate **sample size**.

Box plot

Box plot shows the median, interquartile range (IQR), and outliers of a variable. Boxplot is often used with comparing the same numeric variable over multiple groups.



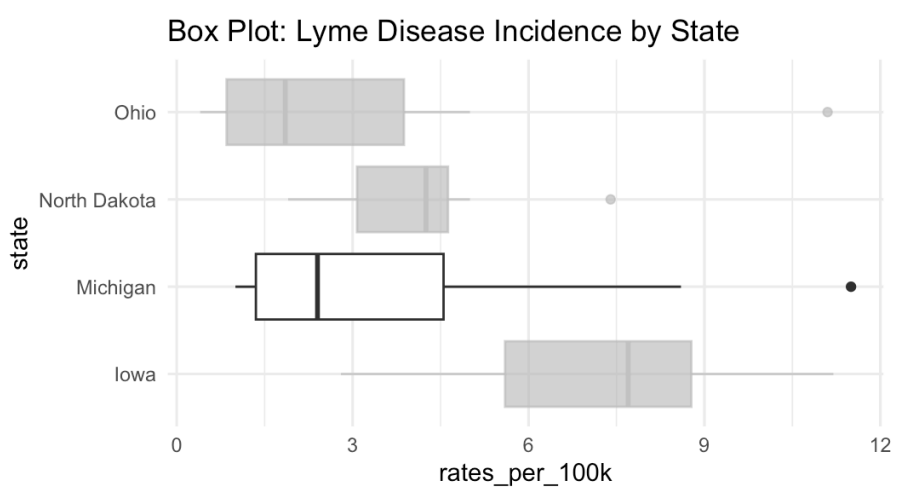

```
1 ggplot(penguins_clean, aes(x = species, y = bo
2   geom_boxplot() +
3   labs(title = "Box Plot: Body Mass by Species
```

Boxplot shows summary statistics, which may hide the distribution of the data.

Enhance box plot

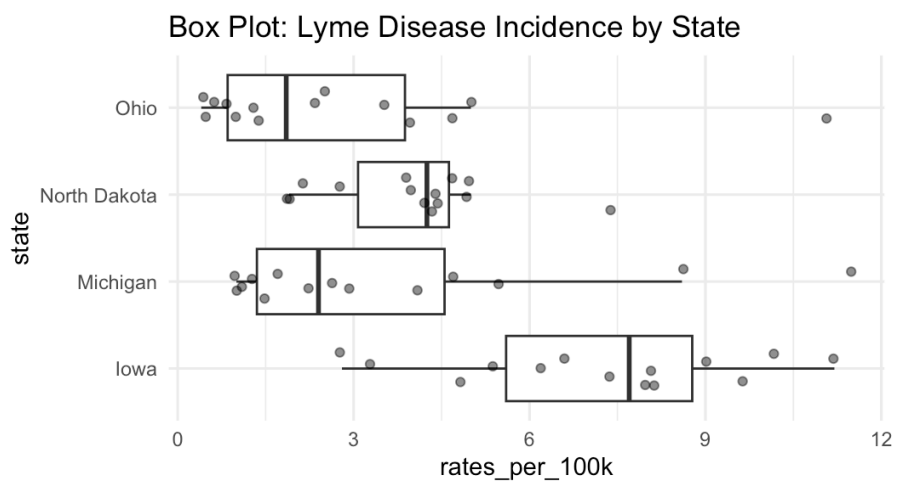
Highlight a group if you have many groups, also flip the coordinate if your categorical variable has long labels.

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iow
  ggplot(aes(x = state, y = rates_per_100k)) +
  geom_boxplot() +
  gghighlight::gghighlight(state == "Michigan"
  coord_flip() +
  labs(title = "Box Plot: Lyme Disease Inciden
  theme_minimal()
```



Jitter raw data, but remember to remove outliers

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iow
  ggplot(aes(x = state, y = rates_per_100k)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(alpha = 0.5, width = 0.2) +
  coord_flip() +
  labs(title = "Box Plot: Lyme Disease Inciden
  theme_minimal()
```



