More Data Visualization Tools

Spring 2023

April 04 2023

So far ..

We know

- ***A basic set of geometries***
- ***How to map variables to aesthetics***
- ***How to layer geoms***
- ***How to change axis labels and titles***
- ***Common statistical graphs***

More to learn ...

Today

- *Changing scales*
- *Changing coordinates*
- *Mapping spatial data*

Changing scales and colors

Functions that control the mapping of your data to aesthetic attributes like color, fill, and shape.

```
scale_<aes>_<method>()
```

Some common scale functions:

- `scale_fill_manual()` : Manually define the fill colors for different categories
- `scale_fill_brewer()` : Use color palettes from the `ColorBrewer` library
- `scale_color_viridis()` : Use the viridis color scale for continuous data.
- `scale_shape_manual()` : Manually define the shapes for different categories.
- `scale_x_log10()`: Transform the x-axis to a logarithmic scale.
- `scale_y_reverse()`: Reverse the direction of the y-axis.
- `scale_color_gradient()`: Define a custom color gradient for continuous data.
- `scale_fill_discrete()`: Use a predefined color palette for discrete data.

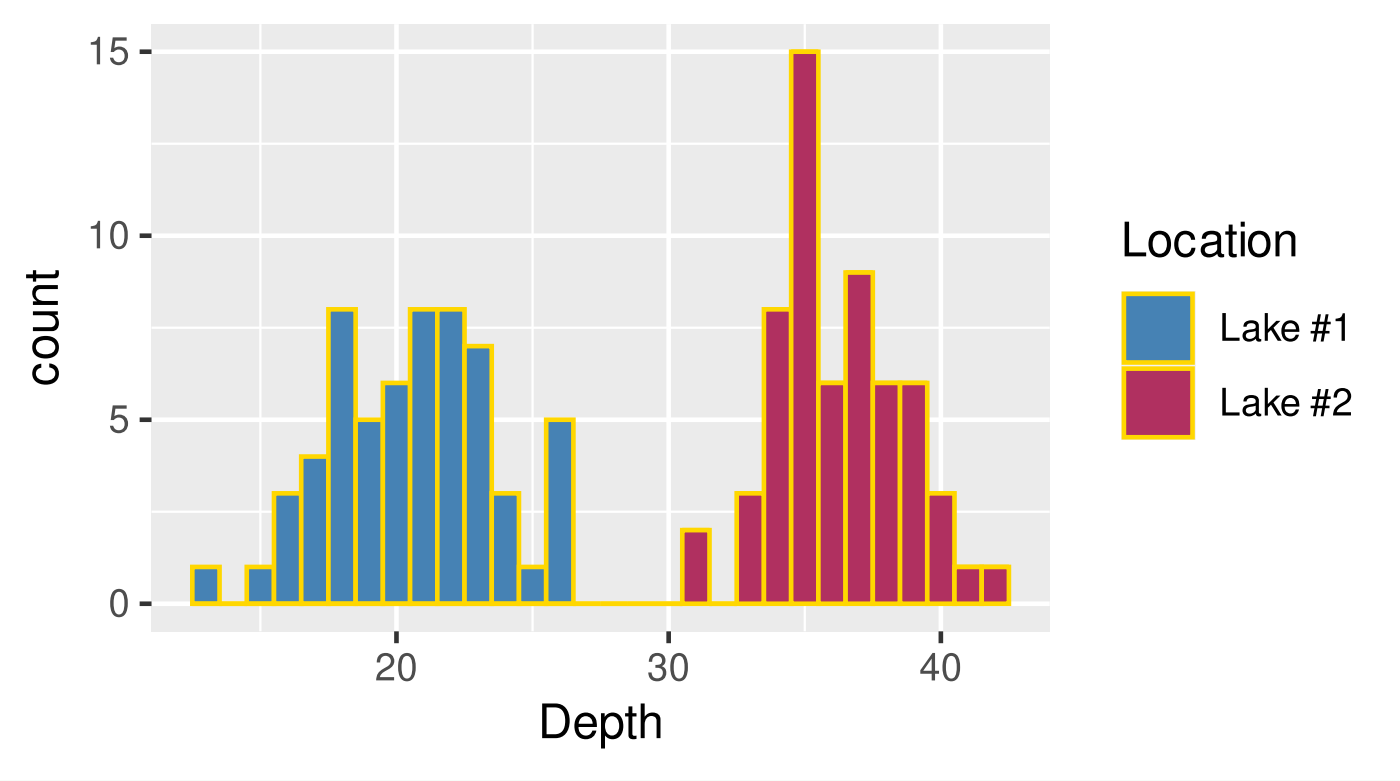
Recommended reading:

- Using colors in R
- Taking control of qualitative colors in ggplot2
- Data journalism

Example

Let's make Lake #1 steelblue and Lake #2 maroon

Plot Code



Example

Let's make Lake #1 steelblue and Lake #2 maroon

Plot Code

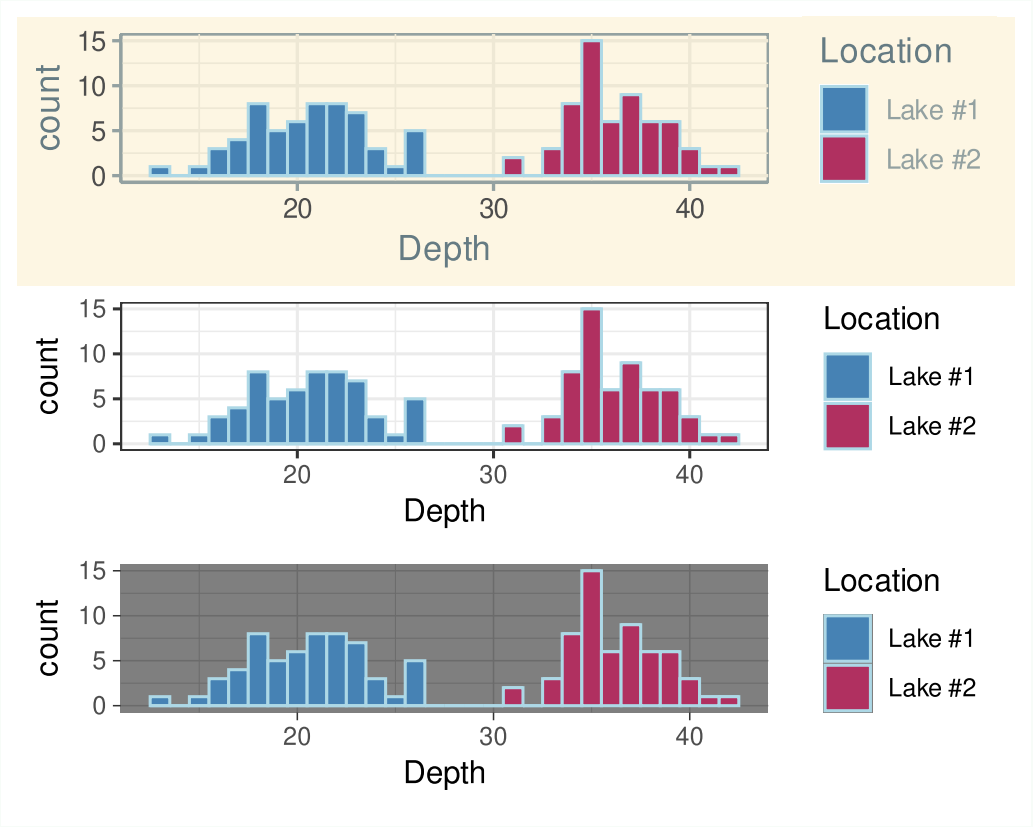
```
ggplot(data) +
  geom_histogram(
    aes(x = Depth, fill = Location),
    binwidth = 1,
    color = "gold")+
    scale_fill_manual(values = c("steelblue", "maroon"))
```

Changing themes

Theme: The non-data ink on your plots

Examples:

- background color
- tick marks
- grid lines
- legend position
- legend appearance

