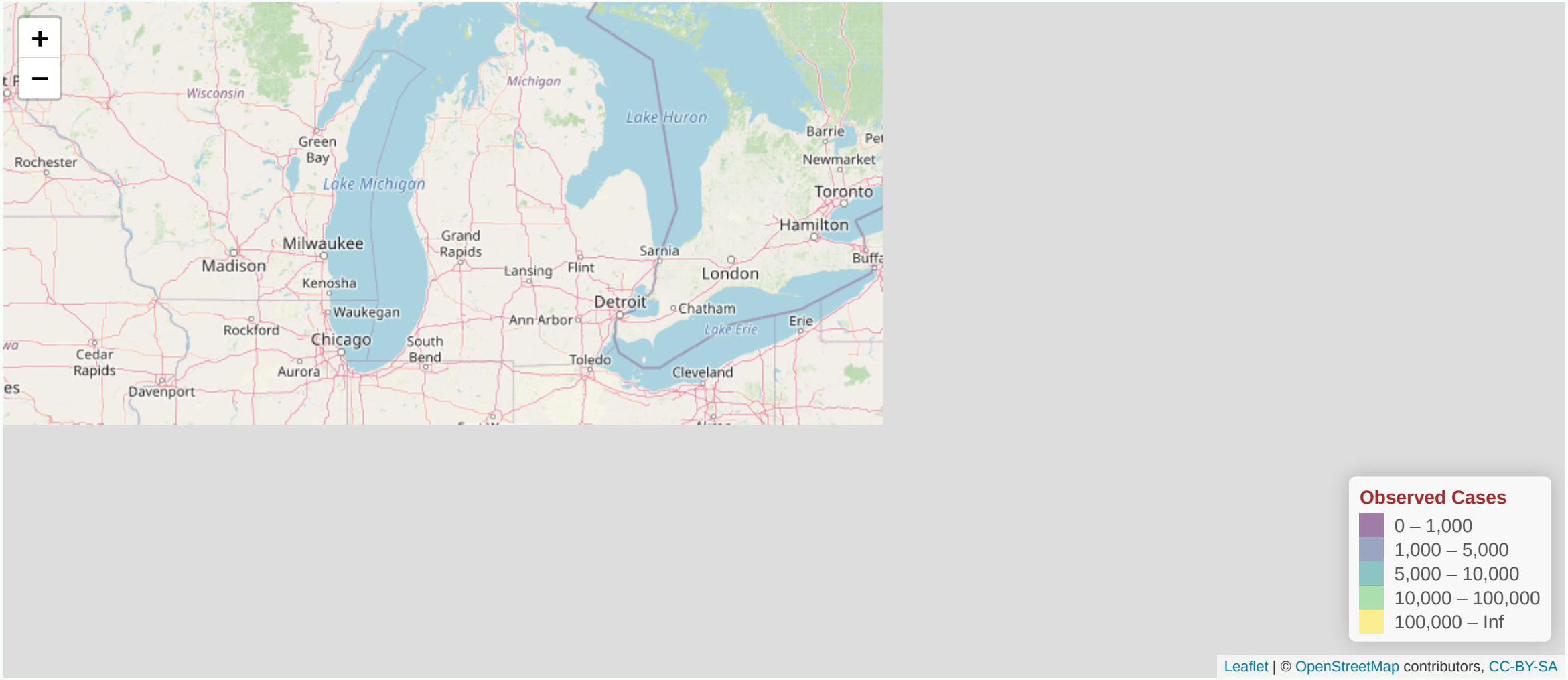
Shiny and Interactive Graphs

Spring 2023

May 12 2023

Interactive leaflet map



Required packages

- ***leaflet*** *is used to create interactive maps*
- ***maps*** *provides geographical data*
- ***sp*** *and* ***maptools*** *are used to manipulate and convert geographical data into formats suitable for* ***leaflet***

