

# Space

## Session 12

PMAP 8921: Data Visualization with R  
Andrew Young School of Policy Studies  
Spring 2025

# Plan for today

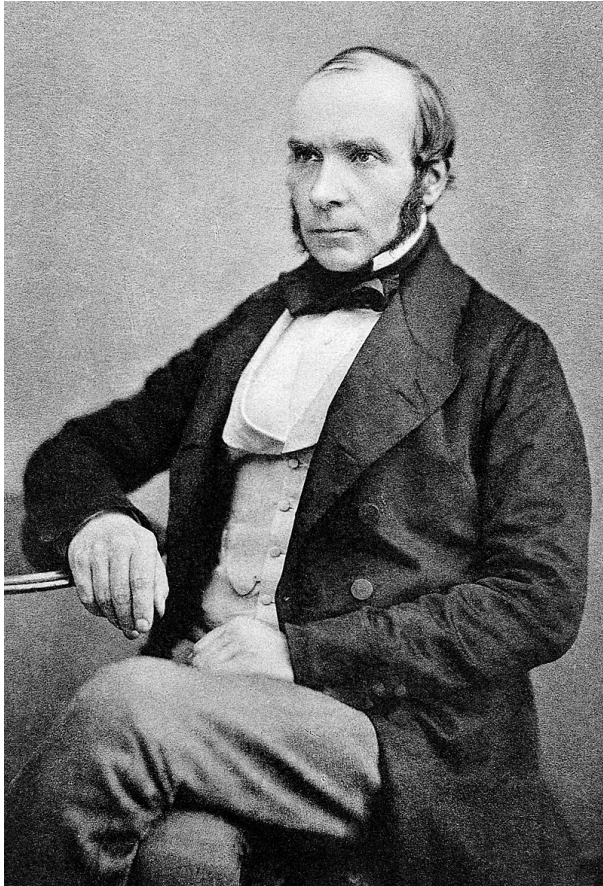
Maps and truth

Putting data on maps

GIS in R with {sf}

# Maps and truth

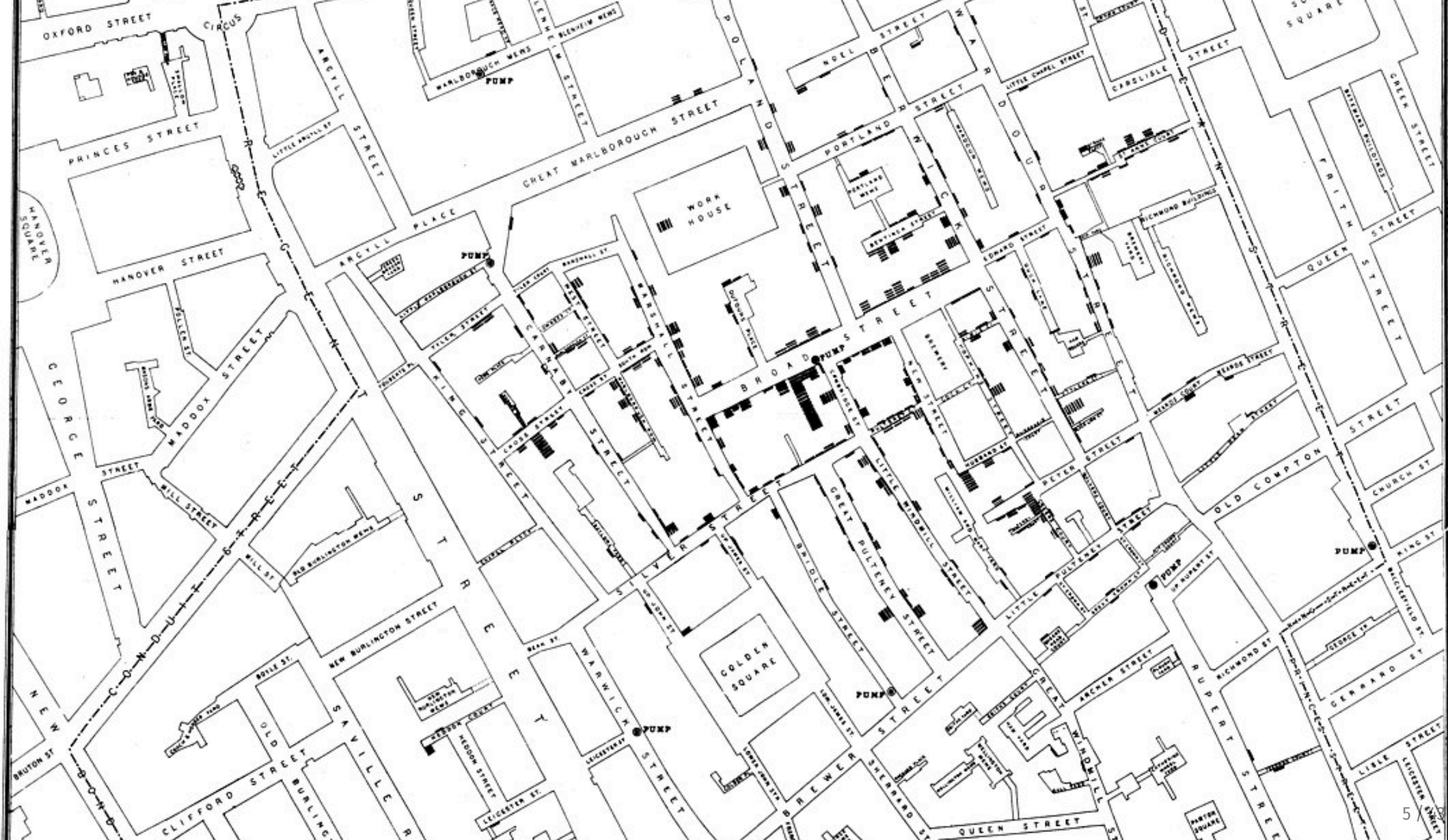
# John Snow and 1854 cholera epidemic



This Jo(h)n Snow knows things

**10% of the population of Soho died in a week (!!)**

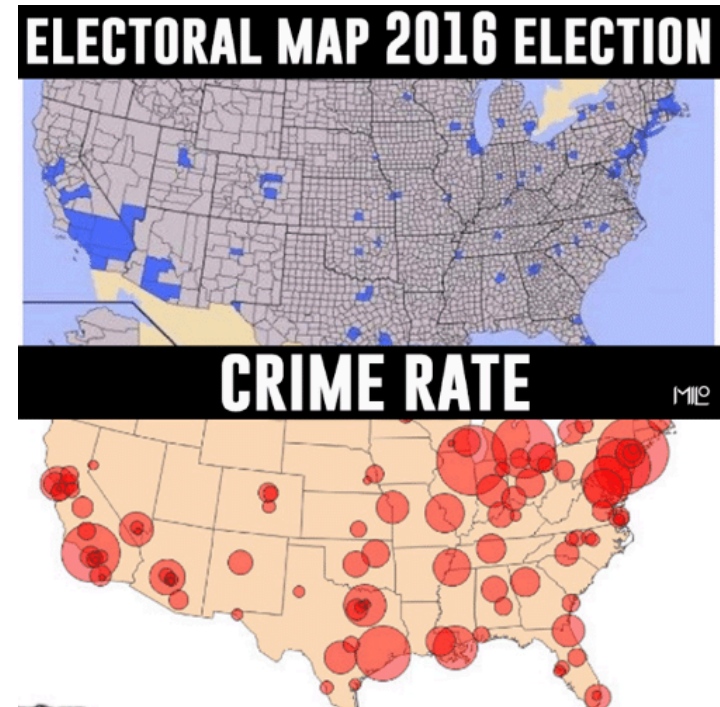
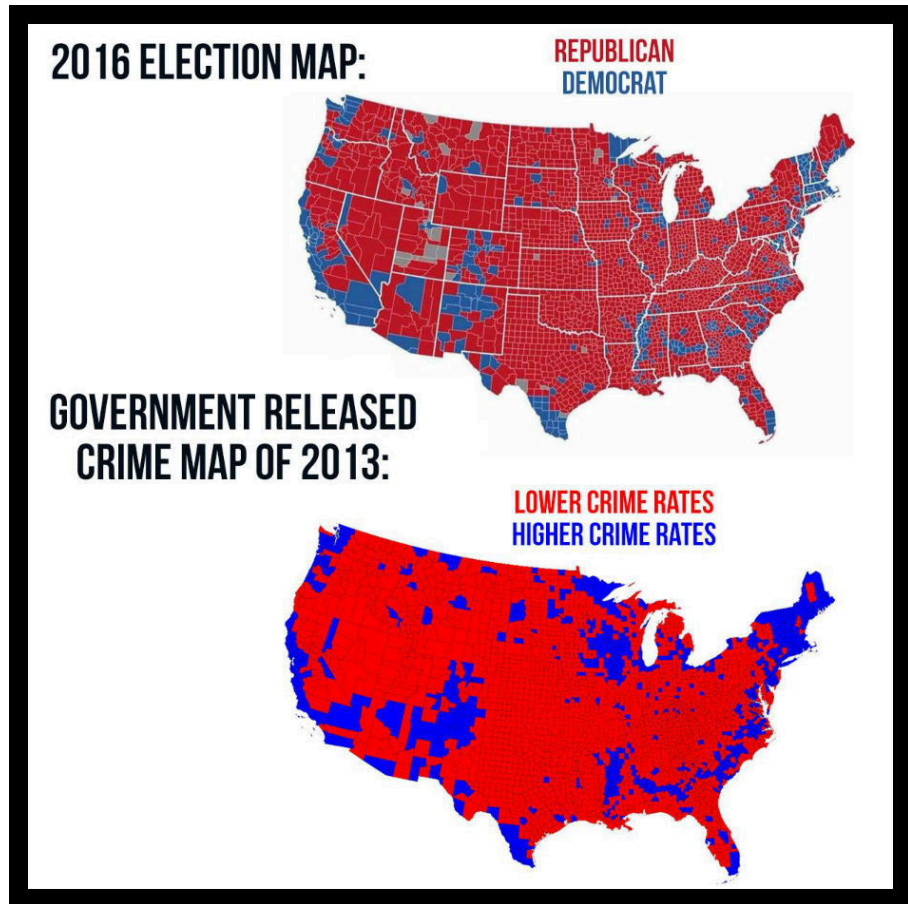
**Miasma theory said it was because the air was bad**



# The Broad Street pump



# Outright lies



Democrats are as consistent in voting as they are in crime I guess...

# Fake maps and junk maps

FAST COMPANY

09-05-18

## The next great fake news threat? Bot-designed maps

A new study reveals how maps go viral—and why they've become the perfect tool for misinformation.