Violin plot

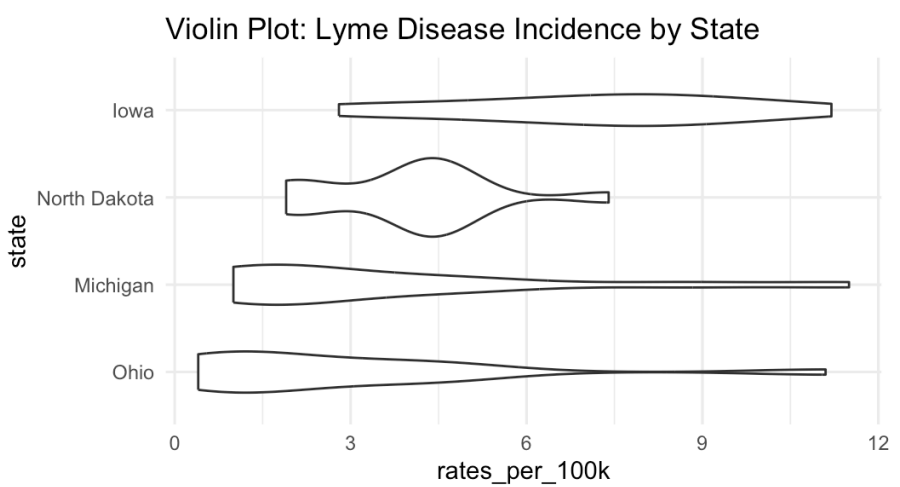
Violin plot shows the density estimate of the variable, similar to a density plot. It is often used with comparing the same numeric variable over multiple groups. It is usually a better alternative than a box plot.

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "North Dakota")) |>
  ggplot(aes(x = state, y = rates_per_100k)) +
  geom_violin() +
  coord_flip() +
  labs(title = "Violin Plot: Lyme Disease Incidence by State") +
  theme_minimal()
```

Enhance violin plot

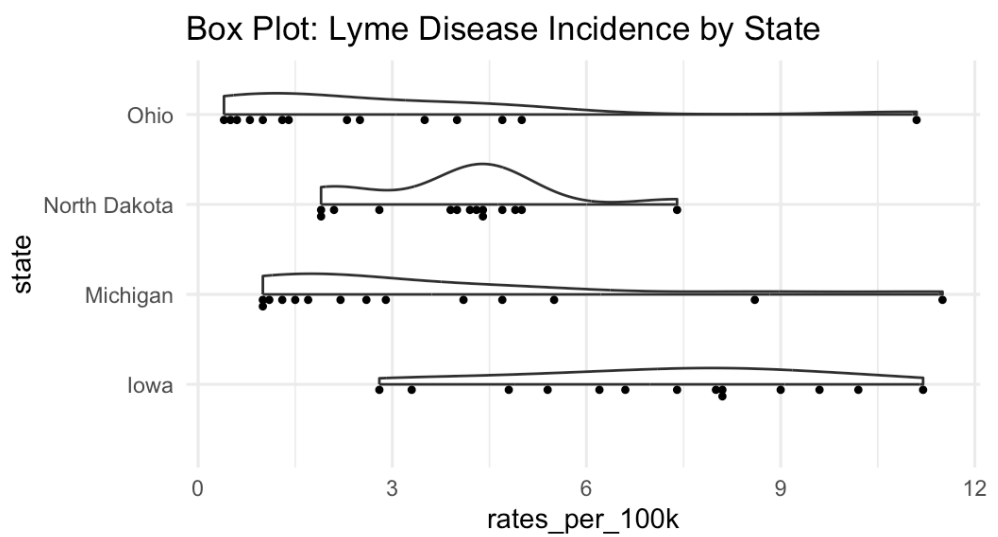
If you have many groups, consider ranking them by median values to make your readers' brain hurt less. Recall `law of continuity`.

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iow
  mutate(state = fct_reorder(state, rates_per_
  ggplot(aes(x = state, y = rates_per_100k)) +
  geom_violin() +
  coord_flip() +
  labs(title = "Violin Plot: Lyme Disease Inci
  theme_minimal()
```



