More Data Visualization Tools

STAT 220

Bastola
January 16 2022

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
- Statistical transformations

More to learn ...

Today

- Changing scales (e.g., color, shape, linetype)
- Changing coordinates
- Changing themes
- Adding annotations
- Mapping spatial data

Changing scales

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

Examples:

- `scale_fill_manual()`
- `scale_fill_brewer()`
- `scale_color_viridis()`
- `scale_shape_manual()`

Recommended reading:

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

Example

Let's make Mountain #1 `green3` and Mountain #2 `lightblue3`

Plot Code

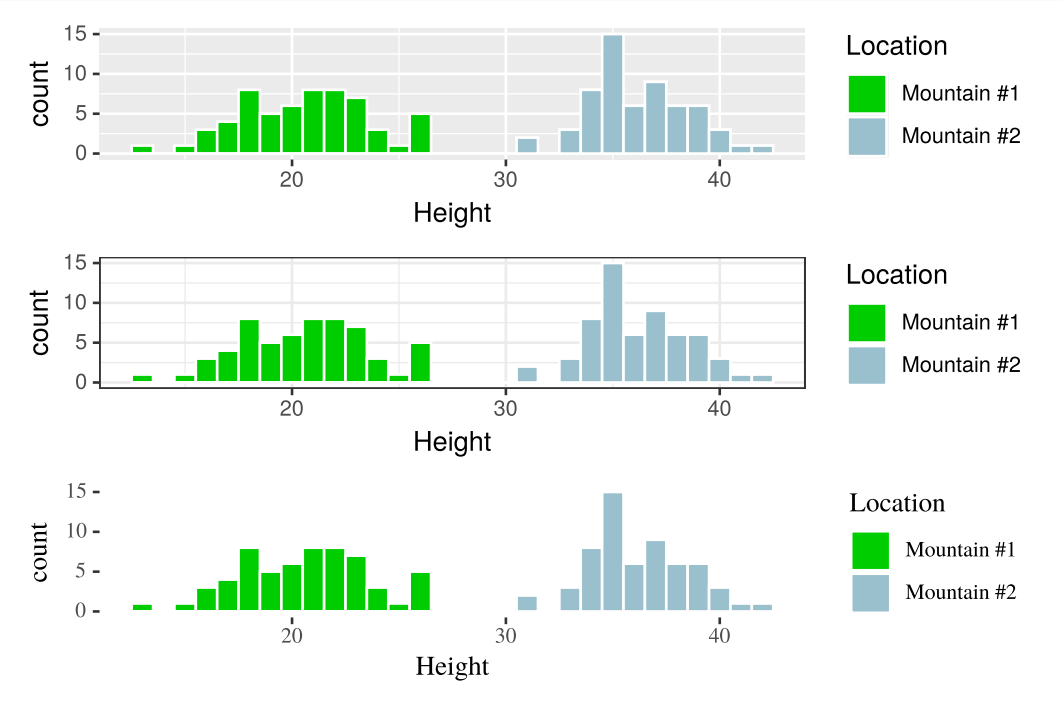
```
ggplot(dat) +
  geom_histogram(
    aes(x = Height, fill = Location),
    binwidth = 1,
    color = "white")+
    scale_fill_manual(values = c("green3", "lightblue3"))
```

Changing themes

Theme: The non-data ink on your plots

Examples:

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



Prepackaged themes

`ggplot2` themes

- `theme_grey()`
- `theme_bw()`
- `theme_linedraw()`
- `theme_light()`
- `theme_dark()`
- `theme_minimal()`
- `theme_classic()`
- `theme_void()`
- `theme_test()`

`ggthemes` themes

- `theme_clean()`
- `theme_economist()`
- `theme_excel()`
- `theme_fivethirtyeight()`
- `theme_gdocs()`
- `theme_solarized()`
- `theme_stata()`
- `theme_tufte()`
- `theme_wsj()`
- And more!

Annotations

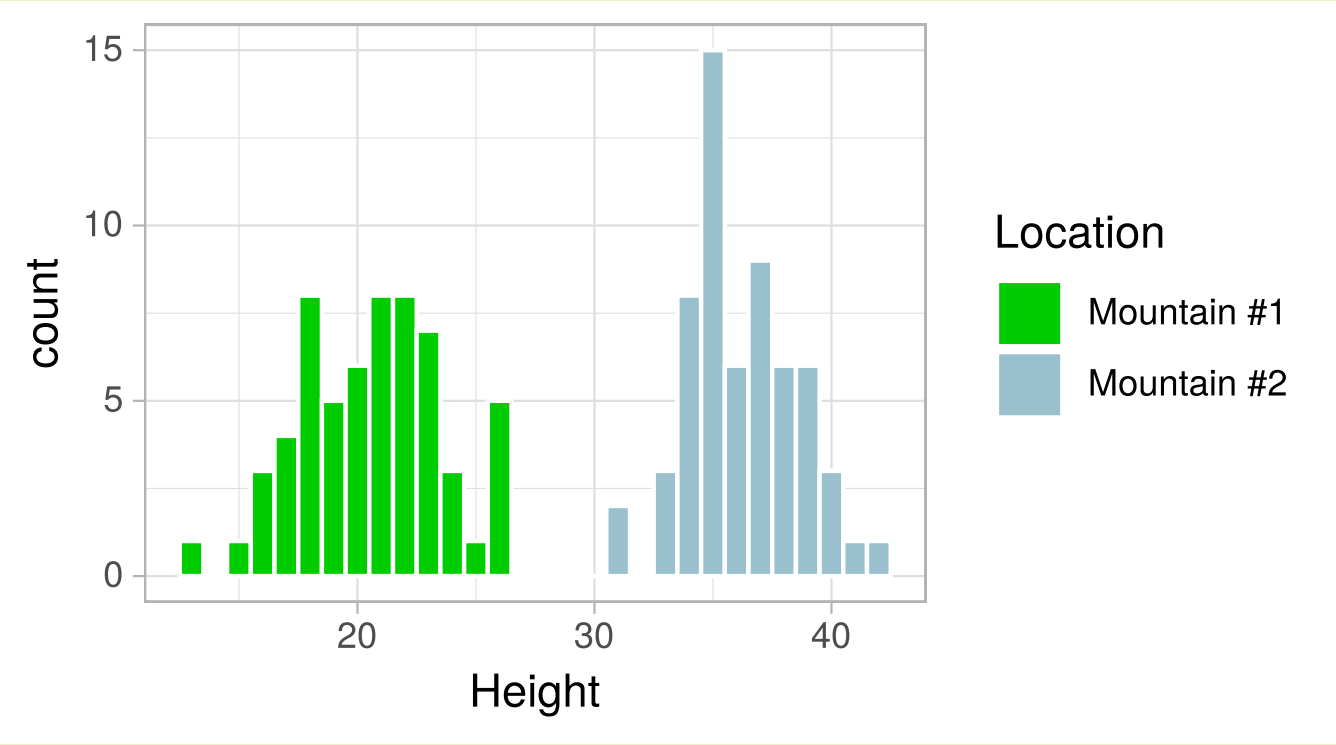
Plot Code

```
last_plot() +
  theme(legend.position = "none") +
  annotate("text", x = 20, y = 15, label = "Mountain #1", color = "green3") +
  annotate("text", x = 36, y = 15, label = "Mountain #2", color = "lightblue3")
```

Your Turn 1

Please git clone the repository More visualizations.

