01 Mini Project: Static Maps

Solveig Senf

This project features three maps (a static and interactive version of the same variable and a single static map of different variable) that display data from the Centers for Disease Control and Prevention (CDC). Data on Covid-19 levels was collected from February 2022 to May 2023 on a county level. This project displays data recorded on May 11, 2023. For the purposes of this state-level project, I will use the average of county-level data for each state to display Covid-19 cases per 100k. Data can be accessed here.

Additionally, this project contains two maps (a static and interactive version) that display the 2024 Presidential Election results. Election data is from the Federal Election Commission (FEC) and can be found at this link.

The interactive maps in this project can be found here (I think…).

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(mdsr)
library(maps)
```

```
Attaching package: 'maps'
```

```
The following object is masked from 'package:purrr':

    map
```

```r
library(viridis)
```

```
Loading required package: viridisLite

Attaching package: 'viridis'

The following object is masked from 'package:maps':

    unemp
```

```r
library(lubridate)
library(leaflet)
library(sf)
```

```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library(RColorBrewer)
```

COVID-19 Data

```r
#data set from the CDC
cdc_data <- read.csv("~/Downloads/01_United_States_COVID-19_Community_Levels_by_County_202502
```

```r
covid_data <- cdc_data |>
  filter(date_updated == "2023-05-11",
         !(state %in% c("Puerto Rico",
                        "American Samoa",
                        "Commonwealth of the Northern Mariana Islands",
                        "United States Virgin Islands",
                        "Guam")))
```

```r
states_polygon <- as_tibble(map_data("state")) |>
  select(region, group, order, lat, long)

states_sf <- read_sf("https://rstudio.github.io/leaflet/json/us-states.geojson") |>
  select(name, geometry)
```

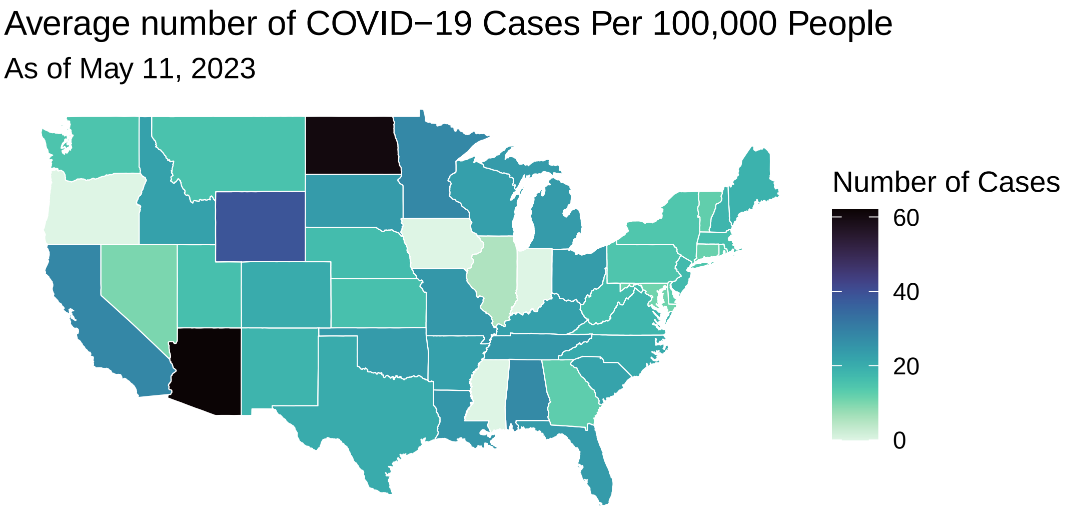
```
#convert county-level data to state-level data
covid_state_level_data <- covid_data |>
  filter(covid_cases_per_100k != "NA") |>
  group_by(state) |>
  summarize(covid_cases_per_100k = mean(covid_cases_per_100k))
```

```
#format state names in all data sets so they match
covid_state_level_data <- covid_state_level_data |>
  mutate(state = str_to_lower(state),
         state = str_replace_all(state, " ", ""),
         state = str_squish(state))

states_sf <- states_sf |>
  mutate(name = str_to_lower(name),
         name = str_replace_all(name, " ", ""))

states_polygon <- states_polygon|>
  mutate(region = str_replace_all(region, " ", ""))
```

