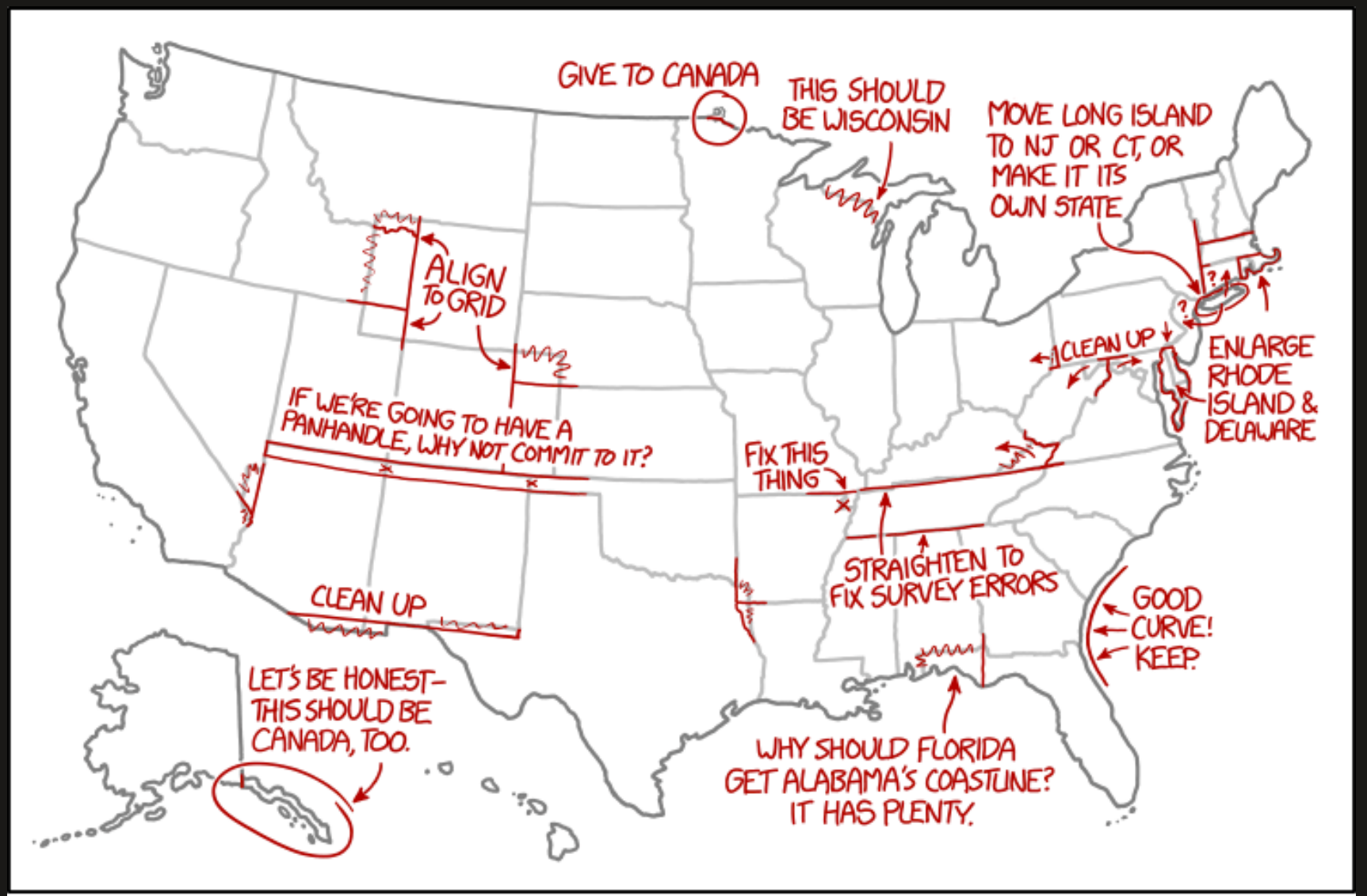


IT WAS SCARY WHEN THE GRAPHIC DESIGNERS SEIZED CONTROL OF THE COUNTRY, BUT IT TURNED OUT THEY JUST WANTED TO FIX SOME THINGS ABOUT THE STATE BORDERS THAT HAD ALWAYS BOTHERED THEM.

Week 8: Maps & geospatial data

EMSE 4197 | John Paul Helveston | March 11, 2020

Quiz 3

45:00

Today's data

```
milk_production  <- read_csv(here::here('data', 'milk_production.csv'))
us_coffee_shops  <- read_csv(here::here('data', 'us_coffee_shops.csv'))
```

New package:

```
install.packages('maps')
install.packages('sf')
install.packages('rgeos')
install.packages('rnaturalearth')
devtools::install_github("ropensci/rnaturalearthhires")
devtools::install_github("ropensci/rnaturalearthdata")
```

Maps & geospatial data

1. Plotting maps
2. Adding data to maps
3. Projections

Maps & geospatial data

1. Plotting maps
2. Adding data to maps
3. Projections

How to make a map

Step 1: Load a shape file

a. Use a library

b. Read in a shape file

Step 2: Plot the shape file

a. `geom_polygon()`

b. `geom_sf()`

Polygon maps

Get the "World" shape file

```
library(ggplot2)

world <- map_data("world")
head(world)
```

```
##        long      lat group order region subregion
## 1 -69.89912 12.45200     1     1  Aruba      <NA>
## 2 -69.89571 12.42300     1     2  Aruba      <NA>
## 3 -69.94219 12.43853     1     3  Aruba      <NA>
## 4 -70.00415 12.50049     1     4  Aruba      <NA>
## 5 -70.06612 12.54697     1     5  Aruba      <NA>
## 6 -70.05088 12.59707     1     6  Aruba      <NA>
```

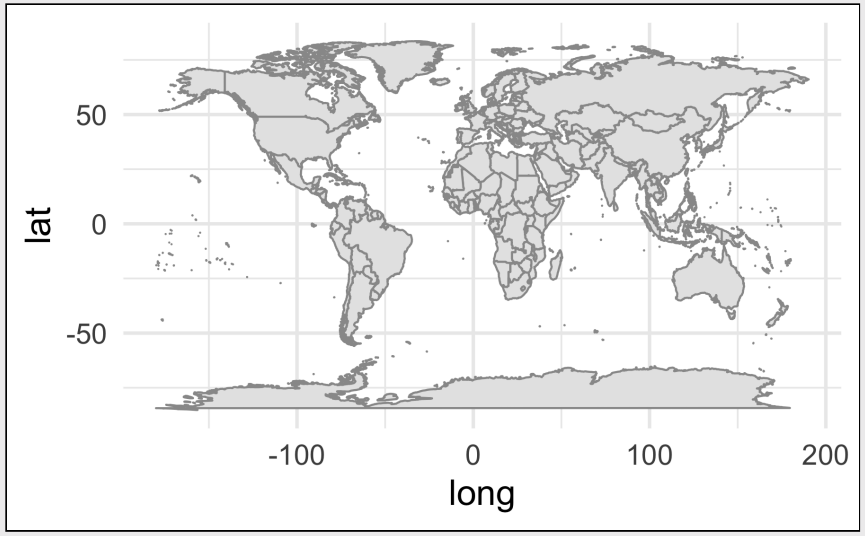
Polygon maps

Get the "World" shape file

```
library(ggplot2)

world <- map_data("world")
```

Make the plot with `geom_polygon()`

```
ggplot(world) +
    geom_polygon(aes(x = long, y = lat, group = group),
                 fill = "grey90", color = "grey60")
```



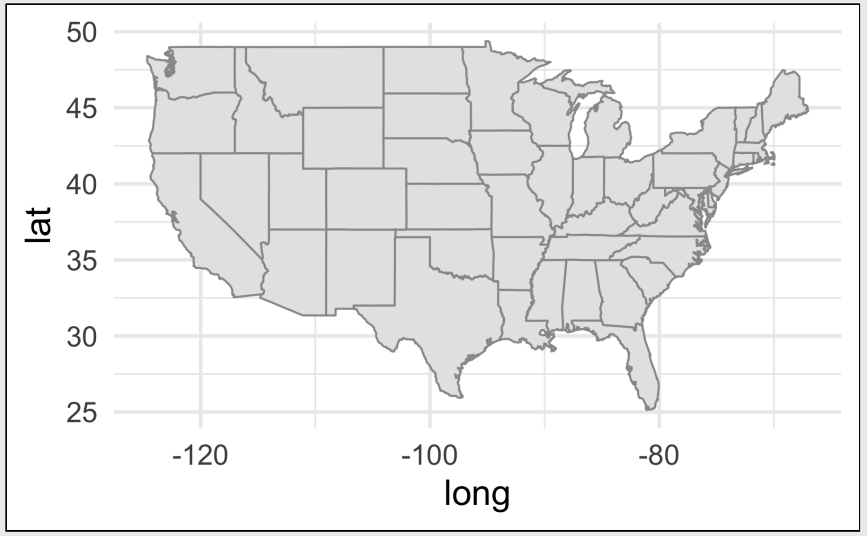
Polygon maps

Get the "US States" shape file

```
library(ggplot2)

us_states <- map_data("state")
```

Make the plot with `geom_polygon()`

```
ggplot(us_states) +
    geom_polygon(aes(x = long, y = lat, group = group),
                 fill = "grey90", color = "grey60")
```



Simple Features (sf) maps

Library data from Natural Earth

```
library(rnaturalearth)
library(rnaturalearthdata)

world <- ne_countries(scale = "medium",
                      returnclass = "sf")

world %>%
    select(name, geometry) %>%
    head()
```

```
## Simple feature collection with 6 features and 1 field
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -70.06611 ymin: -18.01973 xmax: 74.89131
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
##          name                          geometry
## 0       Aruba MULTIPOLYGON (((-69.89912 1...
## 1 Afghanistan MULTIPOLYGON (((74.89131 37...
## 2      Angola MULTIPOLYGON (((14.19082 -5...
## 3    Anguilla MULTIPOLYGON (((-63.00122 1...
## 4     Albania MULTIPOLYGON (((20.06396 42...
```

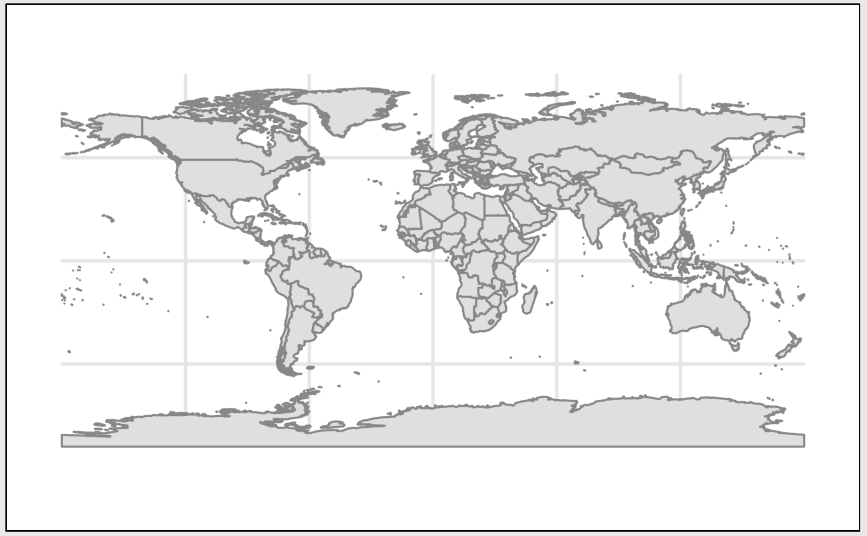
Simple Features (sf) maps

Get the "World" shape file

```
library(rnaturalearth)
library(rnaturalearthdata)

world <- ne_countries(scale = "medium",
                      returnclass = "sf")
```

Make the plot with `geom_sf()`

```
library(sf)

ggplot(data = world) +
    geom_sf(fill = "grey90", color = "grey60")
```



Simple Features (sf) maps

Get the "US States" shape file

```
library(rnaturalearth)
library(rnaturalearthdata)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf')
```