The `{see}` package has `geom_violindot()` function that creates a half-violin half-dot plot, showing both distribution and sample size.

```
lyme |>
  filter(state %in% c("Michigan", "Ohio", "Iowa", "N
  ggplot(aes(x = state, y = rates_per_100k)) +
  see::geom_violindot(size_dots = 2) +
  coord_flip() +
  labs(title = "Box Plot: Lyme Disease Incidence by
  theme_minimal()
```



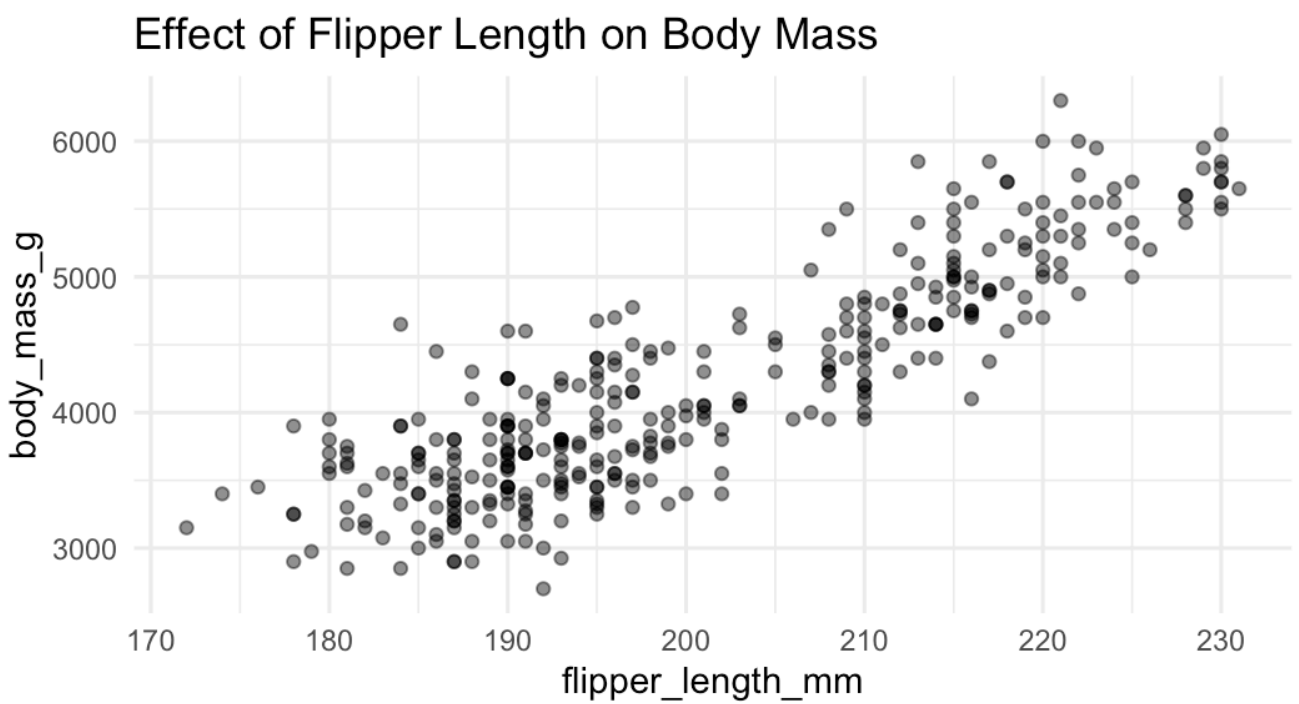
Outline for today

- Introducing the datasets
- Review chart types to visualize basic quantitative information
- Review chart types to visualize trends over time
- Review chart types to visualize distribution
- **Review chart types to visualize relationships**

Scatter plot

> Scatter plot is a good choice to visualize the relationship between two numeric variables. They help us answer questions around the **effect of X on Y**.