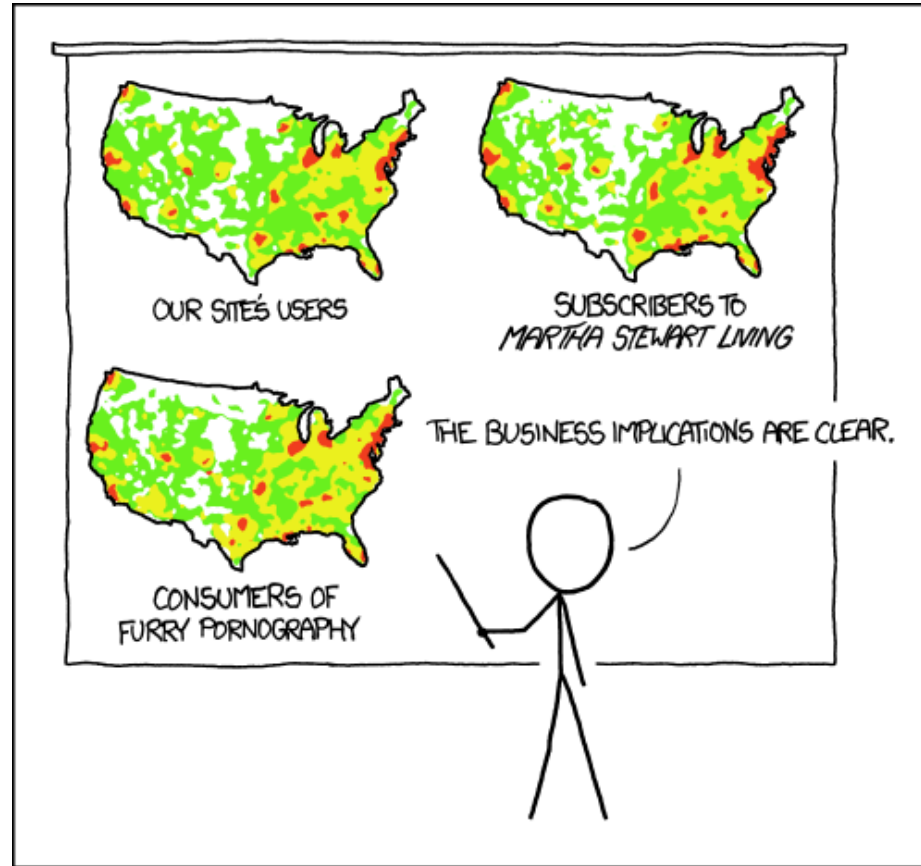
“The next great fake news threat? Bot-designed maps”

## AMERICA'S FAVORITE CANDY BY STATE





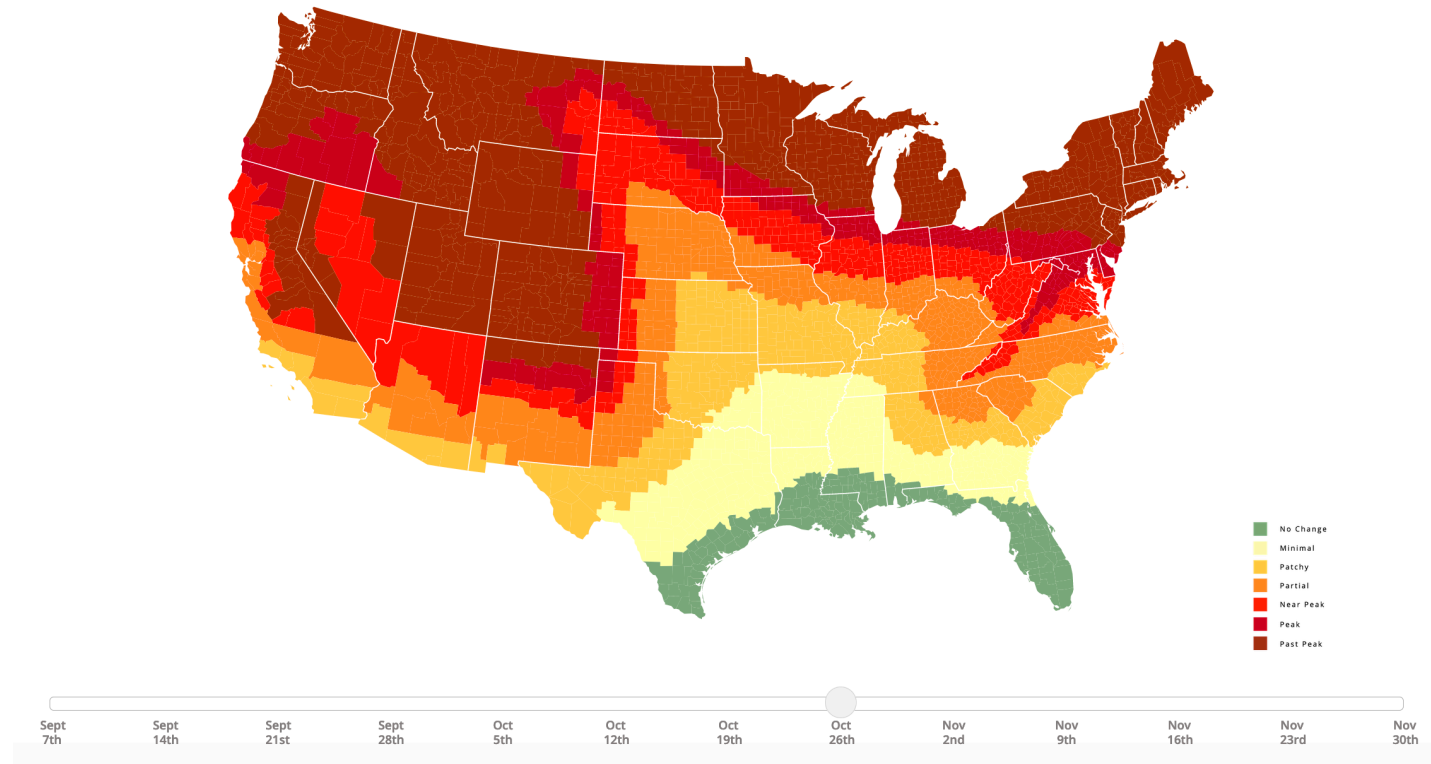
# Points can be useless



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

# Choropleths can be great

THE 2019  
Fall Foliage Prediction Map



Smoky Mountains 2019 Fall Foliage Prediction Map

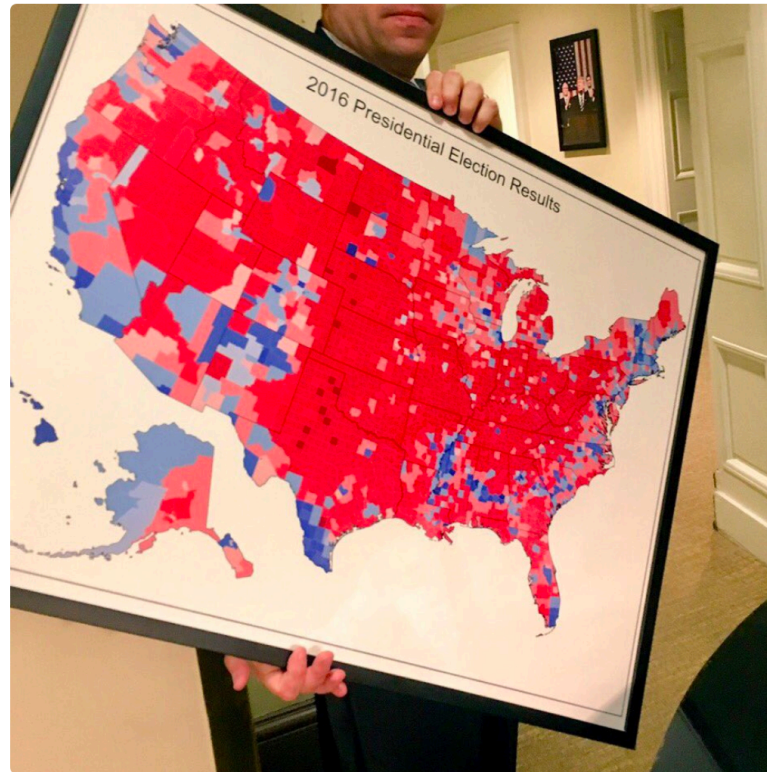
# Choropleths can distort



**Trey Yingst**   
@TreyYingst



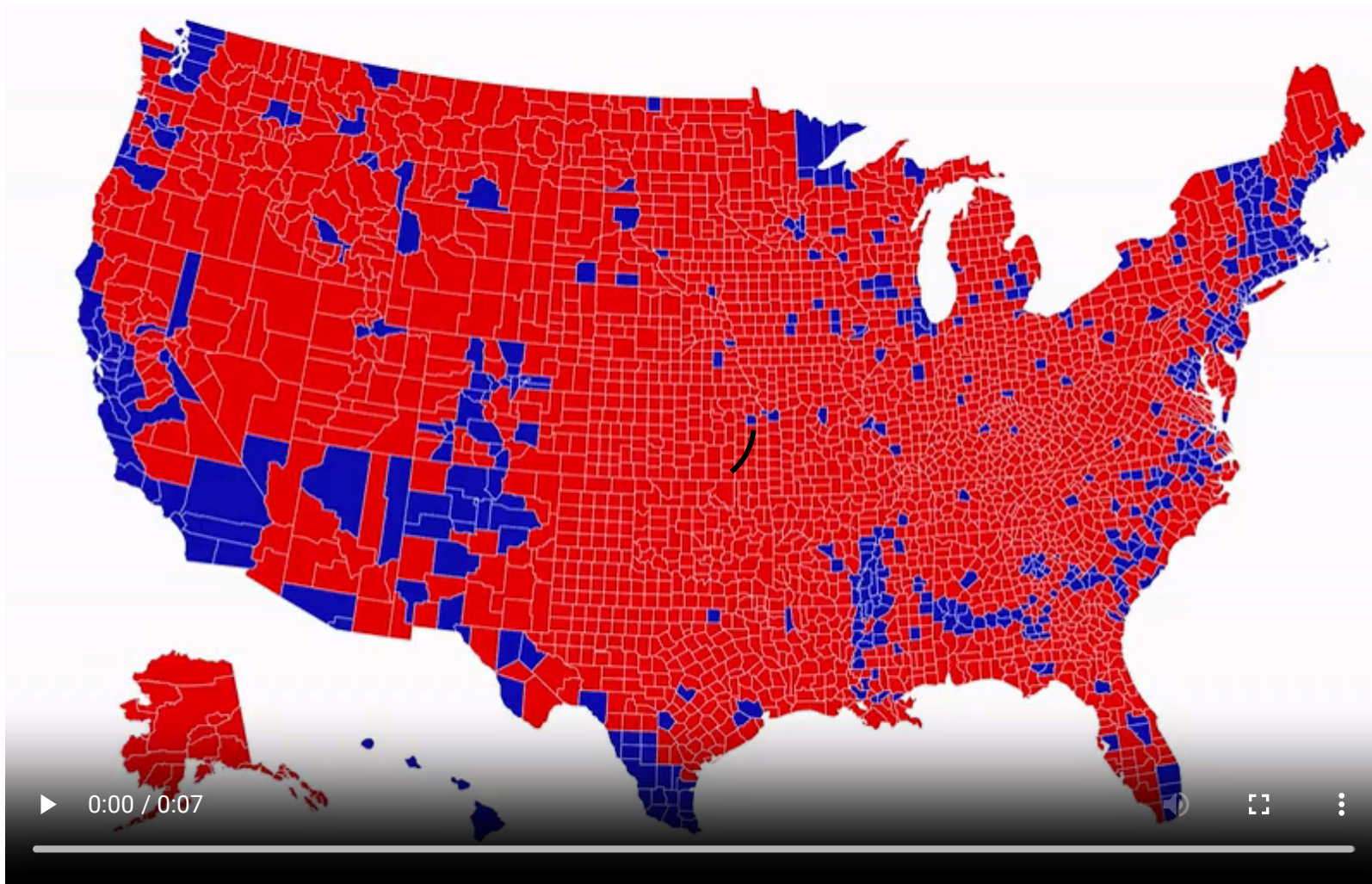
Spotted: A map to be hung somewhere in the West Wing