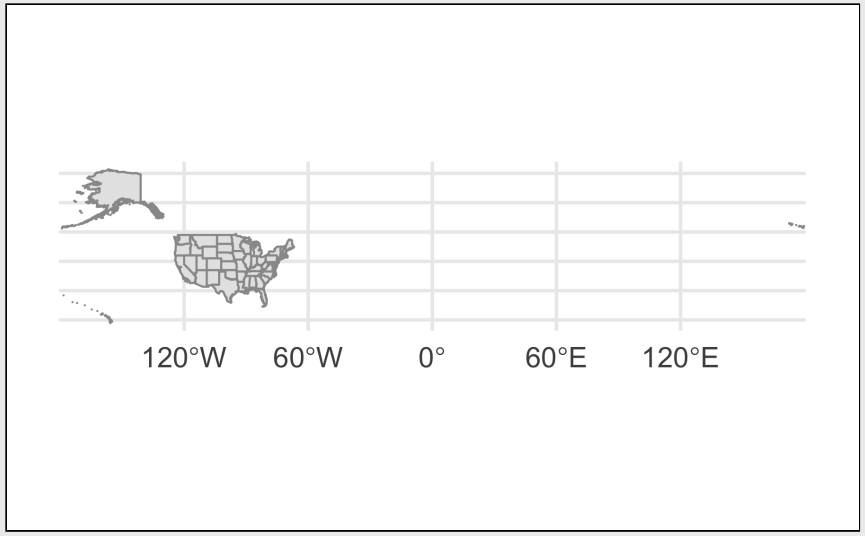
Simple Features (sf) maps

Get the "US States" shape file

```
library(rnaturalearth)
library(rnaturalearthdata)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf')
```

Make the plot with `geom_sf()`

```
library(sf)

ggplot(data = us_states) +
    geom_sf(fill = "grey90", color = "grey60")
```



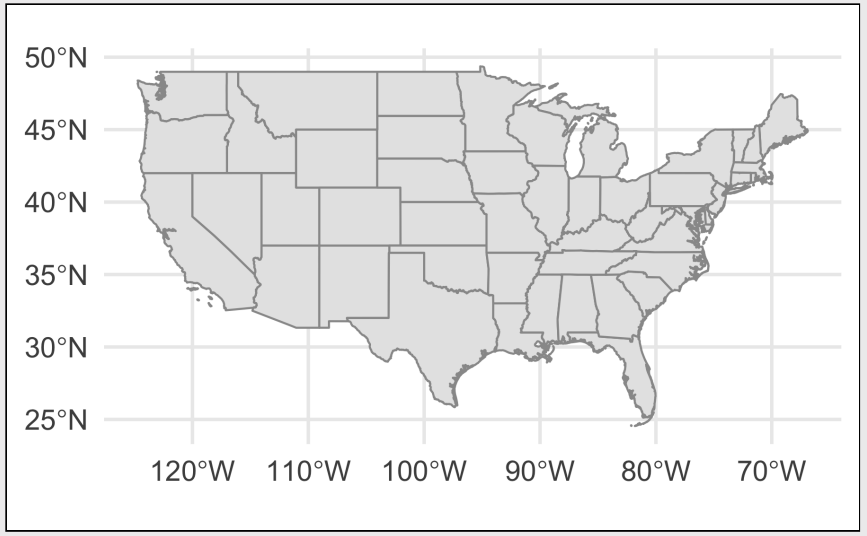
Simple Features (sf) maps

Get the **Continental** "US States" shape file

```
library(rnaturalearth)
library(rnaturalearthdata)

us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))
```

Make the plot with `geom_sf()`

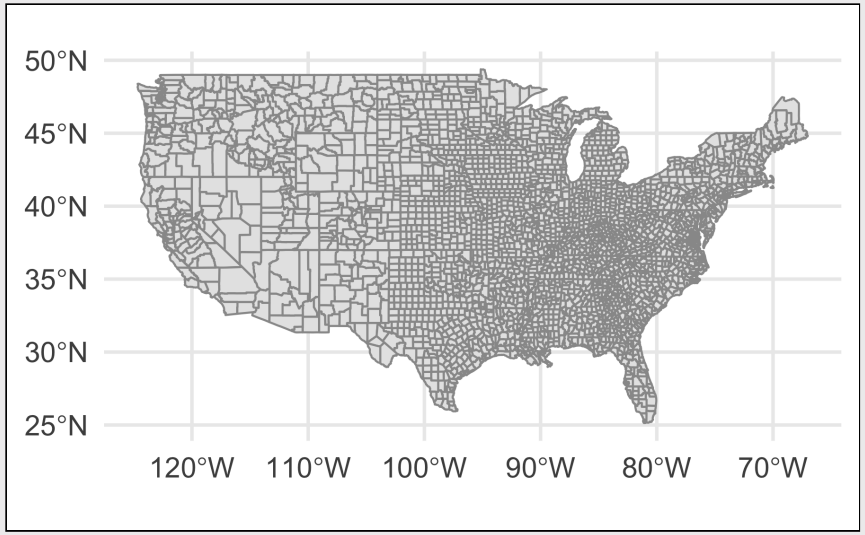
```
library(sf)

ggplot(data = us_states_cont) +
    geom_sf(fill = "grey90", color = "grey60")
```



The `maps` package

Includes data on:

- World: world, world.cities, lakes
- US: states, county, state, usa
- France: france
- Italy: italy
- New zealand: nz



Example:

```
library(maps)

us_counties <- st_as_sf(
    map("county", plot = FALSE, fill = TRUE))

ggplot(data = us_counties) +
    geom_sf(fill = 'grey90', color = 'grey60')
```

Simple Features (sf) maps: `st_read()`

Read in the "World" shape file from Natural Earth

```
library(sf)

world <- st_read(here::here(
    'data', 'natural_earth_countries',
    'ne_50m_admin_0_countries.shp')) %>%
    clean_names()
```

```
## Reading layer `ne_50m_admin_0_countries' from data source `/Use
## Simple feature collection with 241 features and 94 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -180 ymin: -89.99893 xmax: 180 ymax: 83.5
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

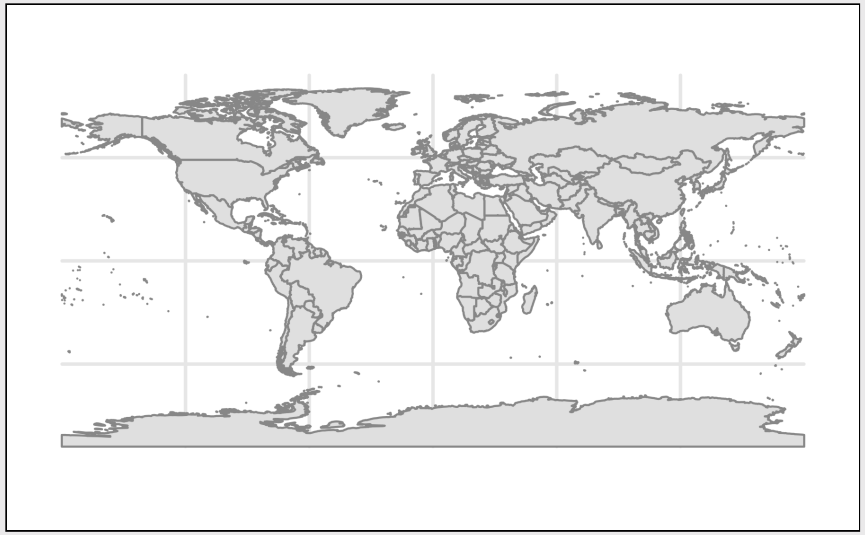
Simple Features (sf) maps: `st_read()`

Read in the "World" shape file

```
library(sf)

world <- st_read(here::here(
    'data', 'natural_earth_countries',
    'ne_50m_admin_0_countries.shp')) %>%
    clean_names()
```

```
## Reading layer `ne_50m_admin_0_countries' from data source `/Use
## Simple feature collection with 241 features and 94 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -180 ymin: -89.99893 xmax: 180 ymax: 83.5
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

```
ggplot(data = world) +
    geom_sf(fill = "grey90", color = "grey60")
```