```
#join covid data with map data
covid_map <- covid_state_level_data |>
  right_join(states_polygon, by =c("state" = "region"))

covid_map <- covid_map |>
  right_join(states_sf, by =c("state" = "name"))
```

Static Map #1

```
covid_map |>
  ggplot(mapping = aes(x = long, y = lat,
                       group = group)) +
  geom_polygon(aes(fill = covid_cases_per_100k), color = "white", linewidth = 0.2) +
  labs(fill = "Number of Cases",
       x = "",
       y = "",
       title = "Average number of COVID-19 Cases Per 100,000 People",
       subtitle = "As of May 11, 2023") +
  coord_map() +
  theme_void() +
  scale_fill_viridis(option = "mako", direction = -1)
```



Average number of COVID−19 Cases Per 100,000 People

As of May 11, 2023

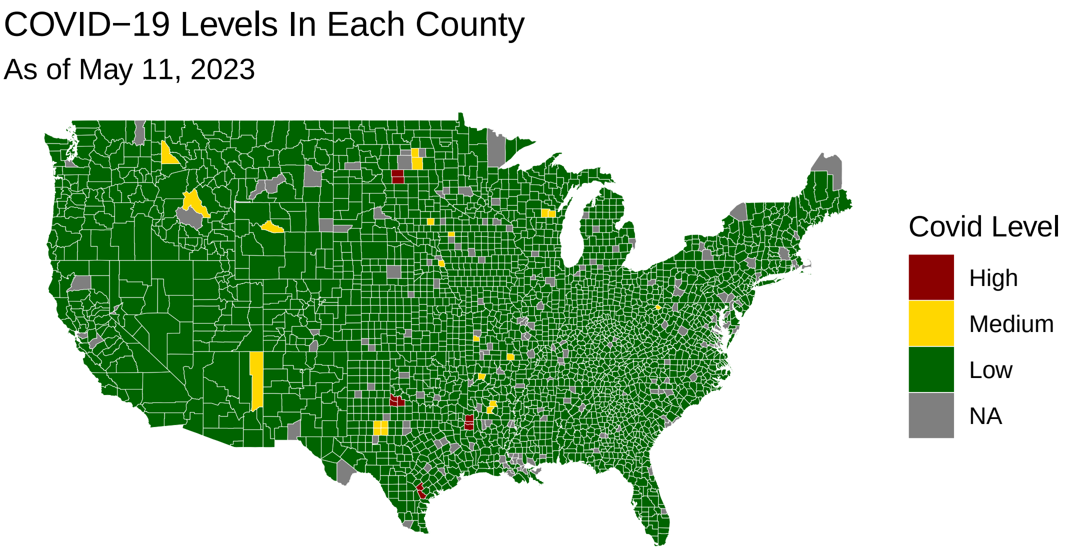
Static Map #2

```
county_map <- map_data("county") |>
  mutate(subregion = str_to_lower(subregion),
         subregion = str_replace_all(subregion, " ", ""),
         subregion = str_squish(subregion)) |>
  mutate(region = str_to_lower(region),
         region = str_replace_all(region, " ", ""))

#fixing odd formating for county names in CDC data
covid_data_clean <- covid_data |>
  select(county, state, covid.19_community_level, covid_cases_per_100k) |>
  mutate(state = str_to_lower(state),
         state = str_replace_all(state, " ", ""),
         state = str_squish(state)) |>
  mutate(county = str_to_lower(county),
         county = str_replace_all(county, " ", ""),
         county = str_squish(county),
         county = str_replace_all(county, "county", ""),
         county = str_replace_all(county, "st.", ""),
         county = str_replace_all(county, "city", ""),
         county = str_replace_all(county, "muni", ""),
         county = str_replace_all(county, "parish", ""))
```

```
covid_county_level_data <- covid_data_clean |>
  right_join(county_map, by =c("county" = "subregion", "state" = "region")) |>
  mutate(covid.19_community_level = fct_relevel(covid.19_community_level, "High", "Medium", "
```

