Shiny and Interactive Graphs

Stat 220

Bastola

February 21 2022

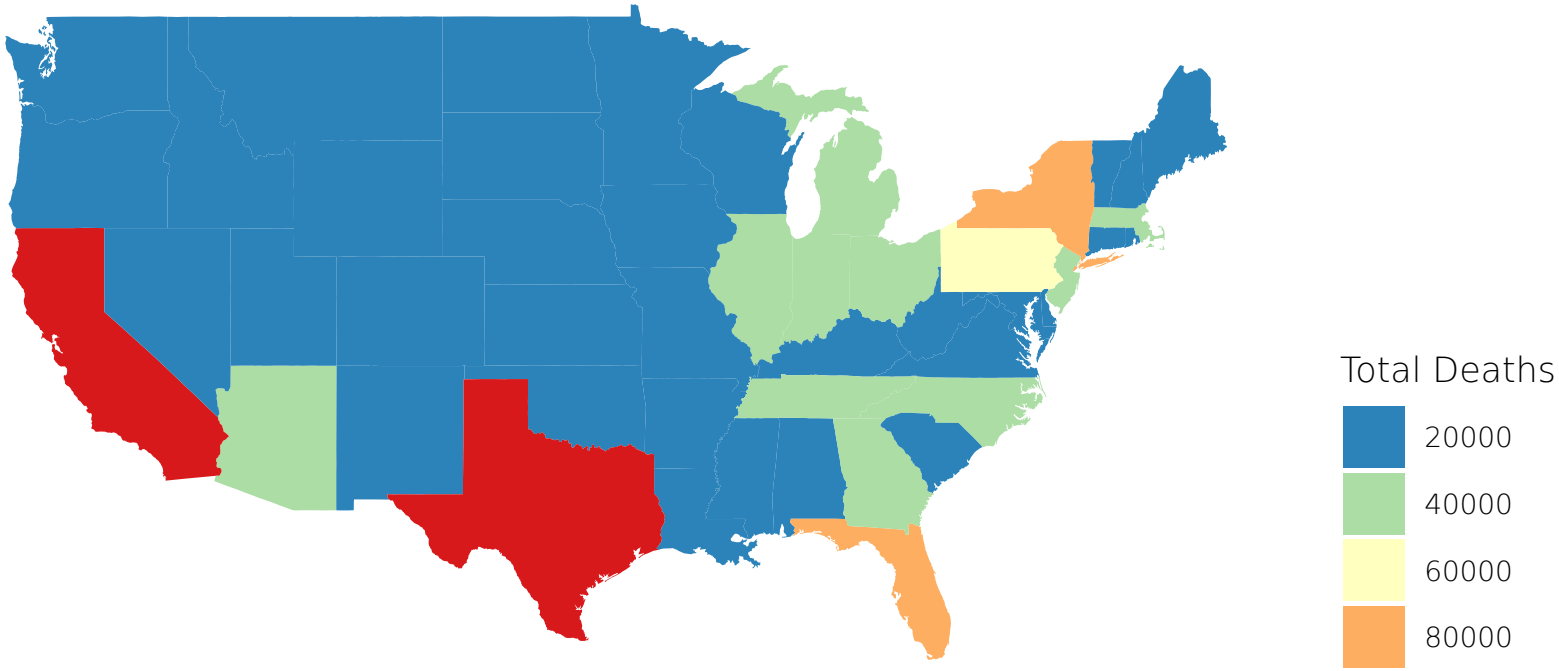
Data Scraping and Cleaning

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

covid <- table_usafacts[[2]]
covid_clean <- covid %>% janitor::clean_names() %>%
  mutate_at(4:6, parse_number) %>% mutate(state = str_to_lower(state))

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

Interactive Plot

```
map1 <- ggplot(covid_data, aes(long, lat, group = group)) + coord_map() + theme_map() +
  geom_polygon_interactive(aes(fill = deaths,
                               tooltip = deaths)) +
  guides(fill=guide_legend(title="7 day average cases")) +
  scale_fill_fermenter(type = "div", palette = "Spectral") +
  theme(legend.position = "right")

ggiraph(code = print(map1))
```