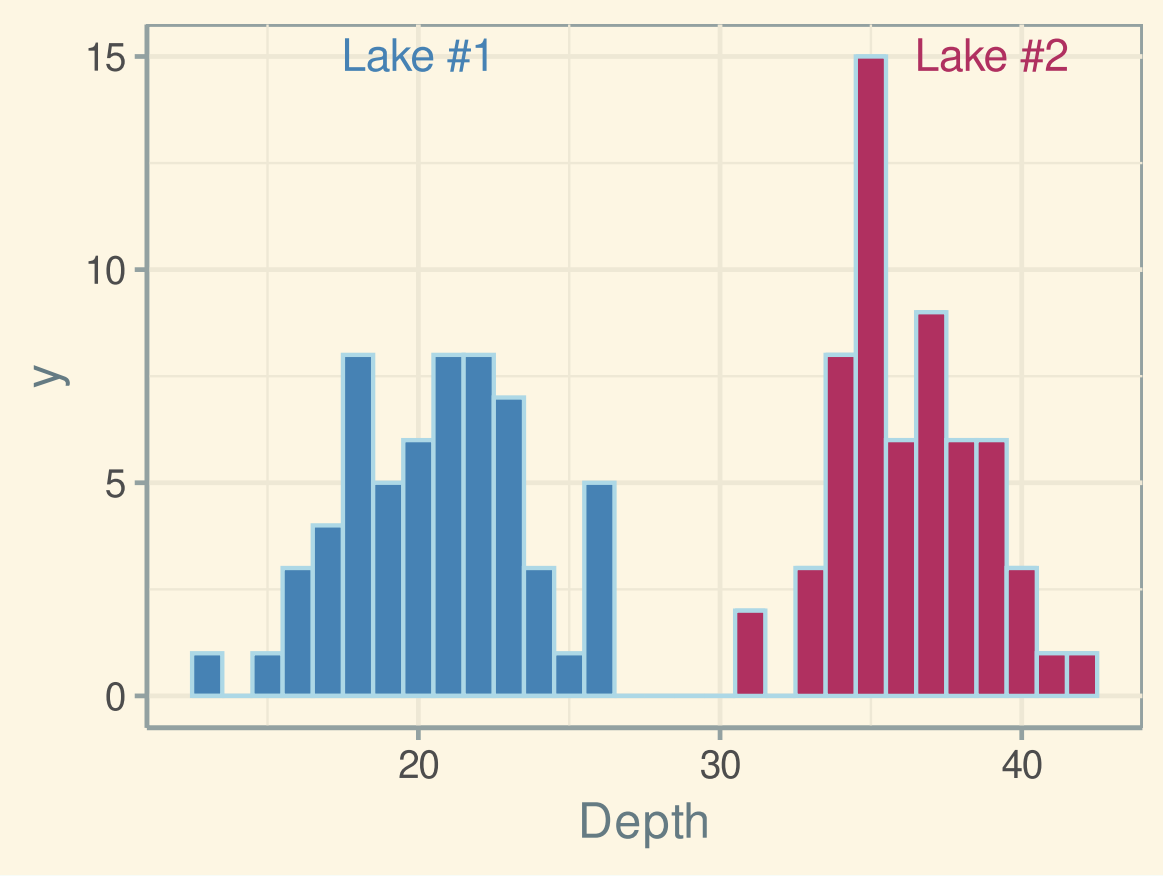


Top to bottom: solarized, bw, and dark themes resp.

Click for various basic ggplot2 themes and for ggthemes

Annotations

Plot Code



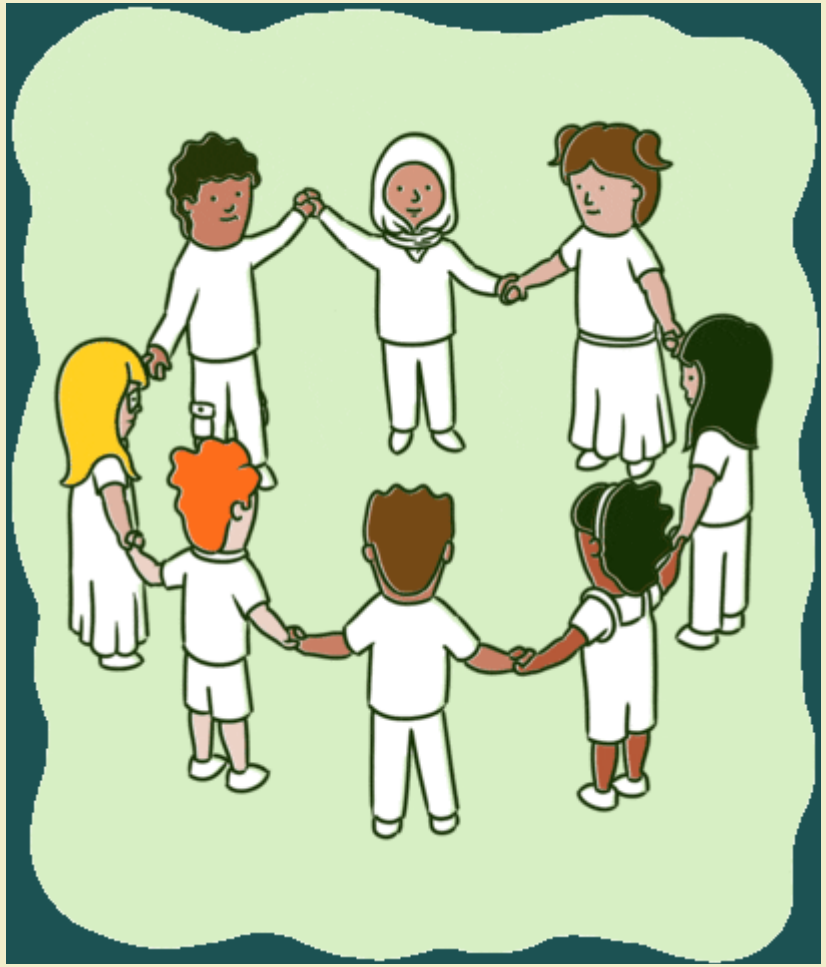
Annotations

Plot Code

```
library(ggthemes)
ggplot(data) +
  geom_histogram(
    aes(x = Depth, fill = Location),
    binwidth = 1,
    color = "lightblue") +
    scale_fill_manual(values = c("steelblue", "maroon")) +
    theme_solarized() +
    theme(legend.position = "none") +
    annotate("text", x = 20, y = 15, label = "Lake #1", color = "steelblue") +
    annotate("text", x = 39, y = 15, label = "Lake #2", color = "maroon")
```

GROUP ACTIVITY 1

06:00



- *Let's go over to maize server/ local Rstudio and our class moodle*
- *Get the class activity 5 .Rmd file*
- *Work on Problem 1*
- *Ask me questions*

Changing coordinates

By default, ggplot2 uses a Cartesian coordinate system, but there are others available!