- Apply `theme_light()` to the histogram
- Close "gap" between bars and axis
- Remove legend title, border and minor grid lines
- Re-position legend & remove the background



05:00

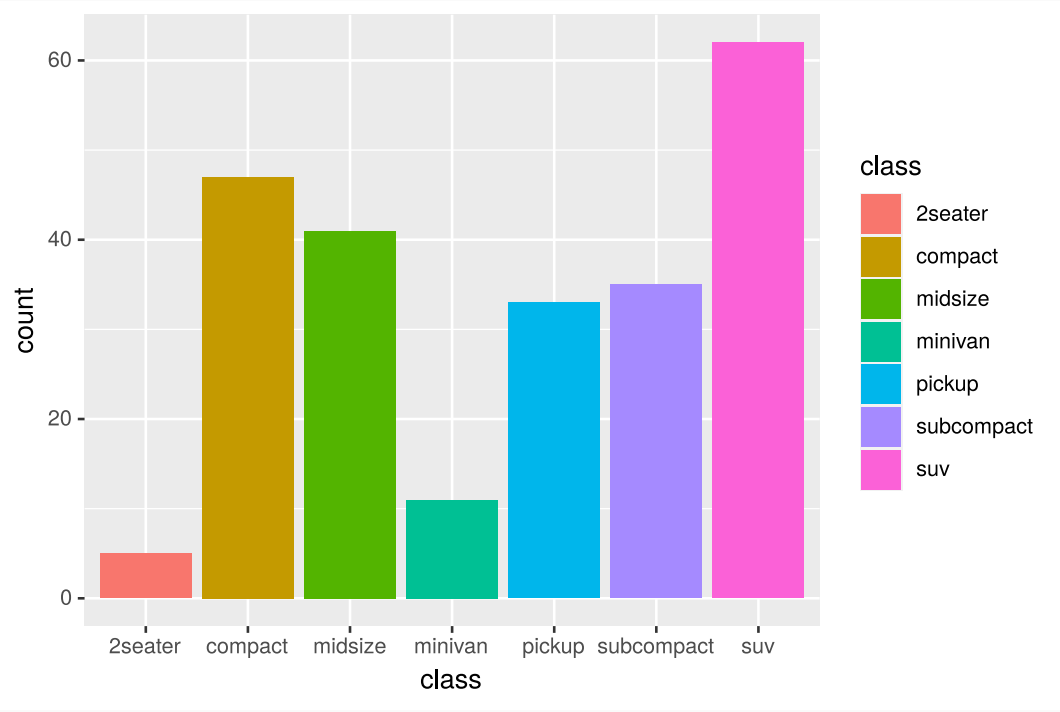
Changing coordinates

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

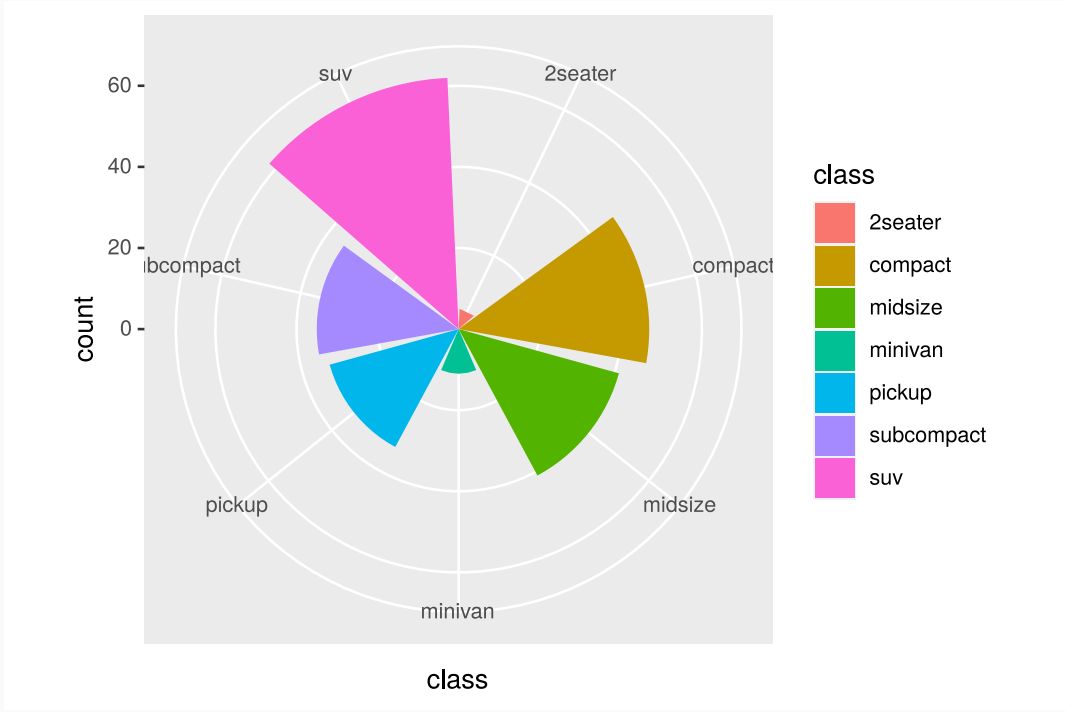
- `coord_cartesian`
- `coord_equal`
- `coord_fixed`
- `coord_flip`
- `coord_map`
- `coord_polar`
- `coord_quickmap`
- `coord_sf`
- `coord_trans`

