

L05 Maps

Data Visualization (STAT 302)

Contents

Overview	1
Datasets	1
Exercises	1
Challenge(s)	6

Overview

The goal of this lab is to explore various ways of building maps with `ggplot2`.

Challenges are not mandatory for students to complete. We highly recommend students attempt them though. *We would expect graduate students to attempt the challenges.*

Datasets

We'll be using data the `US_income.rda` dataset which is already in the `/data` subdirectory in our `data_vis_labs` project. You'll also be downloading your own data to build maps.

```
# Load package(s)
library(tidyverse)
library(raster)
library(sf)
library(viridis)
library(janitor)
library(rgeos)
library(statebins)
library(elevatr)

# Load dataset(s)
load('data/US_income.rda')
```

Exercises

Exercise 1

Plot 1 Make a county map of a US state using `geom_polygon()`. Maybe use your home state (mine is below) or a favorite state. Please do not use the the state in example provided in the book.

Optional: Consider adding a major cities (or your home town).

Hints:

- See section 6.1 in our book.

- Void theme

```
ca_counties <- map_data("county", "california") %>%
  dplyr::select(lon = long, lat, group, id = subregion)

ggplot(ca_counties, aes(lon, lat, group = group)) +
  geom_polygon(fill = "white", color = "grey50") +
  coord_quickmap() +
  theme_void()
```



Plot 2 Now use `geom_sf()` instead. You'll need to get the data to make this happen and either the `tigris` (github page) or `raster` packages can help you so this. `raster`'s `getData()` could be useful.

```
us_data <- getData(name = 'GADM', country = 'USA', level = 2) %>%
  st_as_sf() %>%
  clean_names() %>%
  filter(name_1 == 'California')

ggplot(data = us_data) +
  geom_sf(aes(geometry = geometry), fill = 'white', color = 'grey50') +
  coord_sf() +
  theme_void()
```



Exercise 2

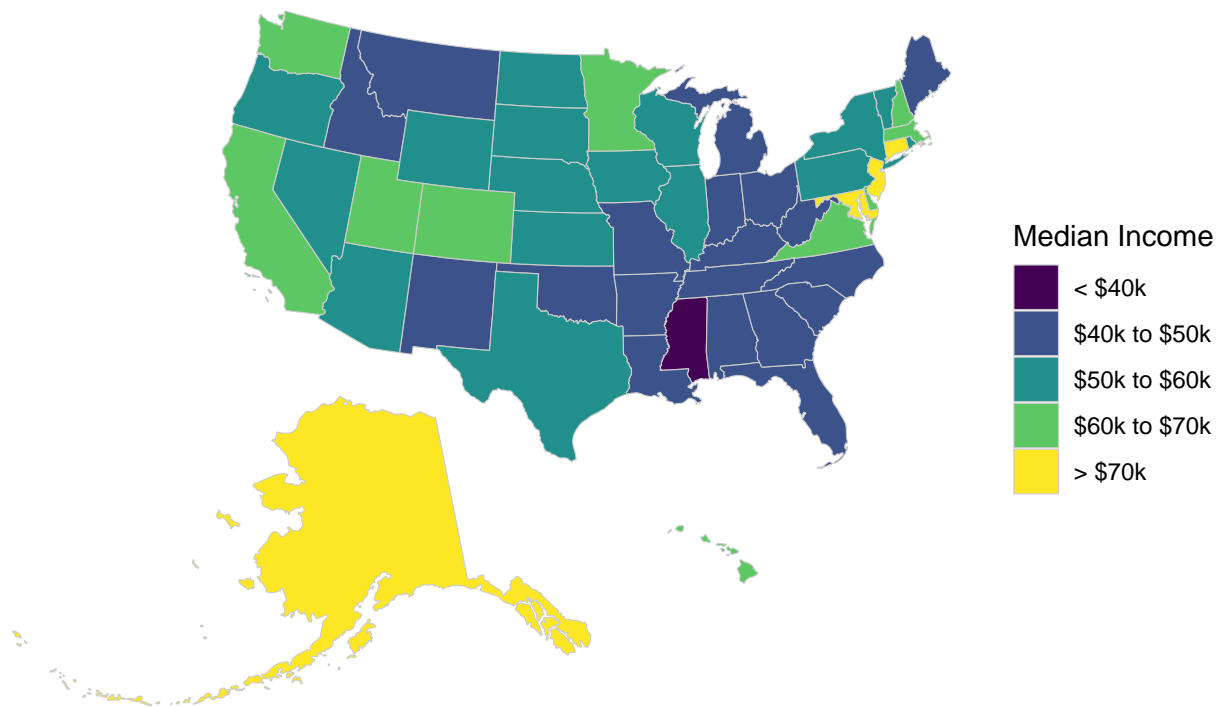
Using `US_income` dataset, recreate the following graphics as precisely as possible.

```
# Setting income levels
US_income <- mutate(
  US_income,
  income_bins = cut(
    ifelse(is.na(median_income), 25000, median_income),
    breaks = c(0, 40000, 50000, 60000, 70000, 80000),
    labels = c("< $40k", "$40k to $50k",
               "$50k to $60k", "$60k to $70k", "> $70k"),
    right = FALSE
  )
)
```