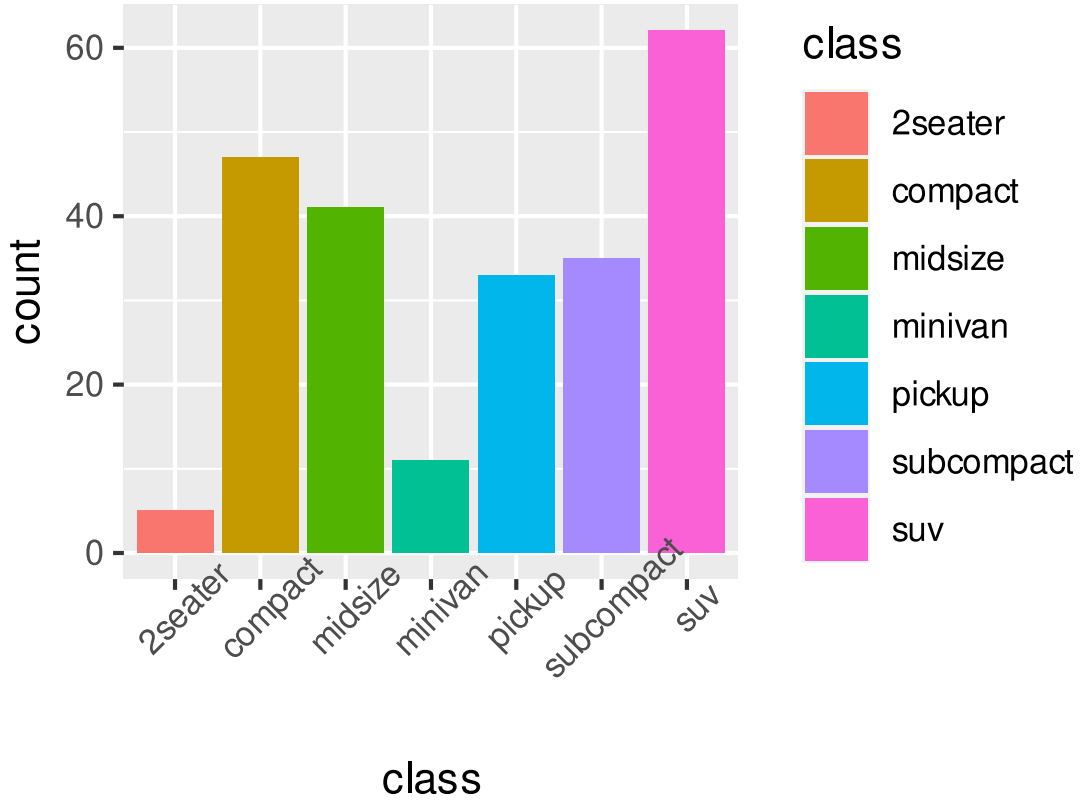
- coord_cartesian: Adjusts the x and y axis limits without modifying the data.
- coord_equal: Ensures equal scaling for the x and y axes.
- coord_fixed: Sets a fixed aspect ratio for the plot.
- coord_flip: Flips the x and y axes.
- coord_map: Projects the plot onto a map projection.
- coord_polar: Transforms the plot to a polar coordinate system.
- coord_quickmap: Provides an approximation for a map projection.
- coord_sf: Designed for use with sf objects (spatial data).
- coord_trans: Transforms the plot's x and y axes using specified transformations.

Cartesian vs. Polar Coordinates

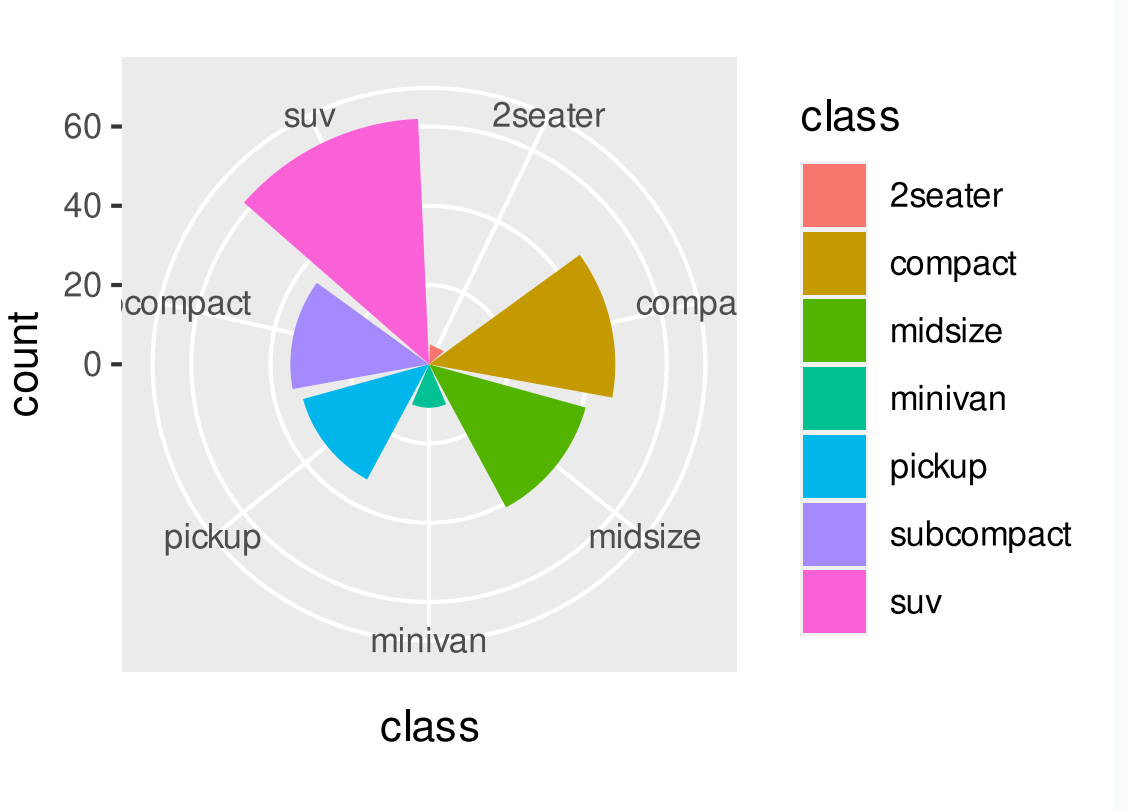
```
ggplot(data = mpg) +
 geom_bar(mapping = aes(x = class, fill = class))
 theme(axis.text.x = element_text(angle = 45))
```

Cartesian vs. Polar Coordinates

```
ggplot(data = mpg) +
 geom_bar(mapping = aes(x = class, fill = class))
 theme(axis.text.x = element_text(angle = 45))
```




```
ggplot(data = mpg) +
 geom_bar(mapping = aes(x = class, fill = class))
 coord_polar(theta = "x")
```



ggplot2 maps

The ggplot2 package contains latitude and longitude to define geographic boundaries

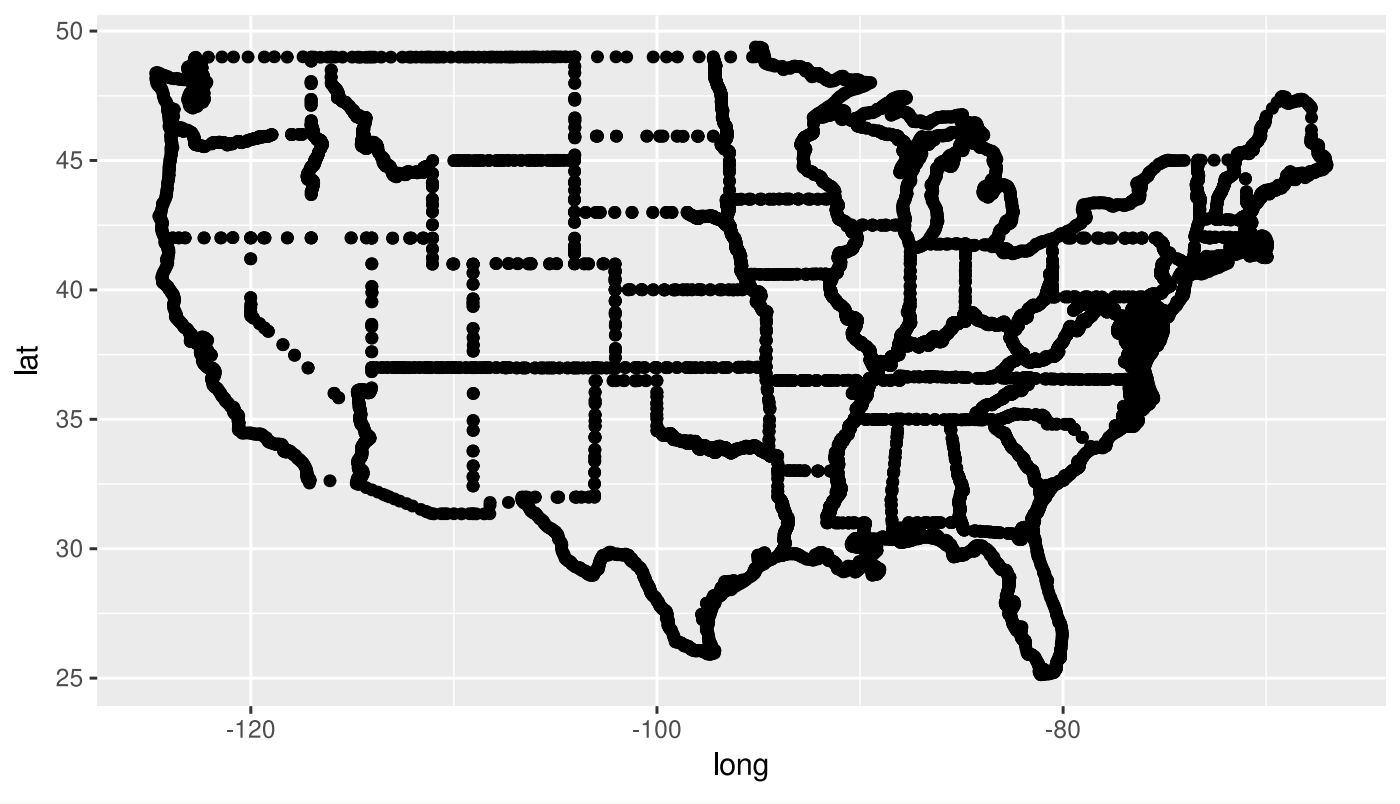
- *some regions: state, usa, world, county*
- *see ?map_data or ?maps for more regions (may need to install maps)*

```
states <- map_data("state")
glimpse(states)
Rows: 15,537
Columns: 6
$ long      <dbl> -87.46201, -87.48493, -87.52503, -87.53076, -87.57087, -87.5…
$ lat       <dbl> 30.38968, 30.37249, 30.37249, 30.33239, 30.32665, 30.32665, …
$ group     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ order     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1…
$ region    <chr> "alabama", "alabama", "alabama", "alabama", "alabama", "alab…
$ subregion <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
```