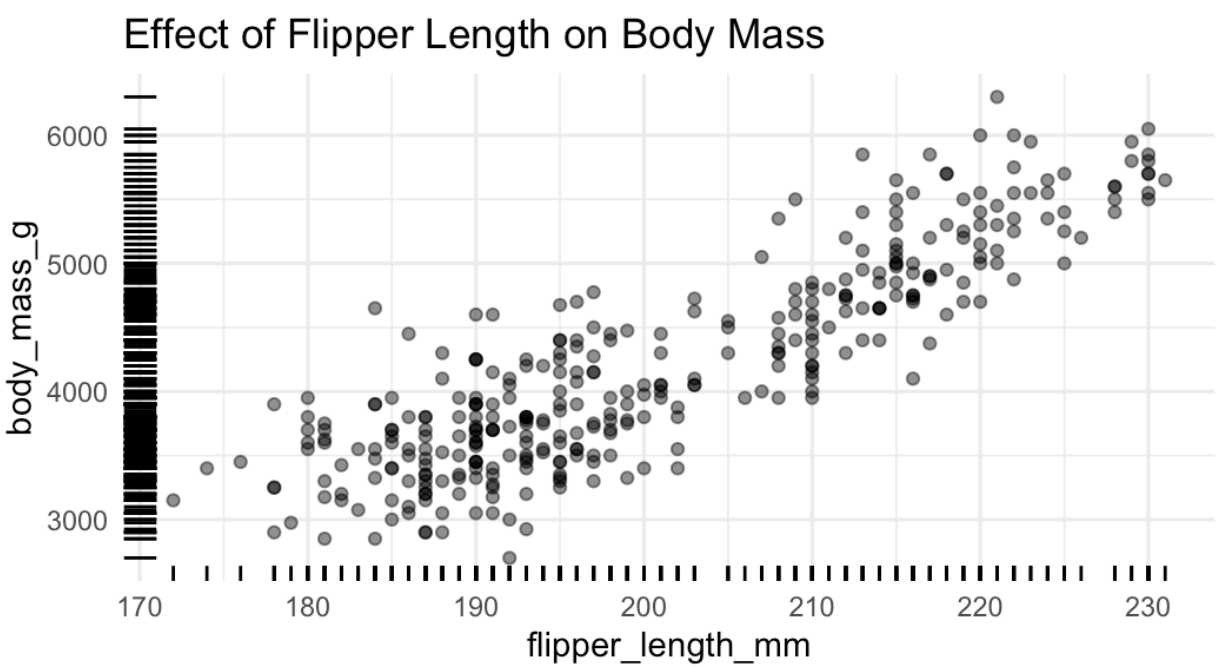
```
penguins_clean %>%
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(alpha = 0.5) +
  labs(title = "Effect of Flipper Length on Body Mass") +
  theme_minimal()
```



Add rug to visualize distribution

> Rug plot uses distribution marks to visualize the distribution of the two numeric variables. Each narrow line represents one data point. It shows the density of the data along the x and y axes.

```
penguins_clean %>%
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(alpha = 0.5) +
  geom_rug() +
  labs(title = "Effect of Flipper Length on Body Mass") +
  theme_minimal()
```

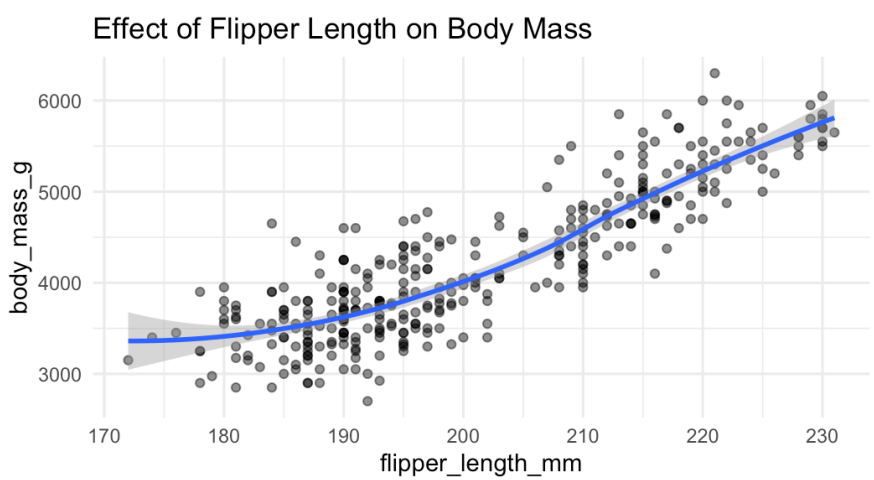


Add trend lines

> Trend lines are used to show the overall trend of the data. Default method for `geom_smooth()` is LOESS (locally estimated scatter plot smoothing), think of it as a moving average.