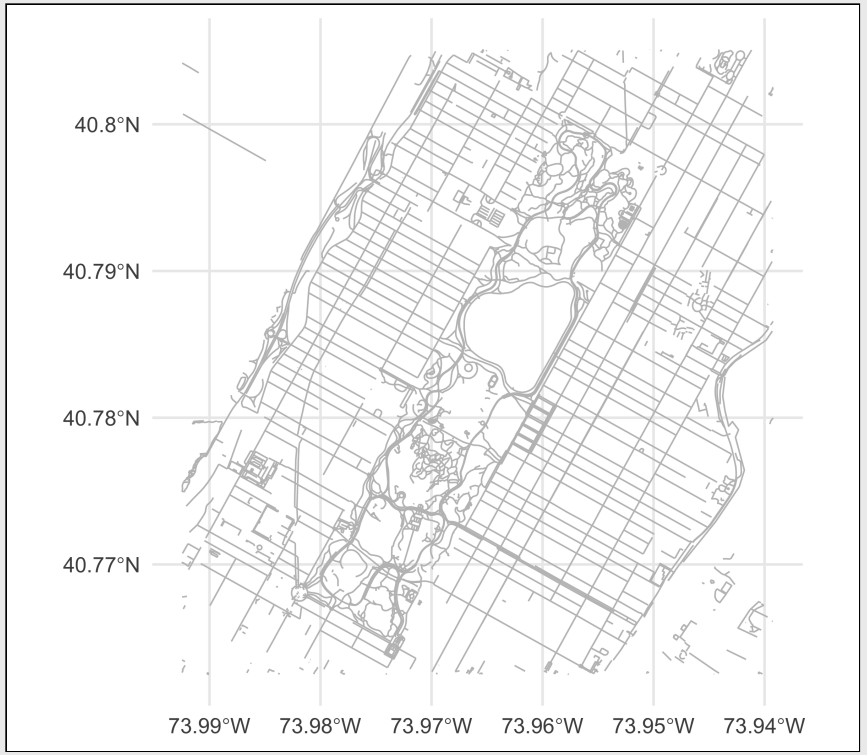
Simple Features (sf) maps: `st_read()`

Read in the "Central Park" shape file [source]

```
library(sf)

central_park <- st_read(here::here(
    'data', 'central_park', 'CentralPark.shp'))
```

```
## Reading layer `CentralPark' from data source `/Users/jhelvy/gh/
## Simple feature collection with 2550 features and 6 fields
## geometry type:  LINESTRING
## dimension:      XY
## bbox:           xmin: -73.99249 ymin: 40.7625 xmax: -73.93922 y
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

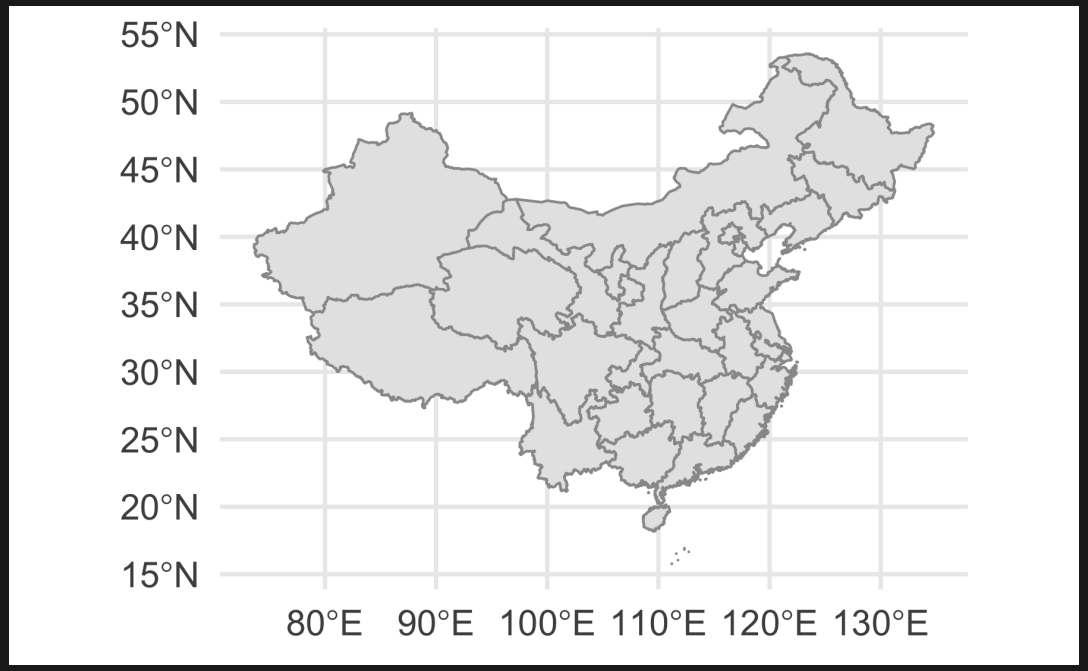
```
ggplot(data = central_park) +
    geom_sf(color = 'grey75')
```

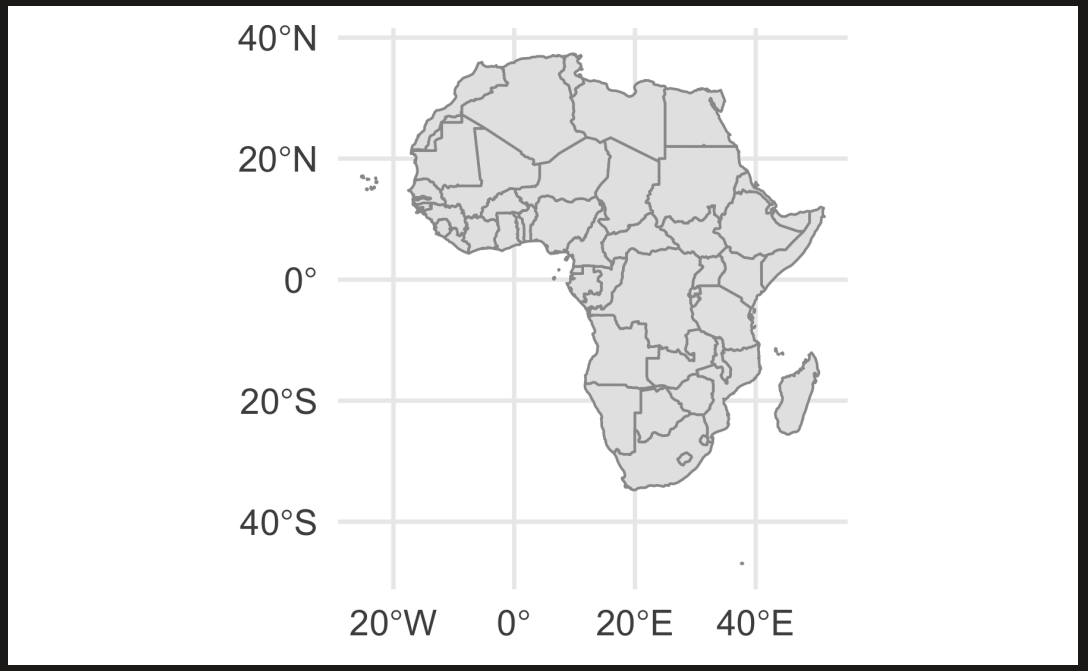


Your turn

10:00

Use the **rnaturalearth** library to extract and plot the shape files for China and Africa:

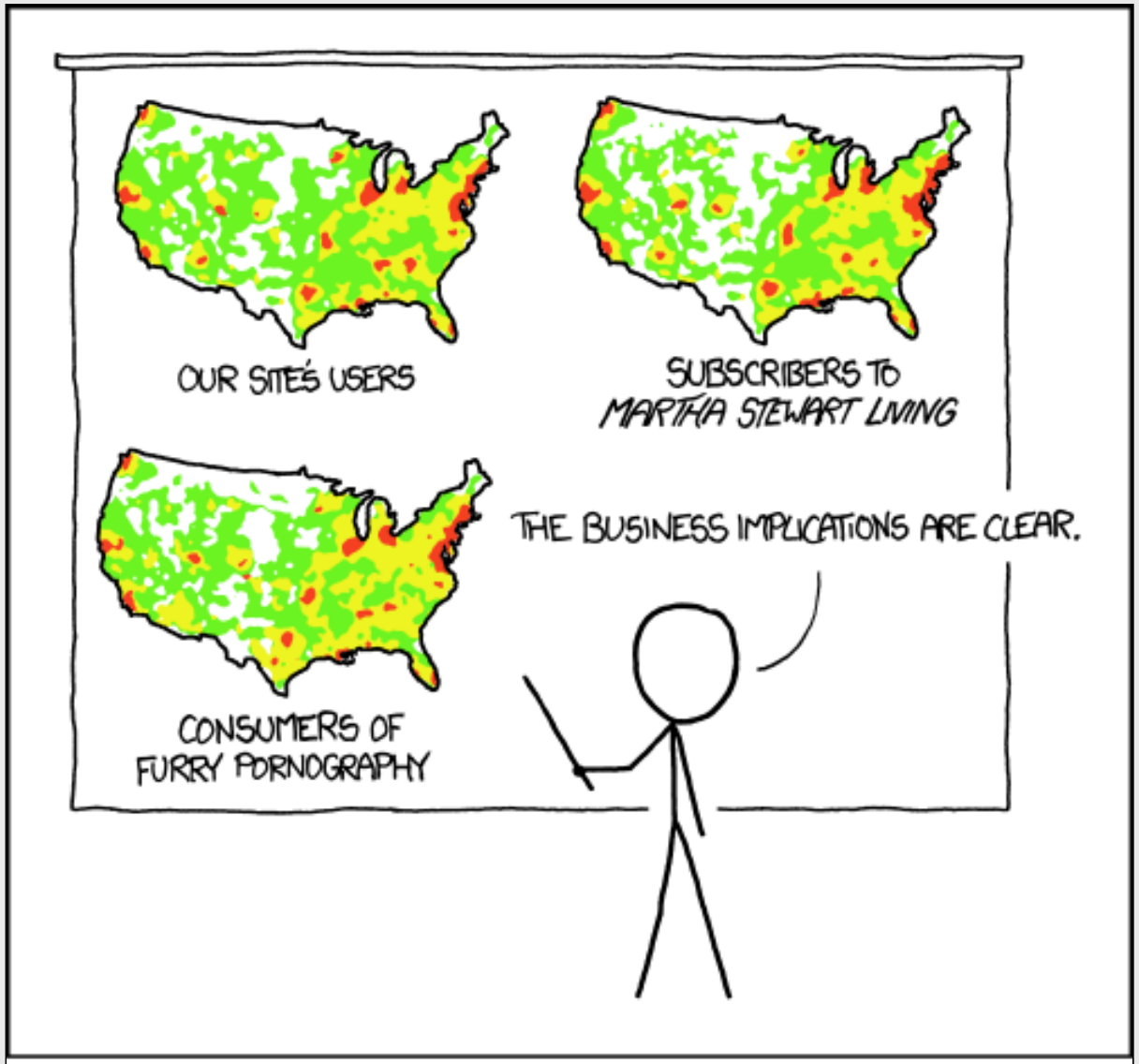




Maps & geospatial data

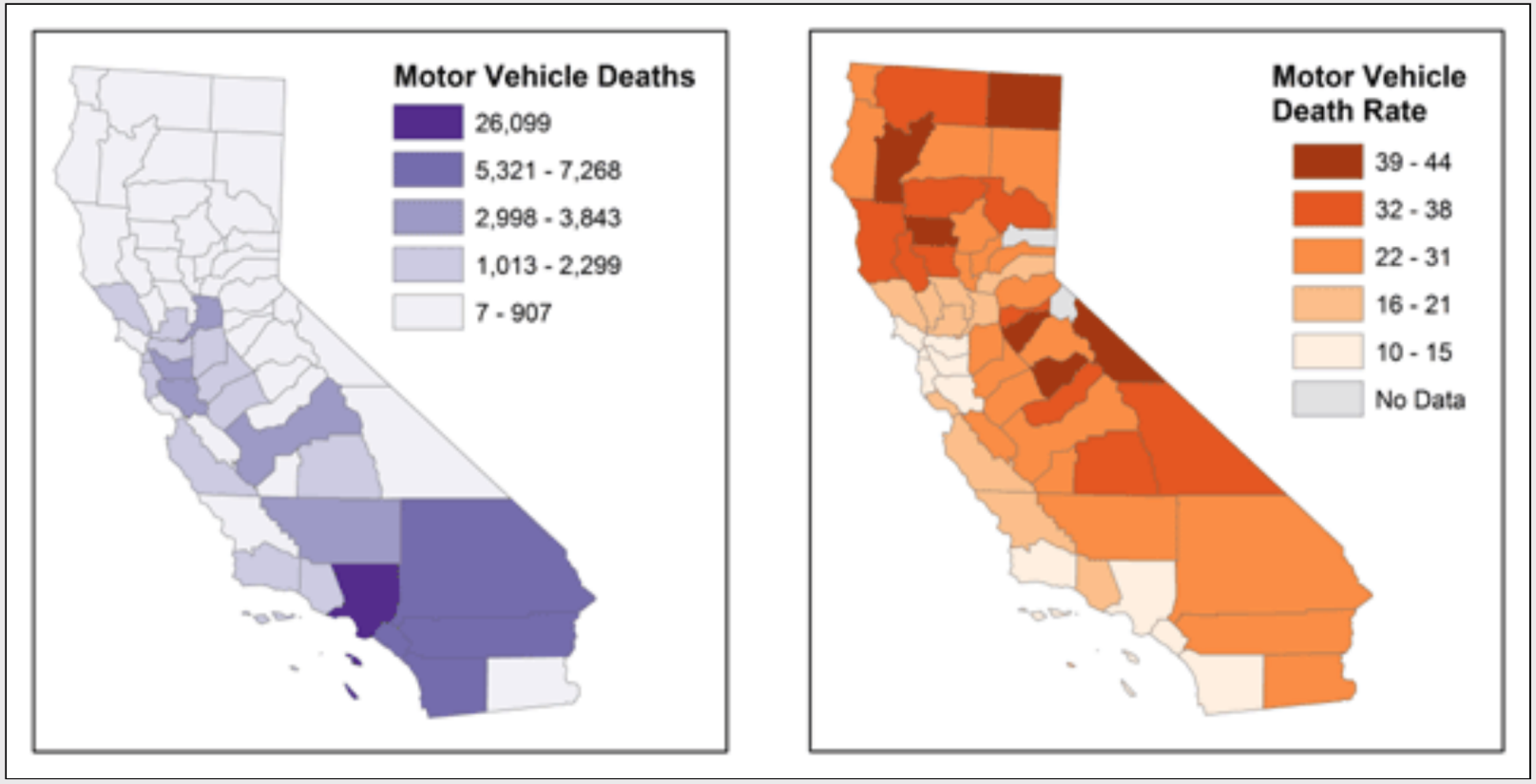
1. Plotting maps
2. Adding data to maps
3. Projections

Choropleth maps

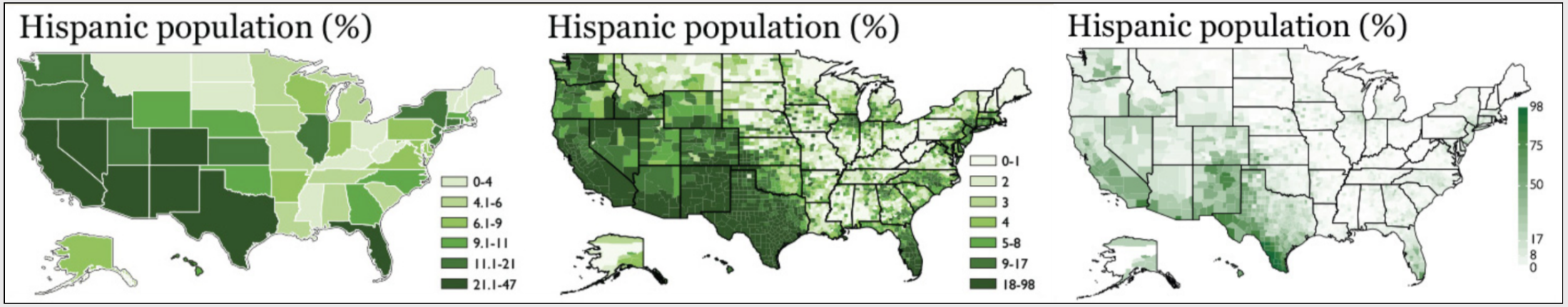


PET PEEVE #208:
GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS

Number of events != Number of vents **per capita**

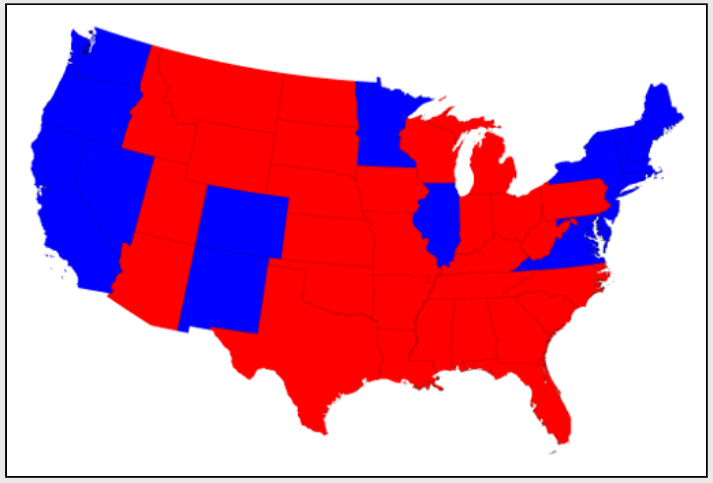


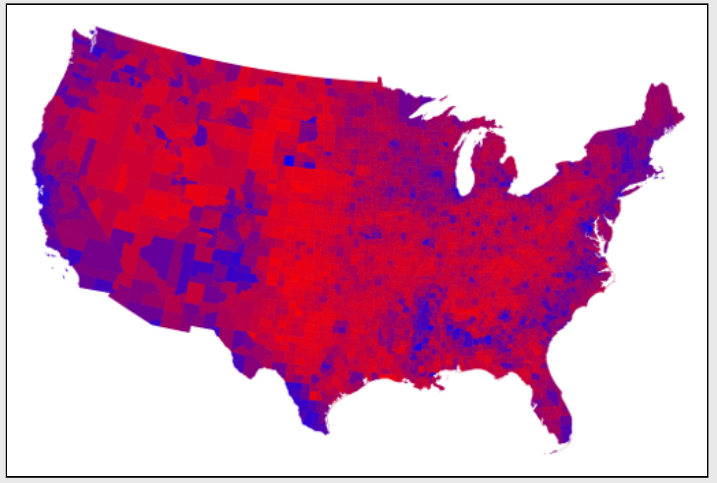
Choropleth maps are easily misleading

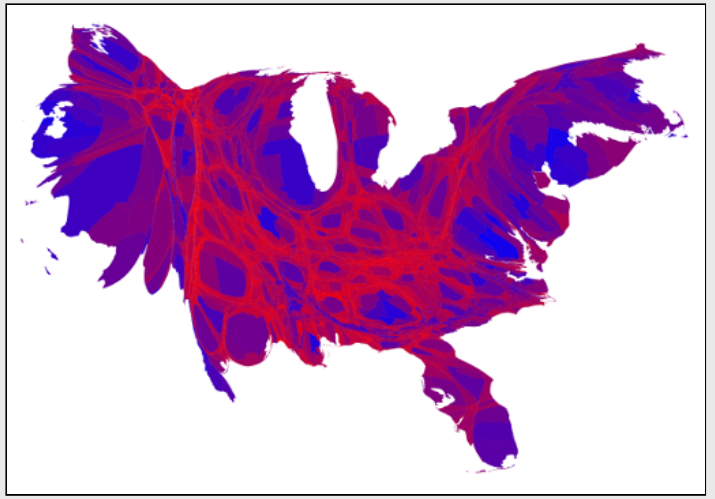


Choropleth maps are easily misleading

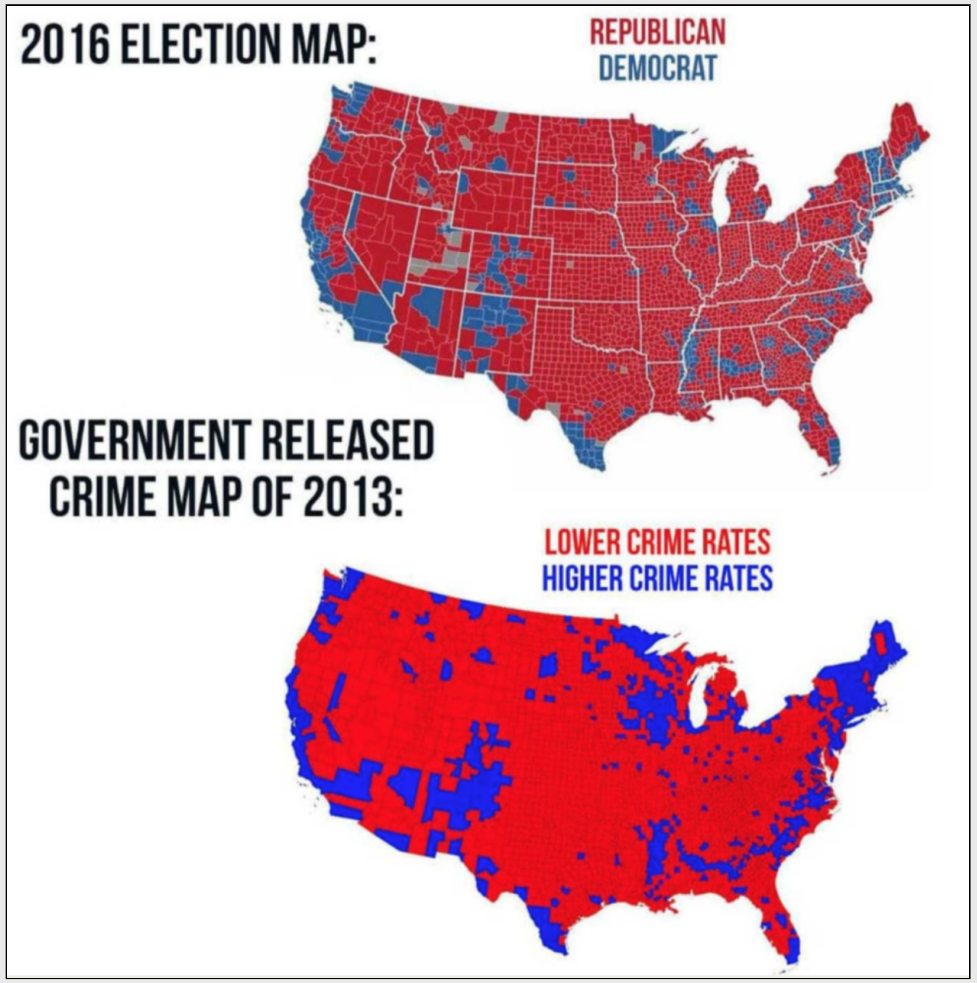
Election maps from: http://www-personal.umich.edu/~mejn/election/2016/







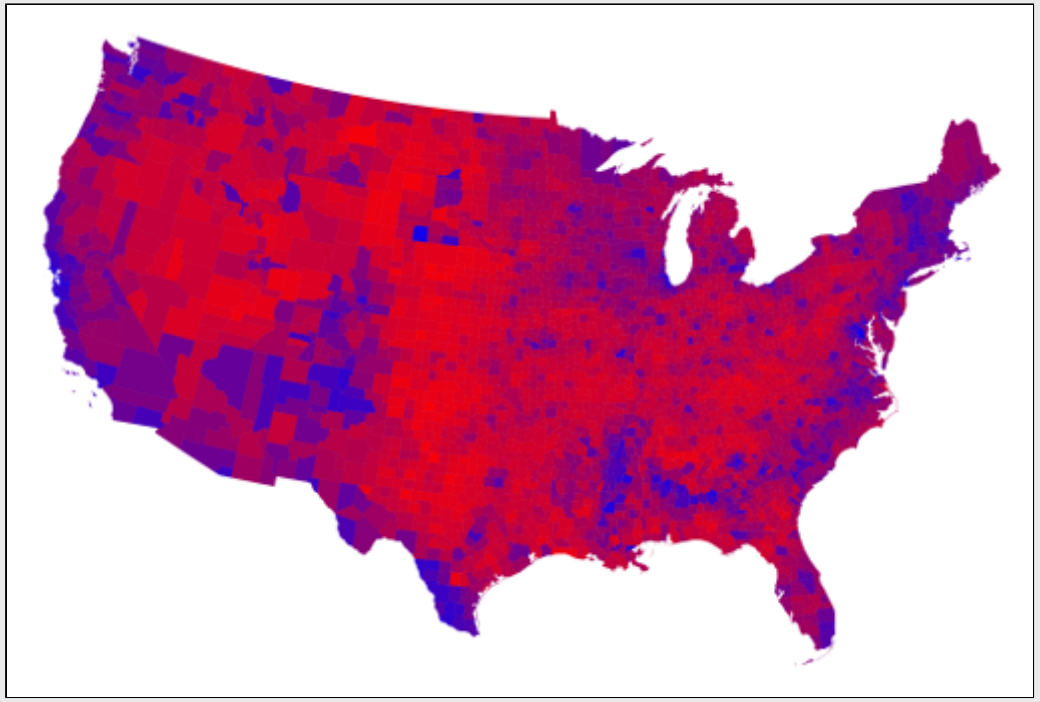
Actual fakenews in dumb memes



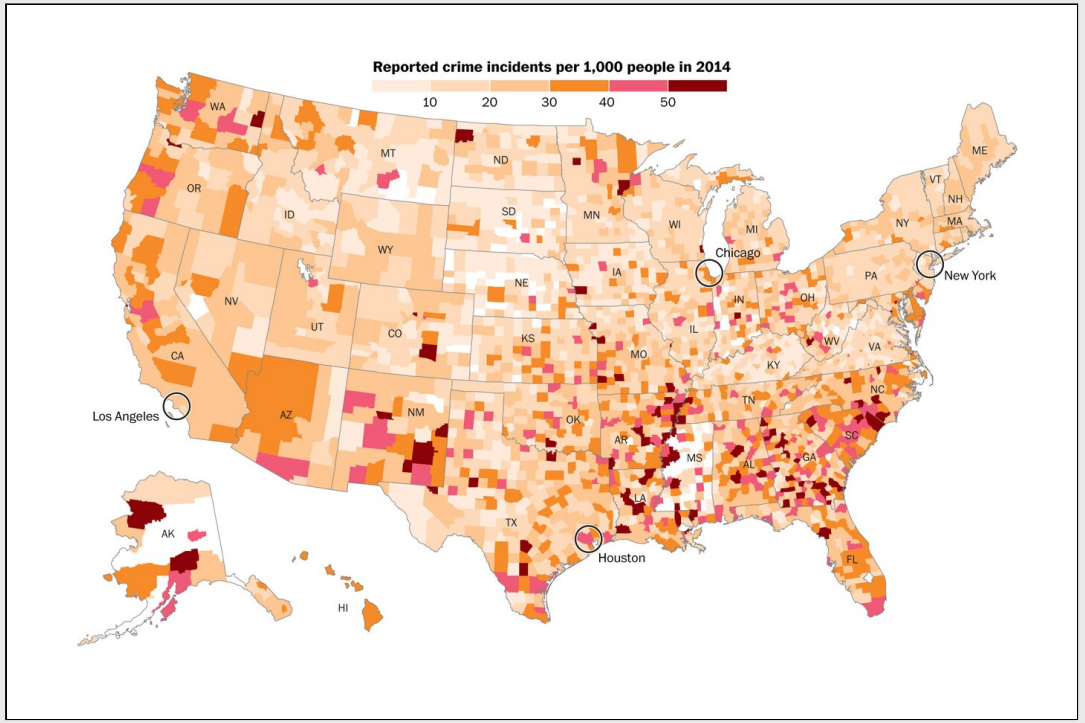
Bottom map is actually the 2012 election map, found here

(here is what actual crime rates look like)

2016 Election map [source]

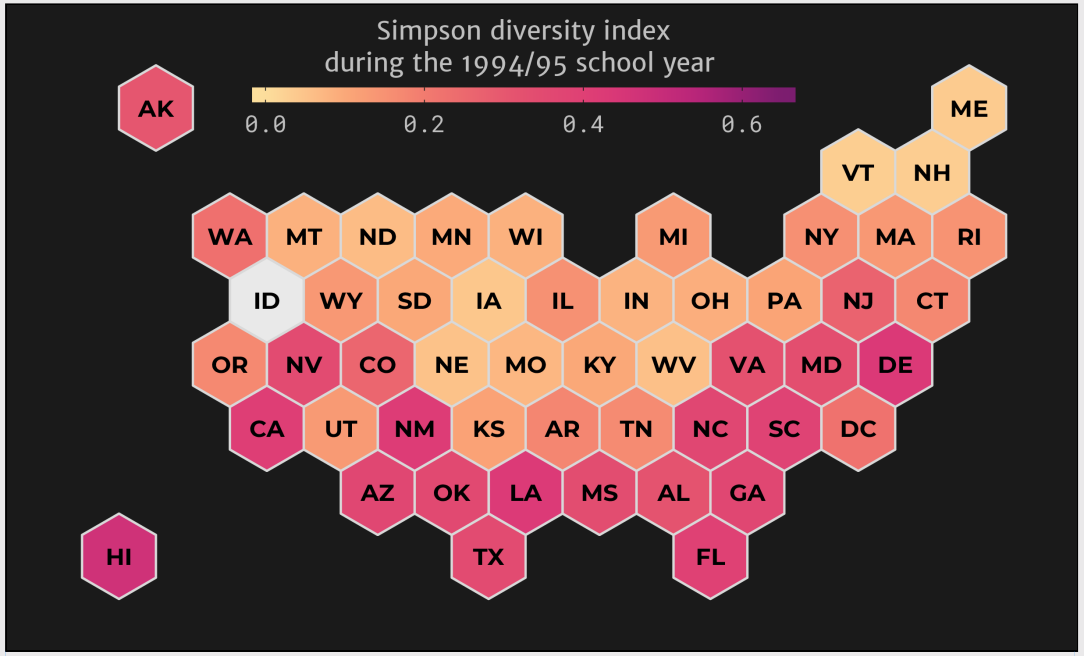


2014 Crime map [source]



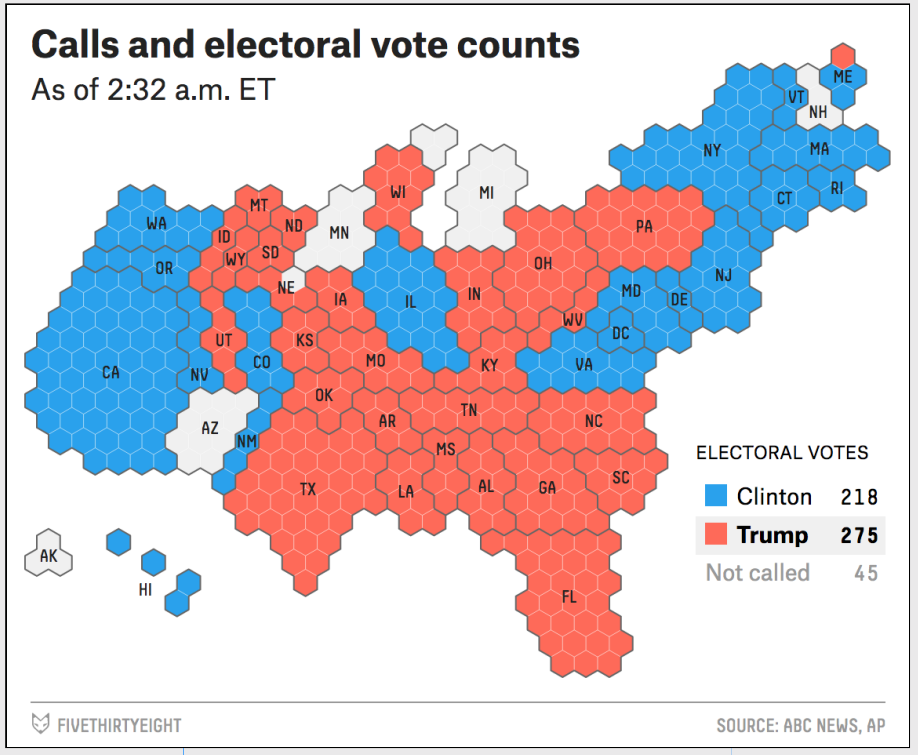
Another alternative: hex maps

1994 Simpson Diversity Index in US Schools



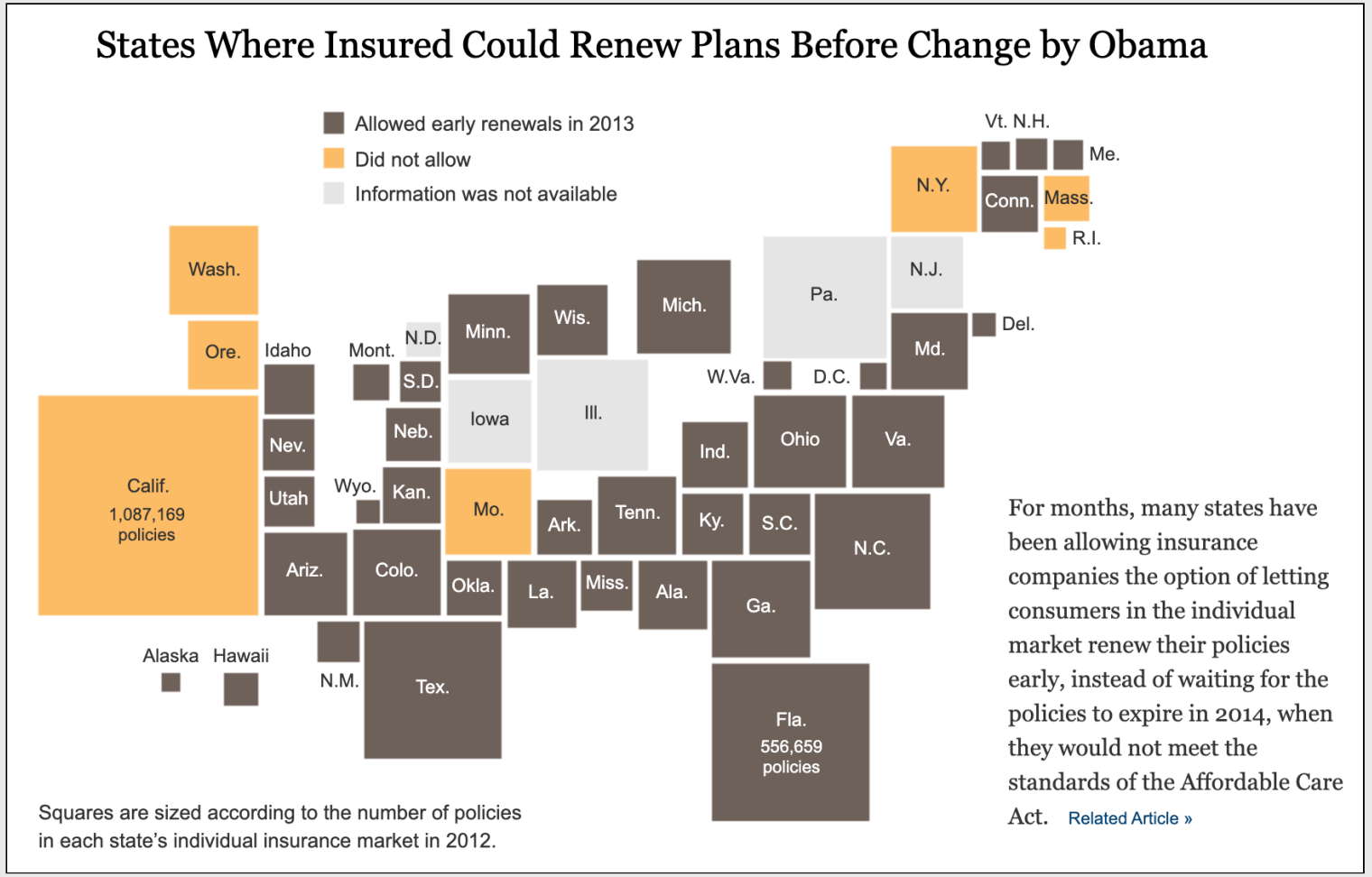
https://github.com/malcolmbarrett/designing_ggplots

2016 Electoral College



https://fivethirtyeight.com/

Another alternative: hex maps



How to make a choropleth map

Get the "fill" data