What is a map?

A set of latitude longitude points...

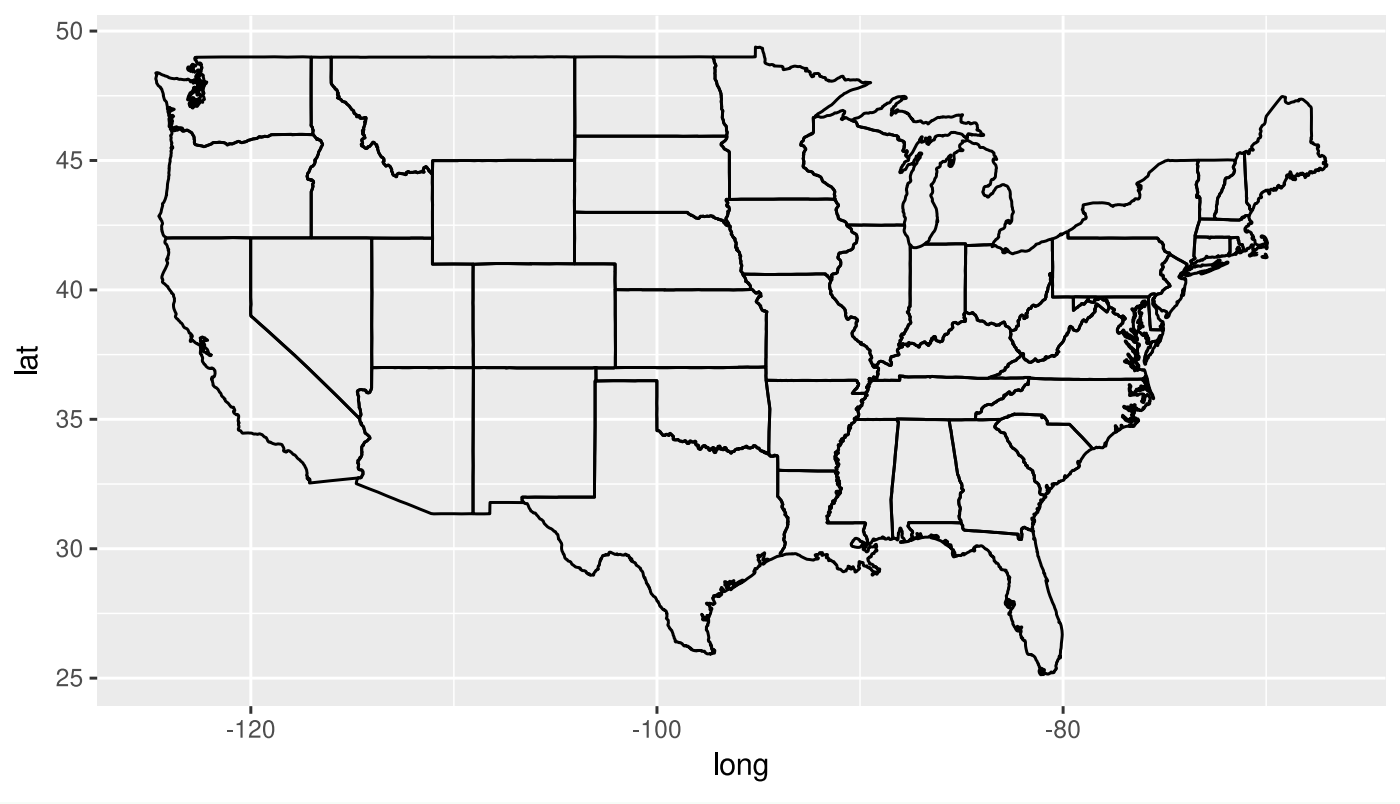
```
ggplot(states) + geom_point(aes(long, lat))
```



What is a map?

... that are connected with lines in a very specific order.

```
ggplot(states) + geom_path(aes(long, lat, group = group))
```



Necessary map data

- *latitude/longitude points for all map boundaries*
- *which boundary group all lat/long points belong*
- *the order to connect points within each group*

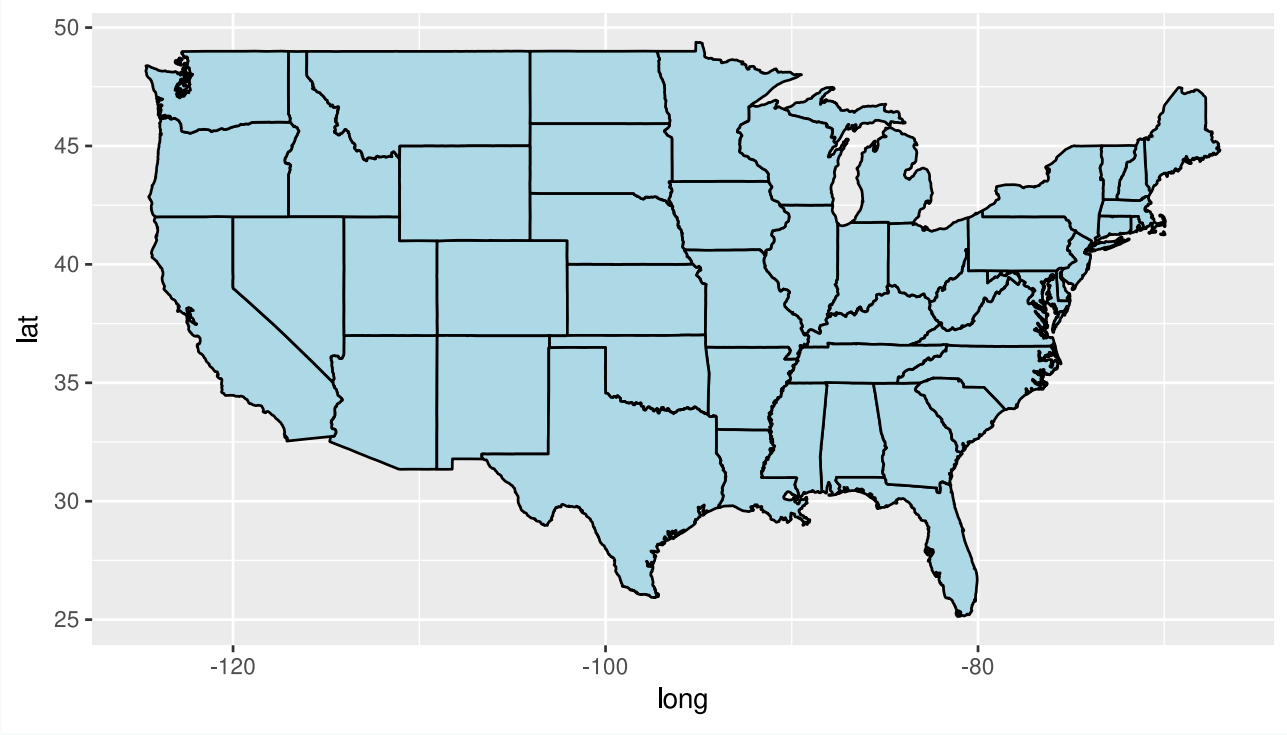
Adding state-level information

- *Add other geographic information by adding geometric layers to the plot*
- *Add non-geographic information by altering the fill color for each state*
  - *Use `geom = "polygon"` to treat states as solid shapes to add color*
  - *Incorporate numeric information using color shade or intensity*
  - *Incorporate categorical information using color hue*