```
Warning in right_join(covid_data_clean, county_map, by = c(county = "subregion", : Detected a
i Row 1 of `x` matches multiple rows in `y`.
i Row 79510 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
covid_county_level_data |>
  ggplot(mapping = aes(x = long, y = lat,
                       group = group)) +
  geom_polygon(aes(fill = covid.19_community_level), color = "white", linewidth = 0.05) +
  labs(fill = "Covid Level",
       x = "",
       y = "",
       title = "COVID-19 Levels In Each County",
       subtitle = "As of May 11, 2023") +
  scale_fill_manual(values = c("High" = "darkred", "Medium" = "gold", "Low" = "darkgreen", "N
  coord_map() +
  theme_void()
```



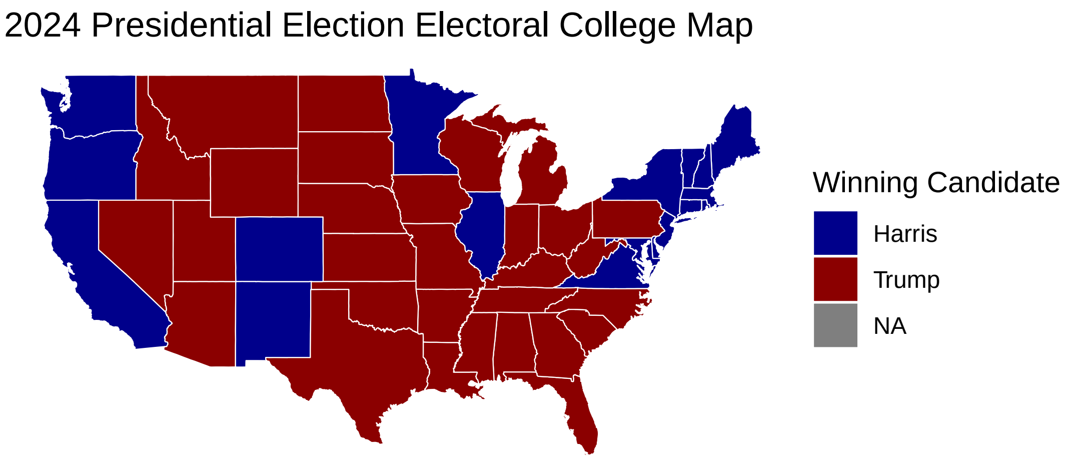
Election Data

Static Map #3

```
#data from the federal election commission
election_data <- read.csv("~/Downloads/2024presgeresults.csv")

election_data <- election_data |>
  select(STATE, ELECTORAL.VOTE..TRUMP..R., ELECTORAL.VOTE..HARRIS..D., HARRIS, TRUMP) |>
  rename(state = STATE,
         Trump = ELECTORAL.VOTE..TRUMP..R.,
         Harris = ELECTORAL.VOTE..HARRIS..D.,
         votes_harris = HARRIS,
         votes_trump = TRUMP) |>
  pivot_longer(cols = c(Trump, Harris),
               names_to ="candidate_won",
               values_to = "electoral_votes") |>
  filter(electoral_votes != "NA") |> #remove observations that show the loosing candidate in
  slice(-30, -20)
#slice to remove rows 30 and 20 which are second observations of maine and nebraska due to th
#the observation recording the winning candidate is kept.

election_data <- election_data |>
  mutate(state = str_to_lower(state),
         state = str_replace_all(state, " ", ""),
         state = str_squish(state)) |>
  mutate(votes_harris = str_squish(votes_harris),
         votes_trump = str_squish(votes_trump))
```

```
electoral_map <- election_data |>
  right_join(states_polygon, by =c("state" = "region"))

electoral_map <- electoral_map |>
  right_join(states_sf, by =c("state" = "name"))
```

```
electoral_map |>
  ggplot(mapping = aes(x = long, y = lat,
                       group = group)) +
  geom_polygon(aes(fill = candidate_won), color = "white", linewidth = 0.2) +
  labs(fill = "Winning Candidate",
       x = "",
       y = "",
       title = "2024 Presidential Election Electoral College Map") +
  coord_map() +
  scale_fill_manual(values = c("Harris" = "darkblue", "Trump" = "darkred")) +
  theme_void()
```


2024 Presidential Election Electoral College Map
Winning Candidate
Harris
Trump
NA