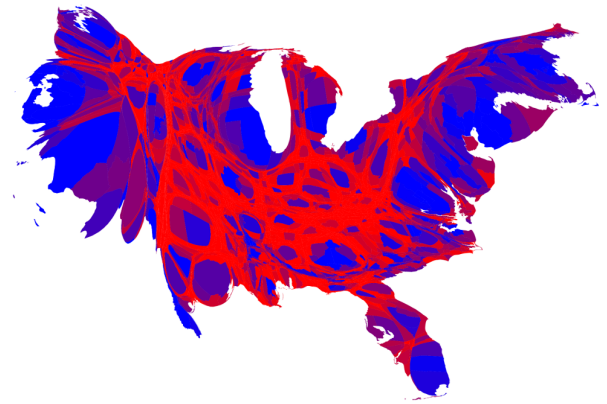
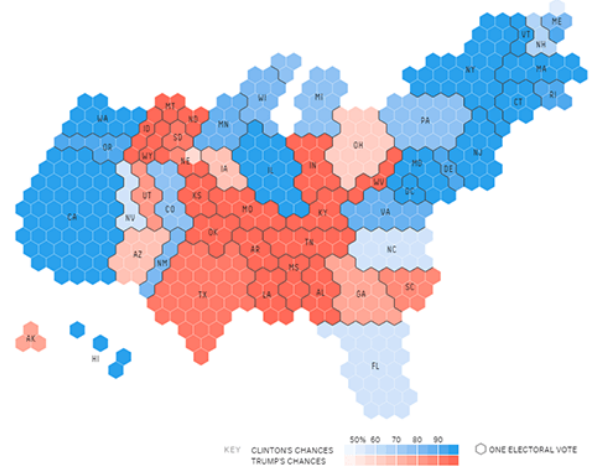
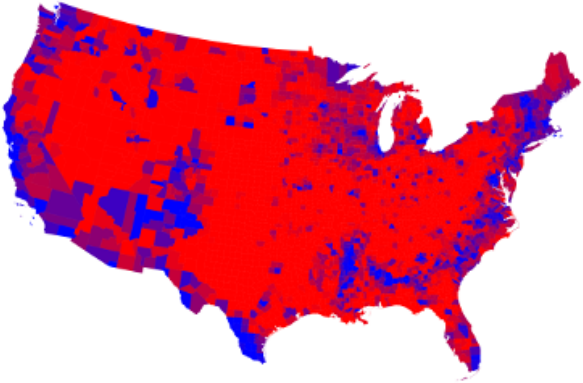
7,929 10:03 AM - May 11, 2017



# Land doesn't vote

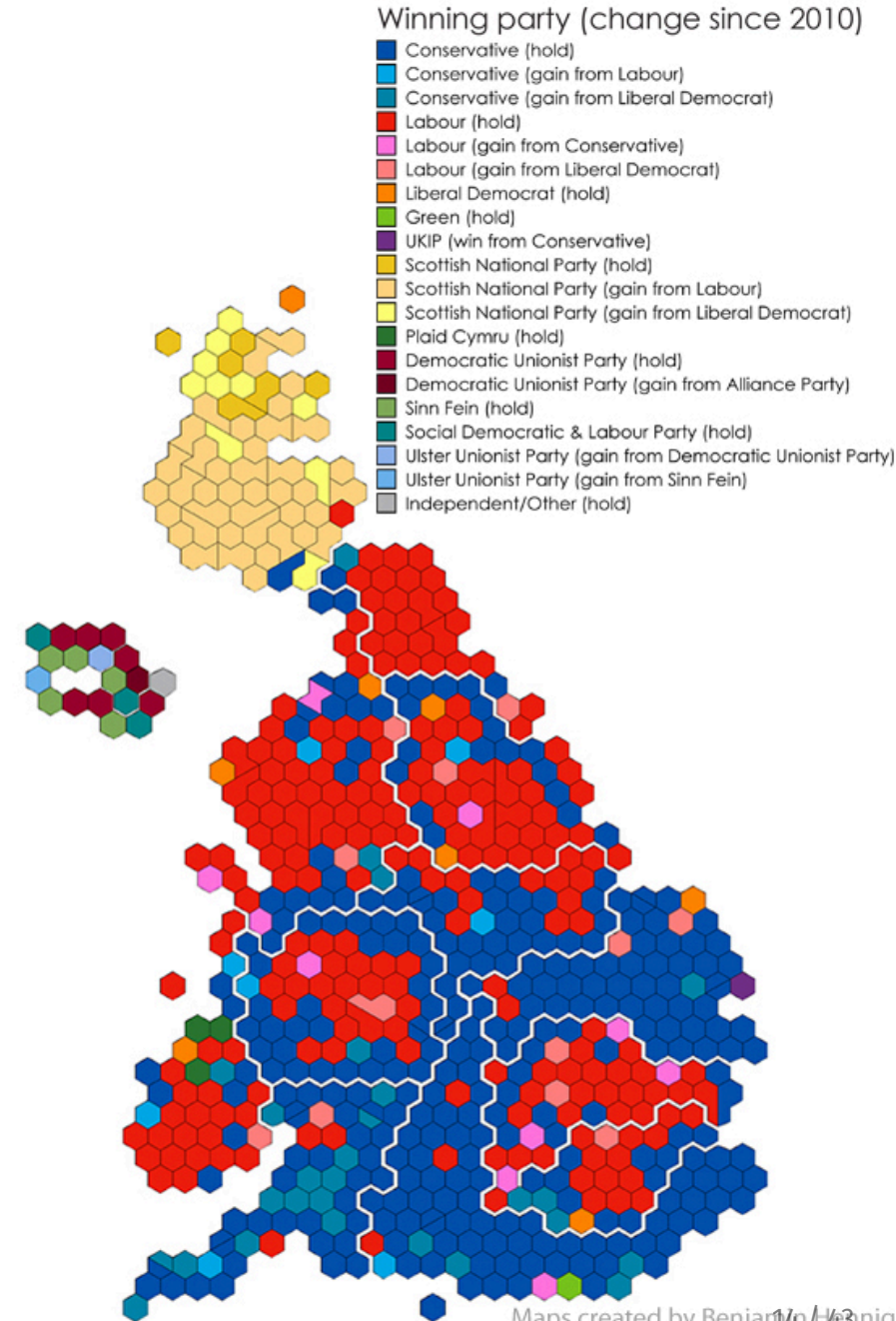
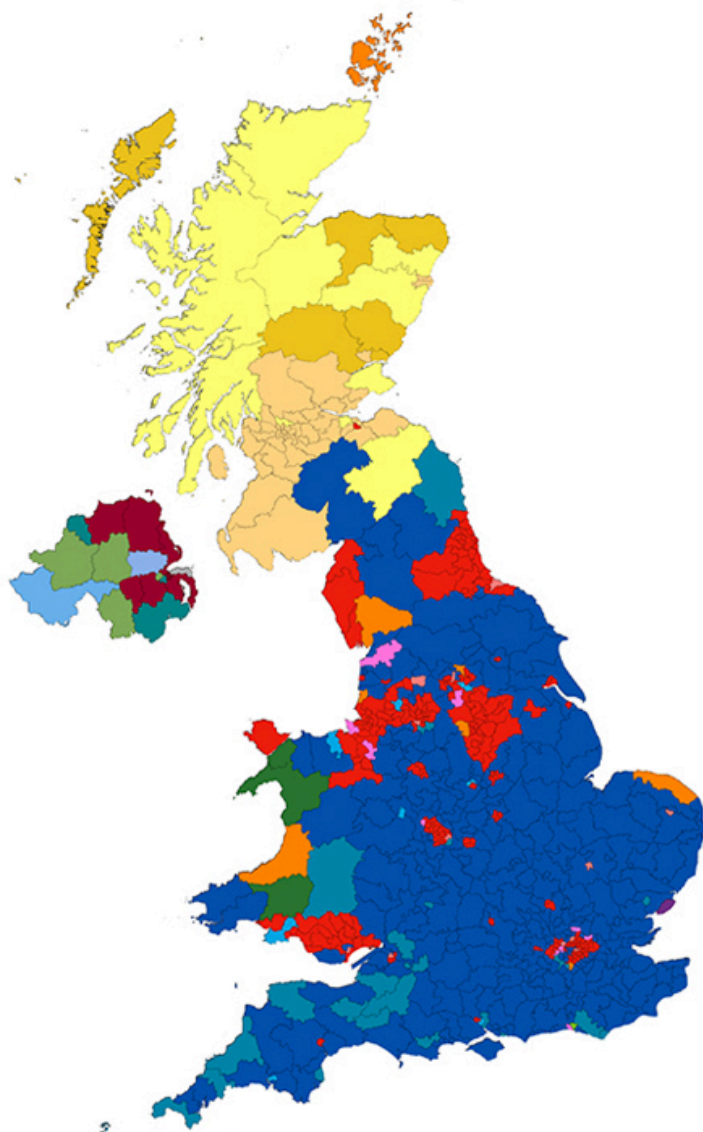


# Cartograms



# Electoral Doctrine

Mapping the  
2015 UK General Election



- Winning party (change since 2010)
- Conservative (hold)
  - Conservative (gain from Labour)
  - Conservative (gain from Liberal Democrat)
  - Labour (hold)
  - Labour (gain from Conservative)
  - Labour (gain from Liberal Democrat)
  - Liberal Democrat (hold)
  - Green (hold)
  - UKIP (win from Conservative)
  - Scottish National Party (hold)
  - Scottish National Party (gain from Labour)
  - Scottish National Party (gain from Liberal Democrat)
  - Plaid Cymru (hold)
  - Democratic Unionist Party (hold)
  - Democratic Unionist Party (gain from Alliance Party)
  - Sinn Fein (hold)
  - Social Democratic & Labour Party (hold)
  - Ulster Unionist Party (gain from Democratic Unionist Party)
  - Ulster Unionist Party (gain from Sinn Fein)
  - Independent/Other (hold)