Maps using *geom_polygon*

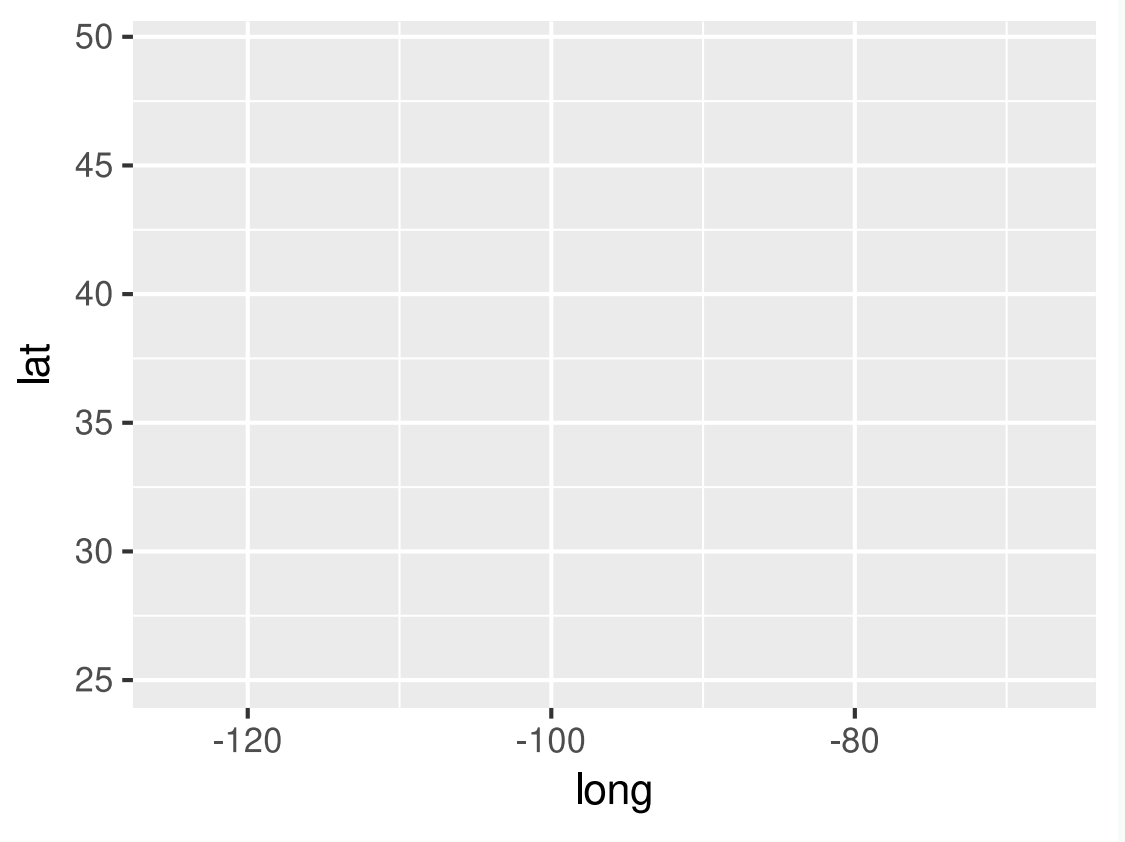
`geom_polygon` connects the dots between lat (`y`) and long (`x`) points in a given `group`. It connects start and end points which allows you to `fill` a closed polygon shape

```
ggplot(states, aes(x=long, y=lat, group=group)) +
    geom_polygon(color="black", fill="lightblue")
```



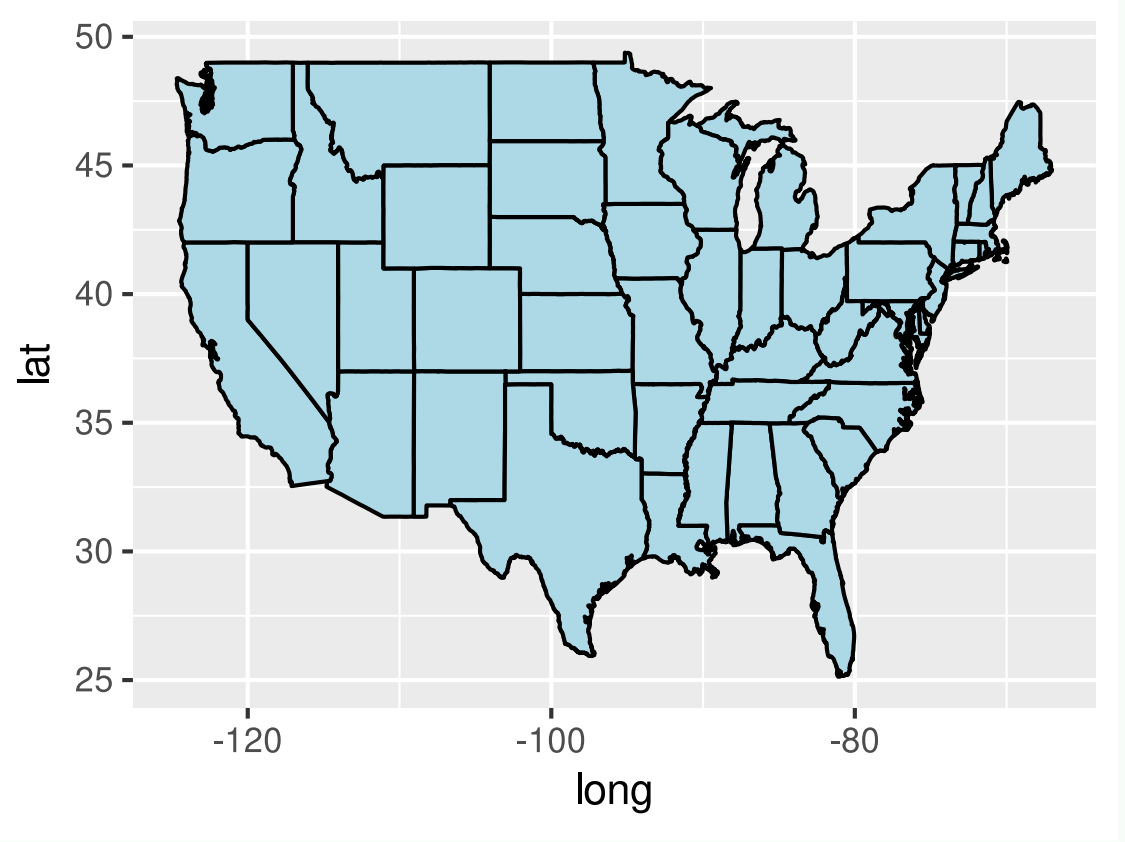
Why is scale so important in a map?

```
ggplot(states, aes(x=long, y=lat, group=group))
```