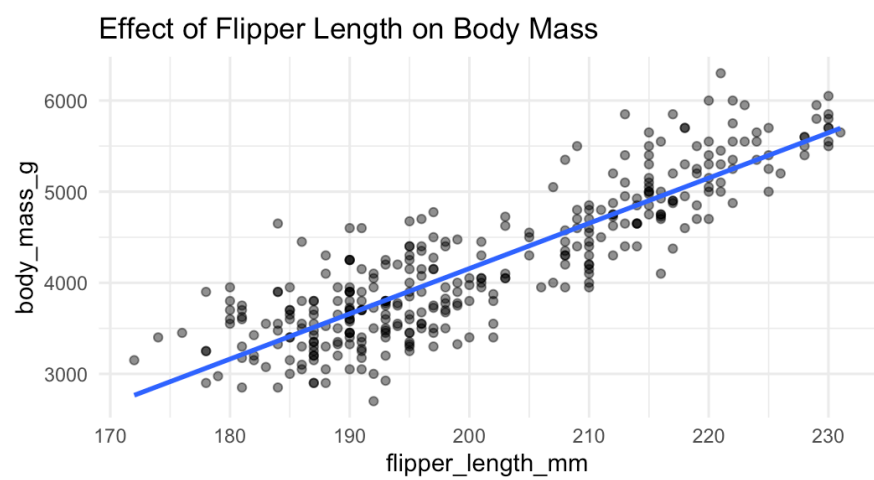
LOESS

```
1 penguins_clean %>%
2   ggplot(aes(x = flipper_length_mm, y = body_
3   geom_point(alpha = 0.5) +
4   geom_smooth() +
5   labs(title = "Effect of Flipper Length on B
6   theme_minimal()
```



Linear regression

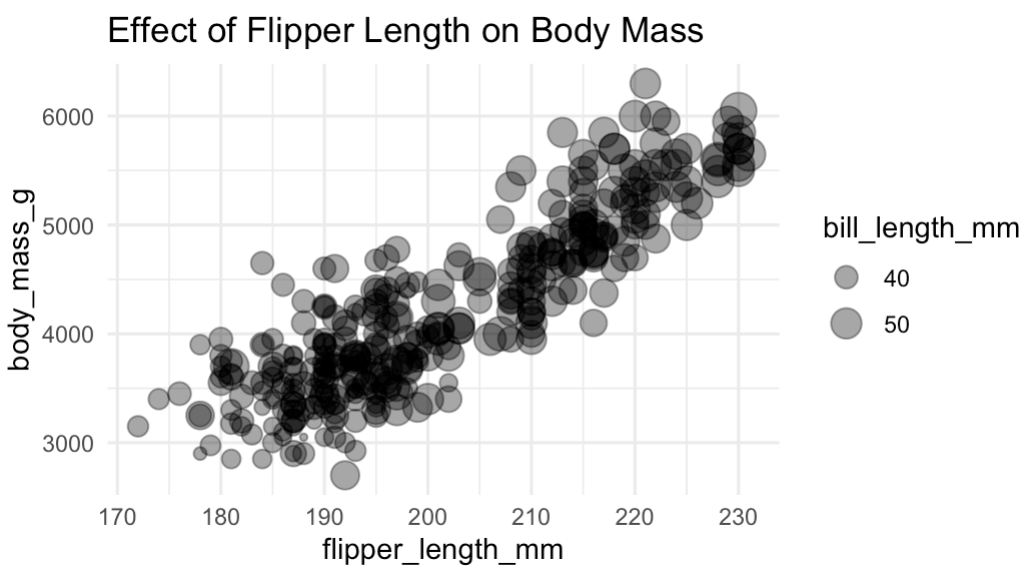
```
1 penguins_clean %>%
2   ggplot(aes(x = flipper_length_mm, y = body_
3   geom_point(alpha = 0.5) +
4   geom_smooth(method = "lm", se = FALSE) +
5   labs(title = "Effect of Flipper Length on B
6   theme_minimal()
```