Interactive Plot

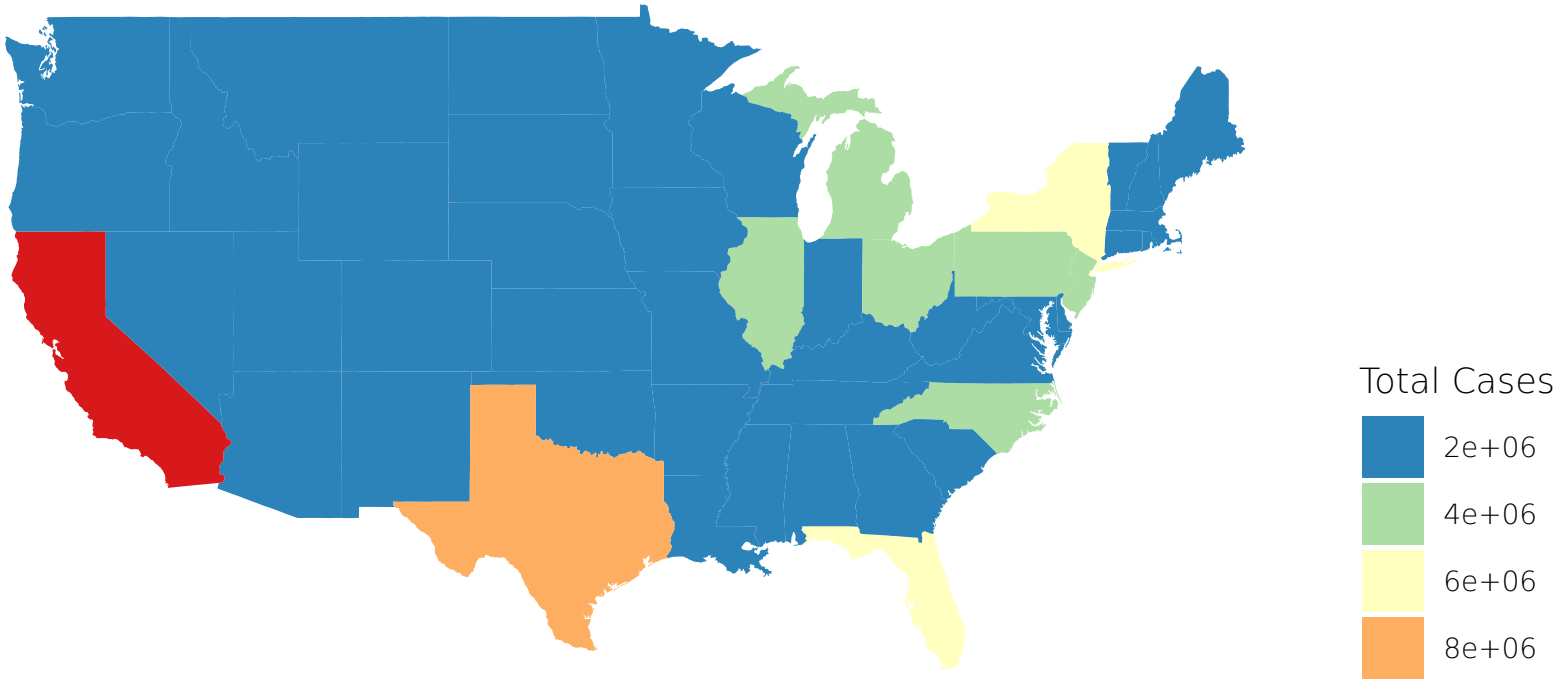


Interactive Plot

```
covid_data$tooltip <- str_c("State = ", str_to_upper(covid_data$region), "\n Cases =",

map2 <- ggplot(covid_data, aes(long, lat)) + coord_map() + theme_map() +
  geom_polygon_interactive(aes(fill = cases, group = group,
                               tooltip = tooltip,
                               data_id = cases)) +
  guides(fill=guide_legend(title="Total Cases")) +
  scale_fill_fermenter(type = "div", palette = "Spectral") +
  theme(legend.position = "right")

ggiraph(code = print(map2),  hover_css = "fill:lightgreen;r:10pt;")
```