```
milk_2017 <- milk_production %>%
    filter(year == 2017) %>%
    select(name = state, milk_produced) %>%
    mutate(milk_produced = milk_produced / 10^9)
```

Get the "map" data

```
us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

```
us_states %>%
    select(name, milk_produced) %>%
    head()
```

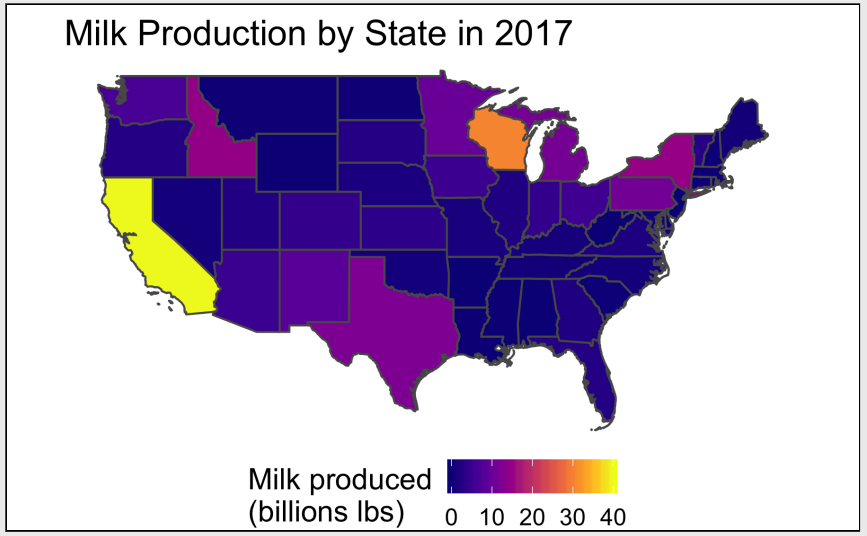
```
## Simple feature collection with 6 fea
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -124.7346 ymin
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum
##           name milk_produced
## 1    Minnesota         9.864 MULTIPO
## 2   Washington         6.526 MULTIPO
## 3        Idaho        14.627 MULTIPO
## 4      Montana         0.288 MULTIPO
## 5 North Dakota         0.345 MULTIPO
## 6     Michigan        11.231 MULTIPO
```

How to make a choropleth map

```
milk_2017 <- milk_production %>%
    filter(year == 2017) %>%
    select(name = state, milk_produced) %>%
    mutate(milk_produced = milk_produced / 10^9)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

Make the plot

```
ggplot(us_states) +
    geom_sf(aes(fill = milk_produced)) +
    scale_fill_viridis(
        option = "plasma",
        limits = c(0, 40)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    labs(fill = 'Milk produced\n(billions lbs)',
         title = 'Milk Production by State in 2017')
```

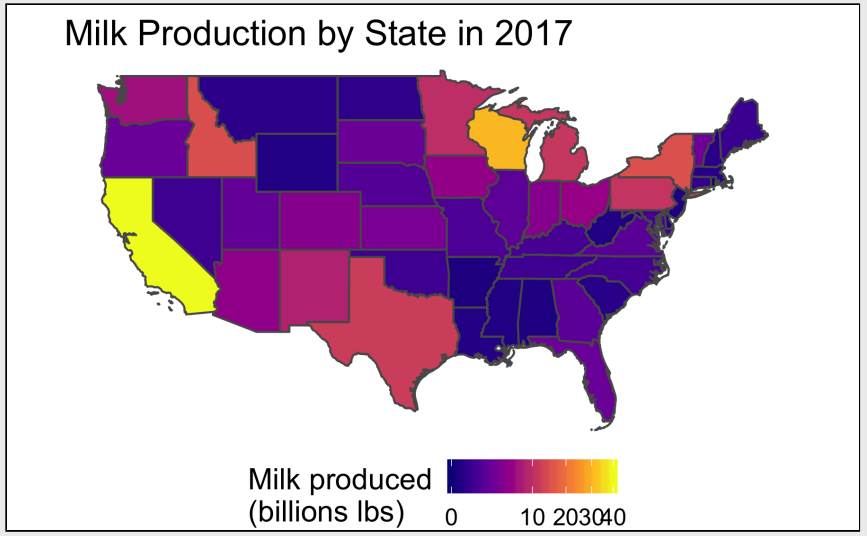


How to make a choropleth map

```
milk_2017 <- milk_production %>%
    filter(year == 2017) %>%
    select(name = state, milk_produced) %>%
    mutate(milk_produced = milk_produced / 10^9)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

Make the plot

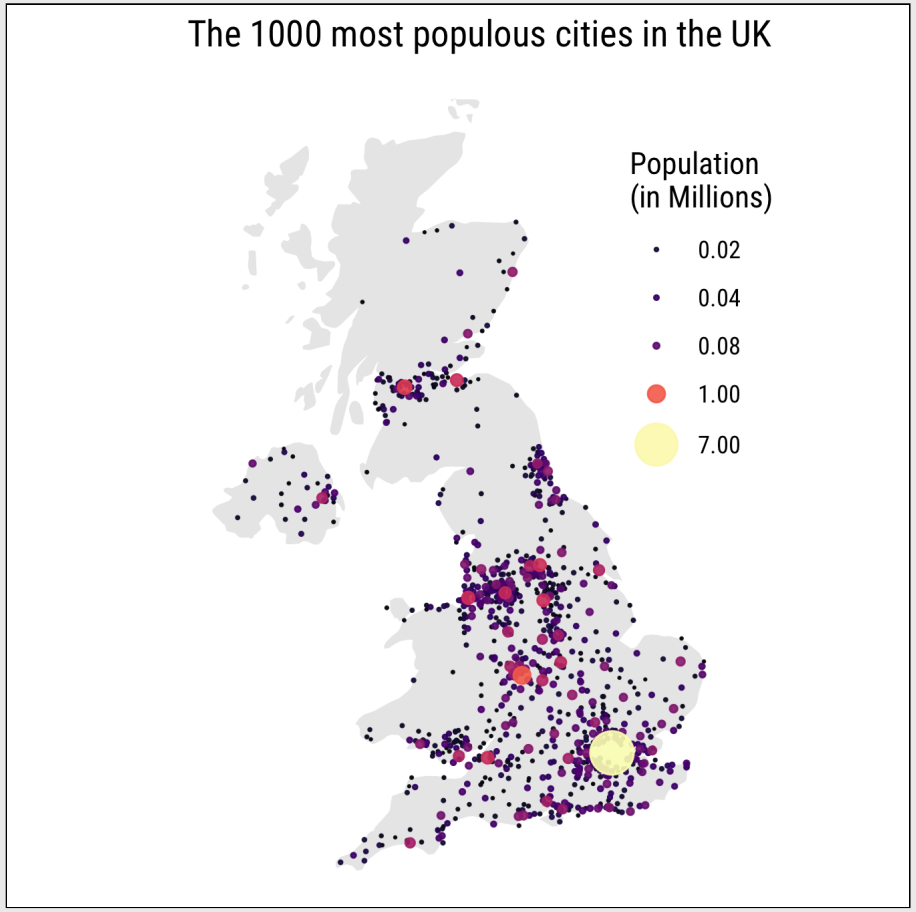
```
ggplot(us_states) +
    geom_sf(aes(fill = milk_produced)) +
    scale_fill_viridis(
        trans = 'sqrt',
        option = "plasma",
        limits = c(0, 40)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    labs(fill = 'Milk produced\n(billions lbs)',
         title = 'Milk Production by State in 2017')
```