Add a third numeric variable with bubble chart

> We can use a bubble chart to show the third numeric variable. The size of the point represents the third variable.

```
penguins_clean %>%
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(size = bill_length_mm), alpha = 0.4) +
  labs(title = "Effect of Flipper Length on Body Mass") +
  theme_minimal()
```



A few caveats:

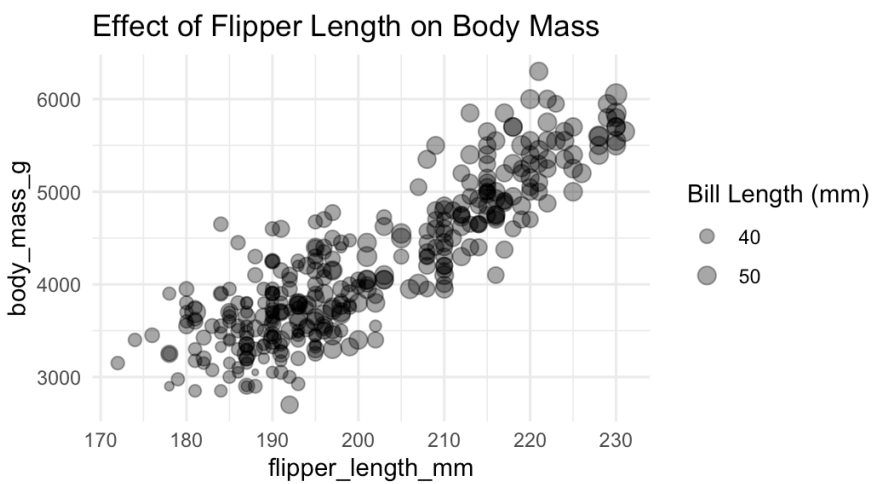
- The relationship between X and Y will be the primary focus
- It may be difficult to distinguish the size of the bubbles

Adjust the size of the bubbles

> Use `scale_size()` to adjust the size of the bubbles. Do not use `scale_radius()`.

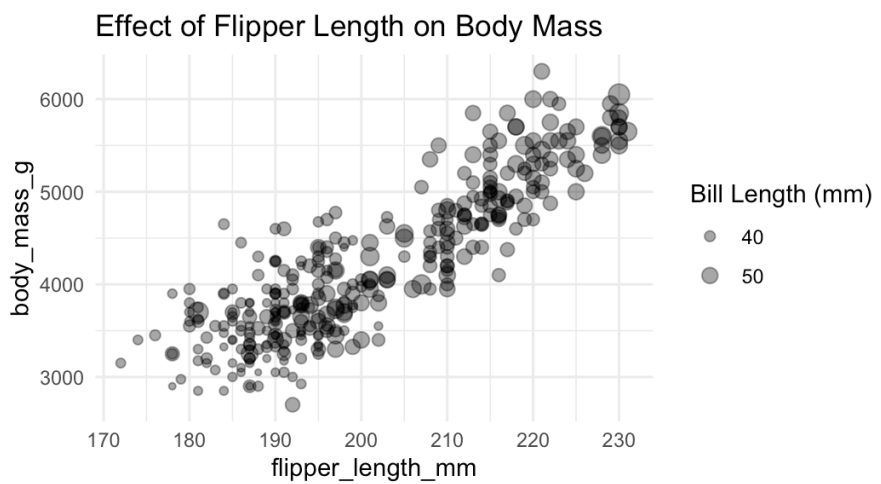
This is good.

```
1 penguins_clean %>%
2   ggplot(aes(x = flipper_length_mm, y = body_m
3   geom_point(aes(size = bill_length_mm), alpha
4   scale_size(range = c(1, 4), name = "Bill Len
5   labs(title = "Effect of Flipper Length on Bo
6   theme_minimal()
```



This is misleading.