Cartesian vs. Polar Coordinates

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




```
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

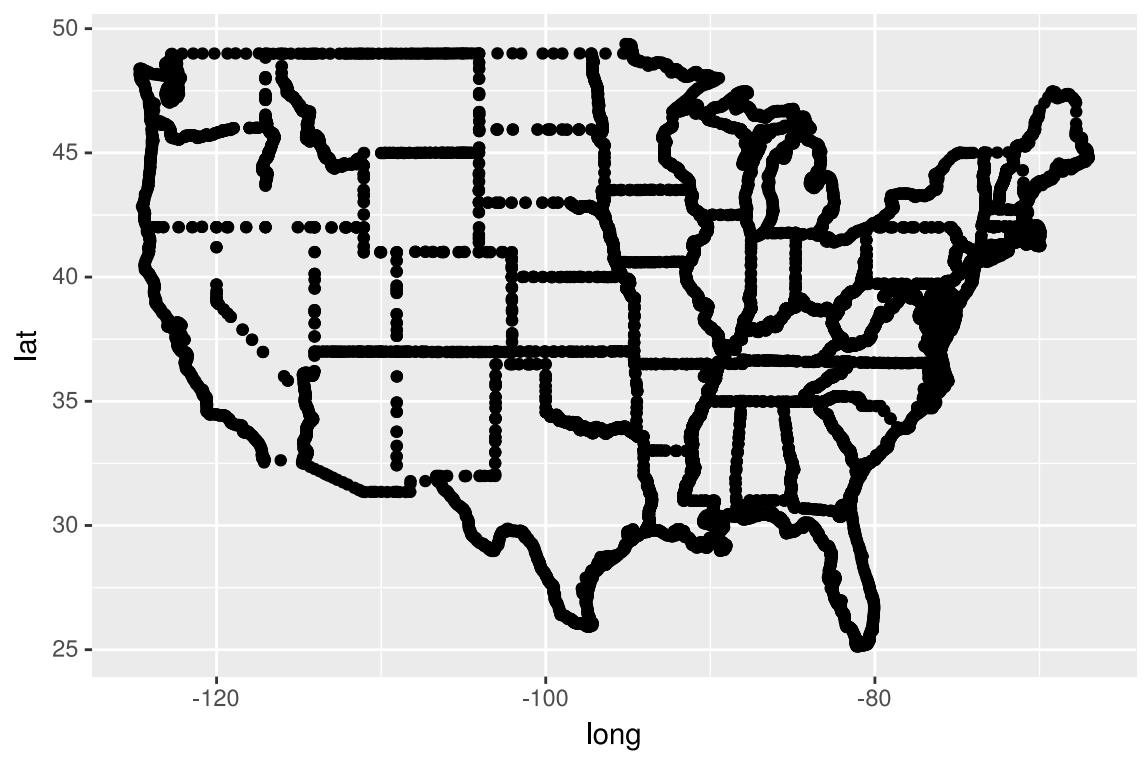
- some regions: `state`, `usa`, `world`, `county`
- see `?map_data` or `?maps` for more regions (may need to install `maps`)

```
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...

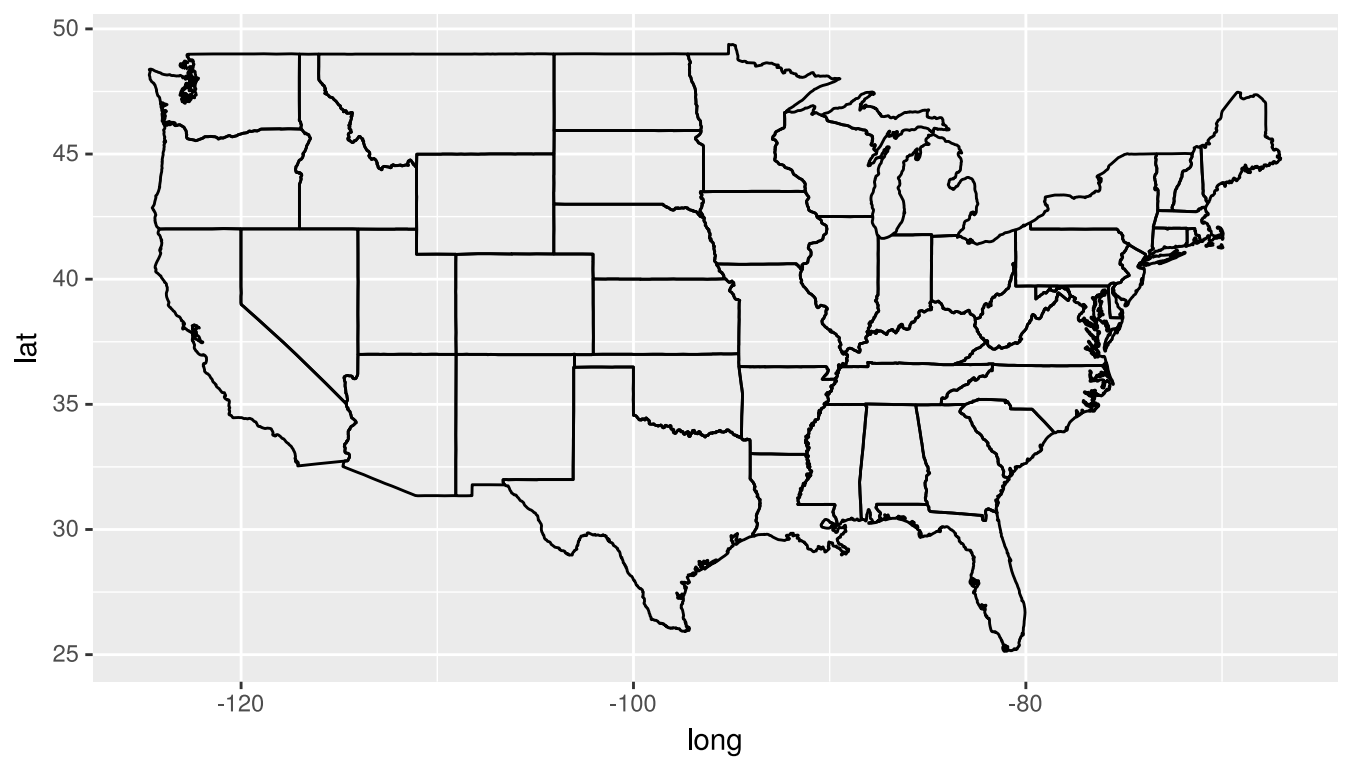
```
ggplot(states) + geom_point(aes(long, lat))
```



