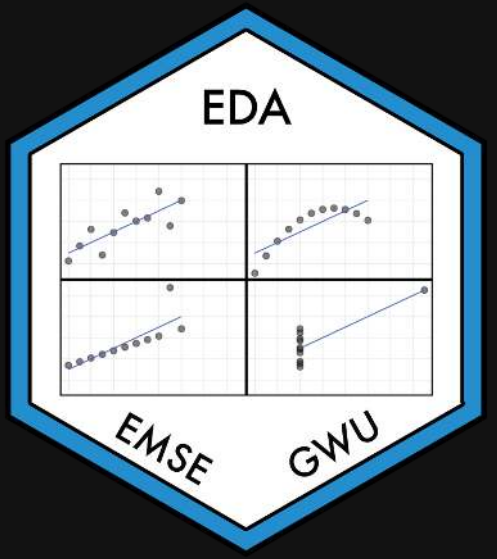


Week 11: *Maps*

EMSE 4572: Exploratory Data Analysis

John Paul Helveston

November 08, 2022

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 packages:

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

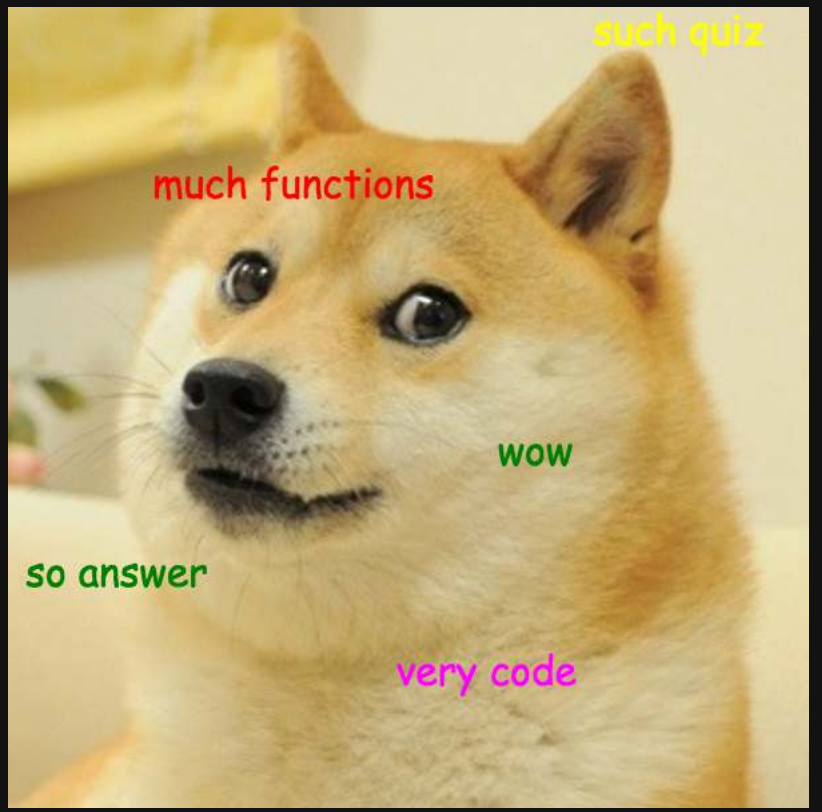
Quiz 4

Download the template from the #class channel

Make sure you unzip it!

When done, submit your `quiz4.qmd` on Blackboard

10:00



Week 11: *Maps*

1. Plotting maps
2. Adding data to maps

BREAK

3. Projections

Week 11: *Maps*

1. Plotting maps
2. Adding data to maps

BREAK

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. Polygon data: `geom_polygon()`

b. Simple Features data: `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

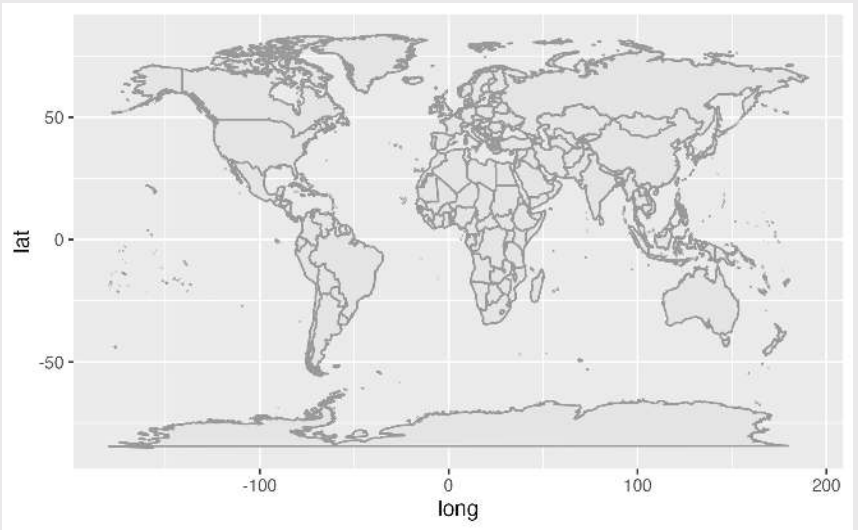
Get the "World" shape file

```
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

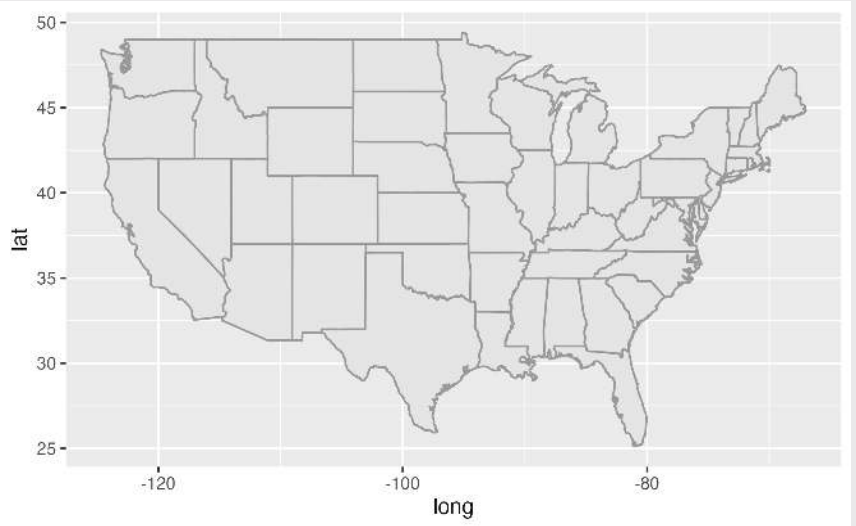
Get the "US States" shape file

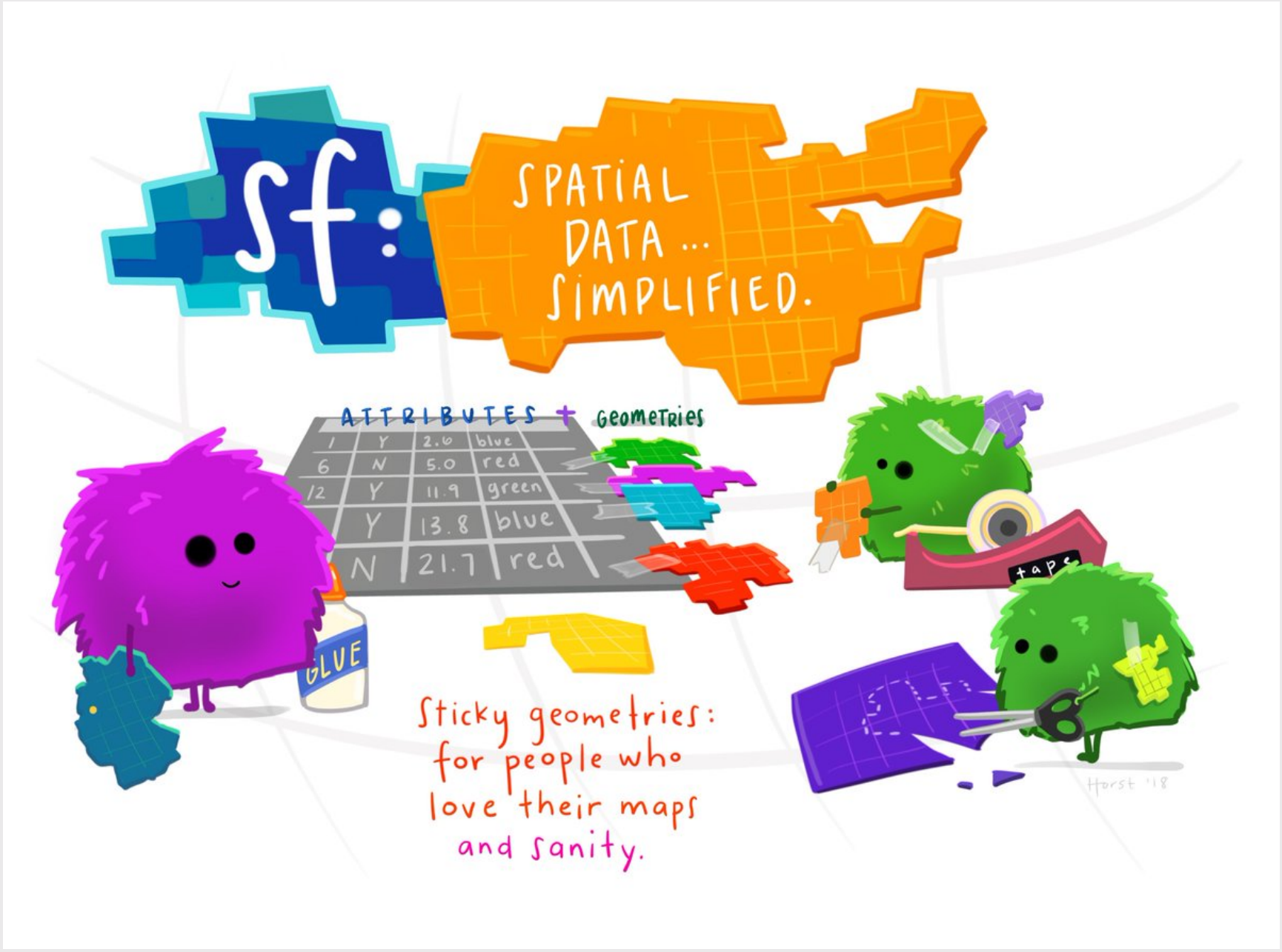
```
library(ggplot2)

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 package: {sf}

Art by Allison Horst

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()
```

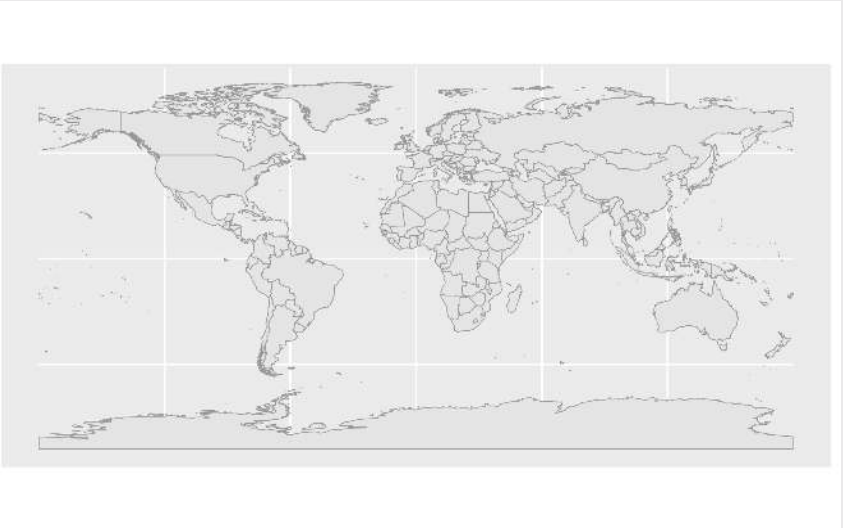
```
#> Simple feature collection with 6 features and 1 field
#> Geometry type: MULTIPOLYGON
#> Dimension:     XY
#> Bounding box:  xmin: -70.06611 ymin: -18.01973 xmax: 7
#> Geodetic CRS:  +proj=longlat +datum=WGS84 +no_defs +e
#>          name                          geometry
#> 0       Aruba MULTIPOLYGON (((-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...
#> 5       Aland MULTIPOLYGON (((20.61133 60...
```

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

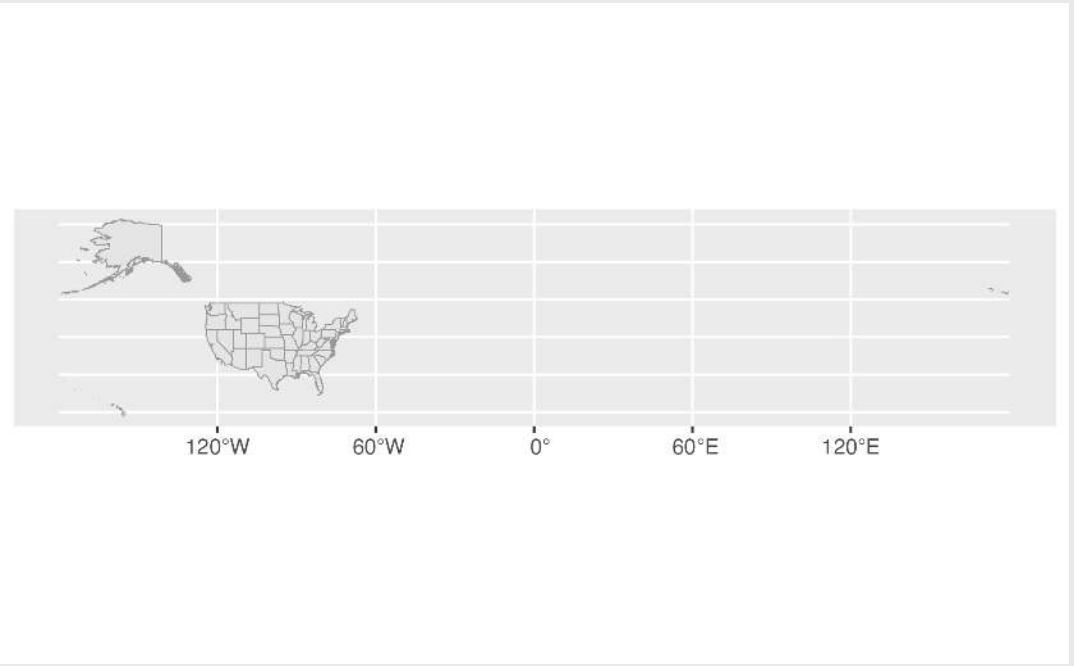
```
library(rnaturalearth)
library(rnaturalearthdata)