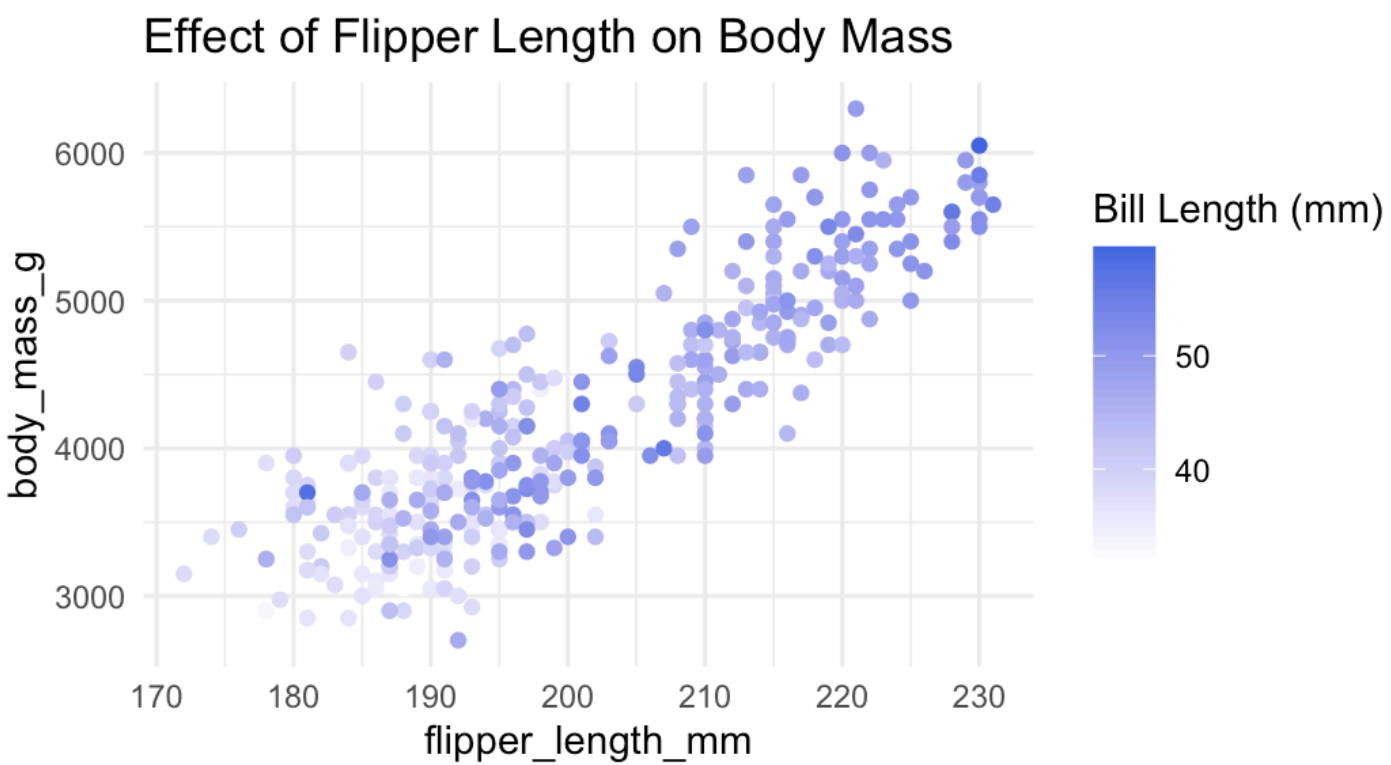
```
1 penguins_clean %>%
2   ggplot(aes(x = flipper_length_mm, y = body_m
3   geom_point(aes(size = bill_length_mm), alpha
4   scale_radius(range = c(1, 4), name = "Bill L
5   labs(title = "Effect of Flipper Length on Bo
6   theme_minimal()
```



Add a third numeric variable with color

> Recall that color hue does not natually have meaning for magnitude, consider using intensity

```
penguins_clean %>%
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = bill_length_mm)) +
  scale_color_gradient(low = "white", high = "royalblue", name = "Bill Length (mm)") +
  labs(title = "Effect of Flipper Length on Body Mass") +
  theme_minimal()
```



🚲 Your turn, get on the bike!

~ Head over to lab2 notebook! ~

End-of-Class Survey

Fill out the end-of-class survey

~ This is the end of Lab 2 ~

10:00