We need numeric data to project!

```
table_usafacts <- bow(url = "https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/state/minn
  scrape() %>%
  html_elements(css = "table") %>%
  html_table()
covidMN <- table_usafacts[[2]]

# tidy it up
covidMN_final <- covidMN %>% janitor::clean_names() %>%
  mutate(cases = as.numeric(str_remove(cases, ","))) %>%
  mutate(county = str_remove(county, " County"))
glimpse(covidMN_final)
```

```
Rows: 87
Columns: 5
$ county             <chr> "Aitkin", "Anoka", "Becker", "Beltrami", "Benton", "…
$ x7_day_avg_cases   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ x7_day_avg_deaths <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ cases              <dbl> 3184, 105106, 9244, 12551, 14910, 1465, 19001, 6832,…
$ deaths             <chr> "63", "808", "96", "128", "177", "9", "105", "81", "…
```

Merging Geographical and Covid-19 Data

```
map <- SpatialPolygonsDataFrame(MNmap, covidMN_final, match.ID = FALSE)
```

- *Create a SpatialPolygonsDataFrame, which merges our geographical data (MNmap) with the Covid-19 case data (covidMN_final).*
  - *This new data frame, 'map', contains both the geographical boundaries of each county and the associated Covid-19 case data.*

Defining Color Palettes

Viridis color palette with colorNumeric:

```
pal <- colorNumeric(palette = "viridis", domain = map$cases)
```

Other options: heat, Dark2, Spectral

RColorBrewer palette (Paired) with colorBin:

```
bins <- c(0, 1000, 5000, 10000, 100000, Inf)
pal <- colorBin(palette = "Paired", domain = map$cases, bins = bins)
```

Other options: Accent

RColorBrewer palette (Set1) with colorQuantile:

```
pal <- colorQuantile(palette = "Pastel1", domain = map$cases, n = 5)
```

Other options: Set1

Creating labels

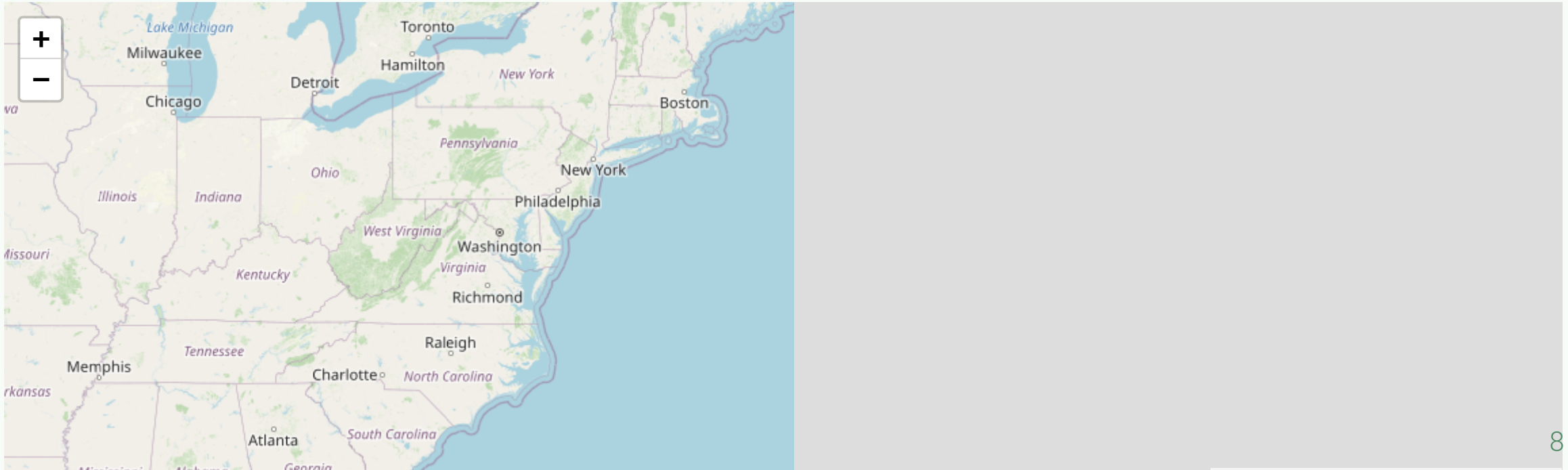
```
labels <- sprintf("<strong> %s </strong> <br/> Observed: %s", map$county, map$cases) %>%
  lapply(htmltools::HTML)
```

```
labels[1:5]
[[1]]
<strong> Aitkin </strong> <br/> Observed: 3184

[[2]]
<strong> Anoka </strong> <br/> Observed: 105106

[[3]]
<strong> Becker </strong> <br/> Observed: 9244

[[4]]
<strong> Beltrami </strong> <br/> Observed: 12551

[[5]]
<strong> Benton </strong> <br/> Observed: 14910
```