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

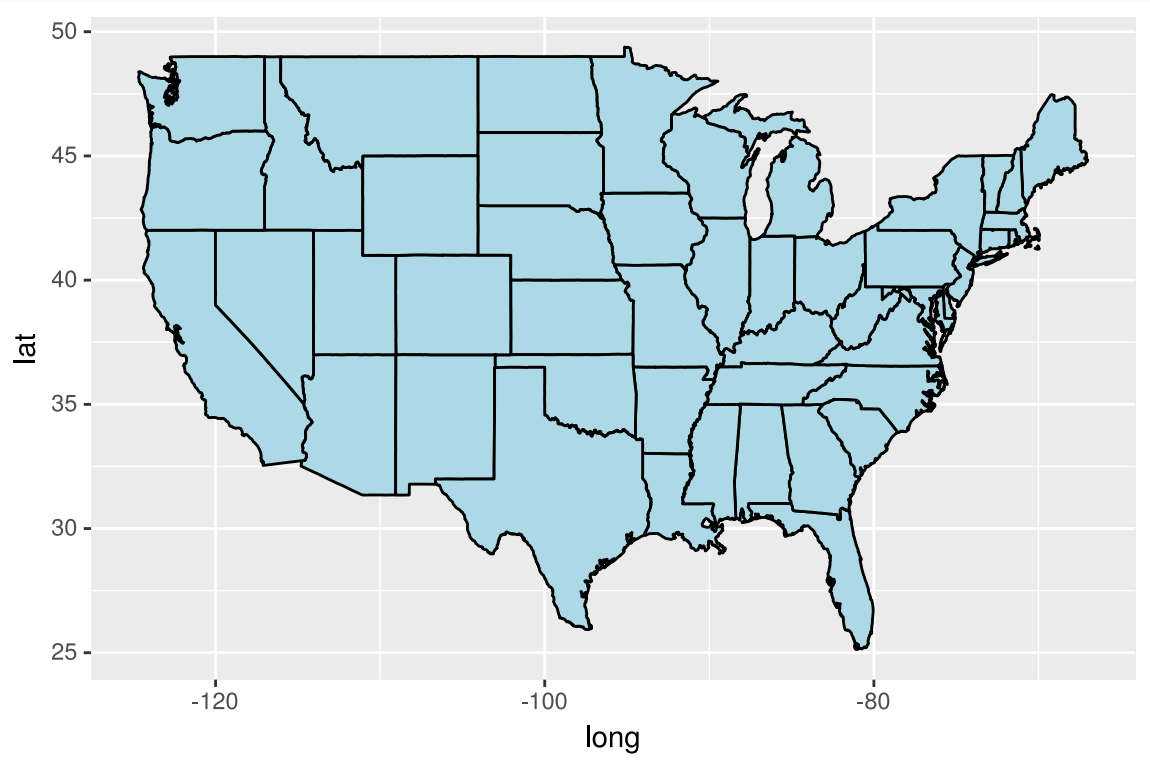
Adding state-level information

- Add other geographic information by adding geometric layers to the plot
- Add non-geopgraphic 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 informaion 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

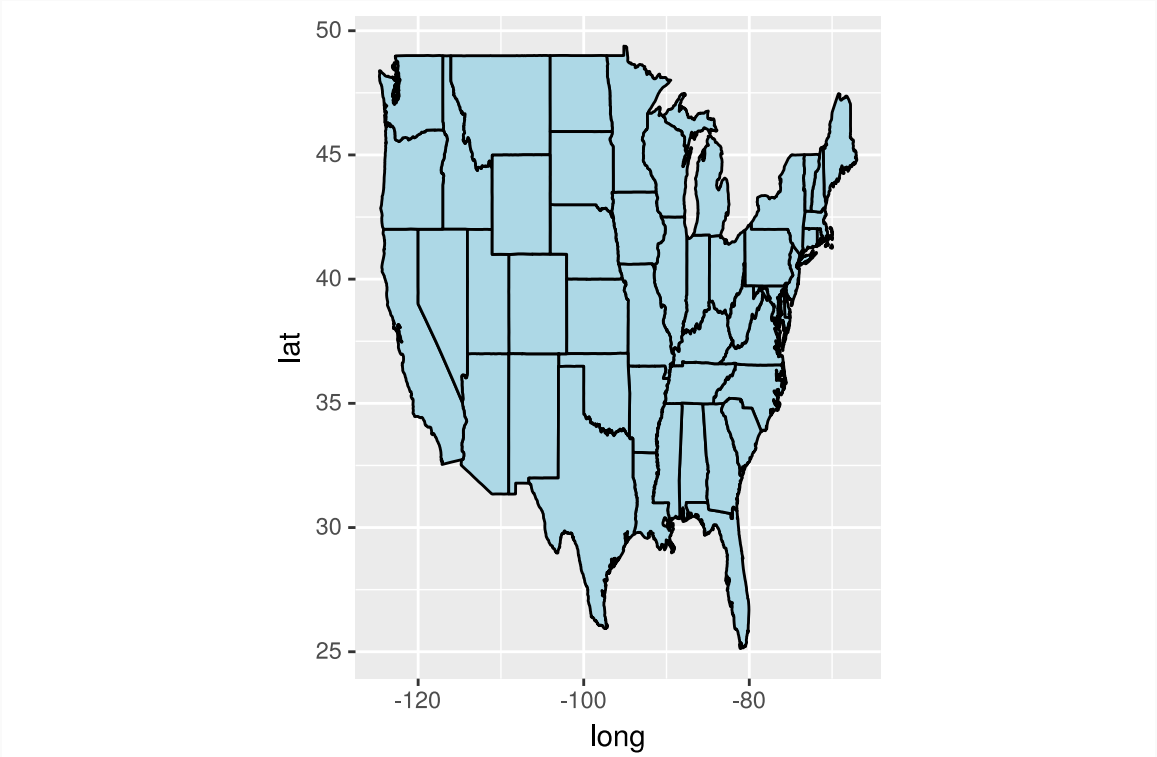
```
ggplot(states, aes(x=long, y=lat, group=group)) +
    geom_polygon(color="black", fill="lightblue")
```



Maps using *geom_polygon*

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 mortality rate

```
Rows: 51
Columns: 8
$ States                 <chr> "alabama", "alaska", "arizona", "arkansas", "ca…
$ 7.day.average.case     <dbl> 10338, 1227, 14971, 7034, 94013, 10562, 9196, 3…
$ 7.day.average.deaths   <int> 19, 0, 60, 16, 88, 16, 23, 13, 2, 28, 40, 2, 9,…
$ TotalCases             <dbl> 1004622, 162955, 1524363, 632743, 6188418, 1062…
$ TotalDeaths            <dbl> 16641, 943, 24992, 9372, 76405, 10528, 9442, 23…
$ Population             <dbl> 4777326, 711139, 6410979, 2916372, 37325068, 50…
$ Deaths.per.100k        <dbl> 348.33294, 132.60417, 389.83126, 321.35818, 204…
$ Cases.per.100k         <dbl> 21028.96, 22914.65, 23777.38, 21696.24, 16579.7…
```