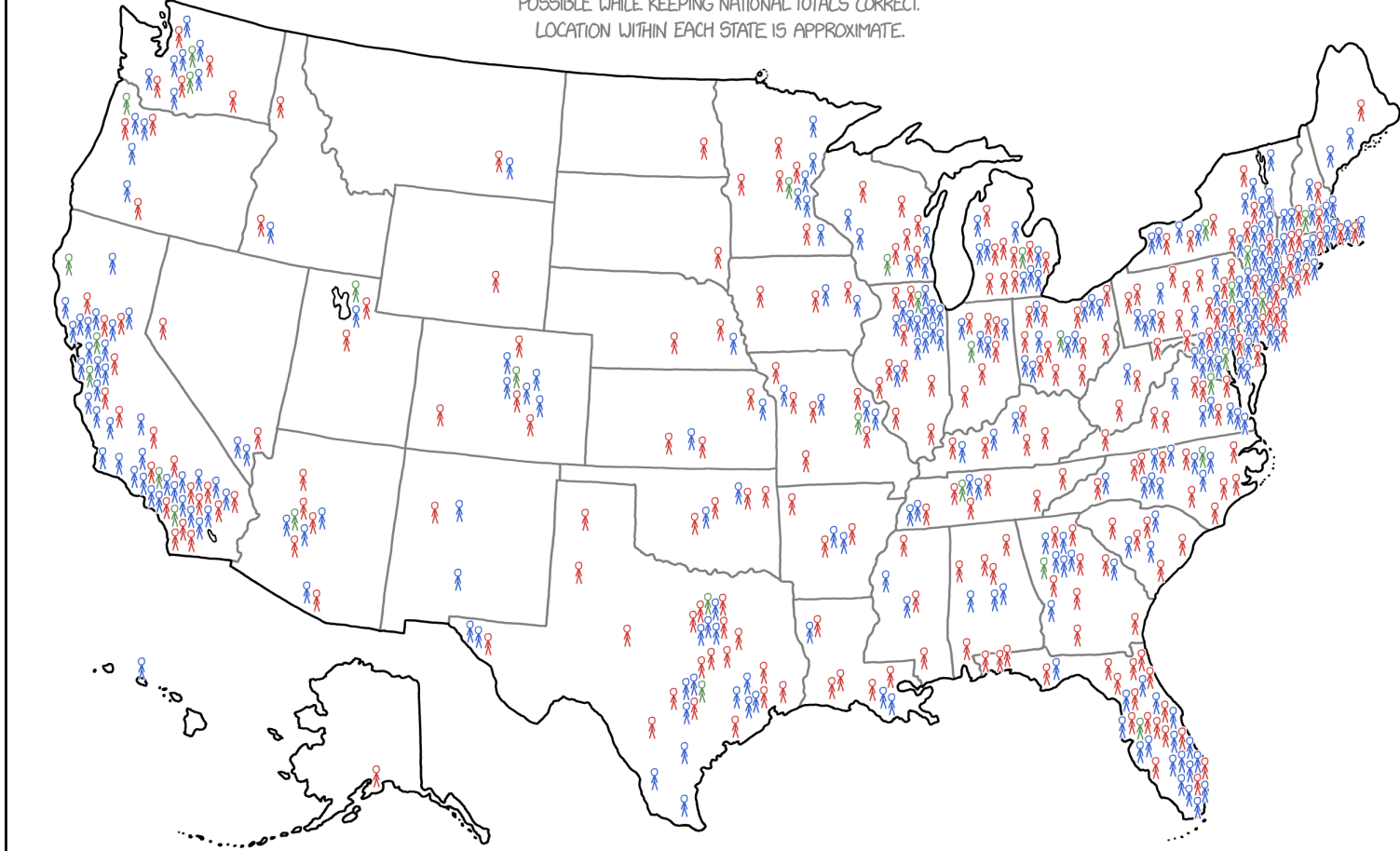
# 2016 ELECTION MAP

EACH FIGURE REPRESENTS 250,000 VOTES

TRUMP CLINTON OTHER

VOTES ARE DISTRIBUTED BY STATE AS ACCURATELY AS POSSIBLE WHILE KEEPING NATIONAL TOTALS CORRECT.

LOCATION WITHIN EACH STATE IS APPROXIMATE.



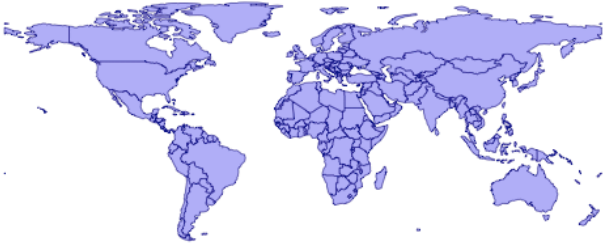
# Projections

## Animated world projections



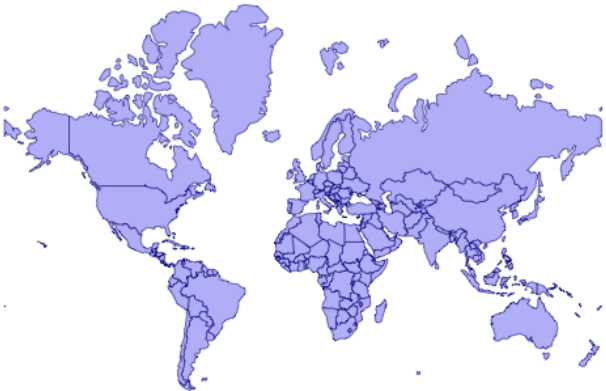
# World projections

## Longitude-latitude



```
crs = "+proj=longlat +ellps=WGS84"
```

## Mercator



```
crs = "+proj=merc"
```

## Gall-Peters



```
crs = "ESRI:54002"
```

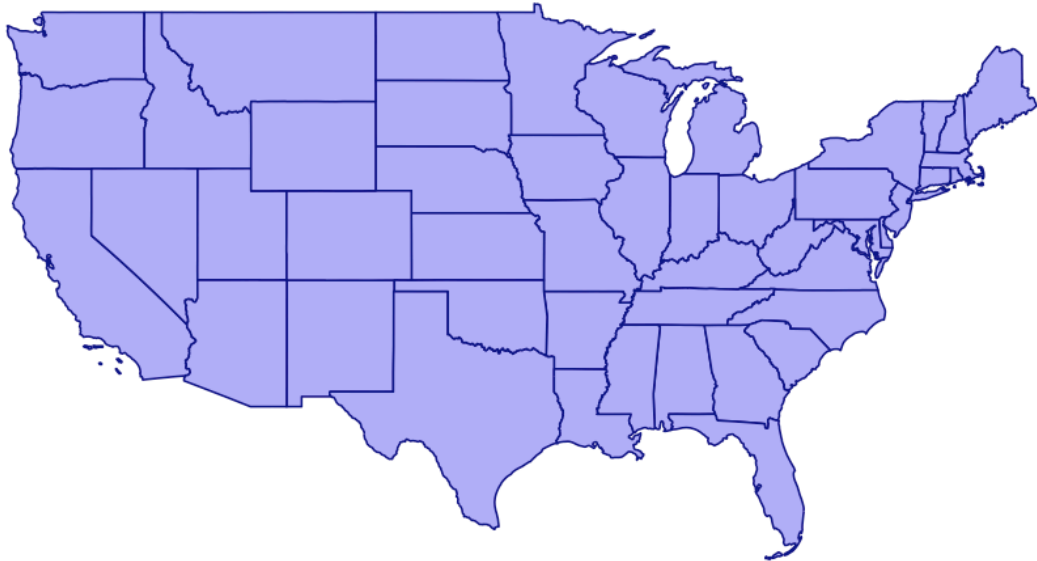
## Robinson



```
crs = "+proj=robin"
```

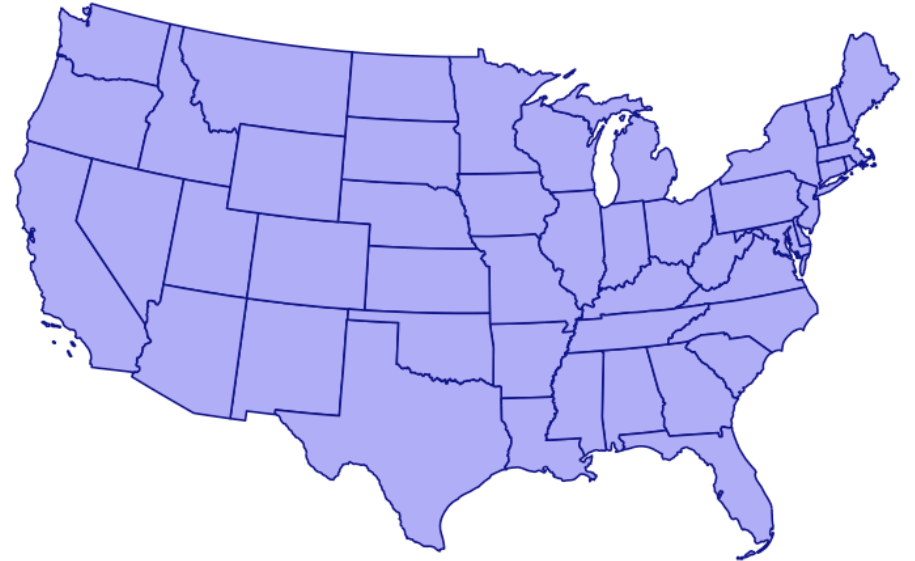
# US projections

**NAD83**



crs = "EPSG:4269"

**Albers**



crs = "ESRI:102003"

# Finding projection codes

[spatialreference.org](https://spatialreference.org)

[epsg.io](https://epsg.io)

[proj.org](https://proj.org)