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")
```

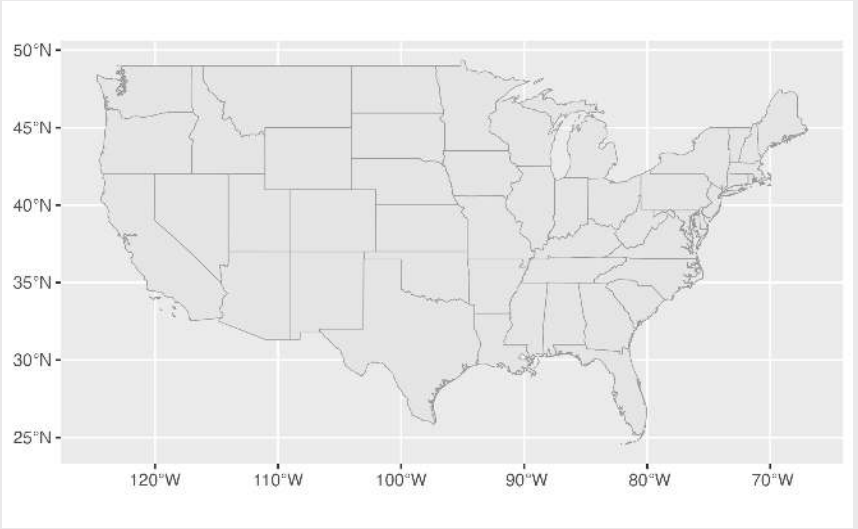


Simple Features (sf) maps

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")
```

Inset Hawaii and Alaska

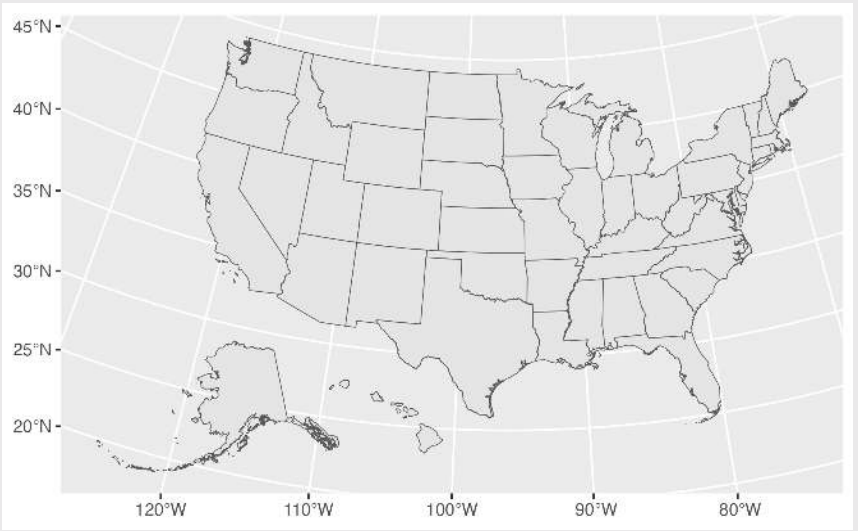
Get the shape file from {tigris} package

```
library(tigris)

us_sf <- tigris::states(class = "sf", cb = TRUE) %>%
    shift_geometry() %>%
    filter(GEOID < 60)
```

Make the plot with `geom_sf()`

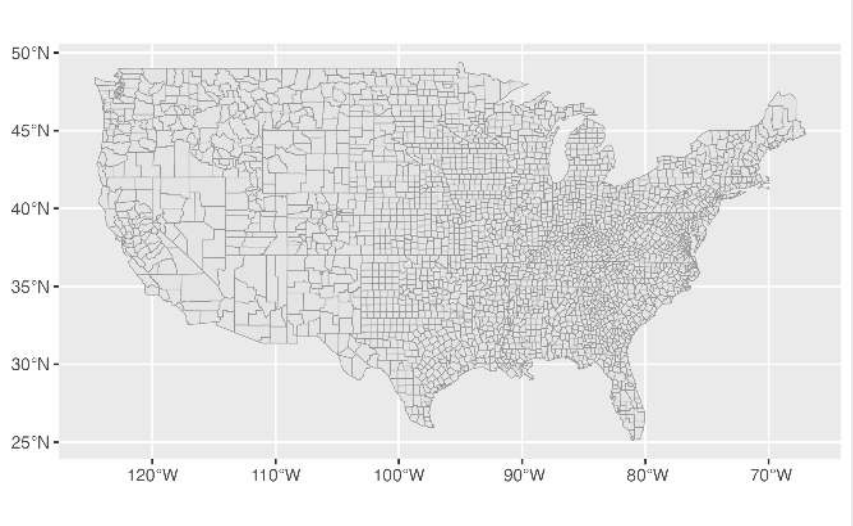
```
us_sf %>%
    ggplot() +
    geom_sf()
```



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 sou
#> Simple feature collection with 241 features and 94 fie
#> Geometry type: MULTIPOLYGON
#> Dimension:     XY
#> Bounding box:  xmin: -180 ymin: -89.99893 xmax: 180 yr
#> Geodetic CRS:  WGS 84
```

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

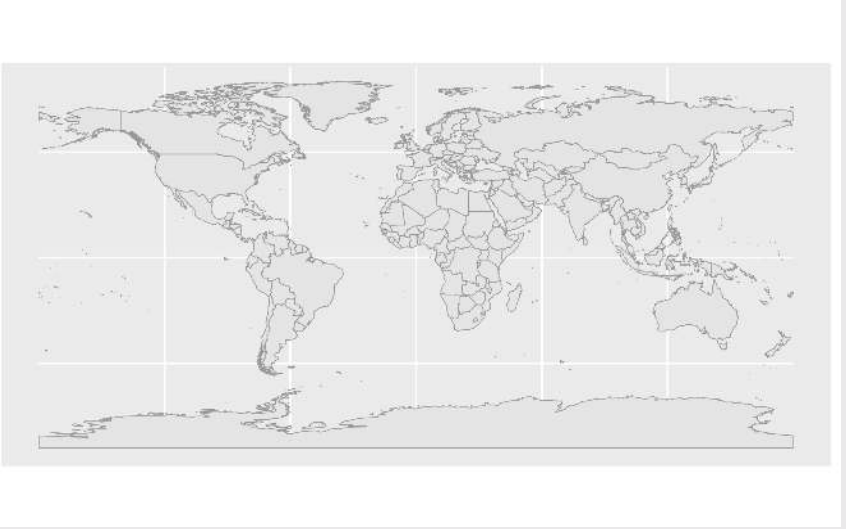
Read in the "World" shape file

```
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 sou
#> Simple feature collection with 241 features and 94 fie
#> Geometry type: MULTIPOLYGON
#> Dimension:     XY
#> Bounding box:  xmin: -180 ymin: -89.99893 xmax: 180 yr
#> Geodetic CRS:  WGS 84
```

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



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

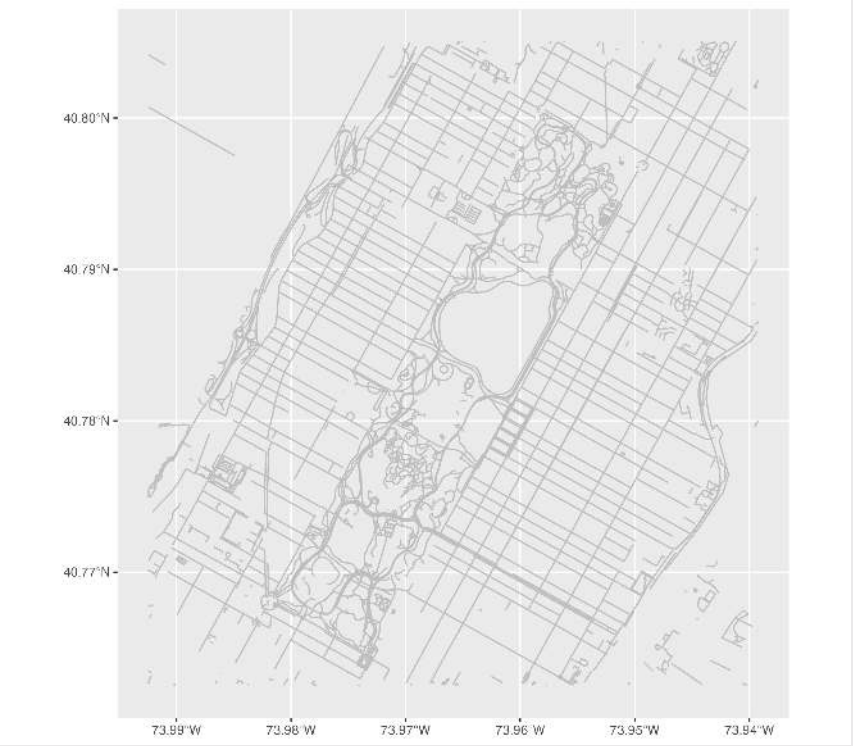
Read in the "Central Park" shape file [source]

```
library(sf)

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

```
#> Reading layer `CentralPark' from data source `/Users/j
#> Simple feature collection with 2550 features and 6 fie
#> Geometry type: LINESTRING
#> Dimension:     XY
#> Bounding box:  xmin: -73.99249 ymin: 40.7625 xmax: -73
#> Geodetic CRS:  WGS 84
```

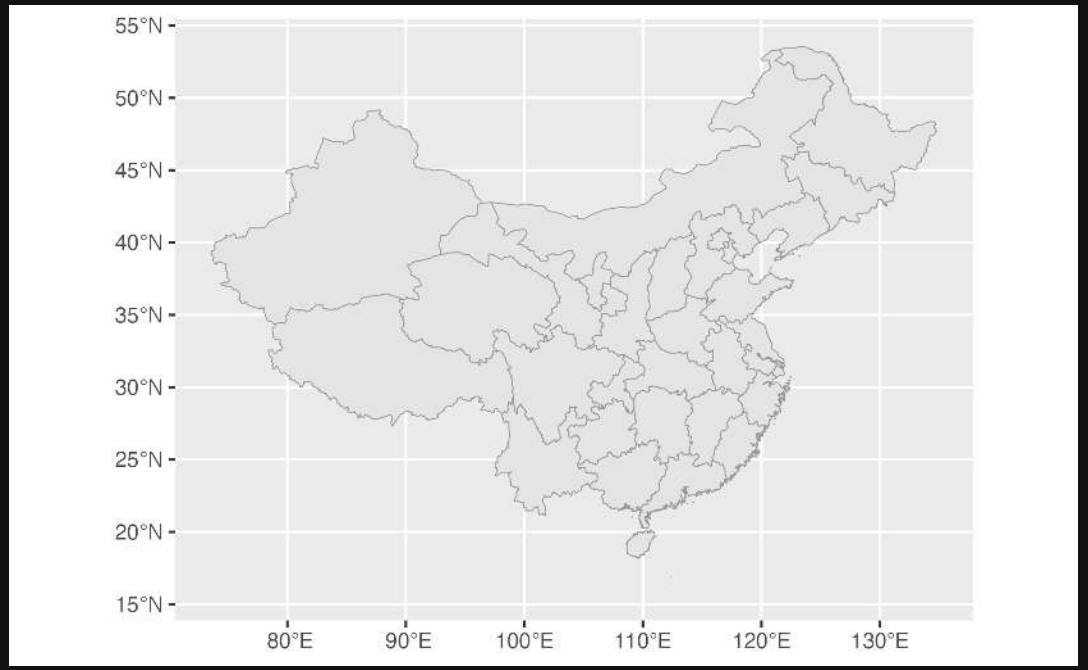
```
ggplot(data = central_park) +
  geom_sf(color = 'grey75')
```