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

Combining datasets

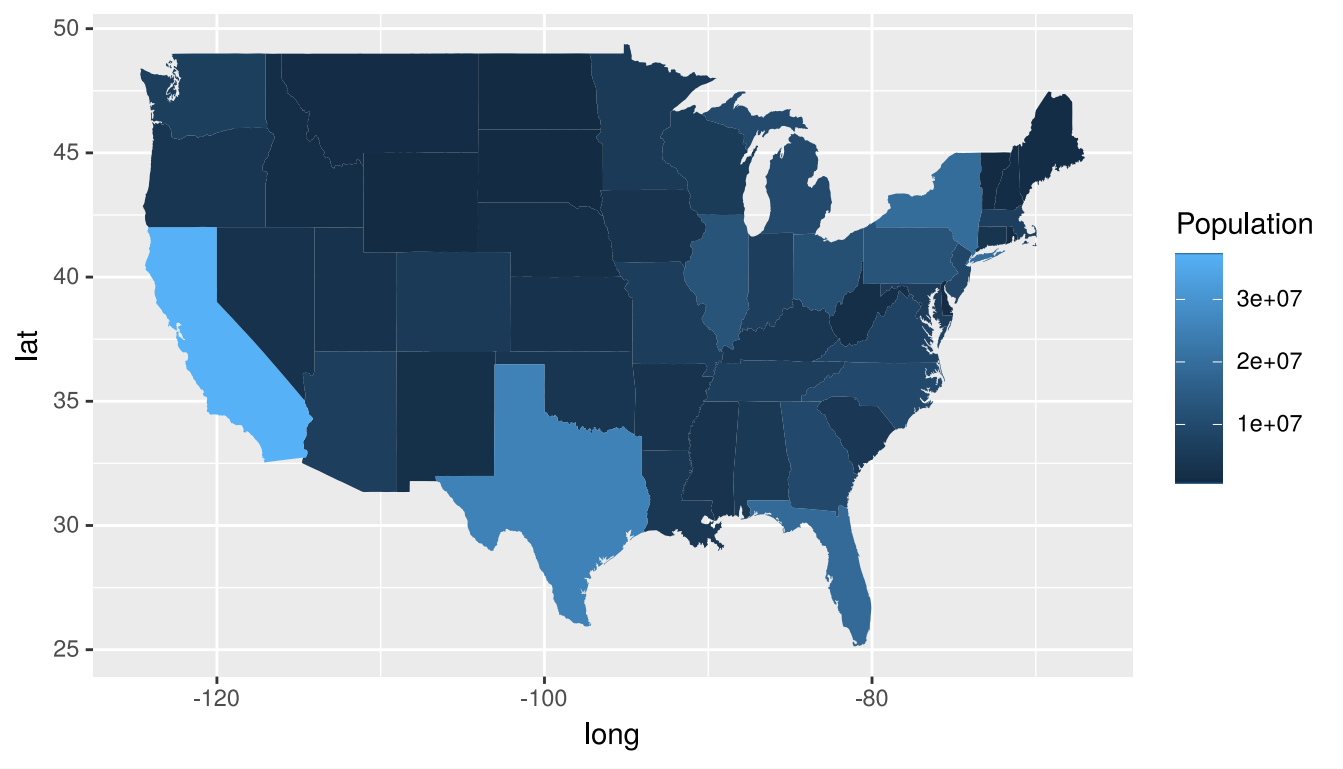
We need to add the covid info to the state polygon data set

```
covid_data <- left_join(states, covid, by = c("region" = "States"))
```

```
# A tibble: 15,537 × 13
   long   lat group order region  subregion 7.day.average.ca… 7.day.average.d…
  <dbl> <dbl> <dbl> <int> <chr>   <chr>                  <dbl>             <int>
1 -87.5  30.4     1     1 alabama <NA>                   10338                19
2 -87.5  30.4     1     2 alabama <NA>                   10338                19
3 -87.5  30.4     1     3 alabama <NA>                   10338                19
4 -87.5  30.3     1     4 alabama <NA>                   10338                19
5 -87.6  30.3     1     5 alabama <NA>                   10338                19
# … with 15,532 more rows, and 5 more variables: TotalCases <dbl>,
#   TotalDeaths <dbl>, Population <dbl>, Deaths.per.100k <dbl>,
#   Cases.per.100k <dbl>
```