Most common ones listed on the  
course website example page

This is an excellent overview of how this all works

And **this** is a really really helpful overview of all these moving parts

# Which projection is best?

**None of them**

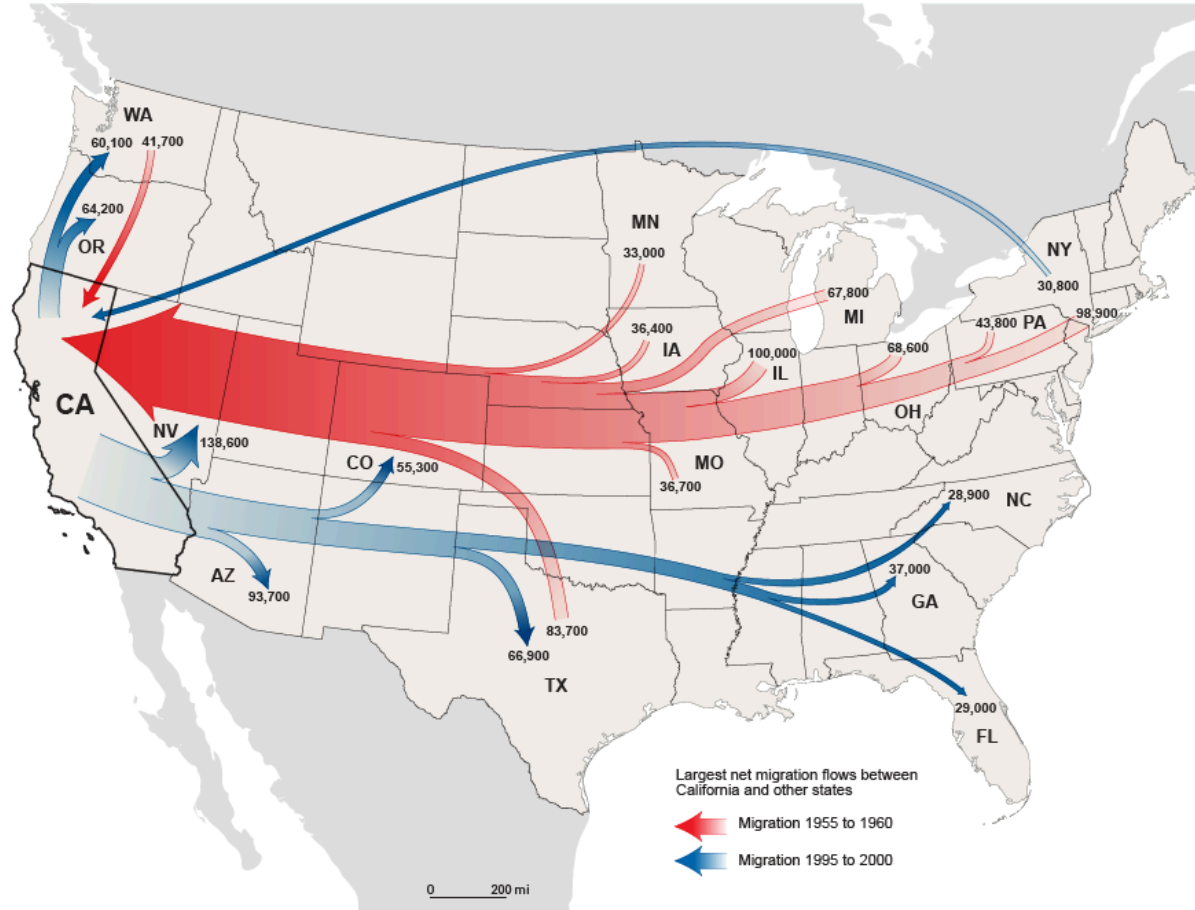
**There are no good or bad projections**

**There are appropriate and  
inappropriate projections**

**(but also ew mercator)**

# Putting data on maps

# Maps with lines



US Census Bureau: Net migration between California and other states

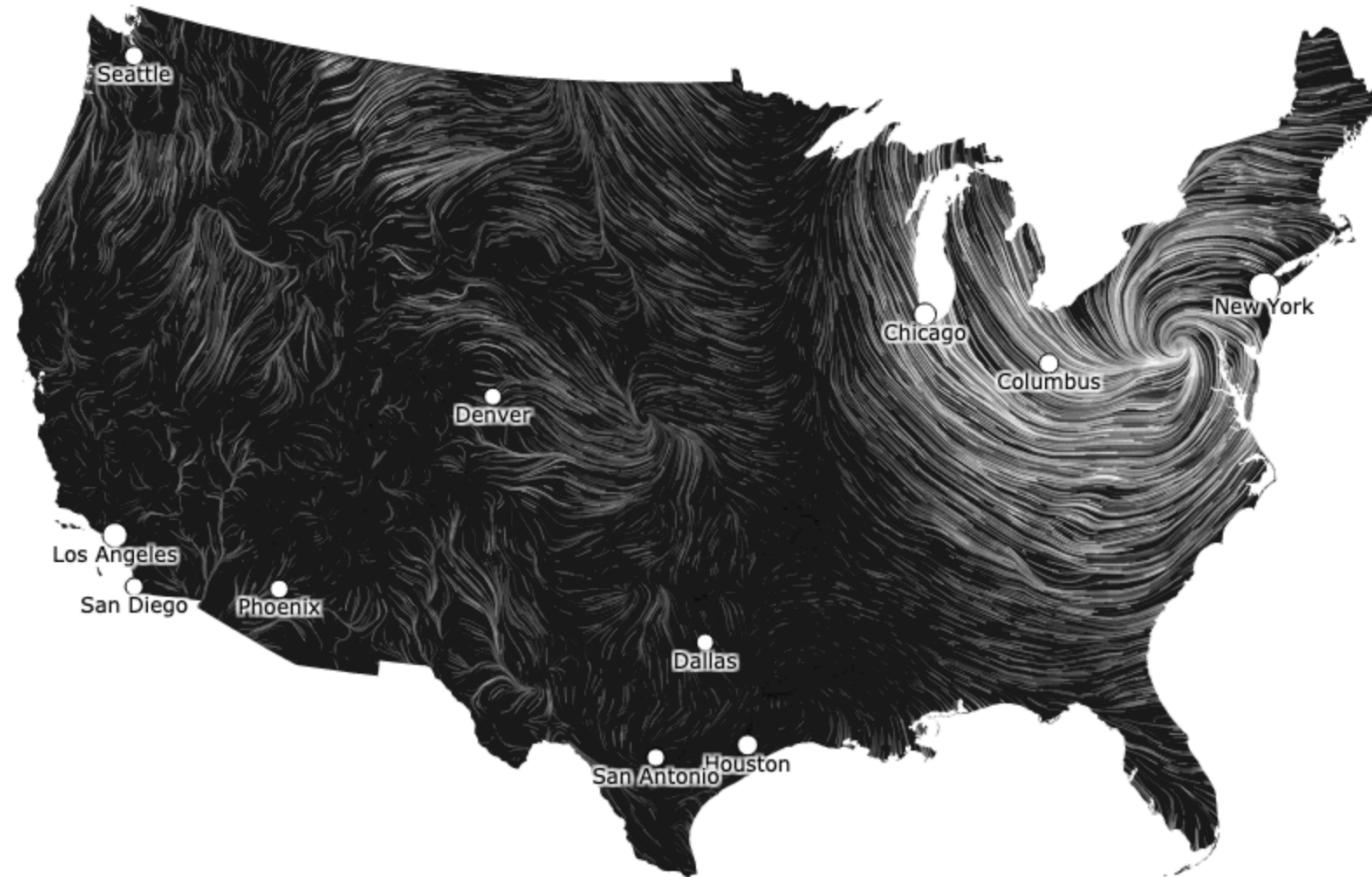
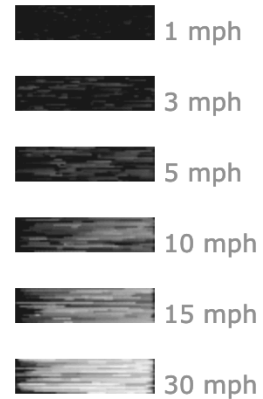
# Maps with lines

**October 30, 2012**

6:59 am EST

(time of forecast download)

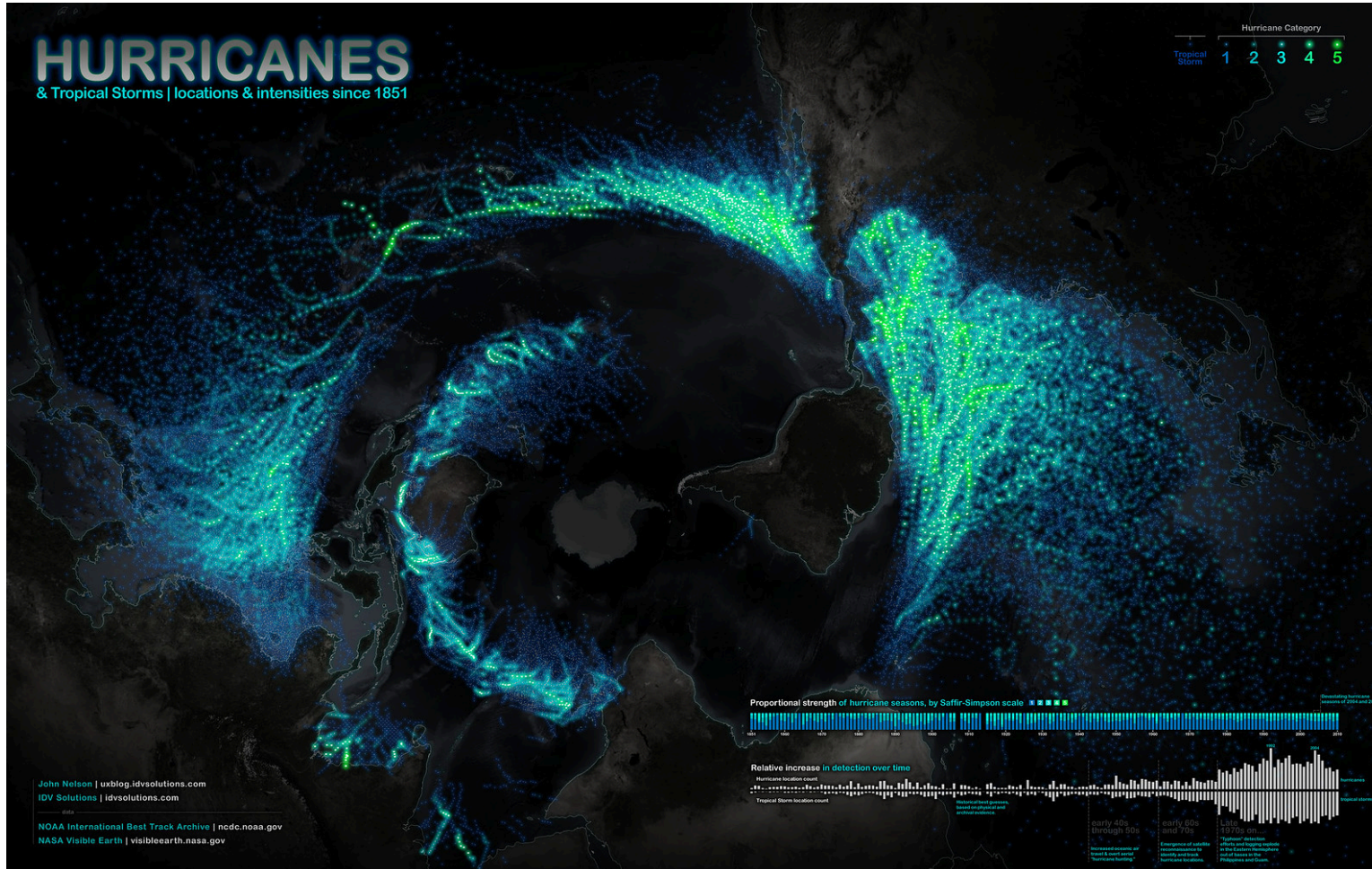
top speed: **39.7 mph**  
average: **8.4 mph**