Points

Points as locations

Points encoding a variable

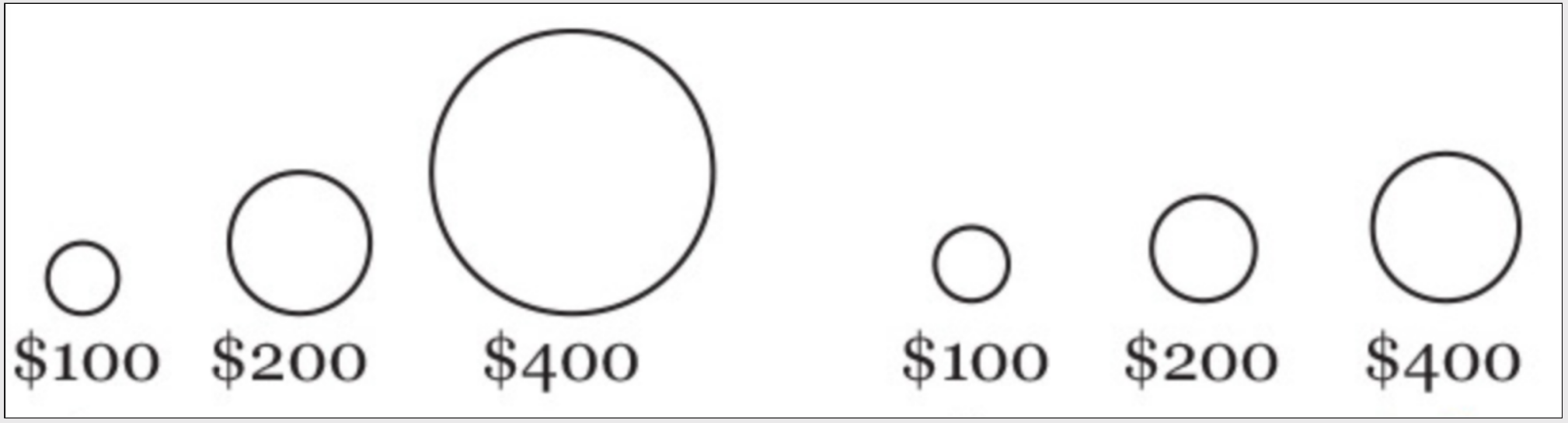


For point size, use **area**, not radius

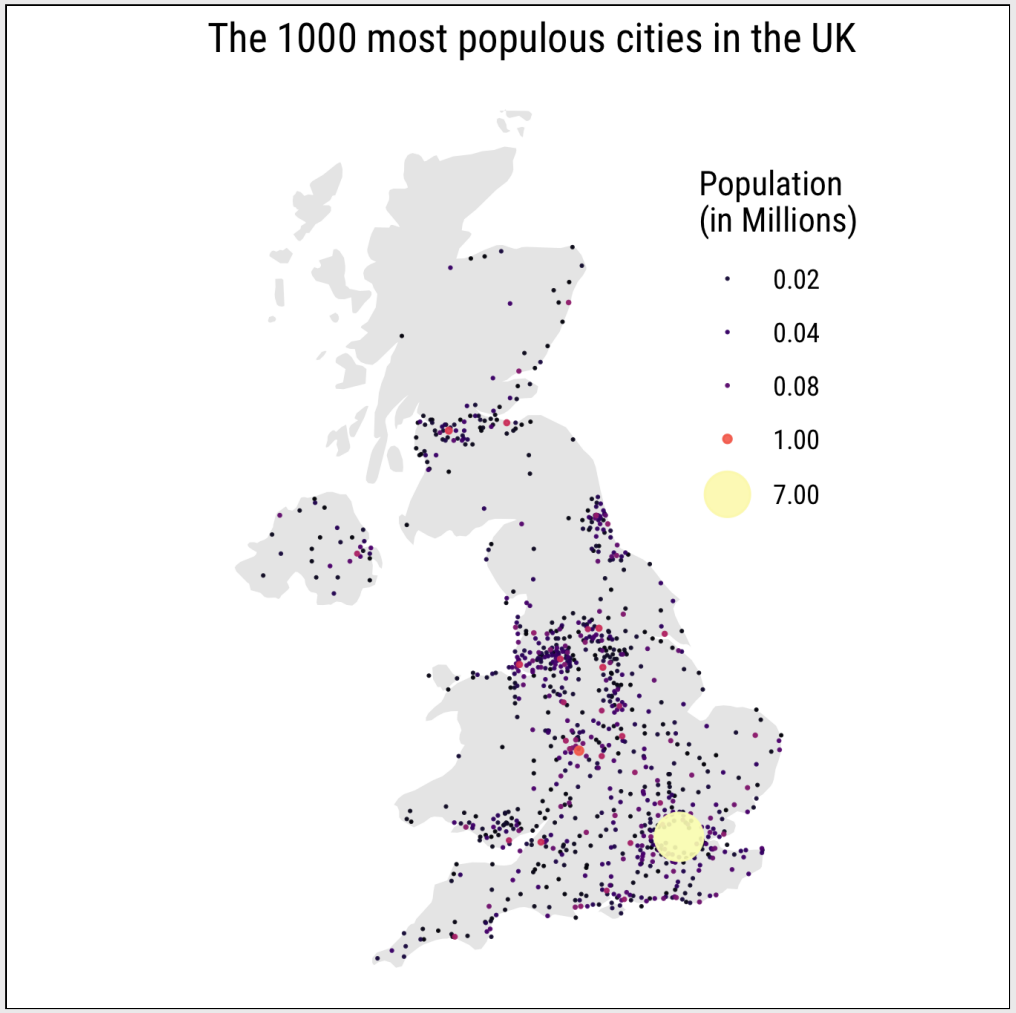
$$Area = \pi r^2$$

Radius

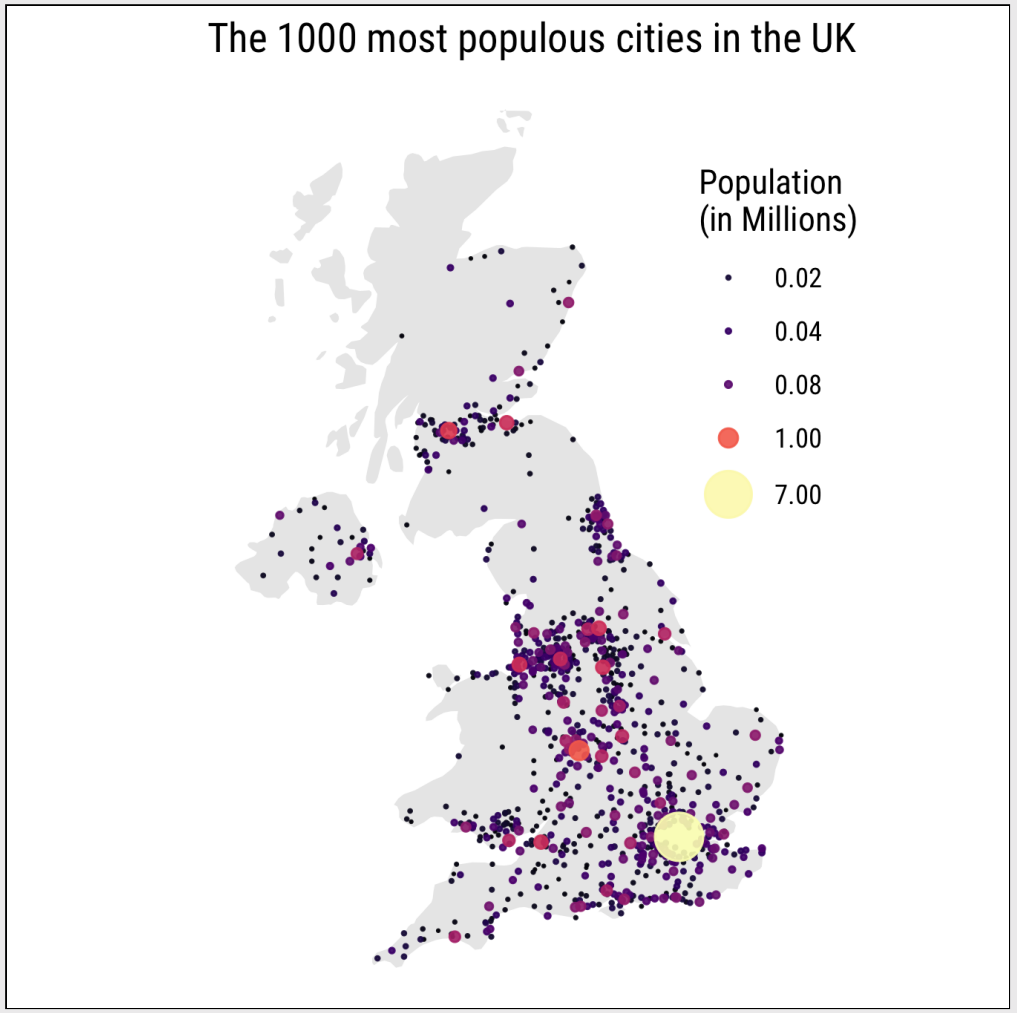
Area



Radius


The 1000 most populous cities in the UK
Population
(in Millions)
0.02
0.04
0.08
1.00
7.00

Area


The 1000 most populous cities in the UK
Population
(in Millions)
0.02
0.04
0.08
1.00
7.00

How to add points to a map

Load the continental US shape file

```
us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))
```

Read in the coffee shop data

```
us_coffee_shops  <- read_csv(here::here(
    'data', 'us_coffee_shops.csv'))

# Only keep data in continental US
us_coffee_shops <- us_coffee_shops %>%
    filter(lat > 22,    lat < 50,
           long > -150, long < -66)
```

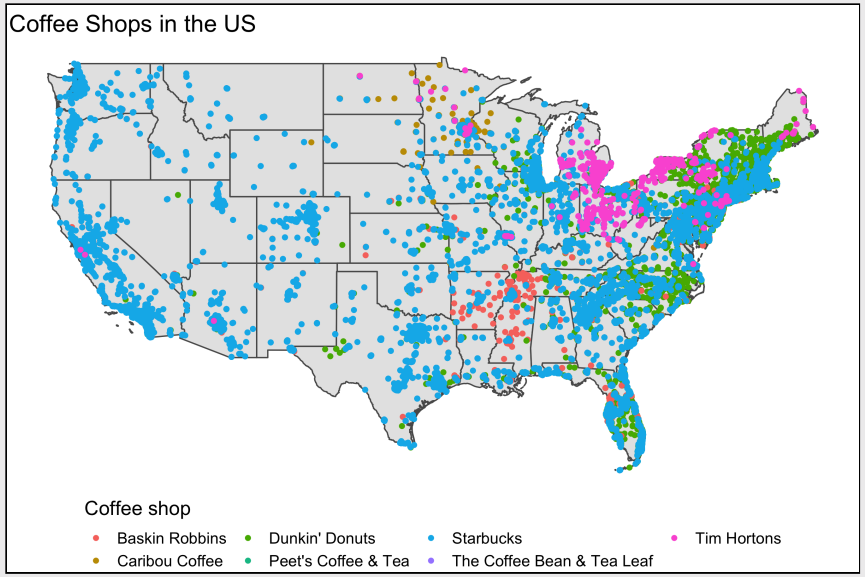
How to add points to a map

Load the continental US shape file & coffee shop data

```
us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))

us_coffee_shops <- us_coffee_shops %>%
    filter(lat > 22,    lat < 50,
           long > -150, long < -66)
```