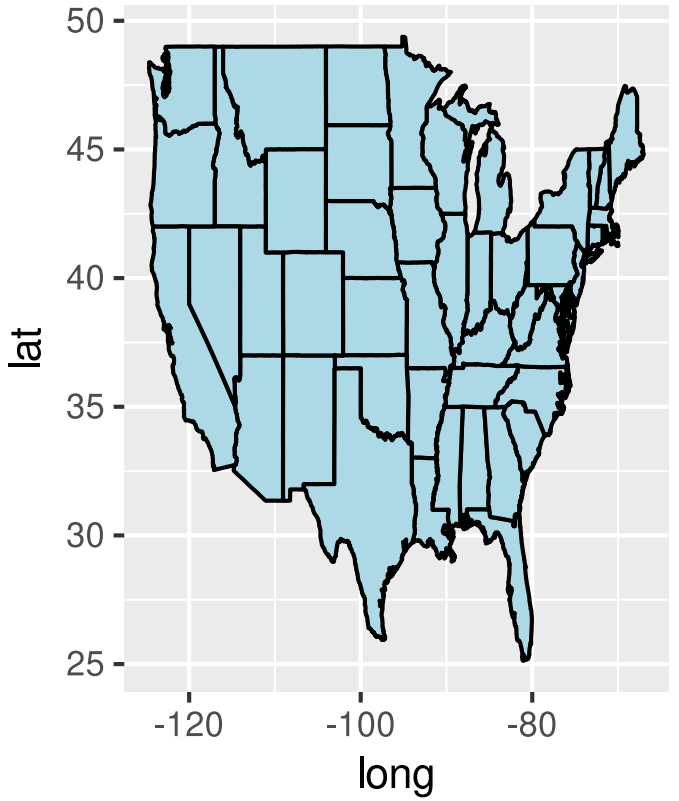
Why is scale so important in a map?

```
ggplot(states, aes(x=long, y=lat, group=group)) +
    geom_polygon(color="black", fill="lightblue")
```



Why is scale so important in a map?

```
ggplot(states, aes(x=long, y=lat, group=group)) +
    geom_polygon(color="black", fill="lightblue")
    coord_fixed(ratio=3)
```



Covid mapping

```
Rows: 51
Columns: 7
$ State                                 <chr> "Alabama", "Alaska", "Arizona",…
$ `7-day avg. cases`                    <int> 294, 0, 465, 0, 173, 320, 147, …
$ `7-day avg. deaths`                   <int> 1, 0, 0, 0, 0, 7, 0, 0, 0, 0, 3…
$ Cases                                 <chr> "1,650,449", "285,075", "2,454,…
$ Deaths                                <chr> "21,127", "1,441", "29,852", "1…
$ `7-day avg. hospitalizations`         <int> 41, 10, 172, 70, 612, 100, 42, …
$ `7-day avg. hospitalizations per 100k` <dbl> 0.8, 1.0, 2.0, 2.0, 1.0, 1.0, 1…
```

Source: https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/

Combining datasets

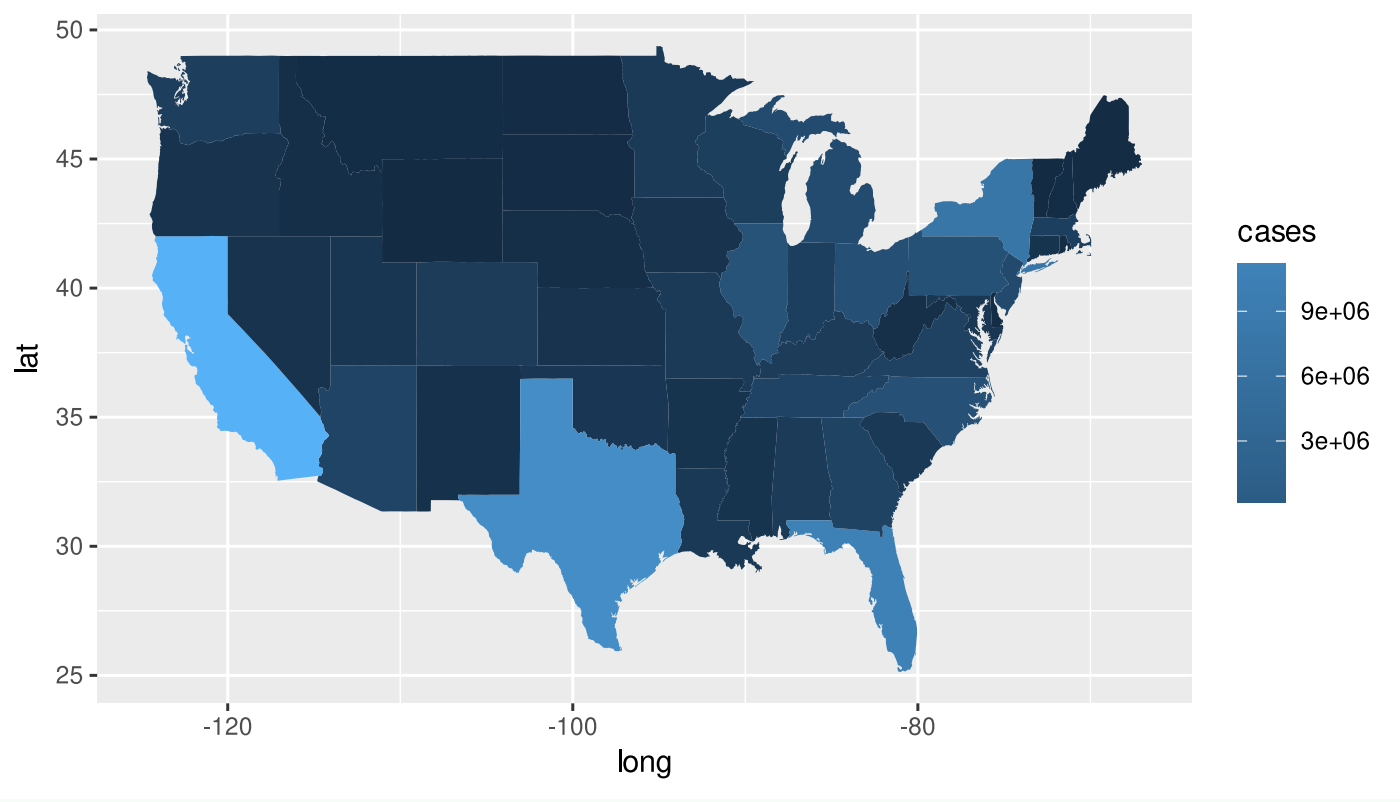
We need to add the covid info to the state polygon data set

```
states <- map_data("state")
covid_data <- left_join(states, covid_clean, by = c("region" = "state"))
```

```
# A tibble: 15,537 × 12
   long   lat group order region  subreg…¹ x7_da…² x7_da…³  cases deaths x7_da…⁴
  <dbl> <dbl> <dbl> <int> <chr>   <chr>      <int>   <int>  <dbl>  <dbl>   <int>
1 -87.5  30.4     1     1 alabama <NA>         294       1 1.65e6  21127      41
2 -87.5  30.4     1     2 alabama <NA>         294       1 1.65e6  21127      41
3 -87.5  30.4     1     3 alabama <NA>         294       1 1.65e6  21127      41
4 -87.5  30.3     1     4 alabama <NA>         294       1 1.65e6  21127      41
5 -87.6  30.3     1     5 alabama <NA>         294       1 1.65e6  21127      41
# … with 15,532 more rows, 1 more variable:
#   x7_day_avg_hospitalizations_per_100k <dbl>, and abbreviated variable names
#   ¹subregion, ²x7_day_avg_cases, ³x7_day_avg_deaths,
#   ⁴x7_day_avg_hospitalizations
```