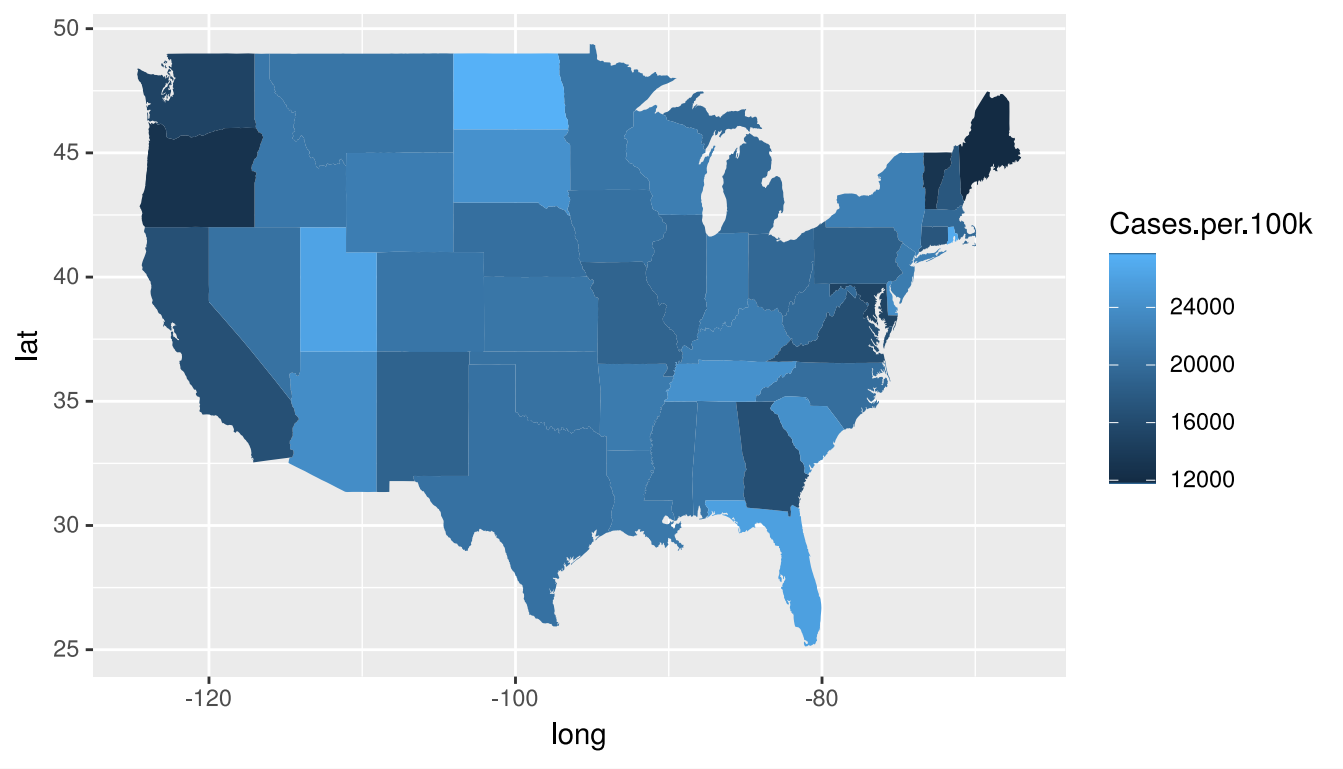
US Population

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



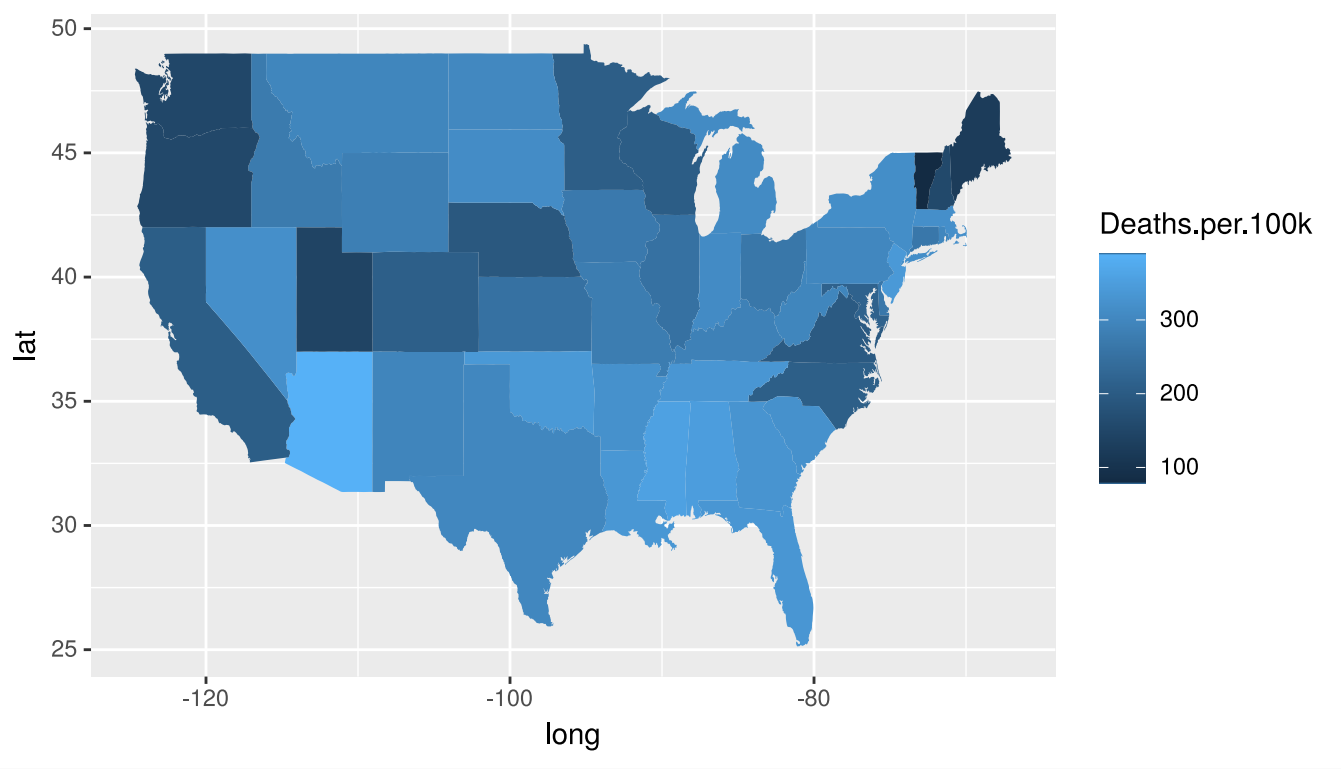
COVID Cases per 100k

```
Covid_cases_map <- ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = Cases.per.100k))
Covid_cases_map
```



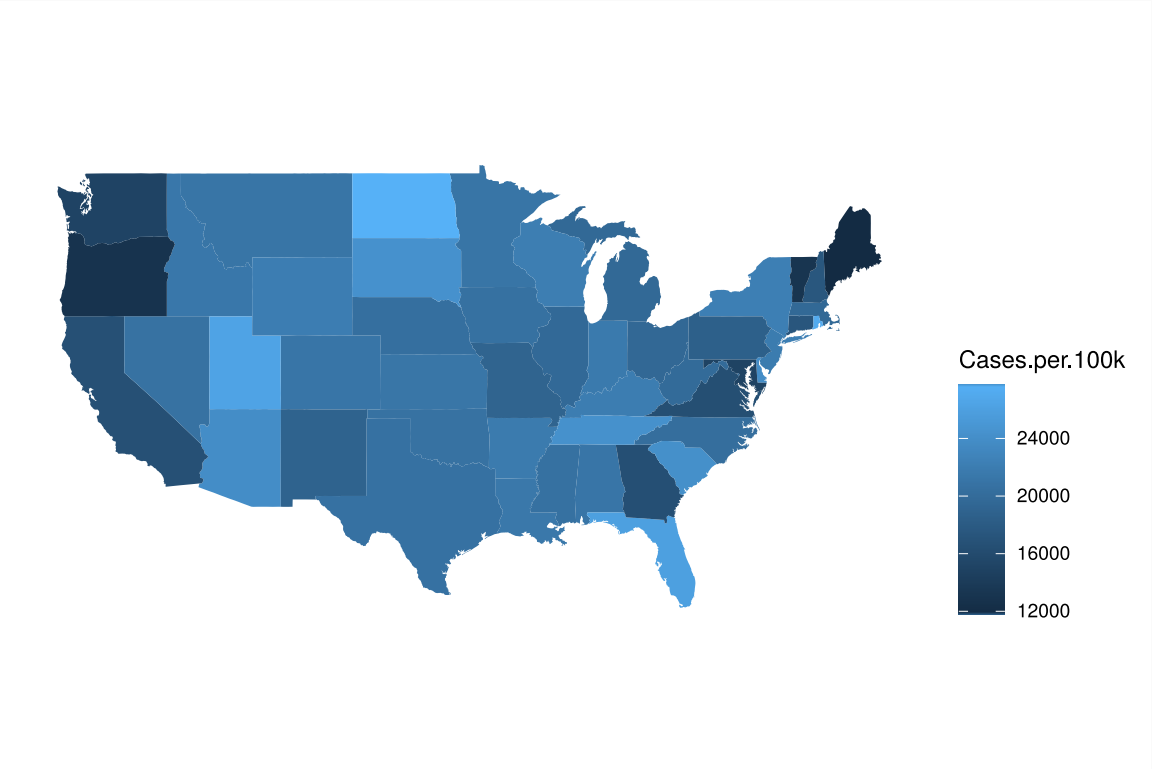
COVID Deaths per 100k

```
Covid_death_map <- ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = Deaths.per.100k))
Covid_death_map
```



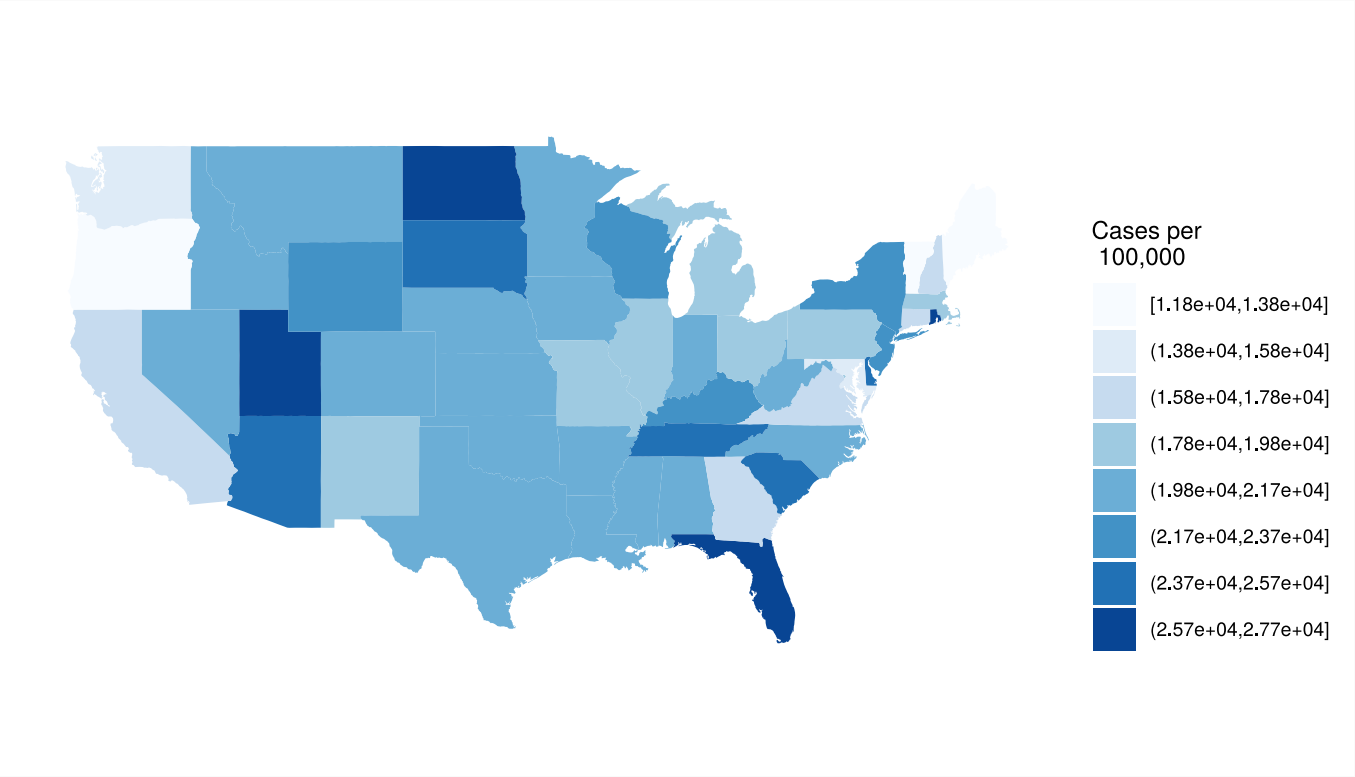
Adjusting the coordinate system + theme

```
Covid_cases_map + coord_map() + theme_map() +  theme(legend.position="right")
```