Plot coffee shop locations over map

```
ggplot() +
    geom_sf(data = us_states_cont) +
    geom_point(data = us_coffee_shops,
               aes(x = long, y = lat, color = name)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    guides(color = guide_legend(title.position = "top")) +
    labs(color = 'Coffee shop',
         title = 'Coffee Shops in the US')
```

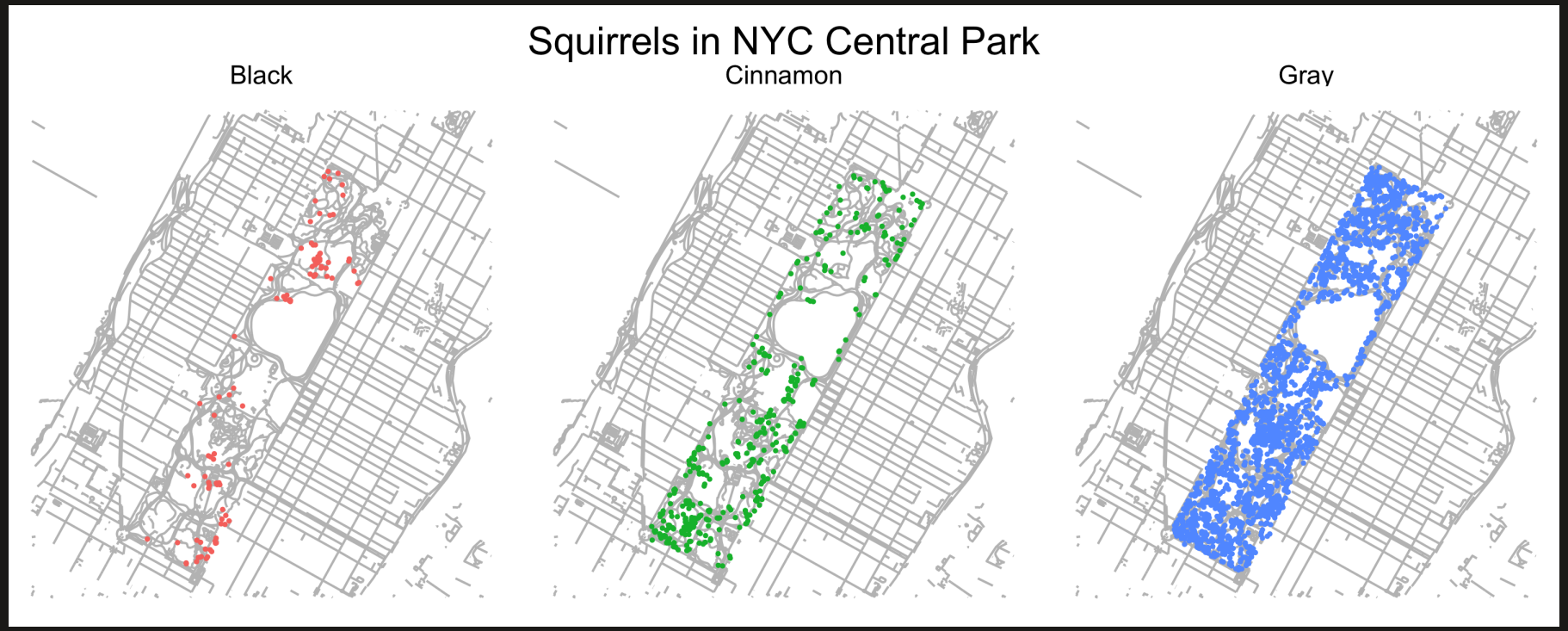


Your turn

15:00

Create the following map of squirrels in NYC's central park using the following data:

- The `CentralPark.shp` file in the `data/central_park` folder.
- The `nyc_squirrels.csv` file in the `data` folder.



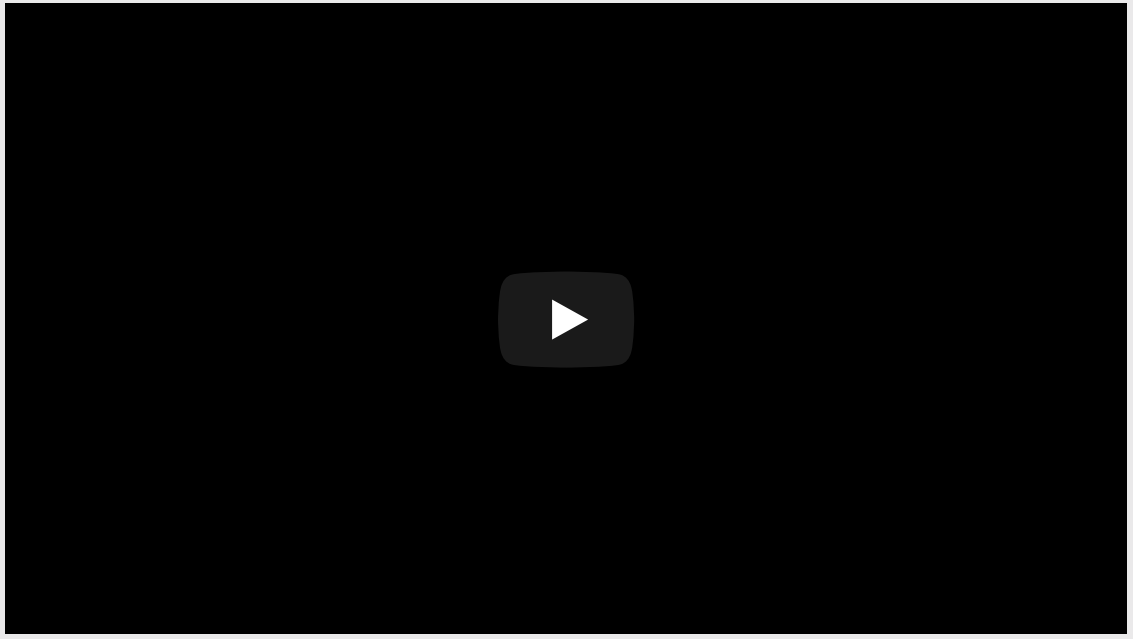
Maps & geospatial data

1. Plotting maps
2. Adding data to maps
3. Projections

1. Compare projections

2. Compare country sizes

What is the best projection?



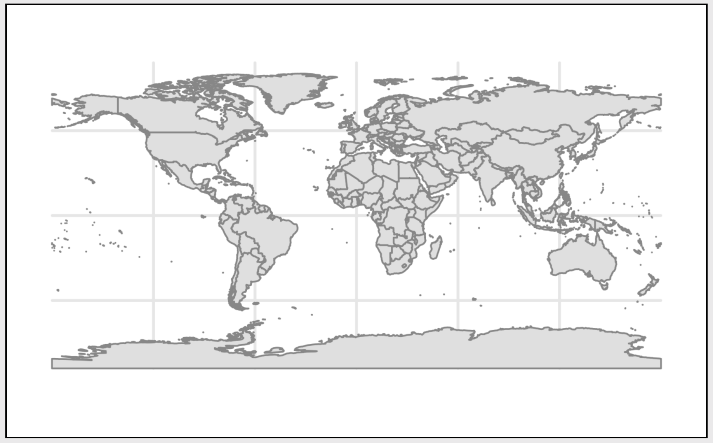
Using projections

To modify the projection of a map, use `coord_sf(crs = st_crs(XXXX))`

```
world <- ne_countries(scale = "medium", returnclass = "sf")
```

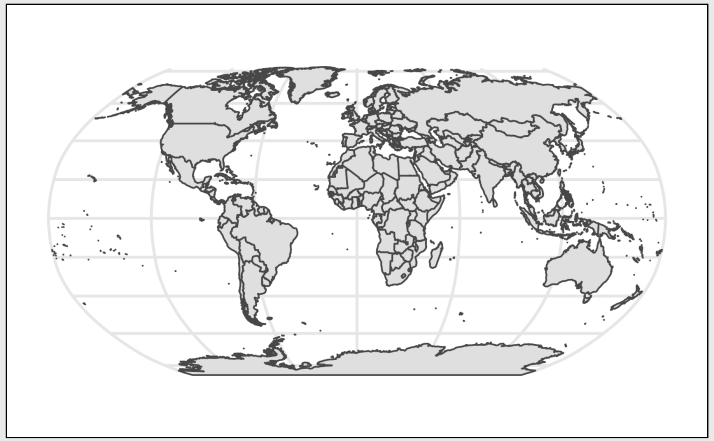
Default (long-lat)

```
ggplot(data = world) +
    geom_sf()
```



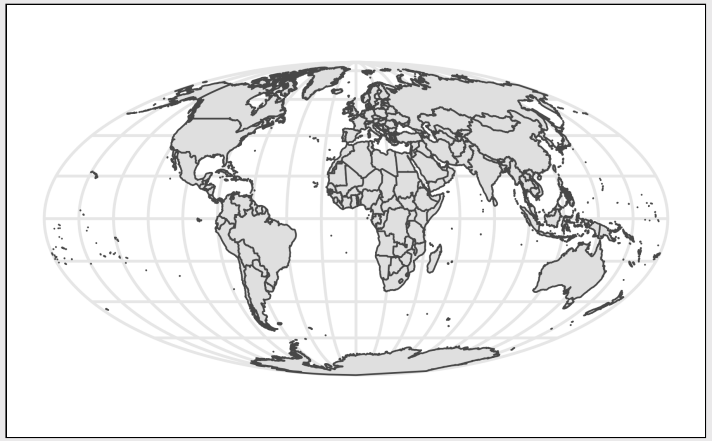
Robinson projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = st_crs(54030))
```



Mollweide projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = st_crs(54009))
```



Common Projections

`coord_sf(crs = st_crs(XXXXX))`

World Projections

Code	Projection
54002	Equidistant cylindrical projection
54004	Mercator projection
54008	Sinusoidal projection
54009	Mollweide projection
54030	Robinson projection

US Projections

Code	Projection
102003	Albers projection
4269	NAD 83

`coord_sf(crs = st_crs(name))`

Name	Projection
"+proj=merc"	Mercator
"+proj=robin"	Robinson
"+proj=moll"	Mollweide

My favorites:

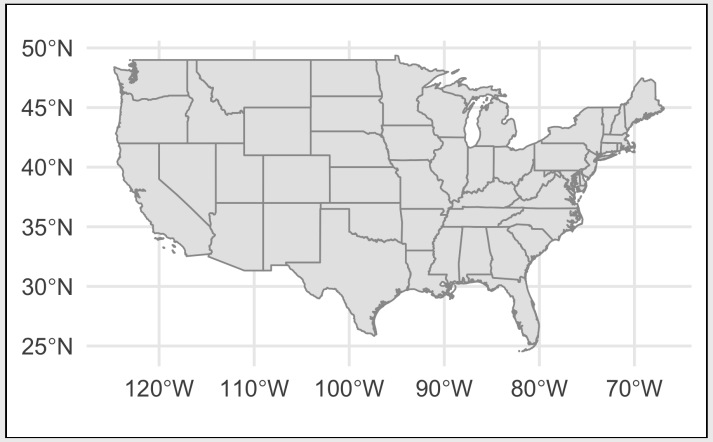
- World: Robinson `54030`
- US: Albers `102003`

US projections

```
us_states_cont <- ne_states(country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))
```

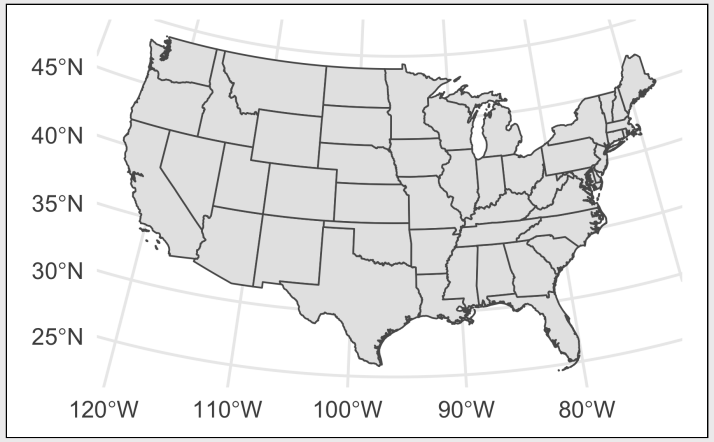
NAD 83 projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = st_crs(4269))
```



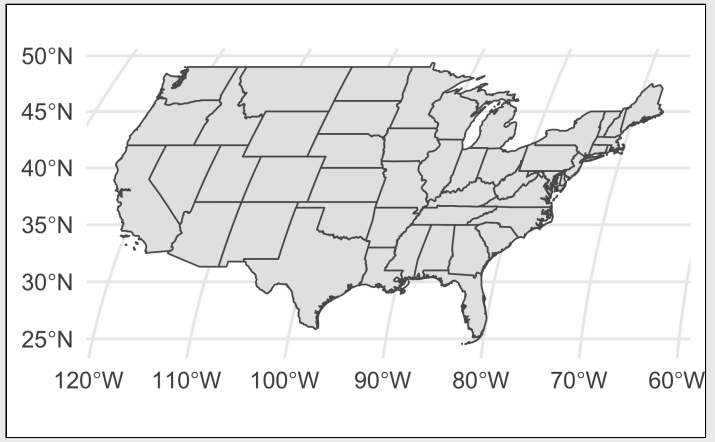
Albers projection