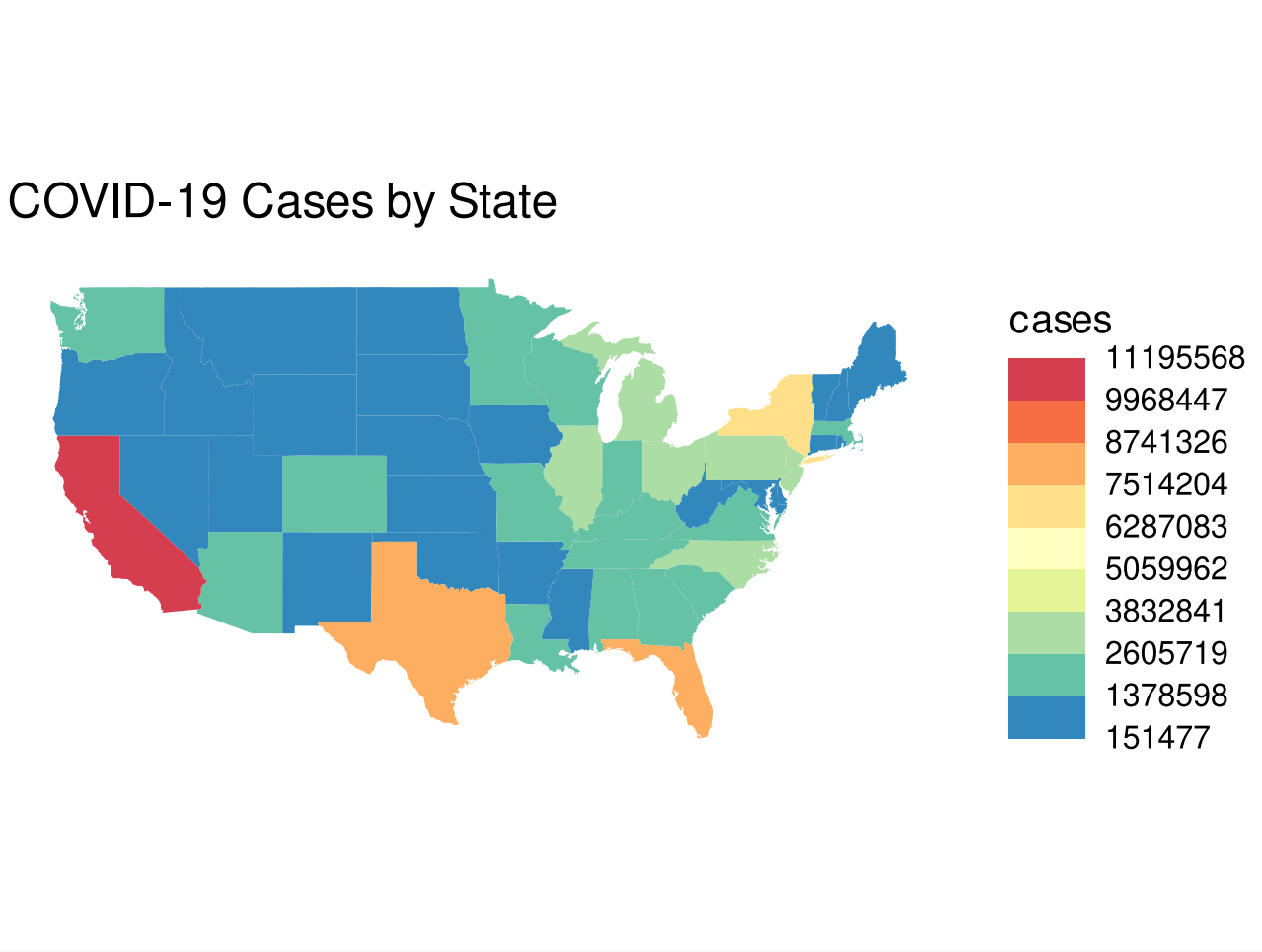
COVID Cases

```
Covid_cases_map <- ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = cases))
Covid_cases_map
```



Adjusting the coordinate system + theme + scale + breaks

Plot Code



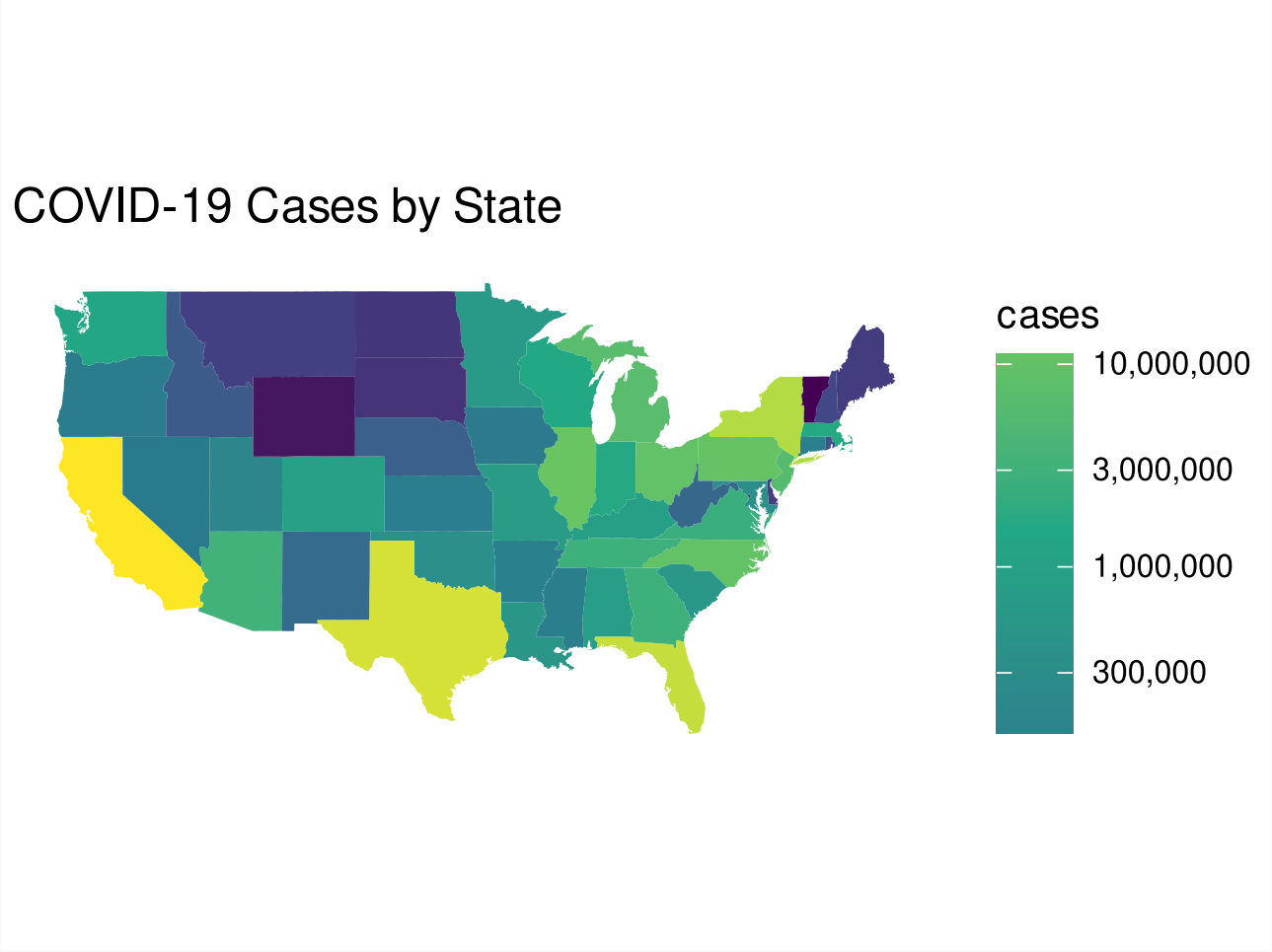
Adjusting the coordinate system + theme + scale + breaks

Plot Code

```
breaks <- round(seq(min(covid_data$cases), max(covid_data$cases), length.out = 10))
Covid_cases_map + coord_map() + theme_map() + theme(legend.position="right") +
    scale_fill_fermenter(type = "div", palette = "Spectral", breaks = breaks) +
    labs(title = "COVID-19 Cases by State") +
    theme(legend.position="right",
          plot.margin = margin(0, 0, 0, 0))  # Adjust the values inside 'margin()' to re
```

Adjusting the color: alternate way

Plot Code



Adjusting the color: alternate way

Plot | Code

```
library(viridis)
ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = cases)) +
  scale_fill_viridis_c(option = "viridis", trans = "log10", labels = scales::comma,
                       guide = guide_colorbar(title.position = "top")) +
  labs(fill = "cases", title = "COVID-19 Cases by State") +
  coord_map() + theme_map() +
  theme(legend.position="right",
        plot.margin = margin(0, 0, 0, 0))
```