Initializing Leaflet Map

Leaflet is a JavaScript library for creating dynamic maps that support panning and zooming along with various annotations like markers, polygons, and popups.

```
l <- leaflet(map) %>% addTiles() %>% setView(lng = -93.1616, lat = 44.4583, zoom = 5)
l
```



Adding Polygons and Highlight Options

`addPolygons` adds geographical shapes to a map.

```
l <- l %>% addPolygons(
  color = "grey",
  weight = 1,
  fillColor = ~pal(cases),
  fillOpacity = 0.7,
  highlightOptions = highlightOptions(
  label = labels
)
```

- Styling arguments include `color` (for border color), `weight` (for border thickness), `fillColor` (for the color inside the polygons), and `fillOpacity` (for the transparency of the fill color).
- `fillColor` means that the fill color of the polygons is determined by the numerical quantity being projected, allowing for a visual representation of the data
- `highlightOptions` argument sets what happens when a user hovers over a polygon, and the `label` argument sets the label that is displayed when this happens.

Adding a Legend

`addLegend` adds a legend to the Leaflet map. It provides necessary context for the map's colors and markers, improving its interpretability.

```
l <- l %>% addLegend(
  pal = pal,
  values = ~cases,
  opacity = 0.5,
  title = "Observed Cases",
  position = "bottomright"
)
```

Color Palette: The `pal` argument is used to specify the color palette. This palette is used to color the items in the legend and, correspondingly, the polygons on the map.

Opacity: The `opacity` argument sets the level of transparency for the legend. Lower values make the legend more transparent, while higher values make it more opaque.

Title and Position: The `title` argument allows you to provide a title for the legend, giving context to the information displayed. The `position` argument controls where the legend is placed on the map, with options like "bottomright", "bottomleft", "topright", and "topleft".

Recap: thins we can do with `leaflet`

- *Make the background map with `leaflet()`, `addTiles()` and `setView()`*
- *Use `addPolygons()` to add the shape of country/states/county*
- *Translate a numeric variable to a palette of color*
  - *Quantile with `colorQuantile`*
  - *Numeric with `colorNumeric`*
  - *Bin with `colorBin`*

For more information, click here

Objects needed for plotting

```
library(leaflet)  # for leaflet maps
library(maps)  # for map data
library(sp)  # for spatial polygons
library(maptools)  # for sp polygon data frame


MNcounty <- map("county", "Minnesota", plot=FALSE, fill=TRUE)
MNmap <- map2SpatialPolygons(MNcounty, IDs = MNcounty$names)
map <- SpatialPolygonsDataFrame(MNmap, covidMN_final, match.ID = FALSE)

bins <- c(0, 1000, 5000, 10000, 100000, Inf)
pal <- colorBin("magma", pretty = TRUE, domain = map$cases, bins = bins)

labels <- sprintf("<strong> %s </strong> <br/> Observed: %s", map$county, map$cases) %>%
  lapply(htmltools::HTML)

l <- leaflet(map) %>% addTiles() %>% setView(lng = -93.1616, lat = 44.4583, zoom = 5)

l %>% addPolygons(color = "grey", weight = 1,
                  fillColor = ~pal(cases), fillOpacity = 0.7,
                  highlightOptions = highlightOptions(weight = 5),
                  label = labels) %>%
  addLegend(pal = pal, values = ~cases, opacity = 0.5,
            title = "Observed Cases",
            position = "bottomright")
```