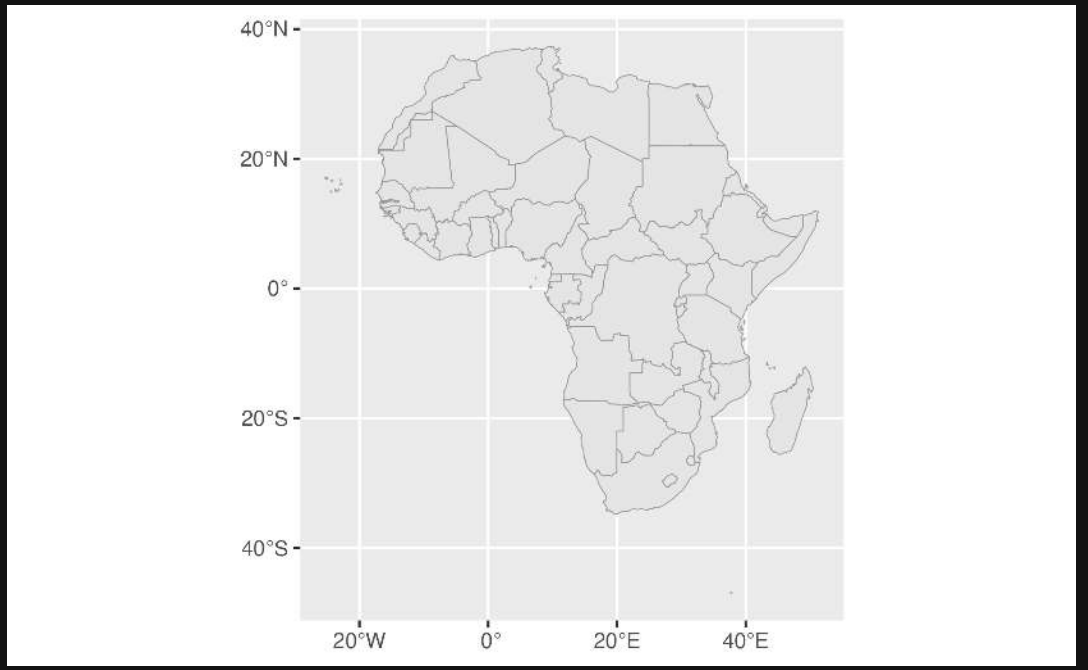


Your turn

10:00

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





Week 11: *Maps*

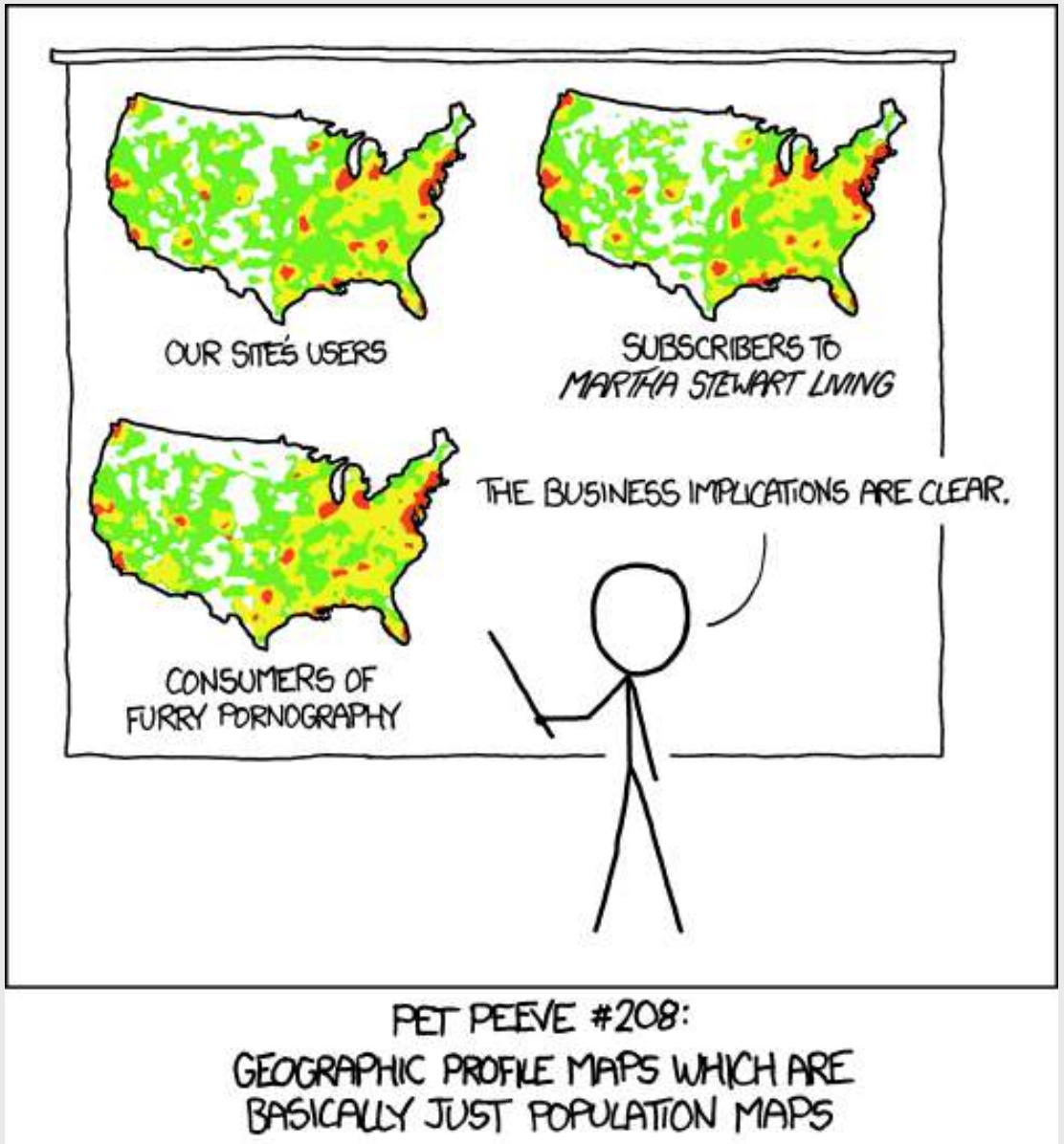
1. Plotting maps
2. Adding data to maps

BREAK

3. Projections

First rule of adding data to maps:

Do you need to make a map?



Not all maps are useful...

https://xkcd.com/1138/

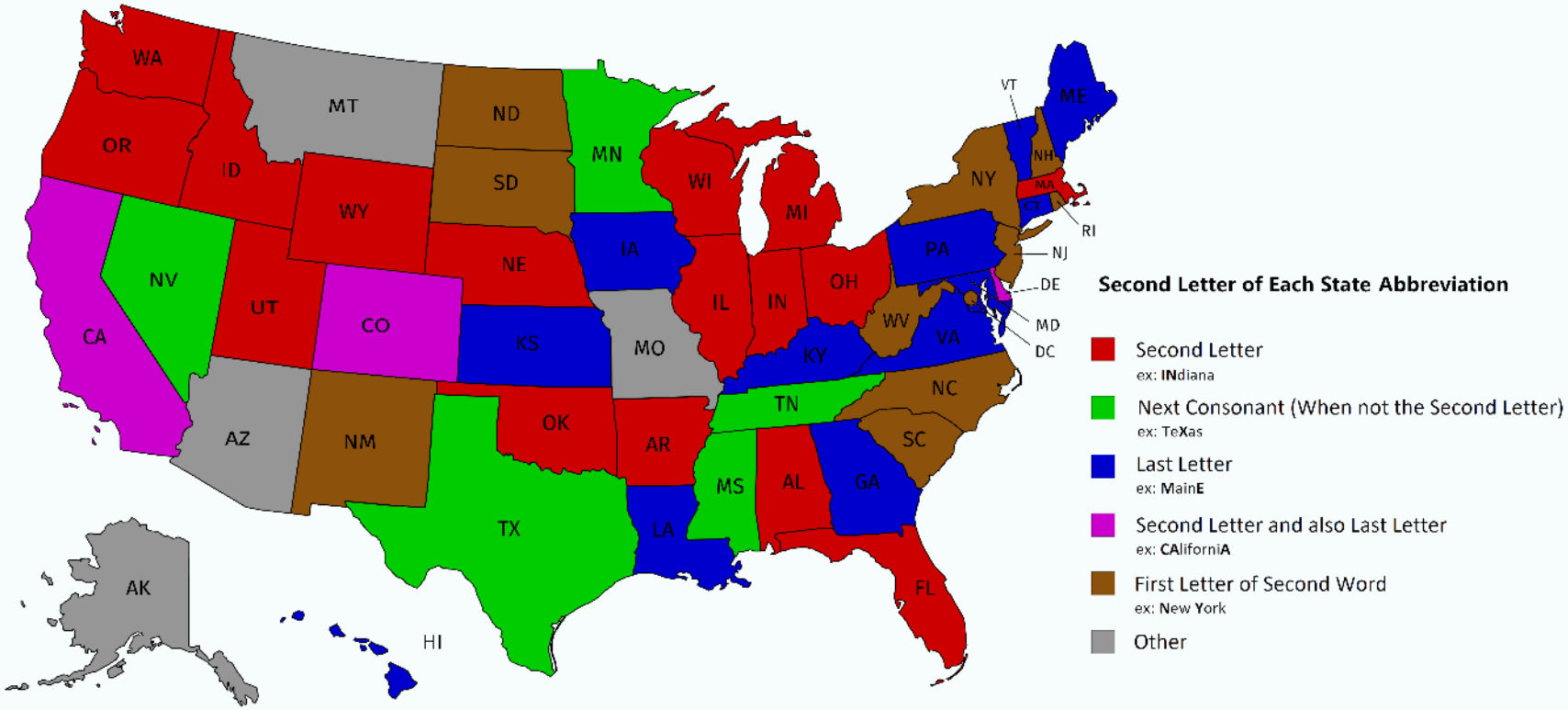
...but some maps are

https://twitter.com/pwang/status/1387228573896003585

1. Choropleth maps
2. Point maps

Choropleth - from Greek:

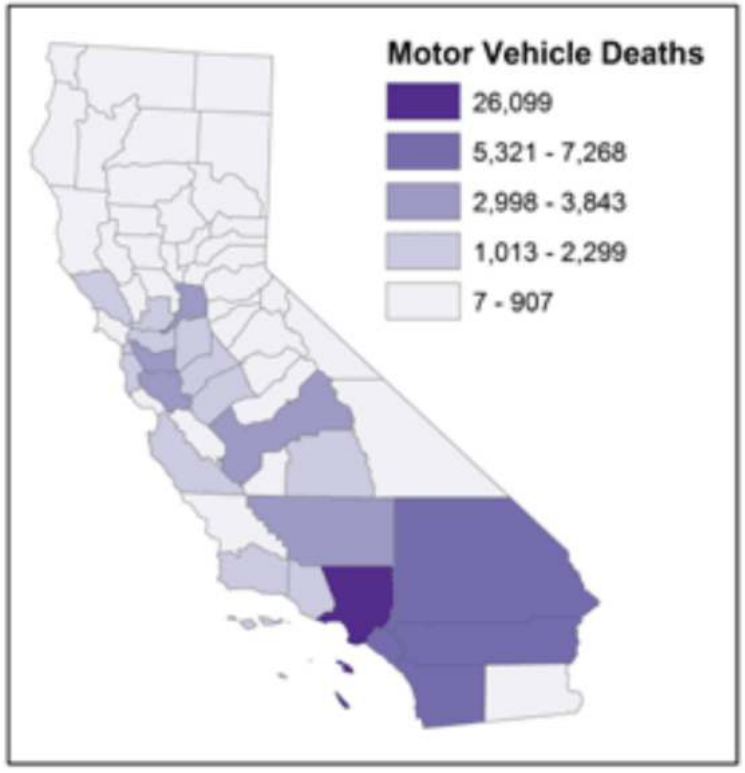
- χῶρος "choros" (area/region)
- πλῆθος "plethos" (multitude)

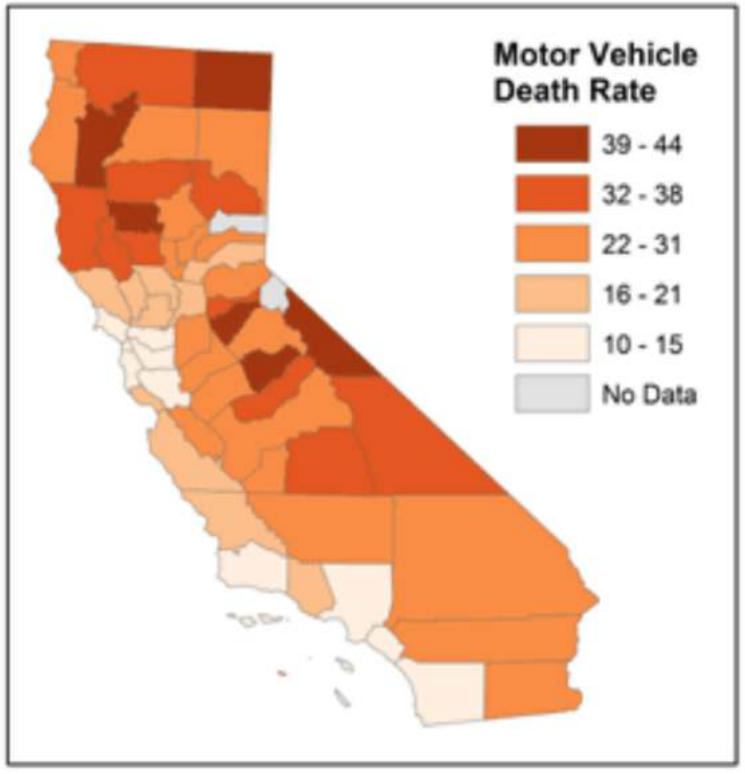


From reddit: r/dataisbeautiful by u/GaudyAudi

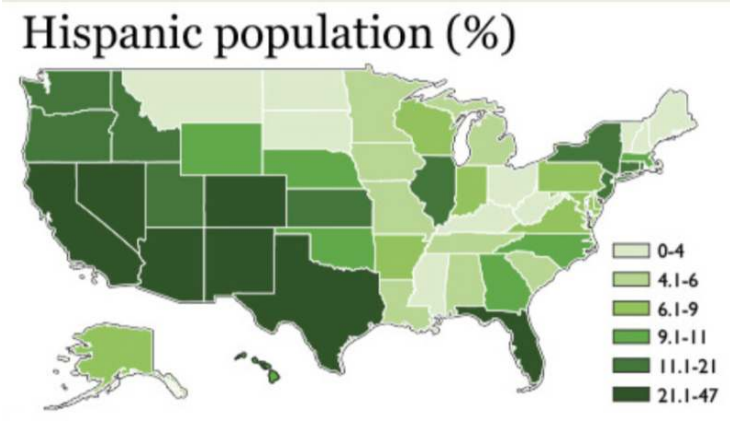
Choropleth maps are easily misleading

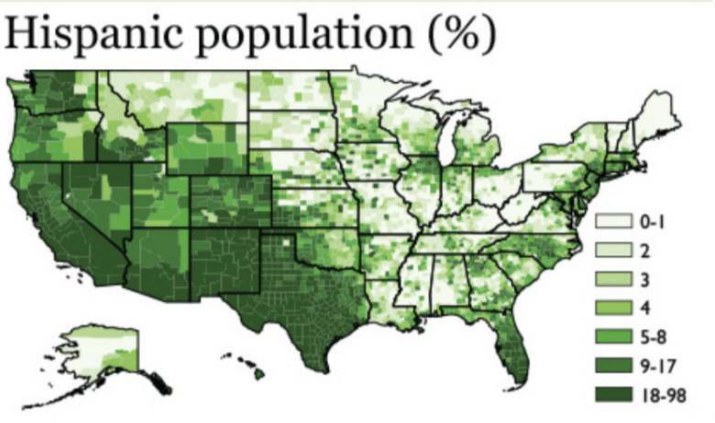
Number of events != Number of events **per capita**