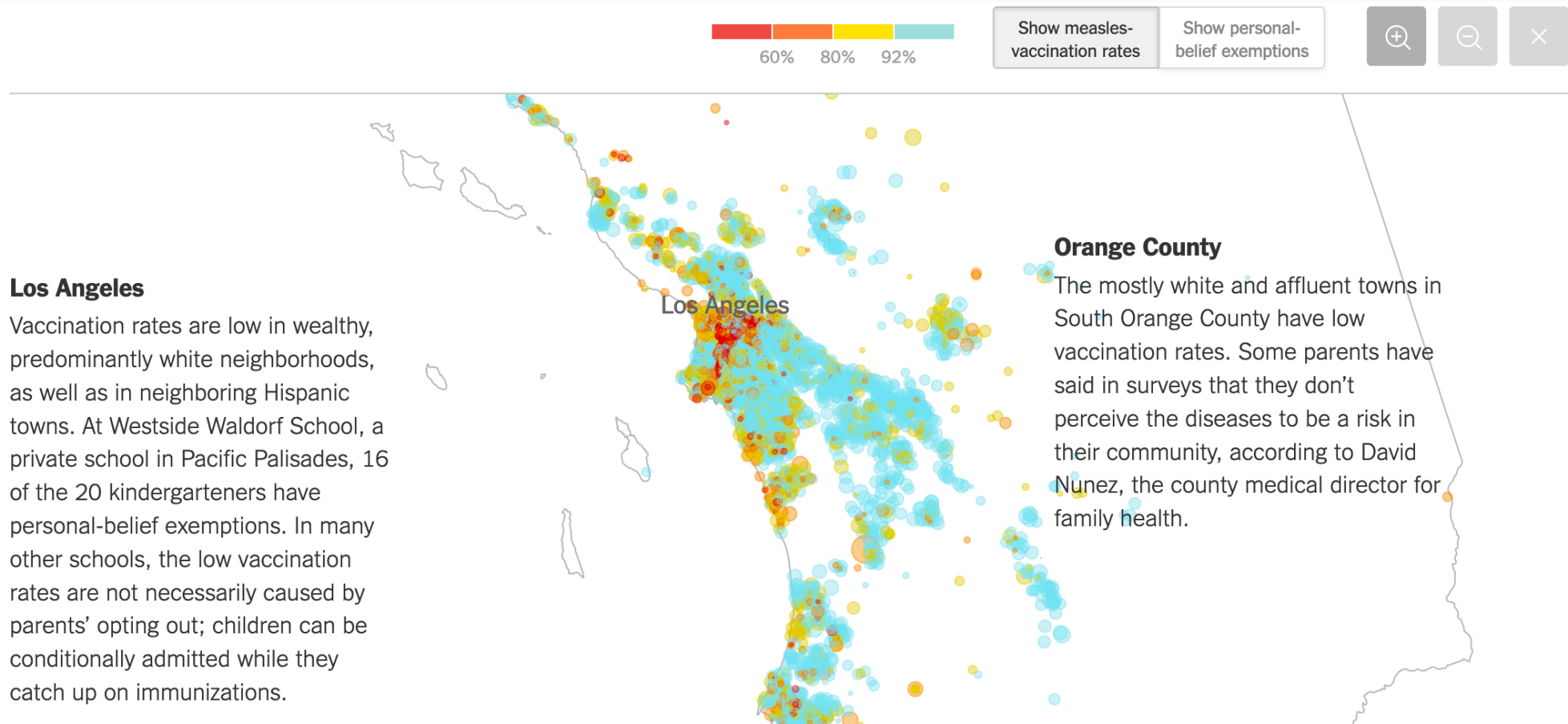


# Maps with points



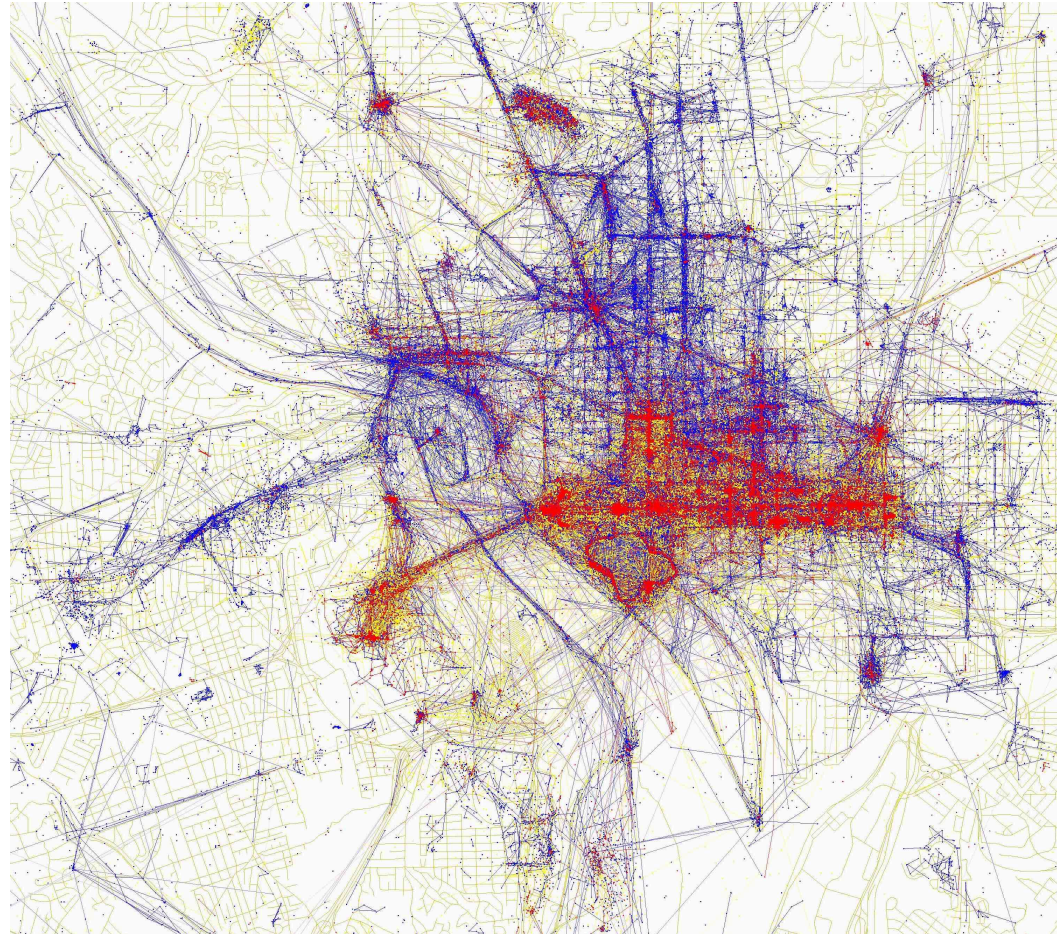
Every hurricane since 1851, by IDV solutions

# Maps with points



The New York Times, "Vaccination Rates for Every Kindergarten in California"

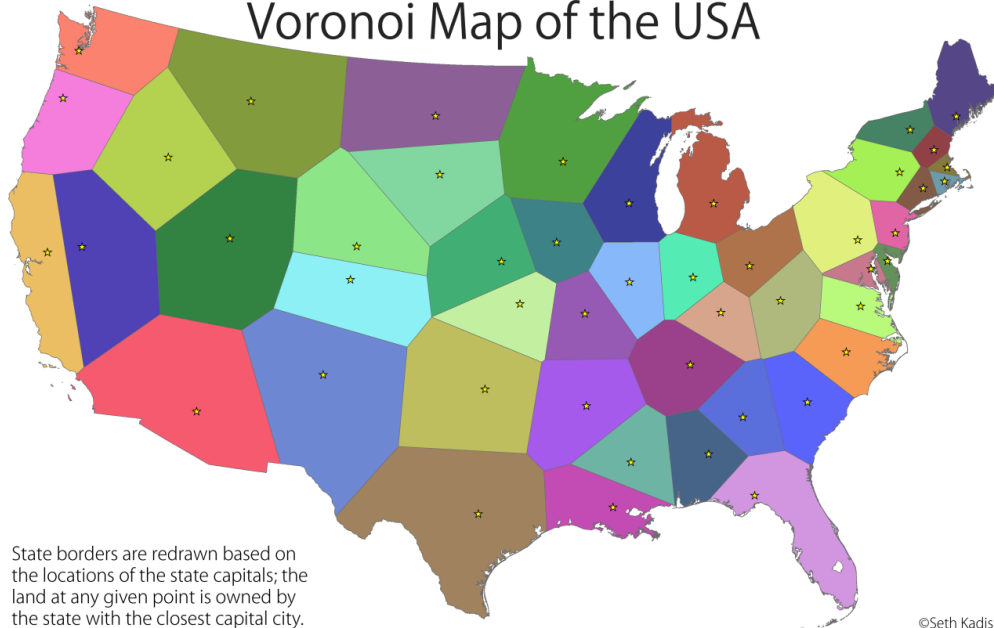
# Maps with points



Locals vs. tourists in DC (blue = locals; red = tourists; yellow = unknown)

# Voronoi maps

Voronoi Map of the USA



State borders are redrawn based on the locations of the state capitals; the land at any given point is owned by the state with the closest capital city.

©Seth Kadish

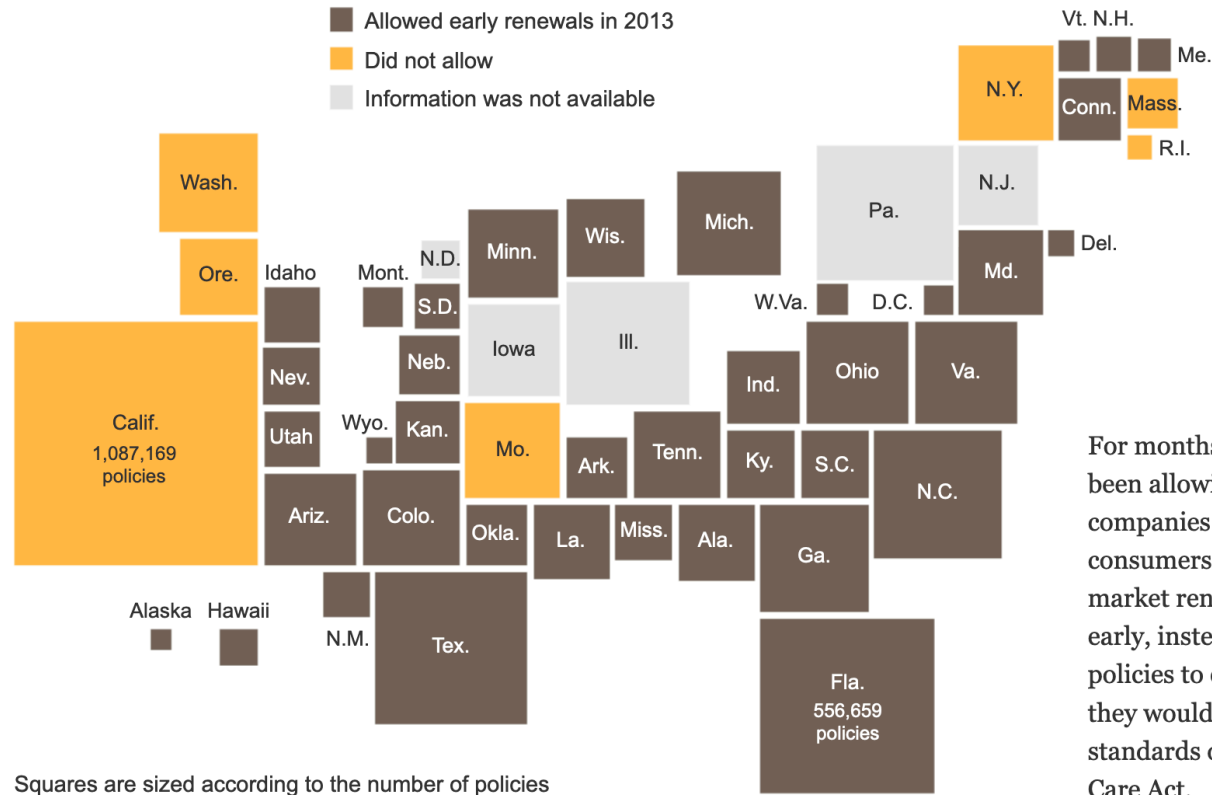
Voronoi state boundaries, by Seth Kadish