Interactive Plot

```
covid_data$tooltip <- str_c("State = ", str_to_upper(covid_data$region), "\n Cases =",

map2 <- ggplot(covid_data, aes(long, lat)) + coord_map() + theme_map() +
  geom_polygon_interactive(aes(fill = cases, group = group,
                               tooltip = tooltip,
                               data_id = cases)) +
  guides(fill=guide_legend(title="Total Cases")) +
  scale_fill_fermenter(type = "div", palette = "Spectral") +
  theme(legend.position = "right")

ggiraph(code = print(map2),  hover_css = "fill:lightgreen;r:10pt;")
```


Total Cases
2e+06
4e+06
6e+06
8e+06

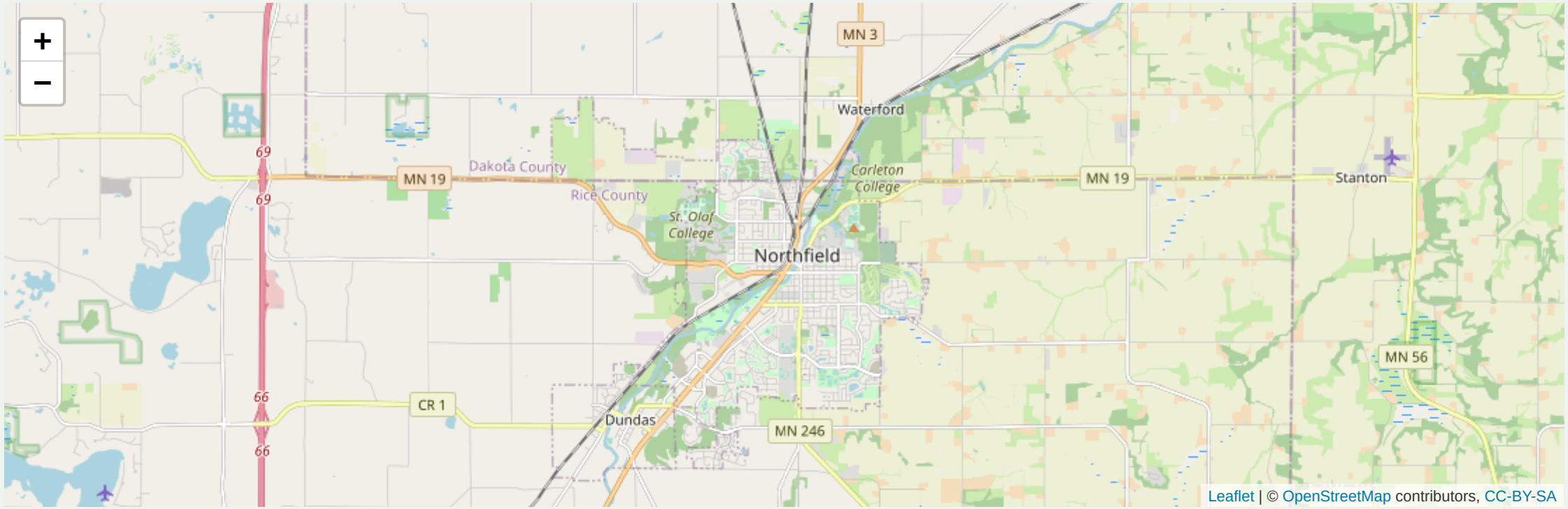
Shiny Implementation

```
# Basic ggiraph using Shiny
ui <- fluidPage(
  girafeOutput("plot")
  )

server <- function(input, output) {

  output$plot <- renderGirafe({
    girafe(ggobj = gg_blahblah )
  })
  })
}
```

Leaflet

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

```
leaflet() %>%
  addTiles() %>%
  setView(lng = -93.1616, lat = 44.4583, zoom = 12)
```



What can you do with `leaflet`?