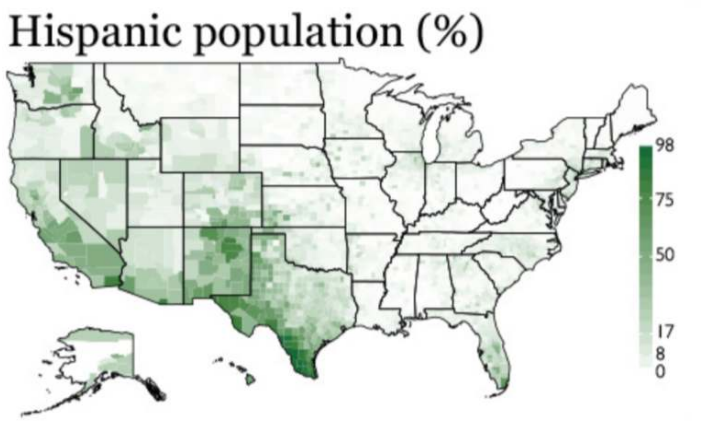




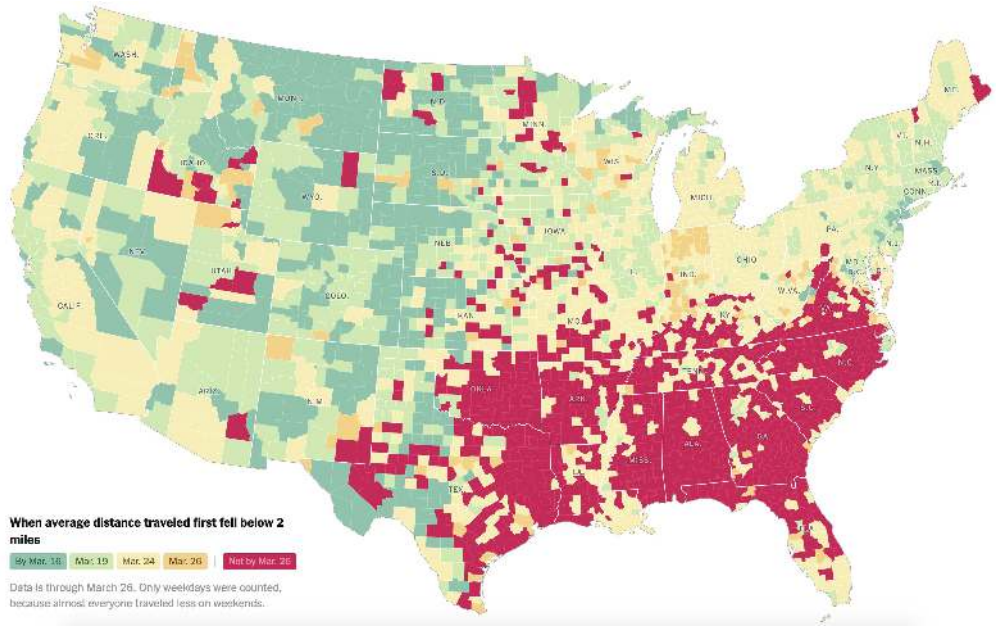
Manipulating fill scale produces wildly different maps

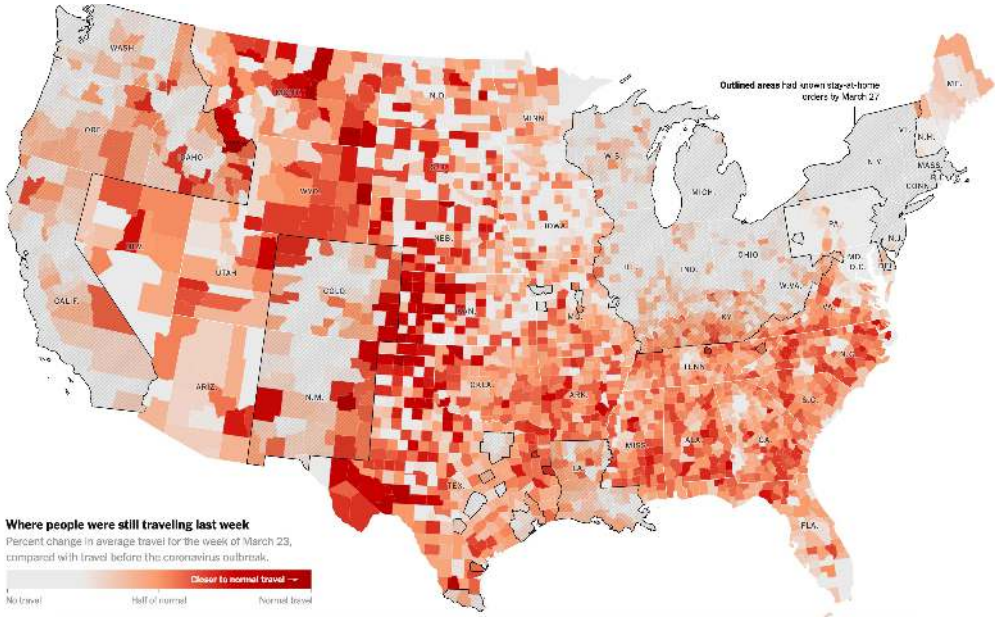






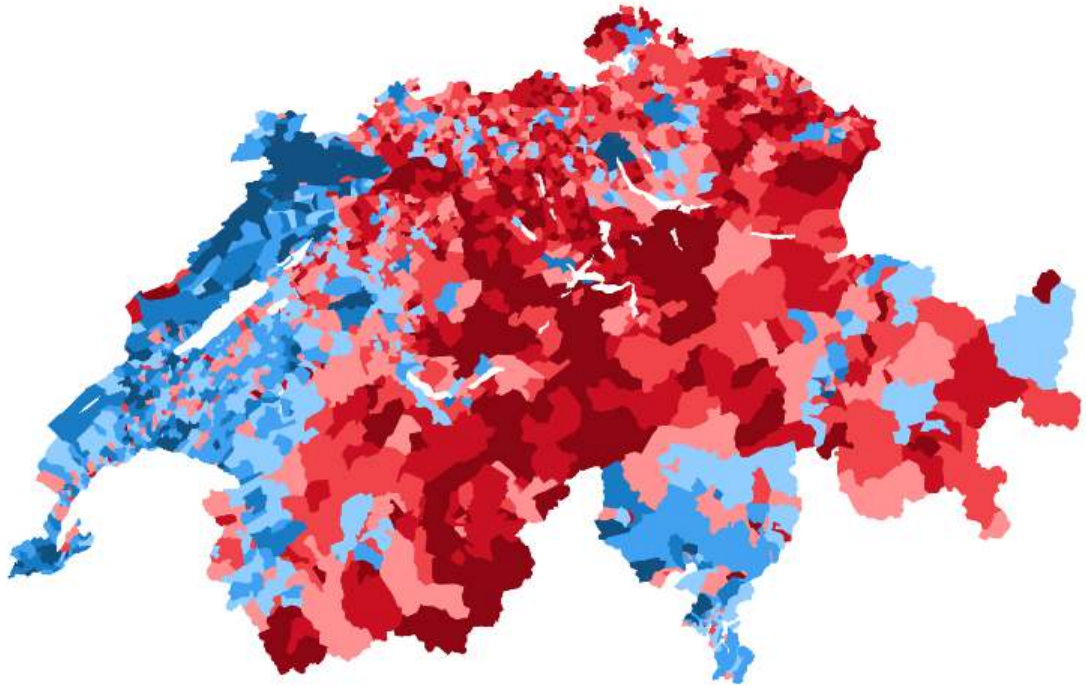
Manipulating fill scale produces wildly different maps