Choropleth maps

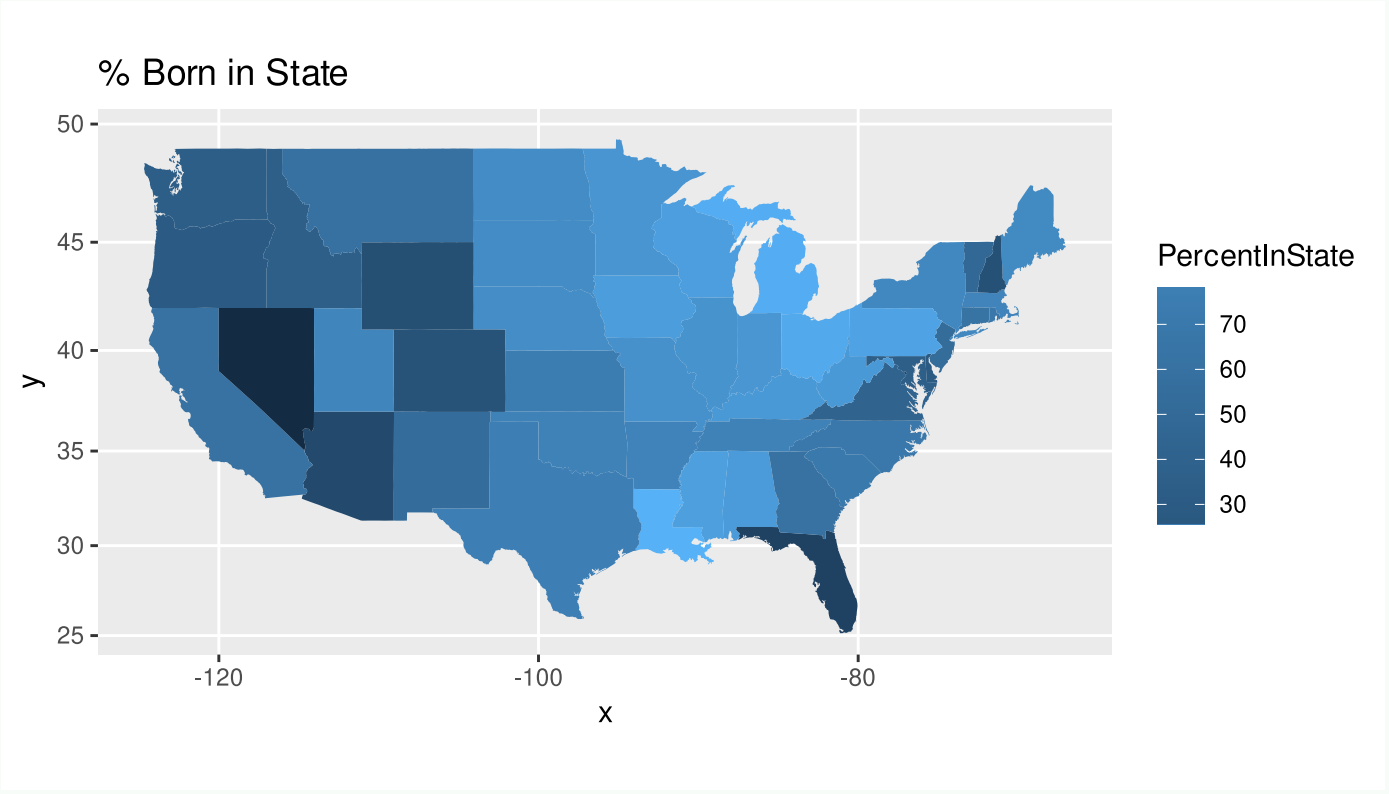
- Uses color or shading of sub regions to visual data
- Displays divided geographical areas or regions that are colored in relation to a numeric variable.

```
ACS <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/ACS.csv")
ACS <- dplyr::filter(ACS, !(region  %in% c("Alaska", "Hawaii"))) # only 48+D.C.
ACS$region <- tolower(ACS$region)  # lower case (match states regions)
glimpse(ACS)
Rows: 49
Columns: 8
$ X             <int> 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18,…
$ region        <chr> "alabama", "arizona", "arkansas", "california", "colora…
$ PopSize       <int> 4841164, 6728577, 2968472, 38654206, 5359295, 3588570, …
$ MedianAge     <dbl> 38.6, 37.1, 37.7, 36.0, 36.4, 40.6, 39.6, 33.8, 41.6, 3…
$ PercentFemale <dbl> 51.5, 50.3, 50.9, 50.3, 49.8, 51.2, 51.6, 52.6, 51.1, 5…
$ BornInState   <int> 3387845, 2623391, 1823628, 21194542, 2294446, 1981427, …
$ MedianIncome  <int> 23527, 26565, 22787, 27772, 31325, 34124, 30648, 41160,…
$ PercentInState <dbl> 69.98, 38.99, 61.43, 54.83, 42.81, 55.21, 45.49, 36.72,…
```

Cloropleth maps using geom_map

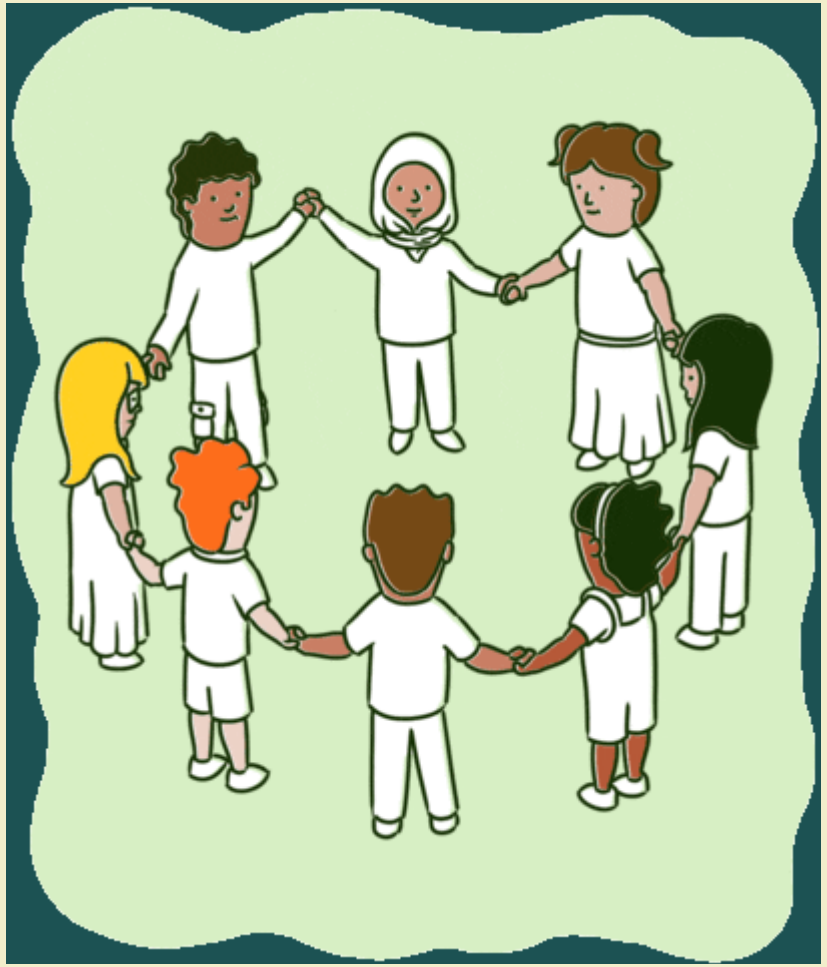
- Don't need to merge ACS and states data!

```
ggplot(data=ACS) + coord_map() +
  geom_map(aes(map_id = region, fill = PercentInState), map = states) +
  expand_limits(x=states$long, y=states$lat) + ggtitle("% Born in State")
```



GROUP ACTIVITY 2

10:00



- *Please work on the remaining problems*
- *Please feel free to talk to me or your neighbor*
- *Submit a pdf to moodle when done*