Closest NBA teams

# Maps with shapes

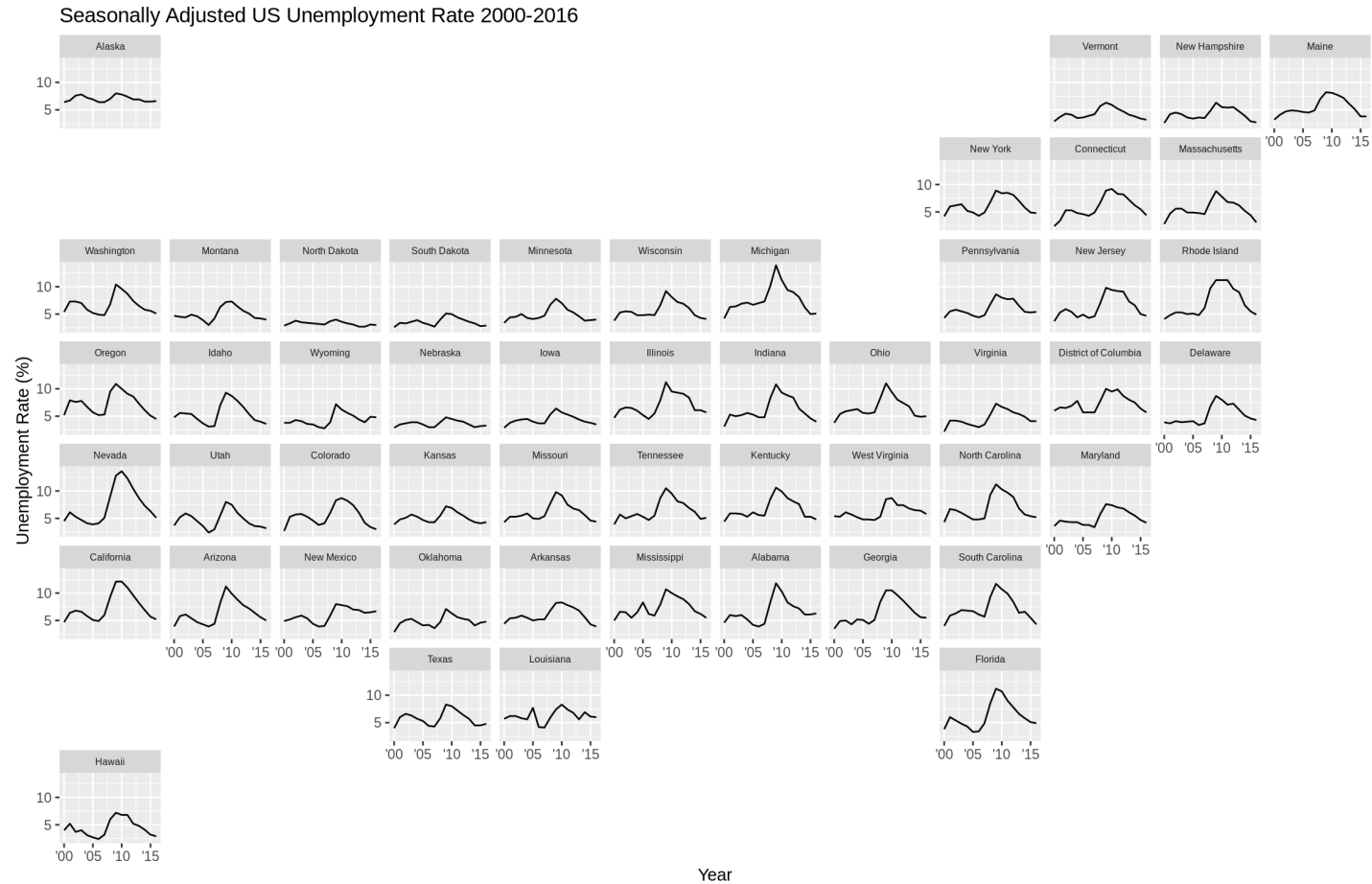
## States Where Insured Could Renew Plans Before Change by Obama



Squares are sized according to the number of policies in each state's individual insurance market in 2012.

For months, many states have been allowing insurance companies the option of letting consumers in the individual market renew their policies early, instead of waiting for the policies to expire in 2014, when they would not meet the standards of the Affordable Care Act. [Related Article »](#)

# Small multiples that look like maps



Data Source: bls.gov

facet\_geo() in the **geofacet** package

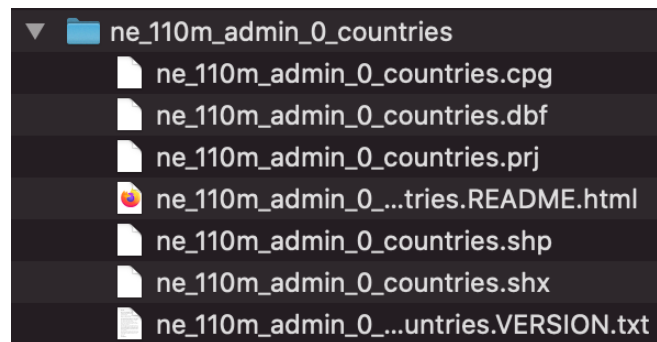
# GIS in R with {sf}

# Shapefiles

Geographic information is shared as **shapefiles**

These are *not* like regular single CSV files!

Shapefiles come as zipped files with a bunch of different files inside





# Structure of a shapefile

```
library(sf)
```

```
world_shapes <- read_sf("data/ne_110m_admin_0_countries/ne_110m_admin_0_countries.shp")
```

```
## Simple feature collection with 7 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:  xmin: -180 ymin: -18 xmax: 180 ymax: 83
## Geodetic CRS:  WGS 84
## # A tibble: 7 × 4
##   TYPE                GEOUNIT                ISO_A3                geometry
##   <chr>                <chr>                <chr>                <MULTIPOLYGON [°]>
## 1 Sovereign country  Fiji                FJI                (((180 -16, 180 -17, 179 -17, 179 -17...
## 2 Sovereign country  Tanzania            TZA                (((34 -0.95, 34 -1.1, 38 -3.1, 38 -3...
## 3 Indeterminate     Western Sahara      ESH                (((-8.7 28, -8.7 28, -8.7 27, -8.7 26...
## 4 Sovereign country  Canada              CAN                (((-123 49, -123 49, -125 50, -126 50...
## 5 Country           United States of America USA                (((-123 49, -120 49, -117 49, -116 49...
## 6 Sovereign country  Kazakhstan          KAZ                (((87 49, 87 49, 86 48, 86 47, 85 47,...
## 7 Sovereign country  Uzbekistan          UZB                (((56 41, 56 45, 59 46, 59 46, 60 45,...
```

# Where to find shapefiles

**Natural Earth** for international maps

**US Census Bureau** for US maps

For anything else...

The Google logo is displayed in its characteristic multi-colored font (blue, red, yellow, green, red).A search bar with a magnifying glass icon on the left and a close button (X) on the right. The text inside the bar is "shapefiles for \_\_\_\_\_".