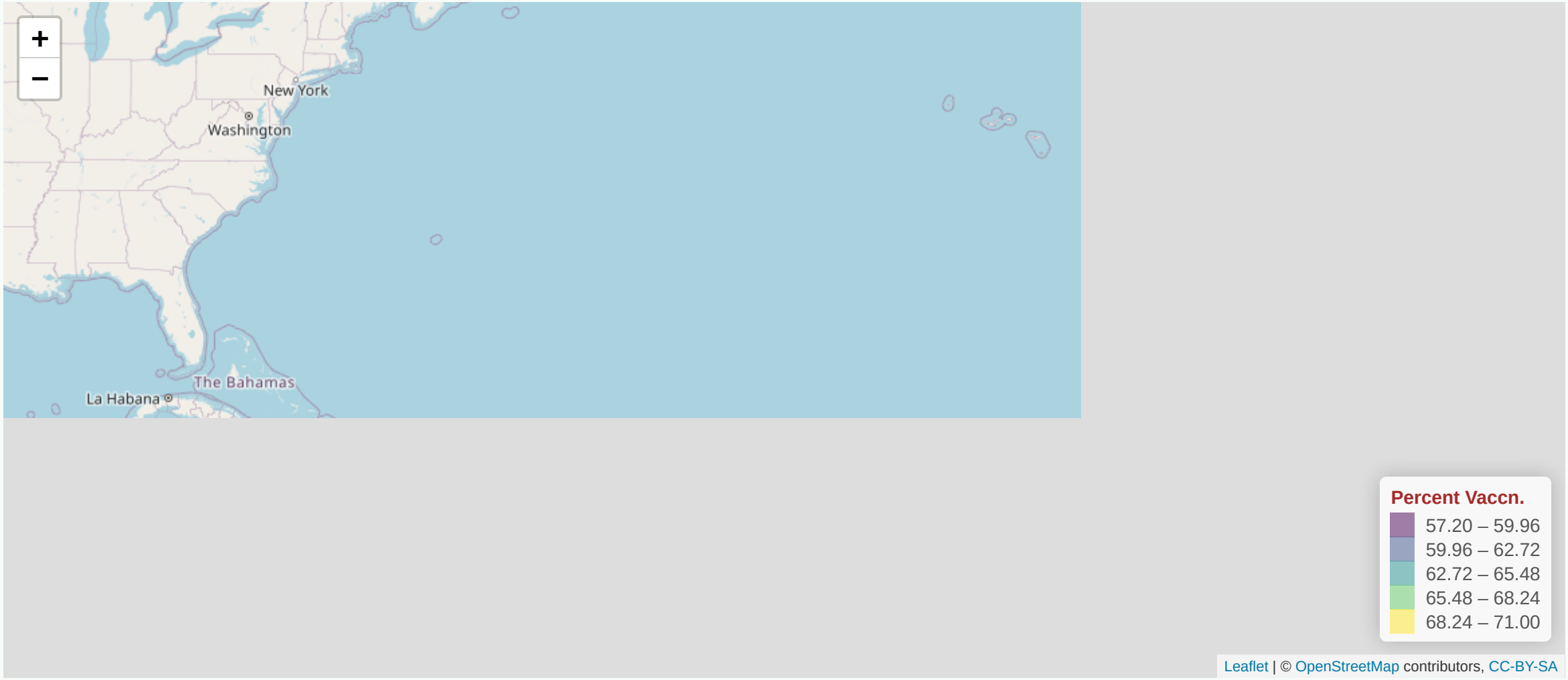
Objects needed for plotting

```
library(leaflet)  # for leaflet maps
library(maps)  # for map data
library(sp)  # for spatial polygons
library(maptools)  # for sp polygon data frame


MNcounty <- map("county", "Minnesota", plot=FALSE, fill=TRUE)
MNmap <- map2SpatialPolygons(MNcounty, IDs = MNcounty$names)
map <- SpatialPolygonsDataFrame(MNmap, covidMN_final, match.ID = FALSE)



pal <- colorNumeric(palette = "viridis", alpha = TRUE, domain = map$cases)
bins <- c(0, 1000, 5000, 10000, 100000, Inf)
pal <- colorBin("viridis", domain = map$cases, bins = bins)

labels <- sprintf("<strong> %s </strong> <br/> Observed: %s", map$county, map$cases) %>%
  lapply(htmltools::HTML)

l <- leaflet(map) %>% addTiles() %>% setView(lng = -93.1616, lat = 44.4583, zoom = 5)

l %>% addPolygons(color = "grey", weight = 1,
                  fillColor = ~pal(cases), fillOpacity = 0.7,
                  highlightOptions = highlightOptions(weight = 5),
                  label = labels) %>%
  addLegend(pal = pal, values = ~cases, opacity = 0.5,
            title = "Observed Cases",
            position = "bottomright")
```