Source: New York Times

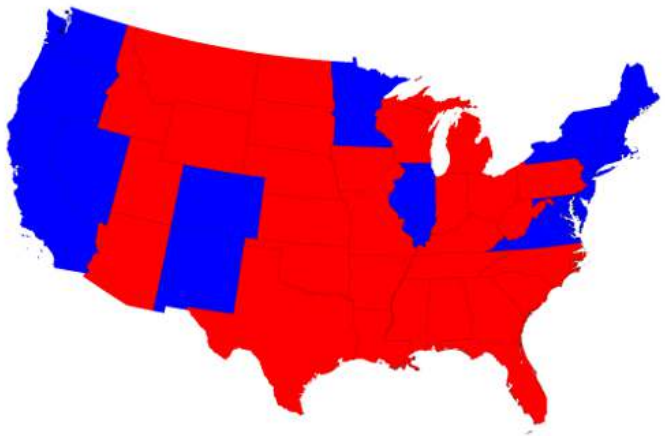
Land doesn't vote - people vote

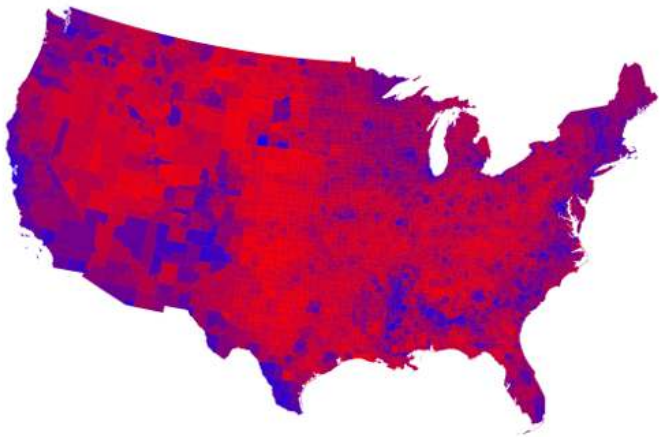


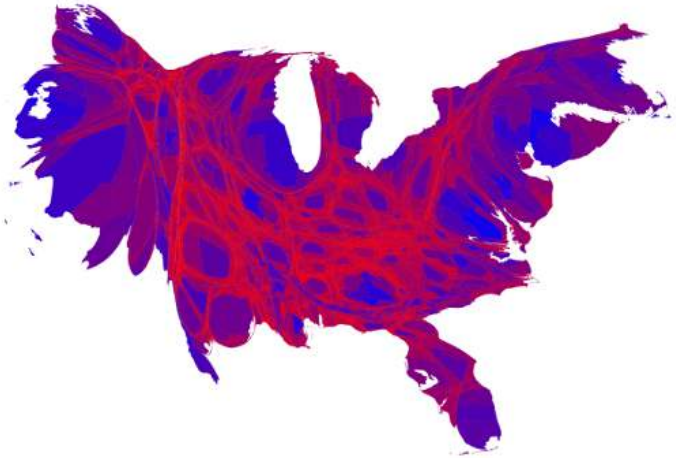
By David Zumbach

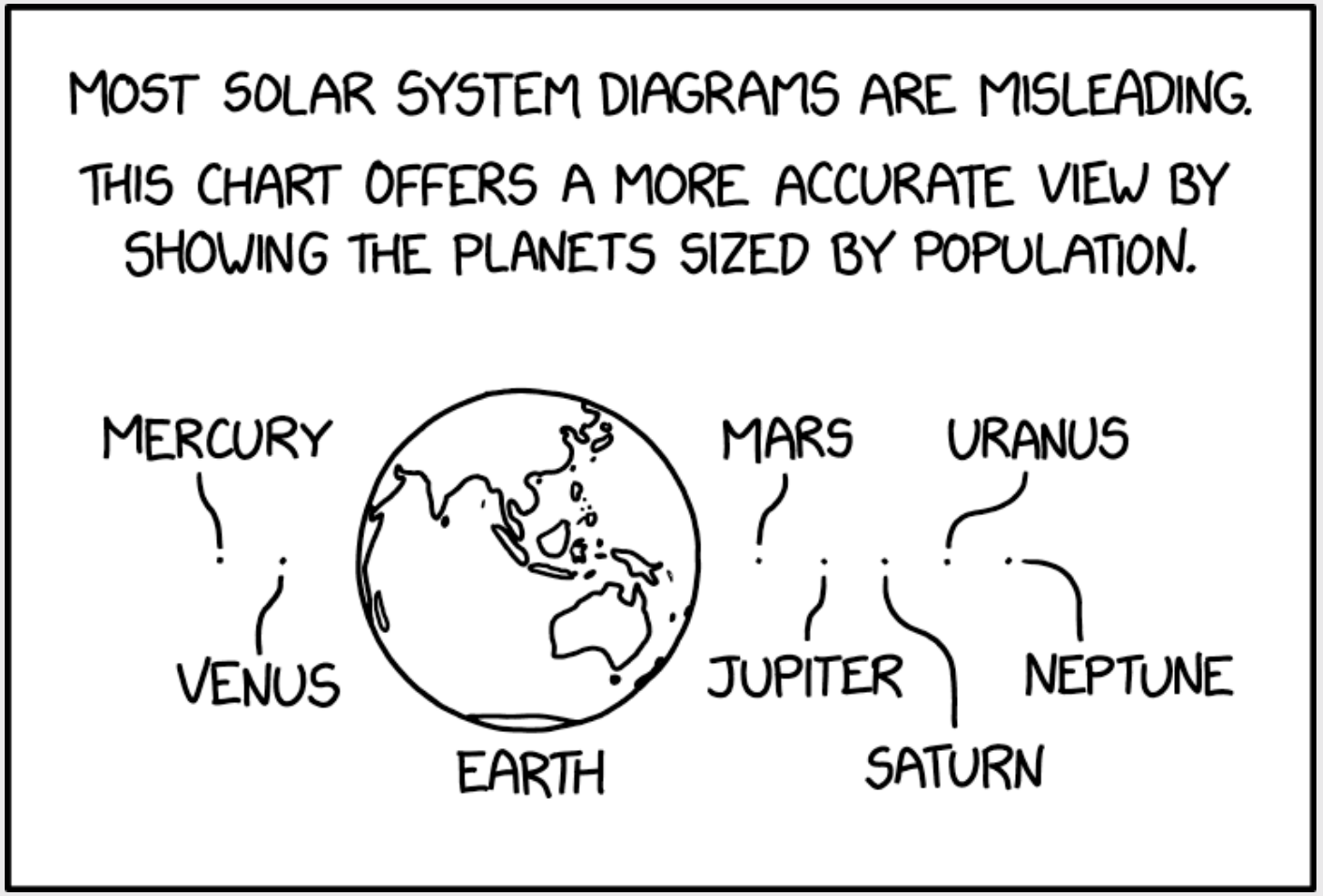
Land doesn't vote - people vote

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









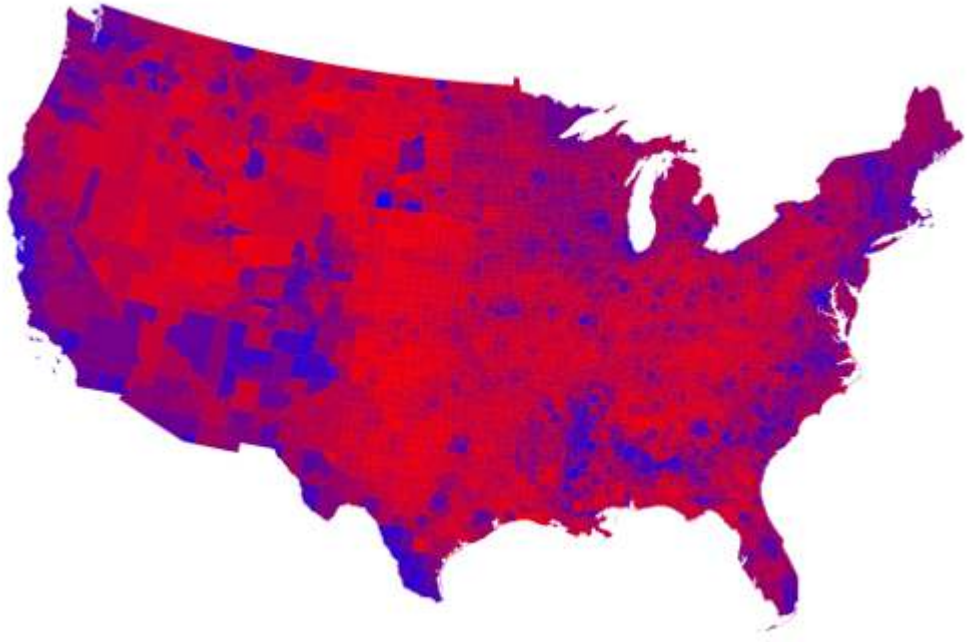
https://xkcd.com/2439/

Easy to lie with fake news

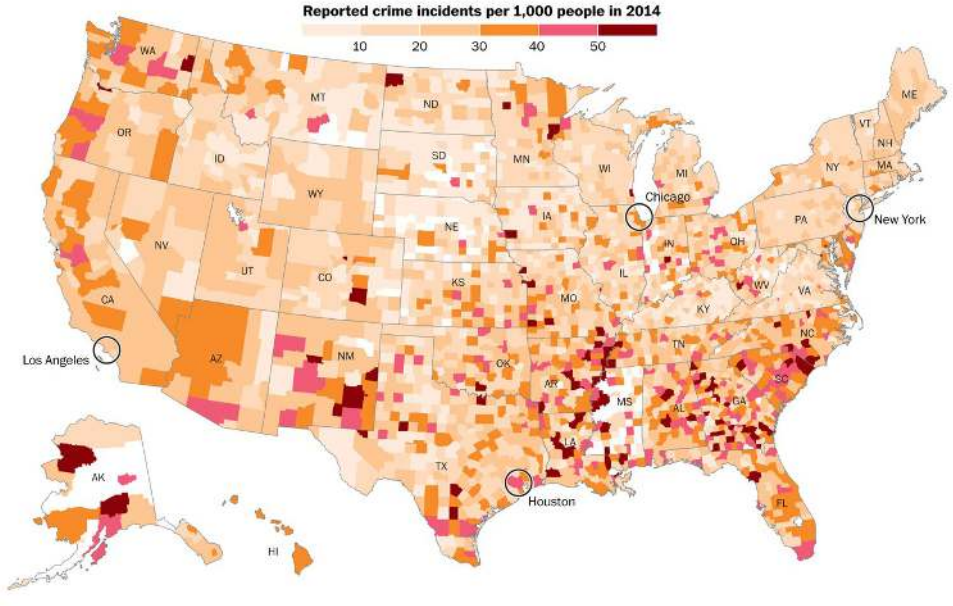
Bottom map is actually this map of the 2012 election

(here is what actual crime rates look like)

2016 Election map [source]

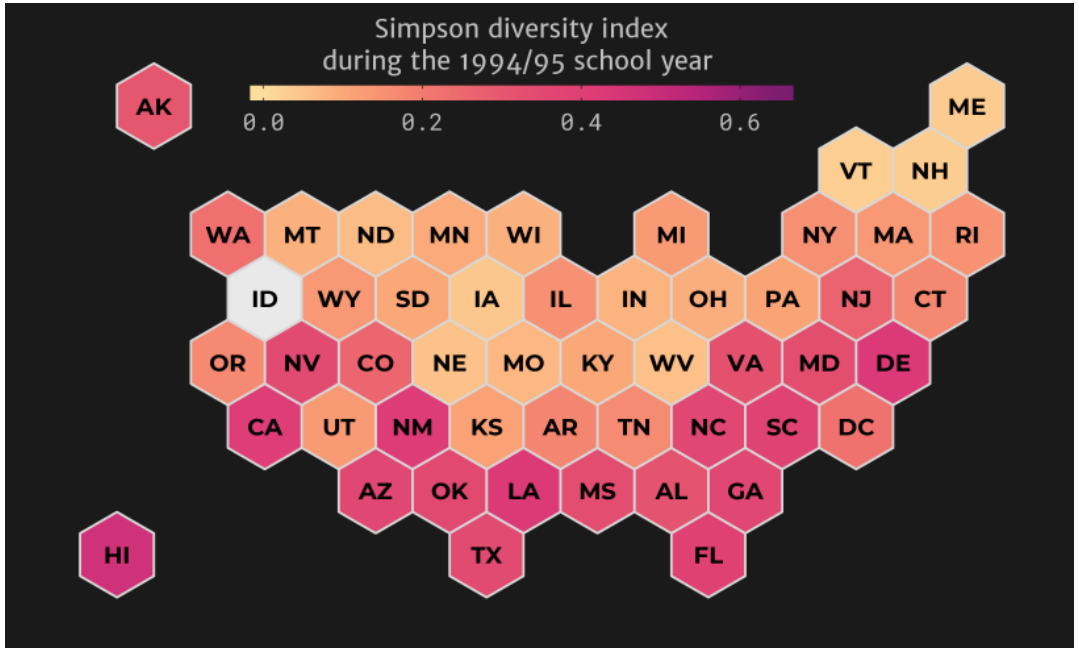


2014 Crime map [source]



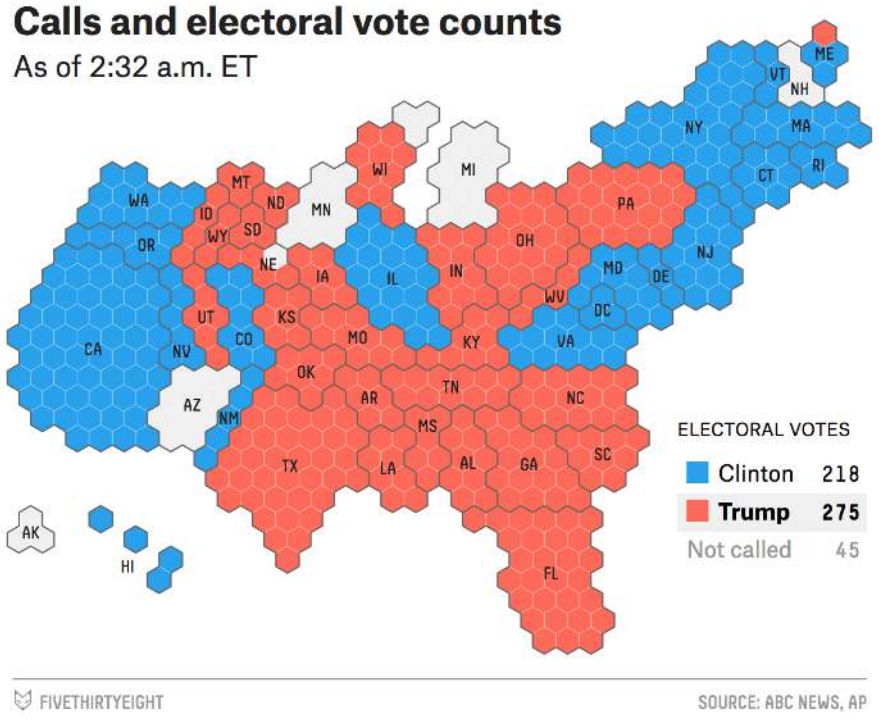
A choropleth alternative: hex maps

1994 Simpson Diversity Index in US Schools



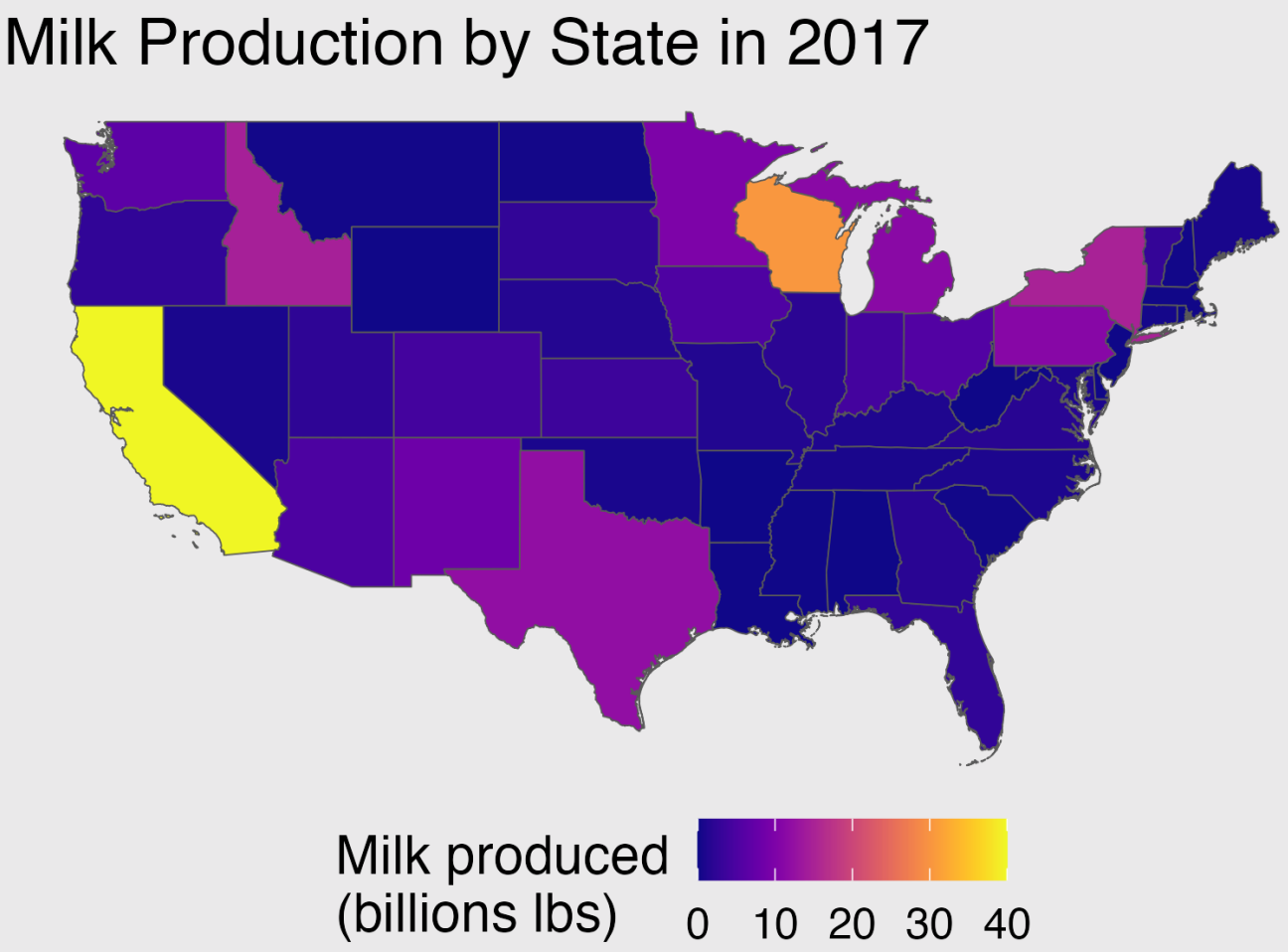
https://github.com/malcolmbarrett/designing_ggplots

2016 Electoral College



https://fivethirtyeight.com/

How to make a choropleth map



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

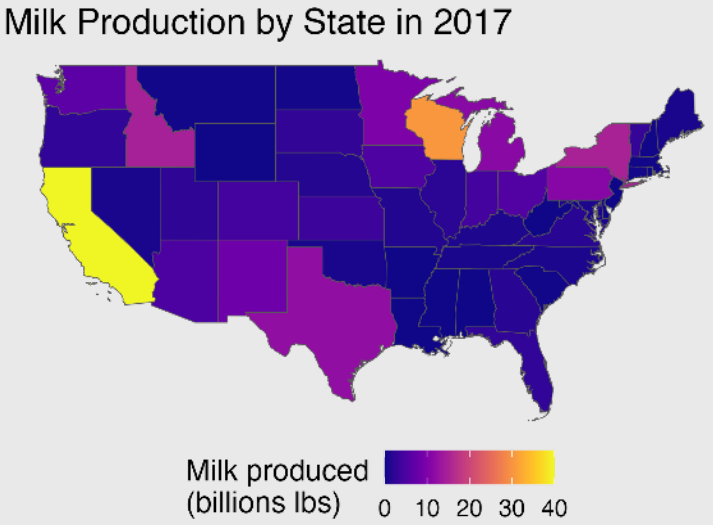
```
us_states <- ne_states(
    country = 'united states of america',
    returnclass = 'sf') %>%
    filter(! name %in% c('Alaska', 'Hawaii')) %>%
    left_join(milk_2017, by = 'name')
```