shapefiles for \_\_\_\_\_

# Scales



1:10m = 1:10,000,000

1 cm = 100 km



1:50m = 1:50,000,000

1cm = 500 km

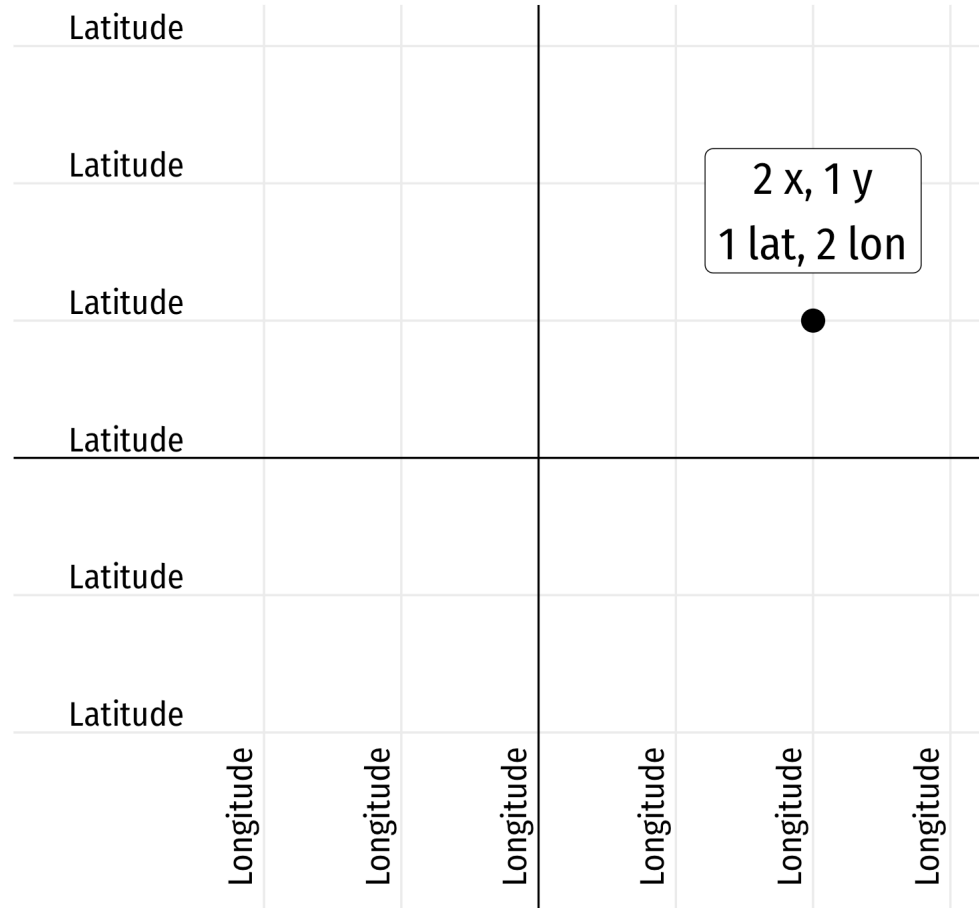


1:110m = 1:110,000,000

1 cm = 1,100 km

Using too high of a resolution  
makes your maps slow and huge

# Latitude and longitude

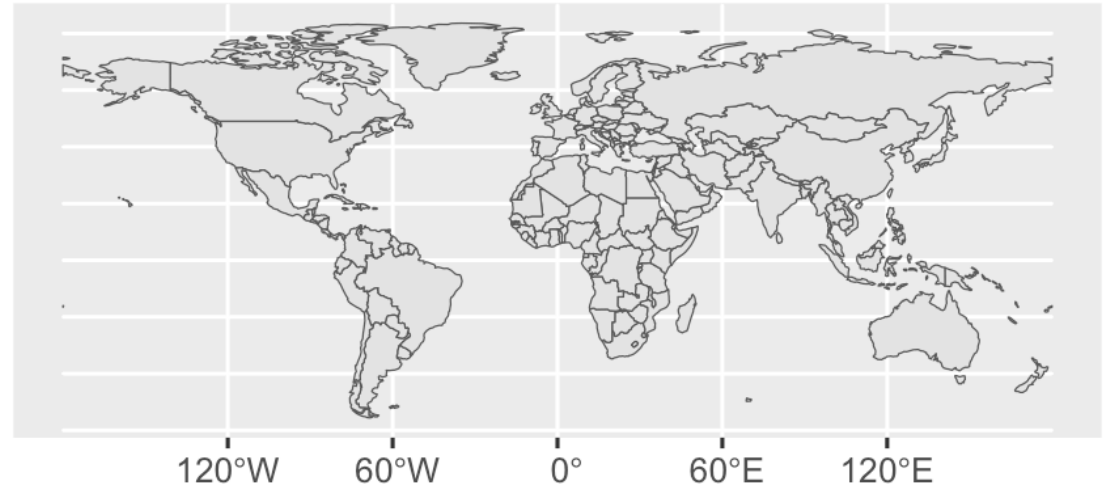


via @sarahbellmaps

# The magic geometry column

As long as you have a magic geometry column,  
all you need to do to plot maps is `geom_sf()`

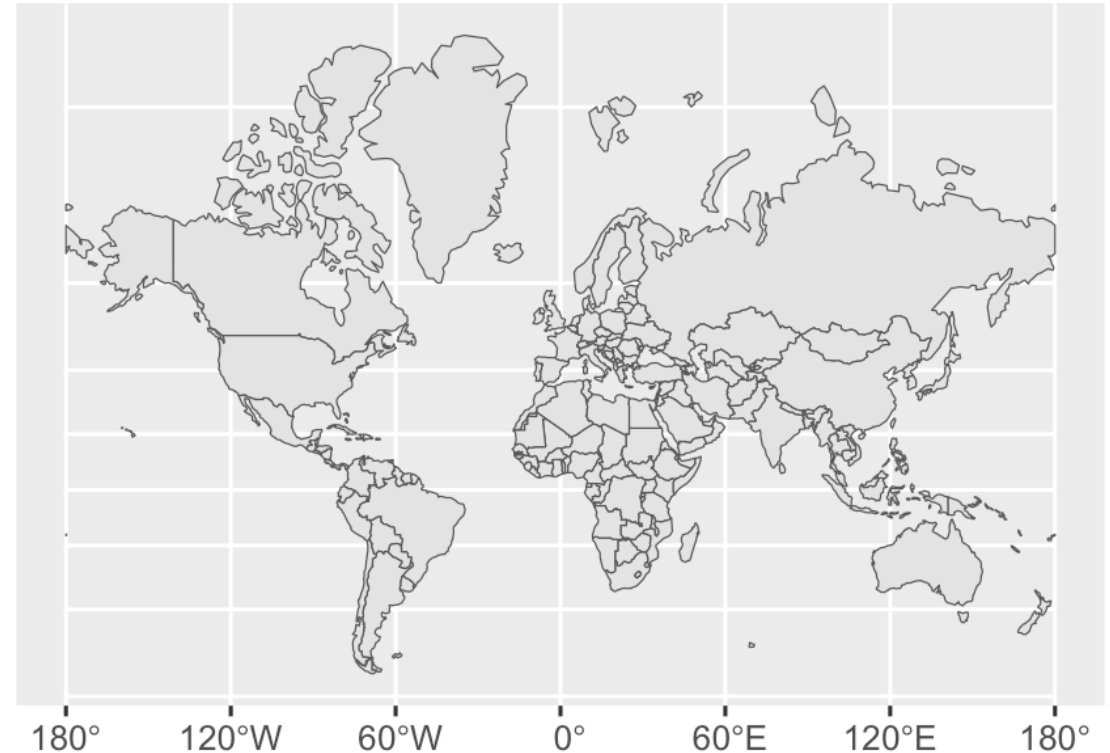
```
ggplot() +  
  geom_sf(data = world_shapes)
```



# The magic geometry column

Use `coord_sf()` to change projections

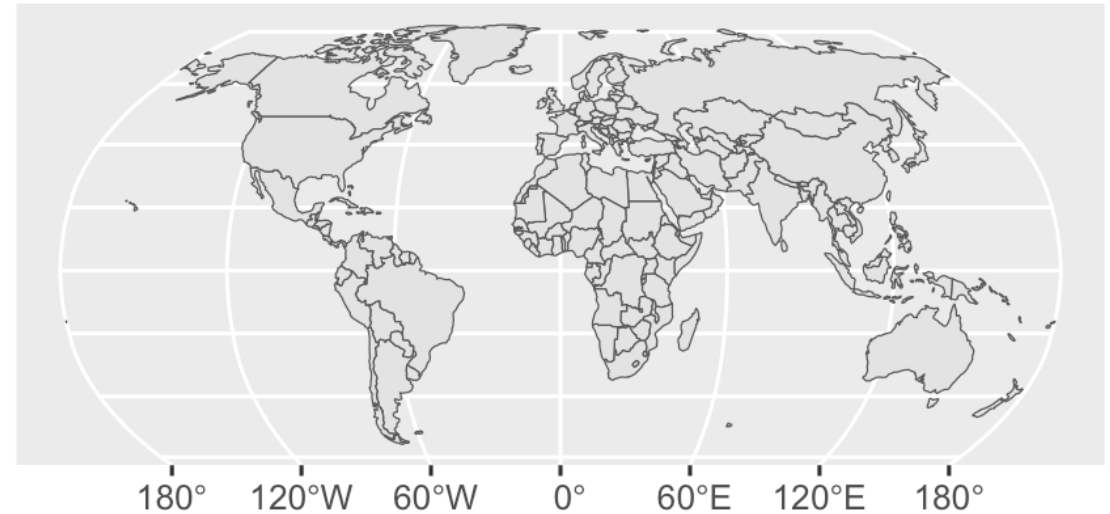
```
ggplot() +  
  geom_sf(data = world_shapes) +  
  coord_sf(crs = "+proj=merc")
```



# The magic geometry column

Use `coord_sf()` to change projections

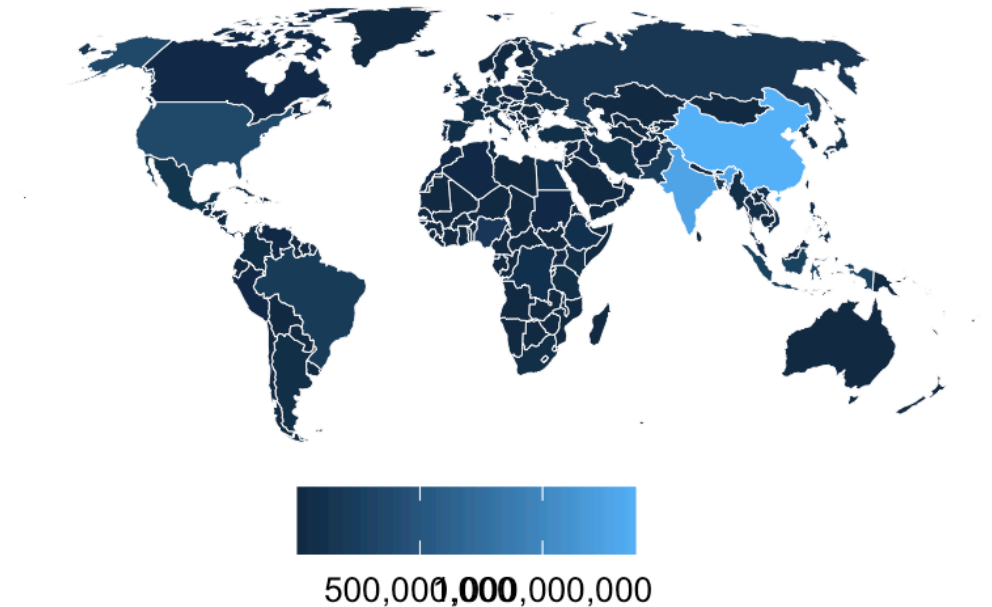
```
ggplot() +  
  geom_sf(data = world_shapes) +  
  coord_sf(crs = "+proj=robin")
```



# Use aesthetics like normal

All regular ggplot layers and aesthetics work

```
ggplot() +  
  geom_sf(data = world_shapes,  
          aes(fill = POP_EST),  
          color = "white", linewidth = 0.15)  
  coord_sf(crs = "+proj=robin") +  
  scale_fill_gradient(labels = scales::label_.,  
                     labs(fill = NULL)) +  
  theme_void() +  
  theme(legend.position = "bottom")
```





# No geometry column?

Make your own with `st_as_sf()`

```
other_data
```

```
## # A tibble: 2 × 3
##   city          long  lat
##   <chr>         <dbl> <dbl>
## 1 Atlanta        -84.4  33.8
## 2 Washington, DC -77.1  38.9
```

```
other_data |>
  st_as_sf(coords = c("long", "lat"),
           crs = st_crs("EPSG:4326"))
```

```
## Simple feature collection with 2 features and 1 field
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: -84 ymin: 34 xmax: -77 ymax: 39
## Geodetic CRS: WGS 84
## # A tibble: 2 × 2
##   city          geometry
## * <chr>         <POINT [°]>
## 1 Atlanta        (-84 34)
## 2 Washington, DC (-77 39)
```

# {sf} is for all GIS stuff

Draw maps

Calculate distances between points

Count observations in a given area

Anything else related to geography!

See [here](#) or [here](#) for full textbooks

# `geom_sf()` is today's standard

You'll sometimes find older tutorials and StackOverflow answers about using `geom_map()` or `{ggmap}` or other things

Those still work, but they don't use the same magical `{sf}` system with easy-to-convert projections and other GIS stuff

**Stick with `{sf}` and `geom_sf()`  
and your life will be easy**