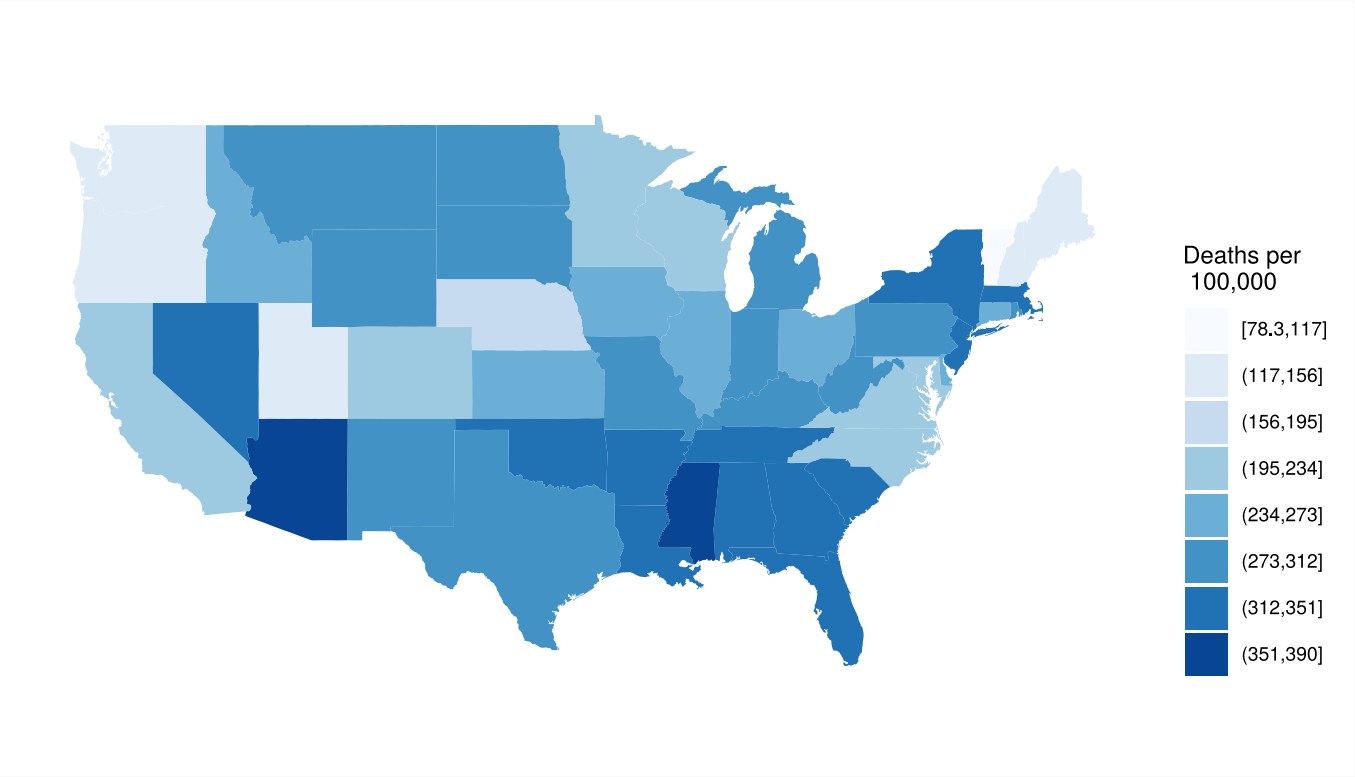
Adjusting the color

```
ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = cut_interval(Cases.per.100k, n=8))) +
  scale_fill_brewer(palette = "Blues") +
  labs(fill = "Cases per \n 100,000") + theme(legend.position="right")
```



Adjusting the color

```
ggplot(covid_data) +
  geom_polygon(aes(long, lat, group = group, fill = cut_interval(Deaths.per.100k, n=8))) +
  scale_fill_brewer(palette = "Blues") +
  labs(fill = "Deaths per \n 100,000") +  theme(legend.position="right")
```



Choropleth maps

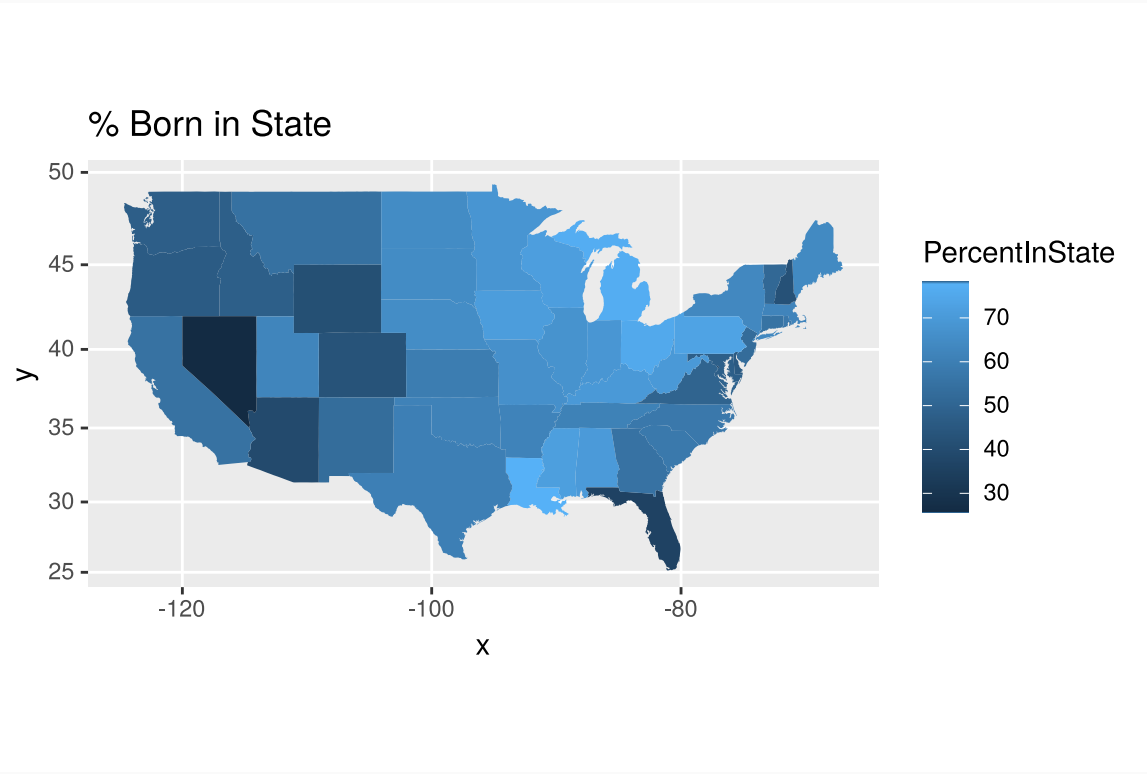
- Uses color or shading of subregions to visual data
- Displays divided geographical areas or regions that are coloured in relation to a numeric variable.

```
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**

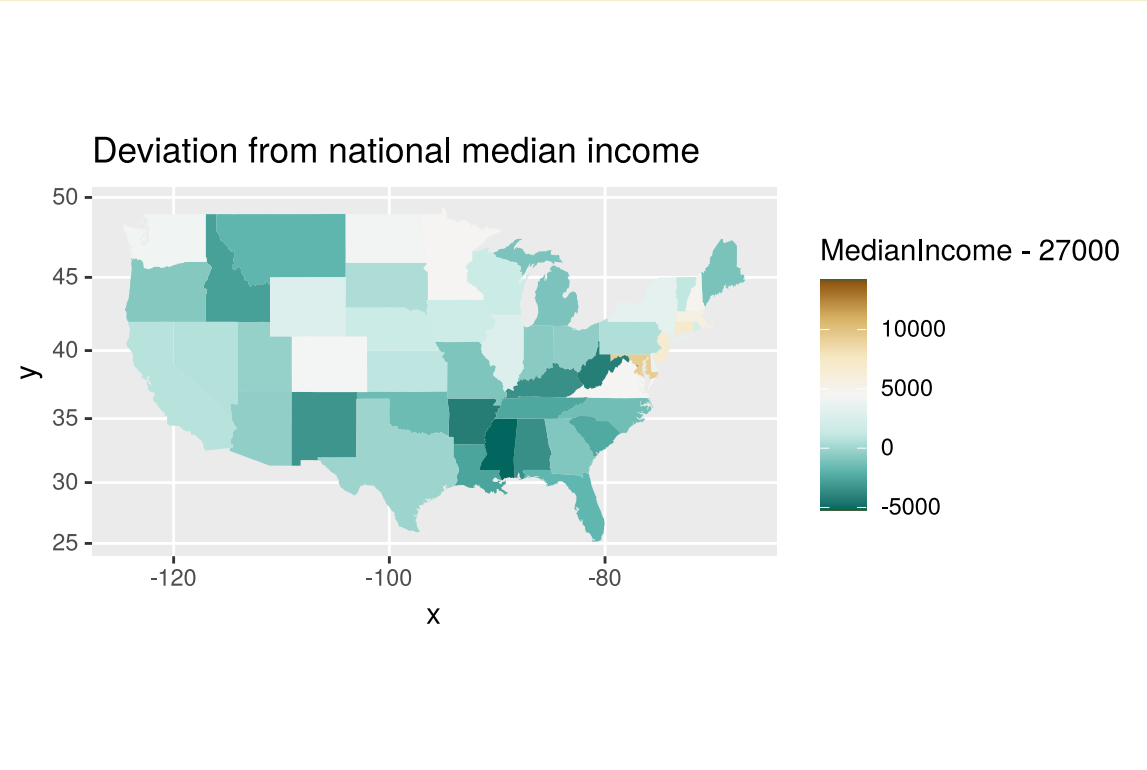
- Don't need to merge `ACS` and `states` data!

```
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")
```



Your Turn 2

Use American Community Survey (ACS) data to complete this exercise.



06:00

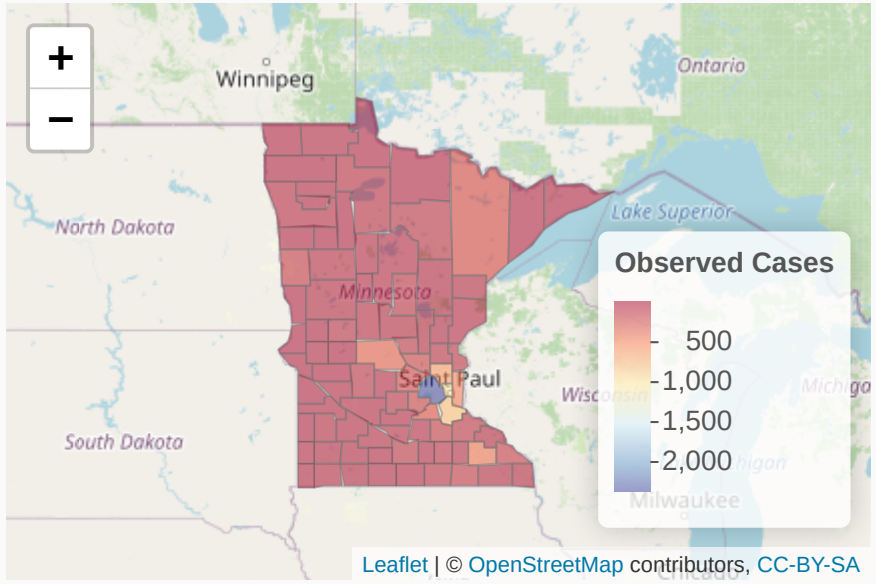
Visualization using **leaflet** in R

```
# Leaflet
data <- data.frame(lat  = 44.4583, long = -93.1616)
leaflet(data) %>%
   addTiles() %>%
   addMarkers(lat = ~lat, lng = ~long, popup="Our Northfield")
```

Leaflet | © OpenStreetMap contributors, CC-BY-SA

Interactive Maps: Covid Cases in Minnesota

```
l <- leaflet(map) %>% addTiles()
l %>% addPolygons(color = "grey", weight = 1, fillColor = ~pal(obs), fillOpacity = 0.5,
    highlightOptions = highlightOptions(weight = 4),
    label = labels,
    labelOptions = labelOptions(style = list("font-weight" = "normal", padding = "3px 8px"),
    textsize = "15px", direction = "auto")) %>%
    addLegend(pal = pal, values = ~obs, opacity = 0.5, title = "Observed Cases", position = "bottomright")
```



Acknowledgement: some of the slides are based on previous works of Adam Loy and Katie St. Clair.