- Create a color palette thanks to the **colorNumeric()** function.
- Translate a numeric variable to a palette of color. Leaflet offers 3 options:
  - Quantile with colorQuantile
  - Numeric with colorNumeric
  - Bin with colorBin
- Make the background map with `leaflet()`, `addTiles()` and `setView()`.
- Use `addPolygons()` to add the shape of all country/states/county, with a color representing some numerical variable

For more information, click here

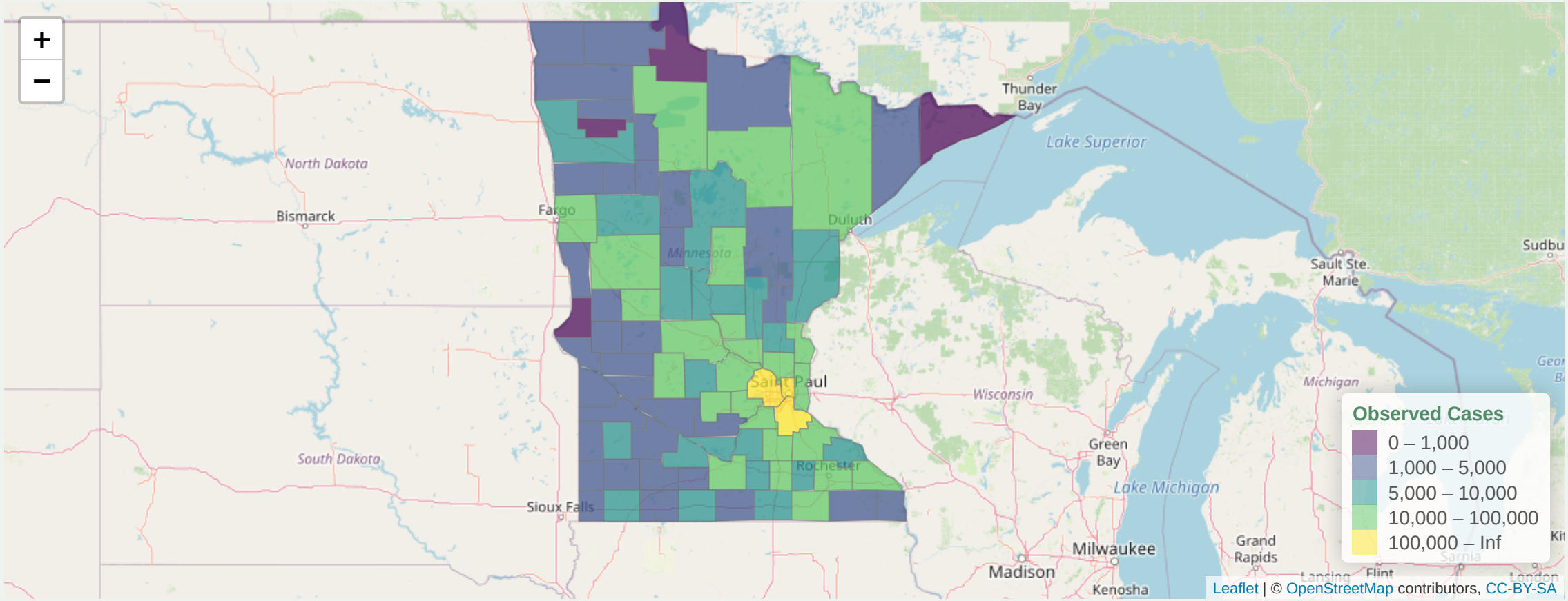
Data Wrangling

```
table_usafacts <- bow(url = "https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/st
  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"))
```

Final plotting code

```
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 = pal2, values = ~cases, opacity = 0.5,
                            title = "Observed Cases",
                            position = "bottomright")
```

Interactive leaflet map



Store leaflet map as html

```
library(htmlwidgets)
saveWidget(l, file="Minnesota.html")
```

Please clone the repository on interactive shiny examples to your local folder. For the remainder of the class, let's go over to in the class activity .Rmd file and practice more examples together.