Plot 1 *Hints:*

- `geom_sf()` — boundary color is "grey80" and size is 0.2
- `viridis` package (`discrete = TRUE` in `scale_*` function)
- Void theme

```
ggplot(data = US_income) +
  geom_sf(mapping = aes(fill = income_bins),
          color = 'grey80', size = 0.2) +
  theme_void() +
  labs(fill = 'Median Income') +
  scale_fill_viridis(discrete = T)
```



Plot 2 *Hints:*

- Download developmental version of `statebins` package
- `geom_statebins()`
- `viridis` package (`discrete = TRUE` in `scale_*` function)
- `Statebins` theme

```
ggplot(data = US_income) +
  geom_statebins(mapping = aes(fill = income_bins, state = name),
                 color = 'grey80', size = 0.2) +
  theme_statebins() +
  labs(fill = 'Median\nIncome') +
  scale_fill_viridis(discrete = T)
```



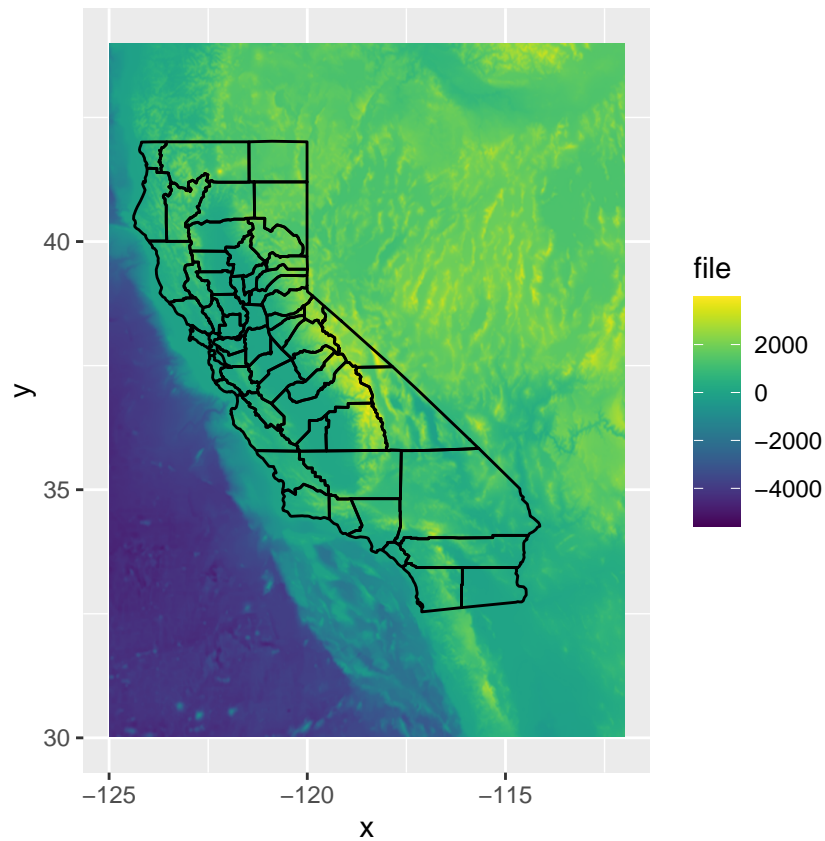
Median Income < \$40k \$40k to \$50k \$50k to \$60k \$60k to \$70k > \$70k

Exercise 3

Pick any city or foreign country to build a map for. You can dress it up or make it as basic as you want. Also welcome to try building a graphic like that depicted at the end of section 6.5 — use a different region though.

```
set.seed(123)
prj_dd <- "+init=EPSG:4326"
elevation_raster <- get_elev_raster(ca_counties, prj = prj_dd, z = 5)
elevation_points <- rasterToPoints(elevation_raster, spatial = T)
elevation_df <- data.frame(elevation_points)
names(elevation_df)[1] <- 'file'

ggplot() +
  coord_quickmap() +
  geom_raster(data = elevation_df, aes(x = x, y = y, fill = file)) +
  geom_polygon(data = ca_counties, aes(x = lon, y = lat, group = group), fill = NA,
    color = 'black') +
  xlim(-125, -112) +
  ylim(30, 44) +
  scale_fill_viridis()
```



Challenge(s)

Using the `tidycensus` package and few others, try to create a map like below using these directions. Try using a different geographical area and a different variable from the ACS.

Hint: You'll need a developmental version of `mapview` package. To get this you need to run the following code in your console

```
remotes::install_github("r-spatial/mapview")
```