```
ggplot(data = us_states_cont) +
    geom_sf() +
    coord_sf(crs = st_crs(102003))
```



Robinson projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = st_crs(54030))
```

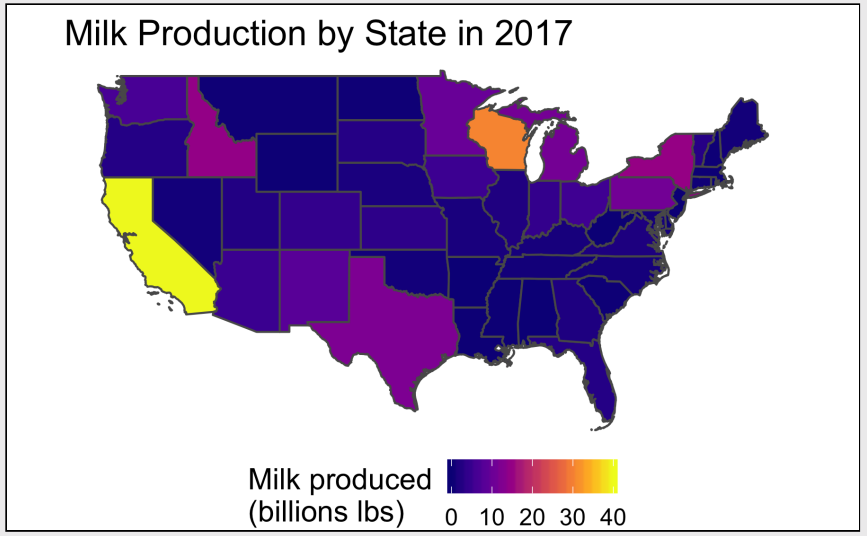


Mapping data to projections - choropleth map

```
milk_2017 <- milk_production %>%
    filter(year == 2017) %>%
    select(name = state, milk_produced) %>%
    mutate(milk_produced = milk_produced / 10^9)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

Make the plot

```
ggplot(us_states) +
    geom_sf(aes(fill = milk_produced)) +
    scale_fill_viridis(
        option = "plasma",
        limits = c(0, 40)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    labs(fill = 'Milk produced\n(billions lbs)',
         title = 'Milk Production by State in 2017')
```

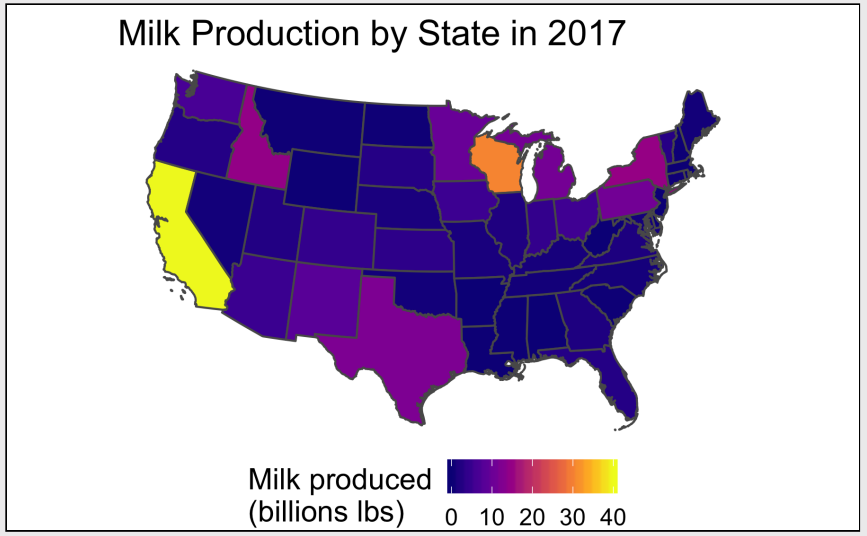


Mapping data to projections - choropleth map

```
milk_2017 <- milk_production %>%
    filter(year == 2017) %>%
    select(name = state, milk_produced) %>%
    mutate(milk_produced = milk_produced / 10^9)

us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

Make the plot

```
ggplot(us_states) +
    geom_sf(aes(fill = milk_produced)) +
    coord_sf(crs = st_crs(102003)) +
    scale_fill_viridis(
        option = "plasma",
        limits = c(0, 40)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    labs(fill = 'Milk produced\n(billions lbs)',
         title = 'Milk Production by State in 2017')
```

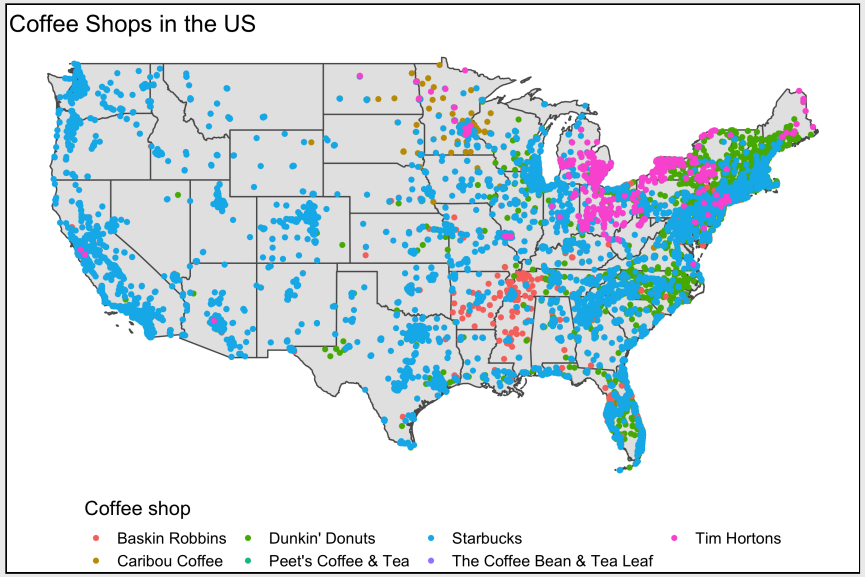


Mapping data to projections - points

```
us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))

us_coffee_shops <- us_coffee_shops %>%
    filter(lat > 22,    lat < 50,
           long > -150, long < -66)
```

Plot coffee shop locations over map

```
ggplot() +
    geom_sf(data = us_states_cont) +
    geom_point(data = us_coffee_shops,
               aes(x = long, y = lat, color = name)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    guides(color = guide_legend(title.position = "top")) +
    labs(color = 'Coffee shop',
         title = 'Coffee Shops in the US')
```



Mapping data to projections - points

```
us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))

us_coffee_shops <- us_coffee_shops %>%
    filter(lat > 22,    lat < 50,
           long > -150, long < -66)
```

Plot coffee shop locations over map...fail!

```
ggplot() +
    geom_sf(data = us_states_cont) +
    geom_point(data = us_coffee_shops,
               aes(x = long, y = lat, color = name)) +
    coord_sf(crs = st_crs(102003)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    guides(color = guide_legend(title.position = "top")) +
    labs(color = 'Coffee shop',
         title = 'Coffee Shops in the US')
```

Mapping data to projections - points

First match `us_coffee_shops` crs to `us_states_cont`

```
us_states_cont <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii'))

us_coffee_shops <- us_coffee_shops %>%
    filter(lat > 22,    lat < 50,
           long > -150, long < -66)

us_coffee_shops_sf <- st_as_sf(us_coffee_shops,
    coords = c("long", "lat"),
    crs = st_crs(us_states_cont))
```

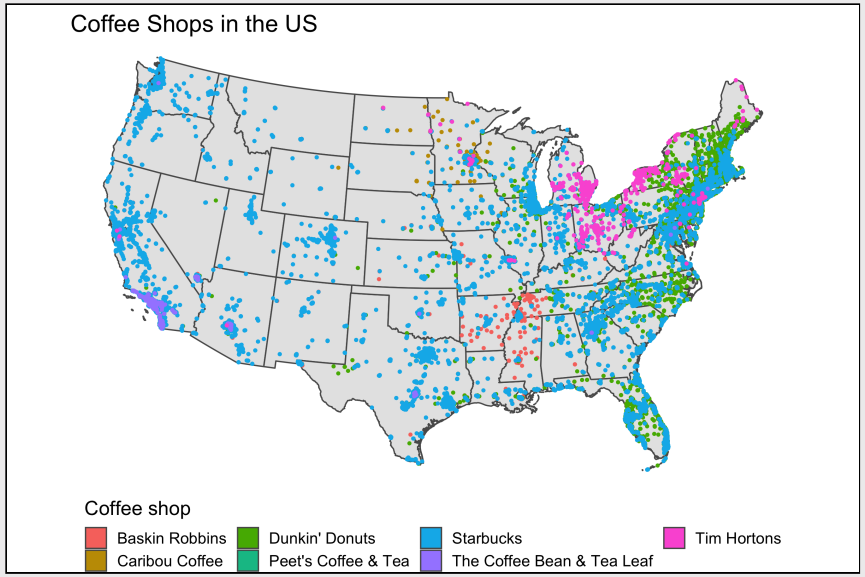
Mapping data to projections - points

First match `us_coffee_shops` crs to `us_states_cont`

```
us_coffee_shops_sf <- st_as_sf(us_coffee_shops,
    coords = c("long", "lat"),
    crs = st_crs(us_states_cont))
```

Plot coffee shop locations over map with `geom_sf()`

```
ggplot() +
    geom_sf(data = us_states_cont) +
    geom_sf(data = us_coffee_shops_sf,
            aes(fill = name),
            shape = 21, stroke = FALSE) +
    coord_sf(crs = st_crs(102003)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    guides(fill = guide_legend(title.position = "top")) +
    labs(fill = 'Coffee shop',
         title = 'Coffee Shops in the US')
```



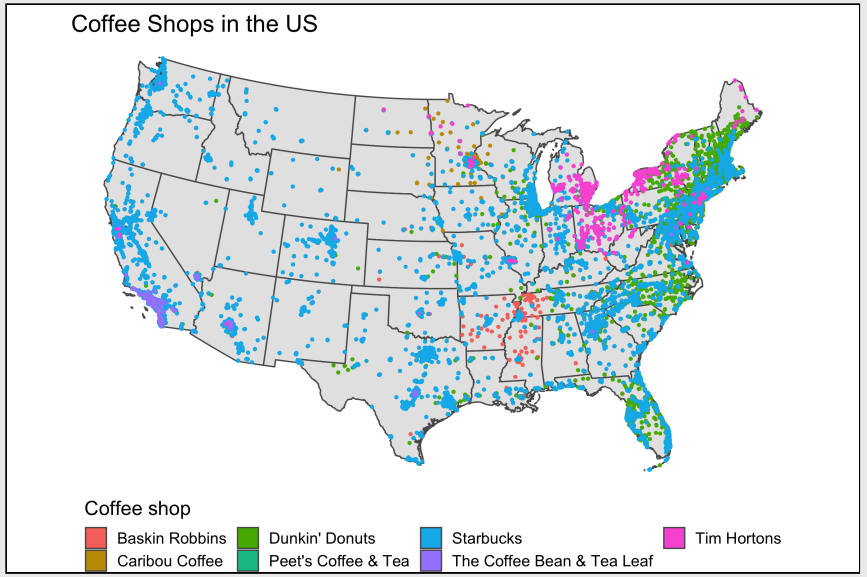
Mapping data to projections - points

First match `us_coffee_shops` crs to `us_states_cont`

```
us_coffee_shops_sf <- st_as_sf(us_coffee_shops,
    coords = c("long", "lat"),
    crs = st_crs(us_states_cont))
```

Plot coffee shop locations over map with `geom_sf()`

```
ggplot() +
    geom_sf(data = us_states_cont) +
    geom_sf(data = us_coffee_shops_sf,
            aes(fill = name),
            shape = 21, stroke = FALSE) +
    coord_sf(crs = st_crs(102004)) +
    theme_void(base_size = 15) +
    theme(legend.position = 'bottom') +
    guides(fill = guide_legend(title.position = "top")) +
    labs(fill = 'Coffee shop',
         title = 'Coffee Shops in the US')
```



Your turn

15:00

Use the `us_states_cont` data frame and the `state_abbs` data frame to create a labeled map of the U.S.:

Proposals & Assignment 7