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

```
#> Simple feature collection with 6
#> Geometry type: MULTIPOLYGON
#> Dimension:     XY
#> Bounding box:  xmin: -124.7346 y
#> Geodetic CRS:  WGS 84
#>           name milk_produced
#> 1   Washington         6.526 MUL
#> 2        Idaho        14.627 MUL
#> 3      Montana         0.288 MUL
#> 4 North Dakota         0.345 MUL
#> 5    Minnesota         9.864 MUL
#> 6     Michigan        11.231 MUL
```

How to make a choropleth map

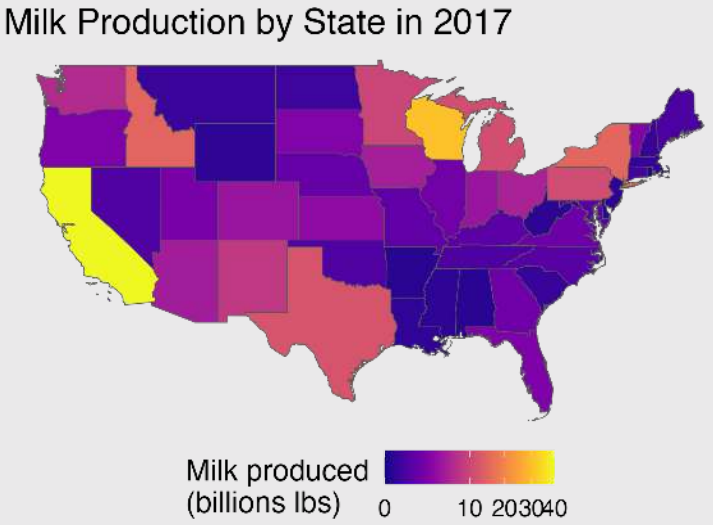
```
ggplot(us_states) +
  geom_sf(aes(fill = milk_produced)) +
  scale_fill_viridis(
    option = "plasma",
    limits = c(0, 40)) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  labs(
    fill = 'Milk produced\n(billions lbs)',
    title = 'Milk Production by State in 2017'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



How to make a choropleth map

```
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'
  )
```

Non-linear scale:

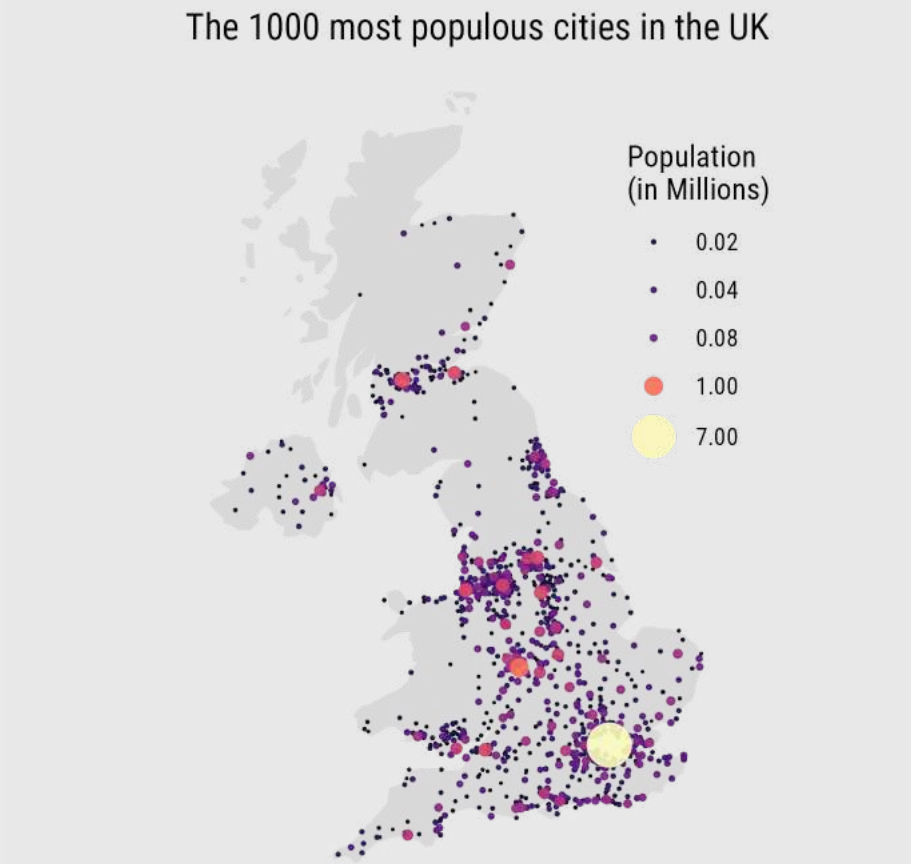


1. Choropleth maps

2. Point maps

Points as locations

Points encoding a variable

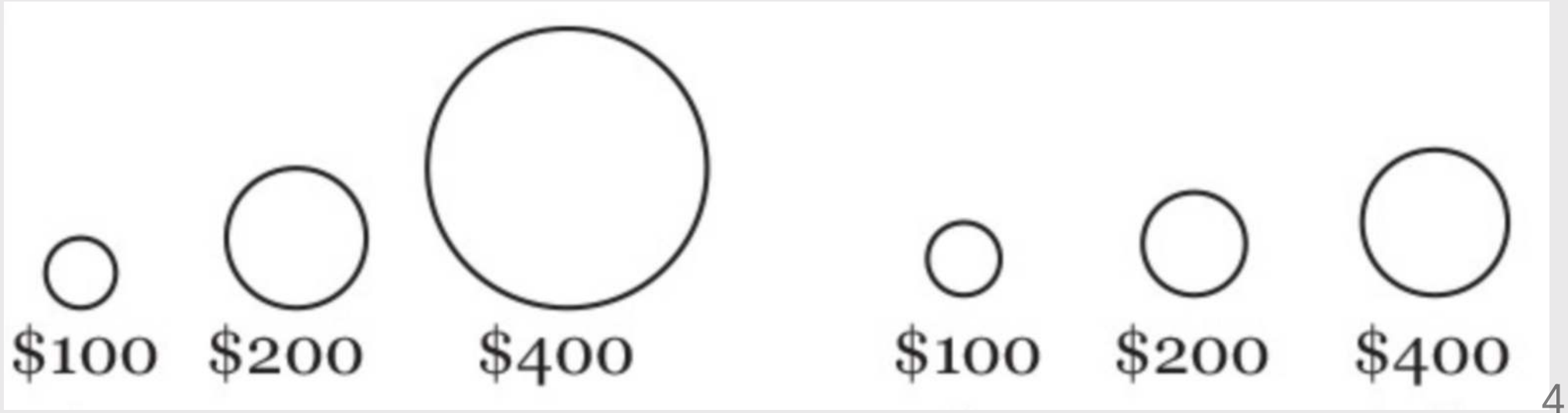


For point size, use **area**, not radius

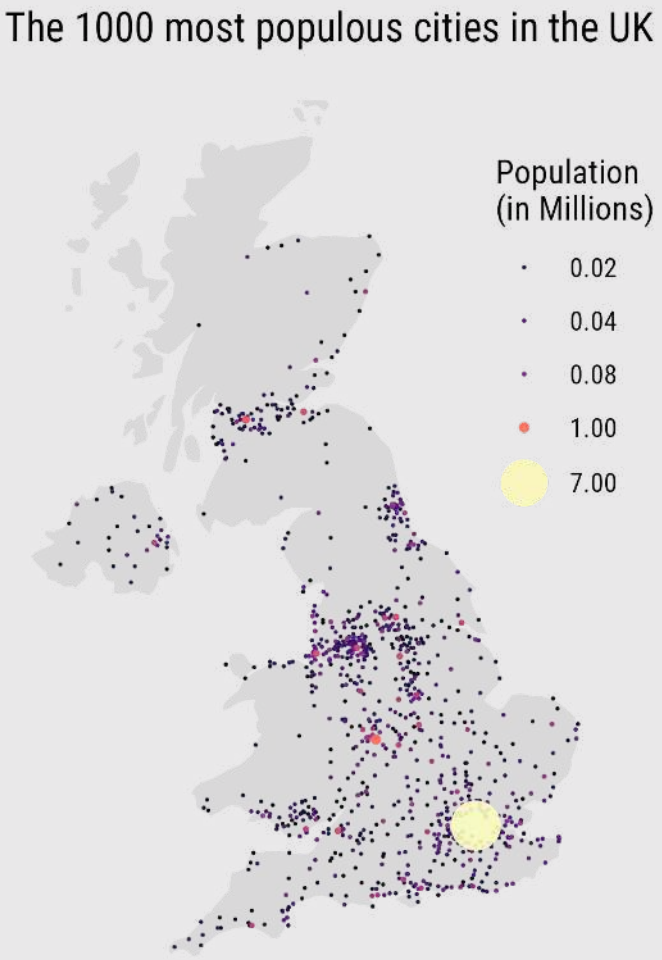
$$Area = \pi r^2$$

Radius

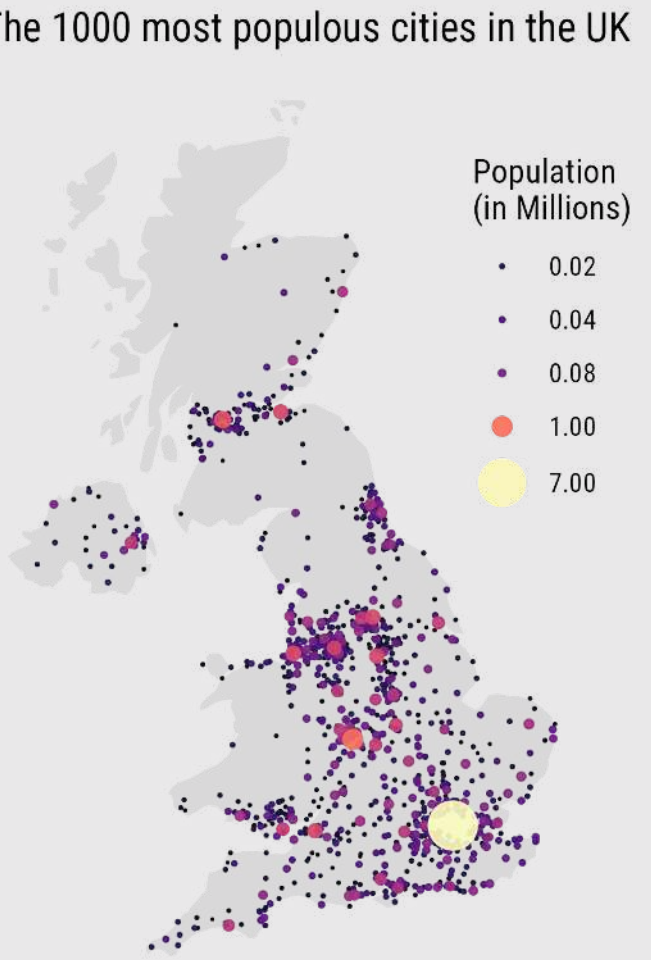
Area



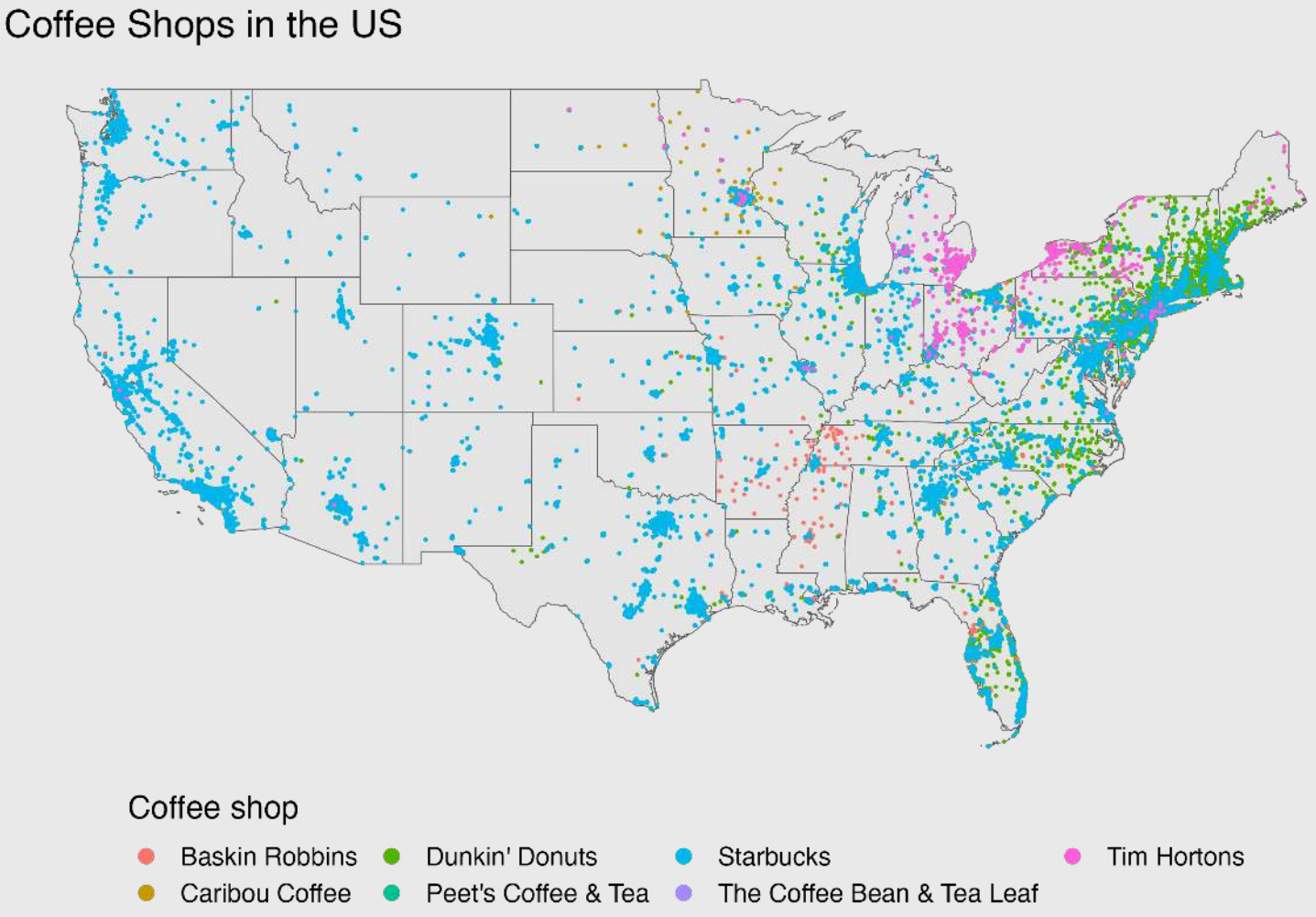
Radius


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



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
    )
```