Another example

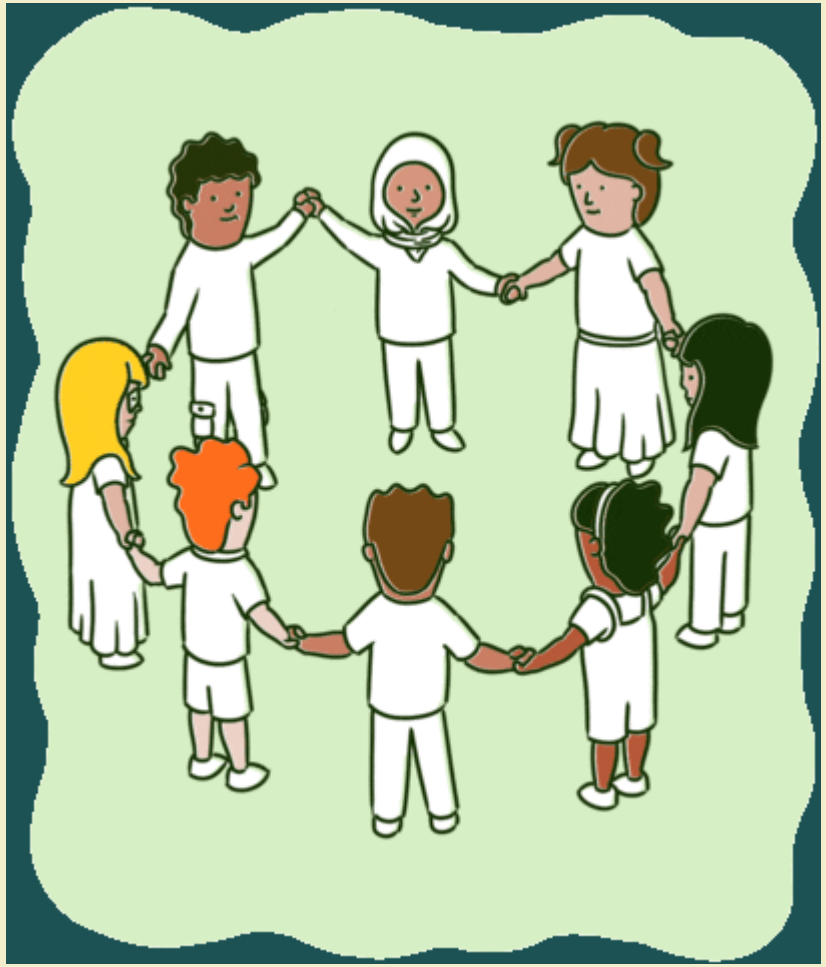
```
# Scrape the data
covid_final <- read_html("https://usafacts.org/visualizations/covid-vaccine-tracker-states/state/minnesot
  html_elements(css = "table") %>% html_table() %>% .[[1]] %>%
  janitor::clean_names() %>%
  mutate_at(2:4, parse_number) %>% mutate(state = str_to_lower(state)) %>%
  filter(state %in% c("minnesota", "wisconsin", "iowa", "michigan", "illinois", "indiana"))

# Prepare the map
USA <- maps::map("state", regions = c("minnesota", "wisconsin", "iowa", "michigan", "illinois","indiana")
  map2SpatialPolygons(IDs = str_remove(.$names, "(?=:).+"))

# Merge the data and the map
map <- SpatialPolygonsDataFrame(USA, covid_final, match.ID = FALSE)

# Create bins and color palette
bins <- seq(min(map$percent_fully_vaccinated), max(map$percent_fully_vaccinated), length.out = 6)
pal <- colorBin("viridis", domain = map$percent_fully_vaccinated, bins = bins)

# Create labels
labels <- sprintf("<strong> %s </strong> <br/> Observed: %s", str_to_upper(map$state), map$percent_fully_
  lapply(htmltools::HTML)

# Plot the map
leaflet(map) %>%
  addTiles() %>%
  setView(lng = -93.1616, lat = 44.4583, zoom = 4) %>%
  addPolygons(
    color = "grey",
    weight = 1,
    fillColor = ~pal(percent_fully_vaccinated),
```

Interactive leaflet map 2



GROUP ACTIVITY 1

15:00



- *Let's go over to maize server/ local Rstudio and our class moodle*
- *Get the class activity 20.Rmd file*
- *Work on activity 1*
- *Ask me questions*