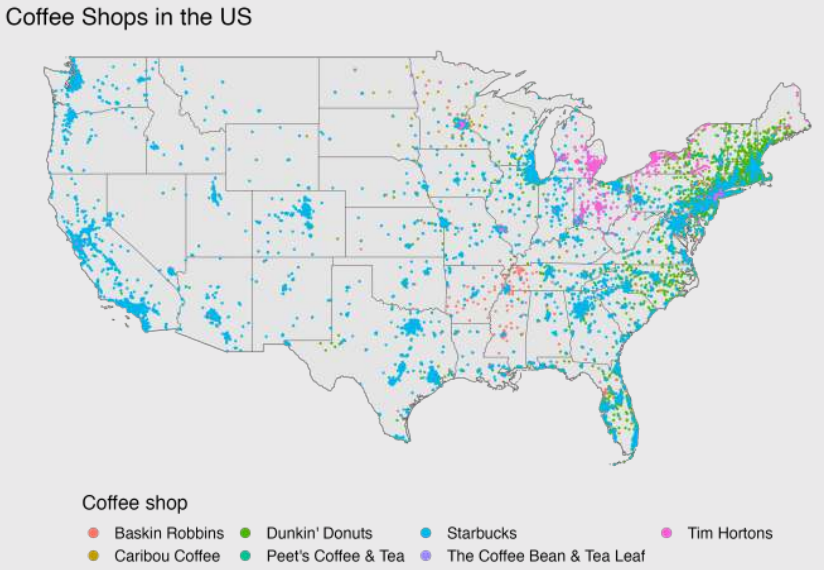
```
head(us_coffee_shops)
```

```
#> # A tibble: 6 × 8
#>   name             lat  long unique_id ci
#>   <chr>          <dbl> <dbl>     <dbl> <c
#> 1 Baskin Robbins  40.8 -73.4   3304448 Hu
#> 2 Baskin Robbins  42.1 -88.0  11342048 Ro
#> 3 Baskin Robbins  34.0 -84.5   3304169 Ma
#> 4 Baskin Robbins  29.8 -95.6   3304006 Ho
#> 5 Baskin Robbins  36.4 -89.5   3303959 Ti
#> 6 Baskin Robbins  40.7 -73.6   3304507 Me
```

How to add points to a map

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),
    size = 0.3
  ) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  labs(
    color = 'Coffee shop',
    title = 'Coffee Shops in the US'
  )
```



Your turn

15:00

Create this map of squirrels in NYC's Central Park using this data from the Squirrel Census:

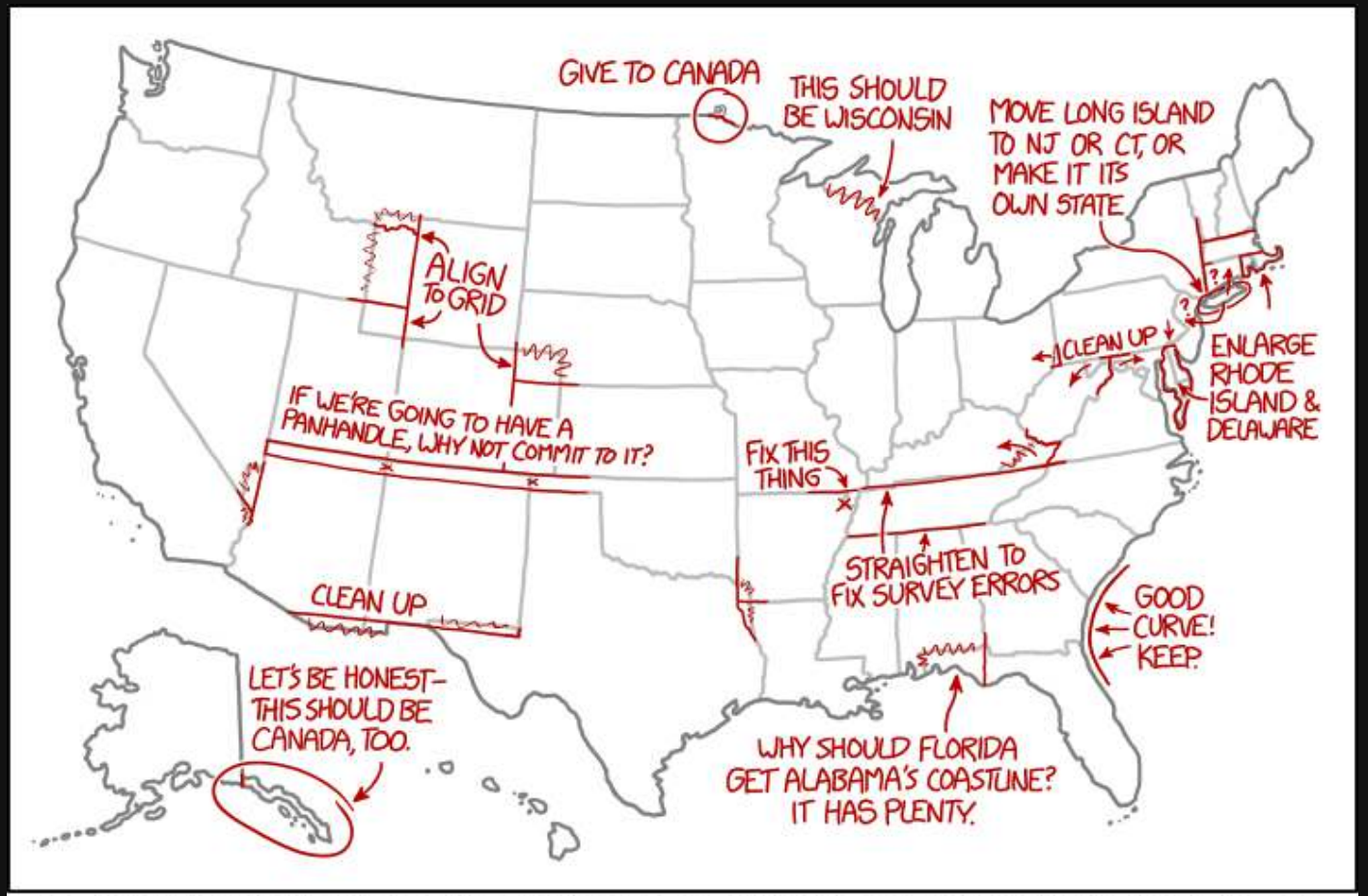
- The `CentralPark.shp` file in the `data/central_park` folder.
- The `nyc_squirrels.csv` file in the `data` folder.

Hint: The color is mapped to the `primary_fur_color` variable

(More about the Squirrel Census here)

Intermission

05:00



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.

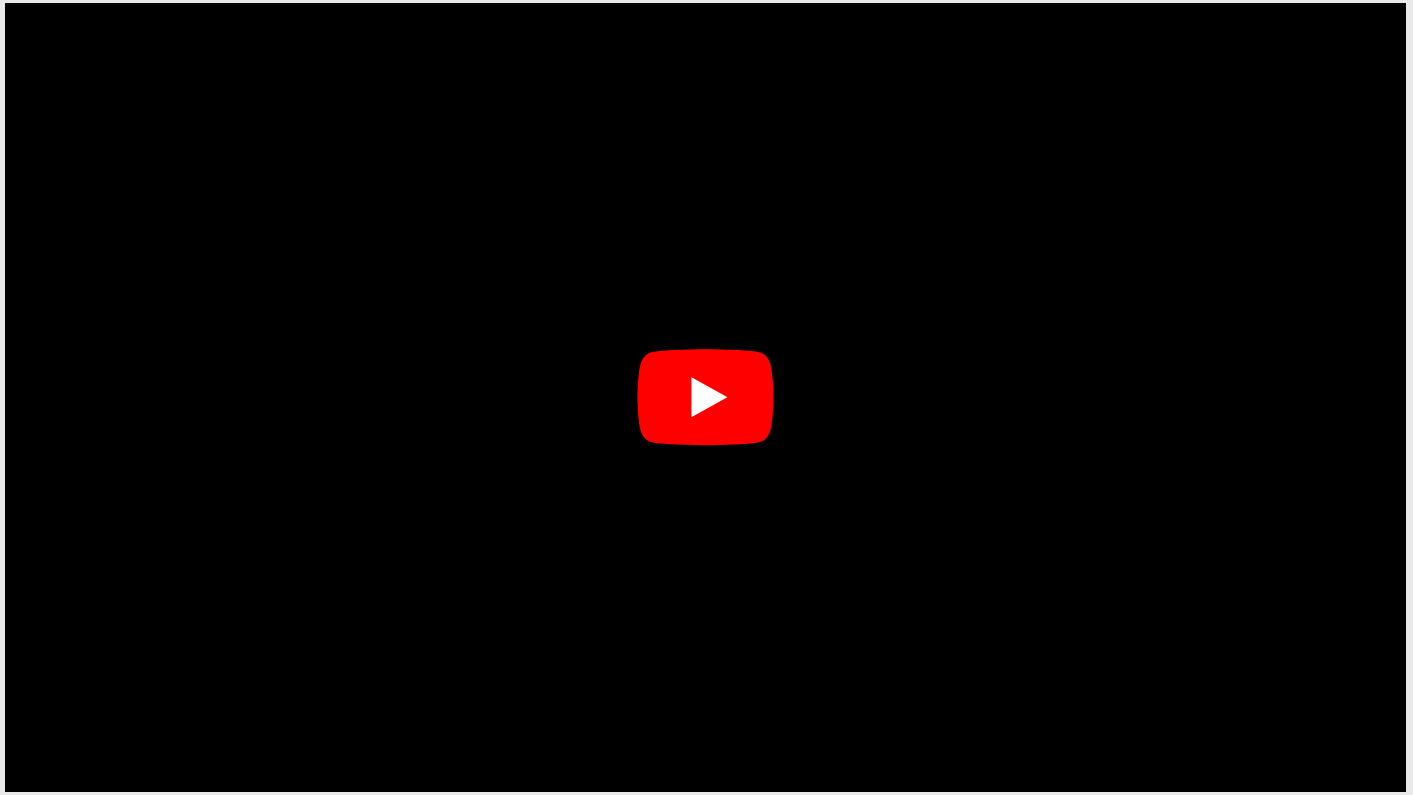
Week 11: *Maps*

1. Plotting maps
2. Adding data to maps

BREAK

3. Projections

What's a map projection?



What is the best projection?...it depends

1. Compare projections

2. Compare country sizes

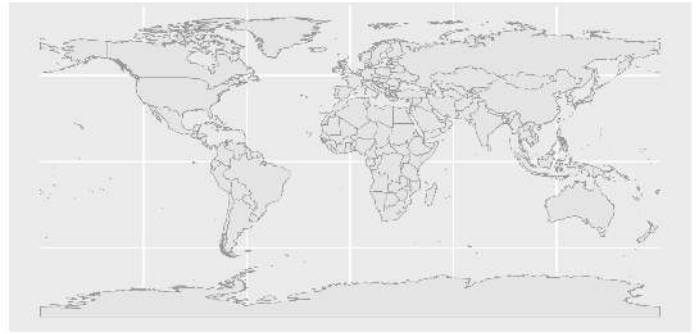
Using projections

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

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

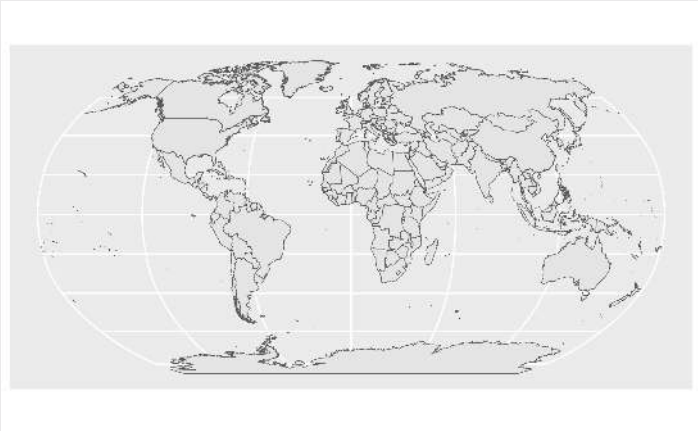
Default (long-lat)

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



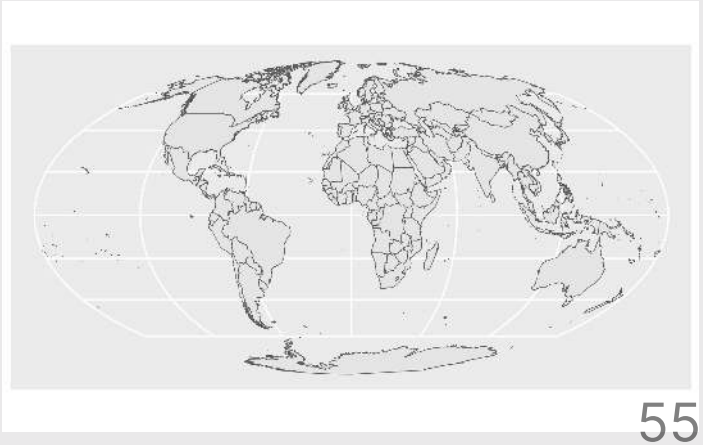
Robinson projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = "ESRI:54030")
```



Mollweide projection

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = "ESRI:54009")
```



Common Projections

ggplot layer:

```
coord_sf(crs = "ESRI:XXXX")
```

World

Code	Projection
`"ESRI:54030"`	**Robinson**
`"ESRI:54002"`	Equidistant cylindrical
`"ESRI:54004"`	Mercator
`"ESRI:54008"`	Sinusoidal
`"ESRI:54009"`	Mollweide

United States

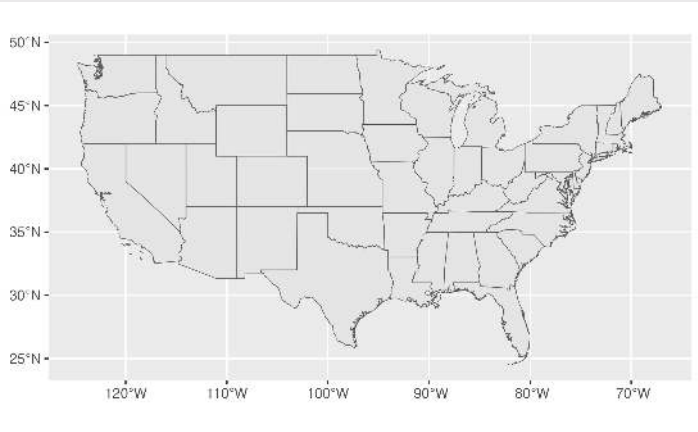
Code	Projection
`"ESRI:102003"`	**Albers**
`"ESRI:102004"`	Lambert Conformal Conic
`4269`	NAD 83

US projections

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

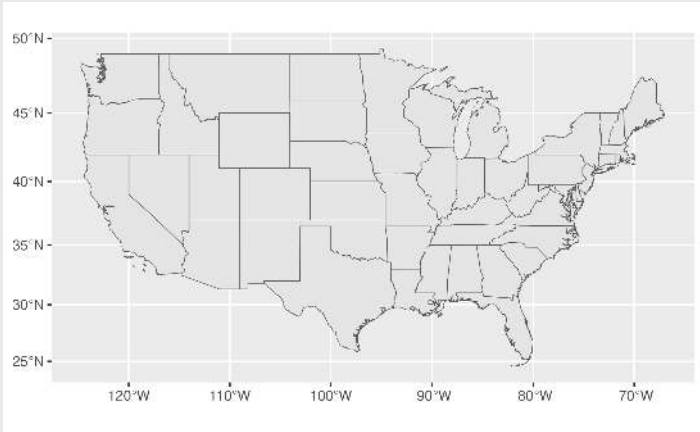
NAD 83 projection

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



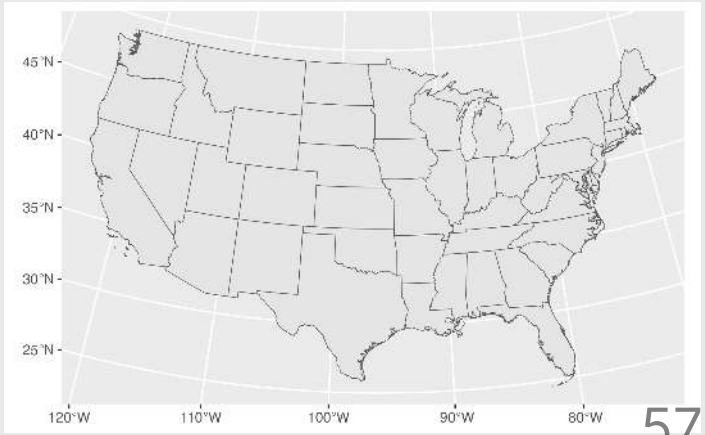
Mercator

```
ggplot(data = world) +
    geom_sf() +
    coord_sf(crs = "ESRI:54004
```



Albers

```
ggplot(data = us_states_cont)
    geom_sf() +
    coord_sf(crs = "ESRI:10200
```

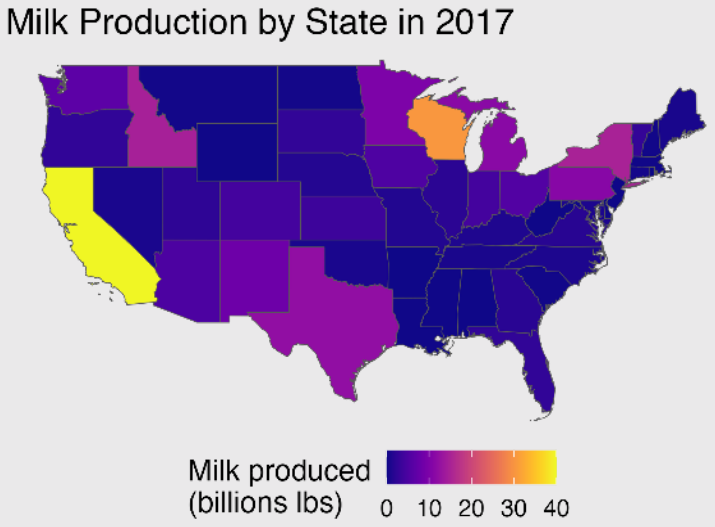


Mapping data to projections - choropleth map

```r
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')

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

```r
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')

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'
  ) +
  coord_sf(crs = "ESRI:102003")
```

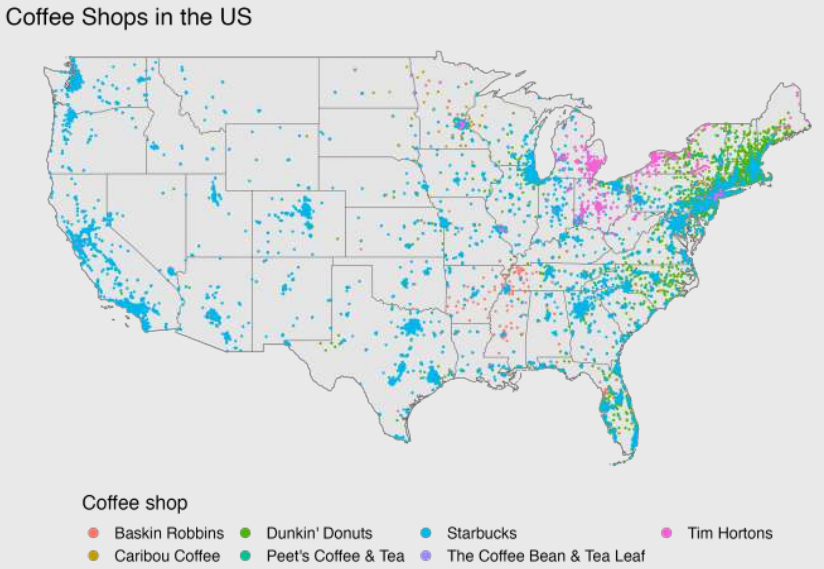
Albers Projection

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)

ggplot() +
  geom_sf(data = us_states_cont) +
  geom_point(
    data = us_coffee_shops,
    aes(x = long, y = lat, color = name),
    size = 0.3) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  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)

ggplot() +
  geom_sf(data = us_states_cont) +
  geom_point(
    data = us_coffee_shops,
    aes(x = long, y = lat, color = name),
    size = 0.3) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  labs(
    color = 'Coffee shop',
    title = 'Coffee Shops in the US'
  ) +
  coord_sf(crs = "ESRI:102003")
```

Fail!

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))
```

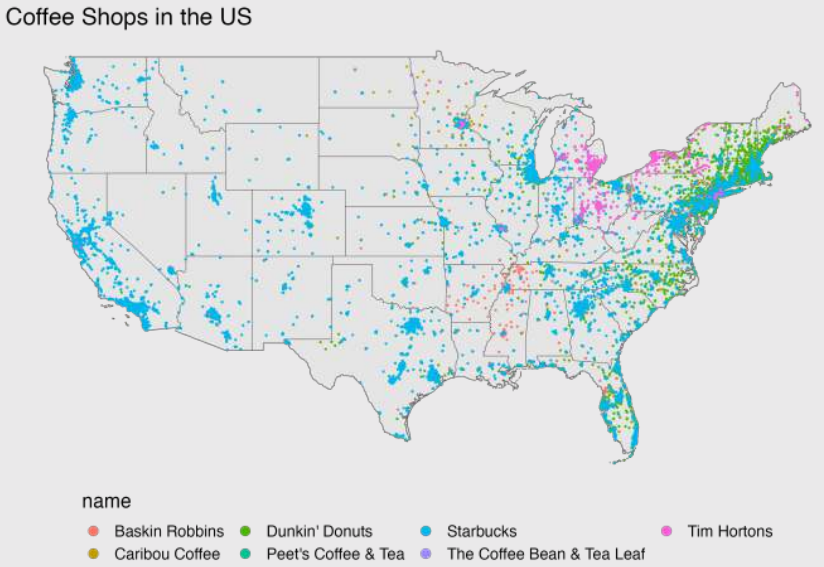
```
head(us_coffee_shops_sf)
```

```
#> Simple feature collection with 6
#> Geometry type: POINT
#> Dimension:     XY
#> Bounding box:  xmin: -95.60337 y
#> Geodetic CRS:  WGS 84
#> # A tibble: 6 × 7
#>   name            unique_id city
#>   <chr>               <dbl> <chr>
#> 1 Baskin Robbins   3304448 Hunti
#> 2 Baskin Robbins  11342048 Rolli
#> 3 Baskin Robbins   3304169 Marie
#> 4 Baskin Robbins   3304006 Houst
#> 5 Baskin Robbins   3303959 Tipto
#> 6 Baskin Robbins   3304507 Merri
```

Mapping data to projections - points

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

```
ggplot() +
  geom_sf(data = us_states_cont) +
  geom_sf(
    data = us_coffee_shops_sf,
    aes(color = name),
    size = 0.3) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  labs(
    fill = 'Coffee shop',
    title = 'Coffee Shops in the US'
  )
```



Mapping data to projections - points

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

```
ggplot() +
  geom_sf(data = us_states_cont) +
  geom_sf(
    data = us_coffee_shops_sf,
    aes(color = name),
    size = 0.3) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  labs(
    fill = 'Coffee shop',
    title = 'Coffee Shops in the US'
  ) +
  coord_sf(crs = "ESRI:102003")
```

Albers Projection

Mapping data to projections - points

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

```
ggplot() +
  geom_sf(data = us_states_cont) +
  geom_sf(
    data = us_coffee_shops_sf,
    aes(color = name),
    size = 0.3) +
  theme_void(base_size = 15) +
  theme(legend.position = 'bottom') +
  guides(color = guide_legend(
    # Move legend title to top
    title.position = "top",
    # Increase legend point size
    override.aes = list(size = 3))) +
  labs(
    fill = 'Coffee shop',
    title = 'Coffee Shops in the US'
  ) +
  coord_sf(crs = "ESRI:102004")
```

LCC Projection

20:00

Your turn

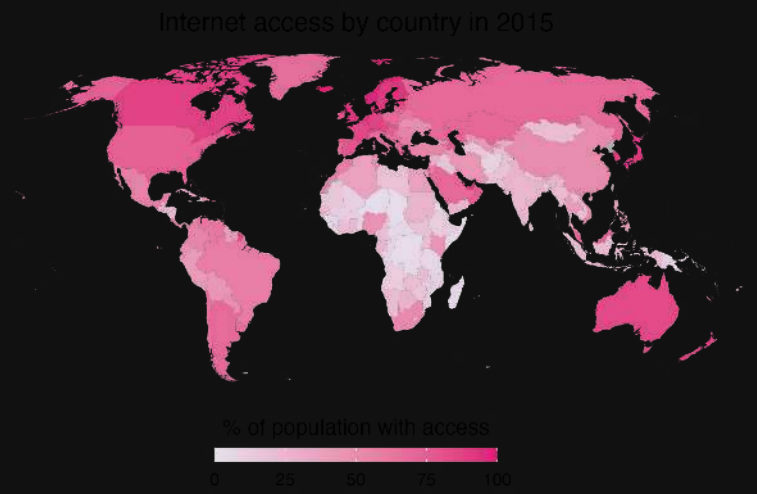
Use the `internet_users_country.csv` data and the `world` data frame from the **rnaturalearth** library to create these two versions of internet access by country in 2015.

Hints:

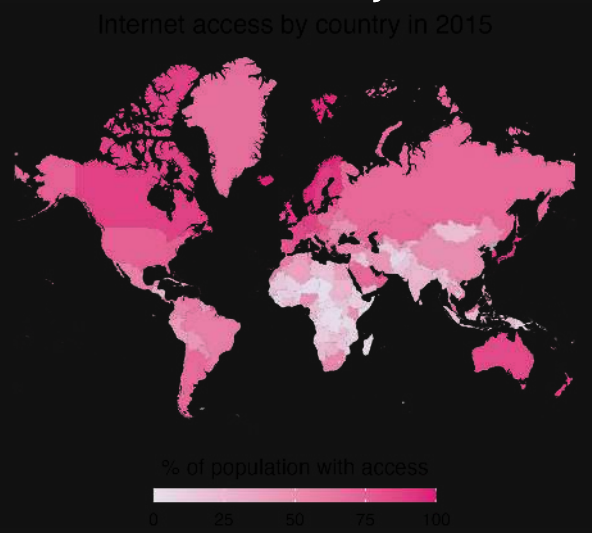
- The `iso_a3` variable in the `worlds` data frame corresponds with the `code` variable in the `internet_users_country.csv` data frame (use this for joining).
- Use `scale_fill_gradient()` to fill the color:

```
scale_fill_gradient(
    low = "#e7e1ef",
    high = "#dd1c77",
    na.value = "grey70",
    limits = c(0, 100))
```

Robinson Projection



Mercator Projection



Remember: Mini Project 3 Due 11/21

(That's 2 weeks)

https://eda.seas.gwu.edu/2023-Fall/mini/3-redesign.html

Extra practice

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.: