Spatial Data and Mapping

Grayson White

Math 241

Week 5 | Spring 2026

Week 5 Goals

Mon Lecture

- Introduction to spatial data
  - point data
  - polygon data
- Static and interactive maps with point and polygon data
  - base layers (almost as good as merino)
  - coordinate reference systems / projections
  - popups and additional context
  - static graphs with `sf` and `ggplot2`
  - interactive graphs with `leaflet`

Wed Lecture

- Geospatial computations
  - distance
  - spatial joins
- Raster data
  - what is it?
  - extracting values by polygons

Spatial Data

- Data that is linked to the physical world.
- **Example**: Crash data in Portland from Problem Sets 2 and 3!
- Visualizing these data on a map (or map-like graph) is a great way to add useful context!

Focus on creating the following types of maps:

- Point maps
- Choropleth maps
- Cartograms
- Hexbin maps (cartogram heatmaps)
- Interactive maps
- Raster maps (Wednesday)

Spatial Data Example: Cholera Outbreak in 1854

The most famous early analysis of geospatial data was done by physician John Snow in 1854.

Cholera outbreaks were common in 19th century London

- *believed* to be an airborne disease caused by breathing foul air

Outbreak in 1854: over 600 deaths in Soho, London

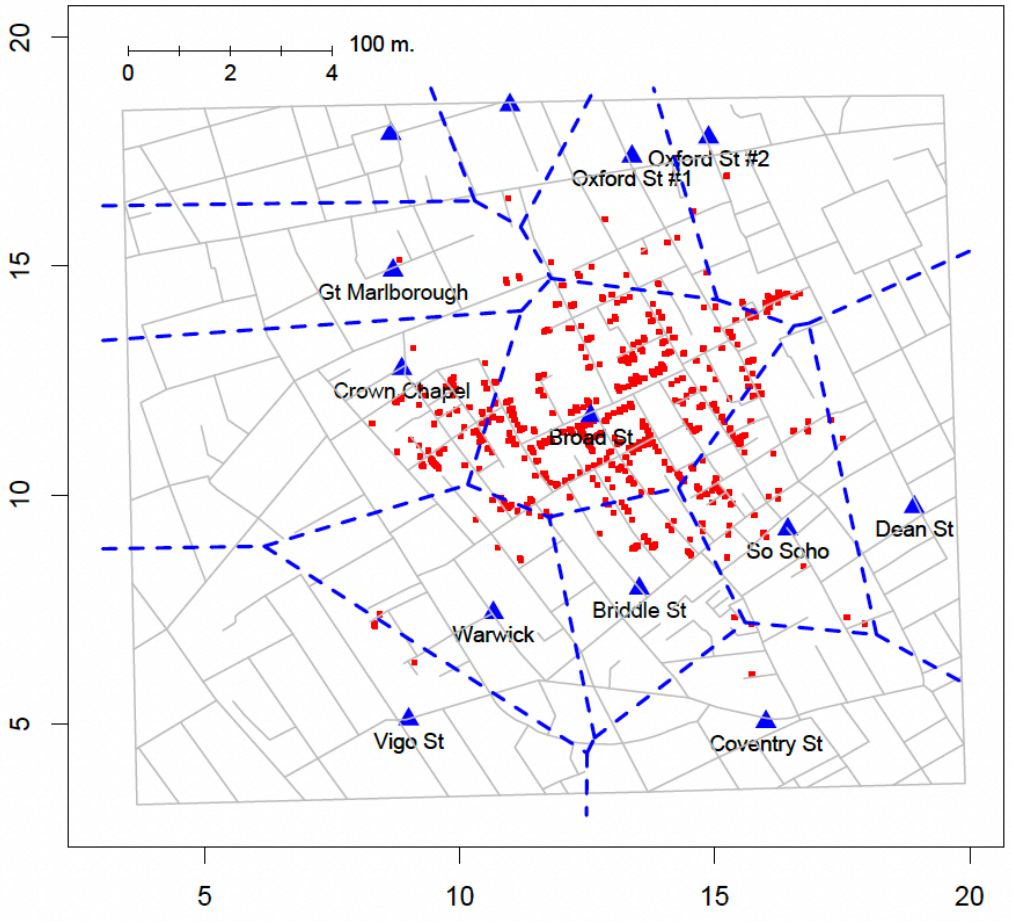
- John Snow collects data, thinks *contaminated water* is the cause
- At the time water was provided by competing private firms *from water pumps*
- Snow recorded locations of (i) deaths and (ii) water pumps

Reproduced from *Visual and Statistical Thinking*, ©E.R. Tufte 1997, based on Snow's drawing .

A simple visual model (Tobler 1994):

This figure:

- **Red Dots:** Cholera cases
- **Blue Triangles:** Pumps
- **Blue Lines:** Closest pump is contained in each section

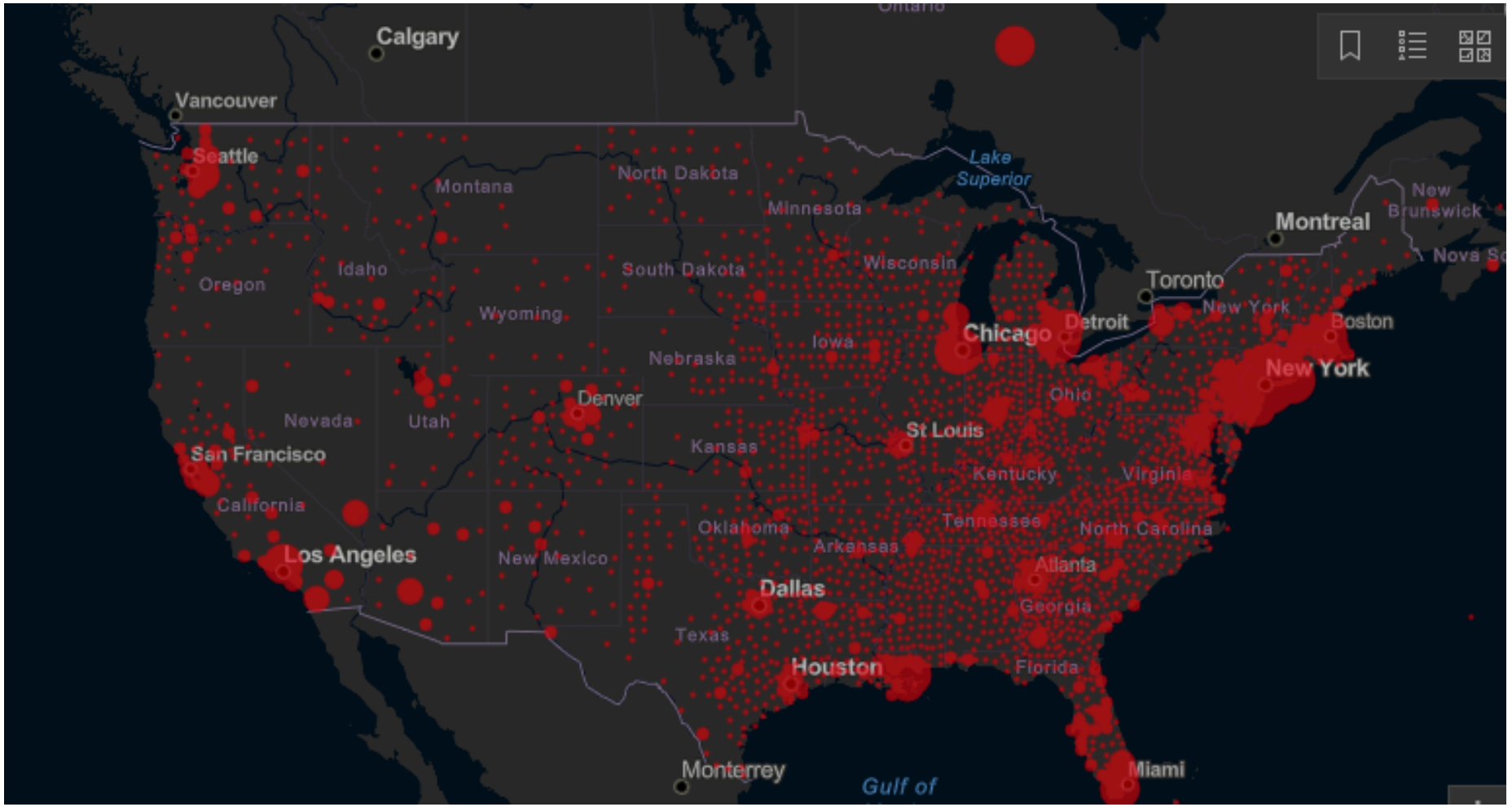


Q: What pattern do you notice?

Q: What are some possible explanations for the "outliers"? (deaths that don't match up with the pattern)

Conclusion: Visualizing spatial data can help us see key patterns

Static Maps



Source: JHU

Choropleth Maps

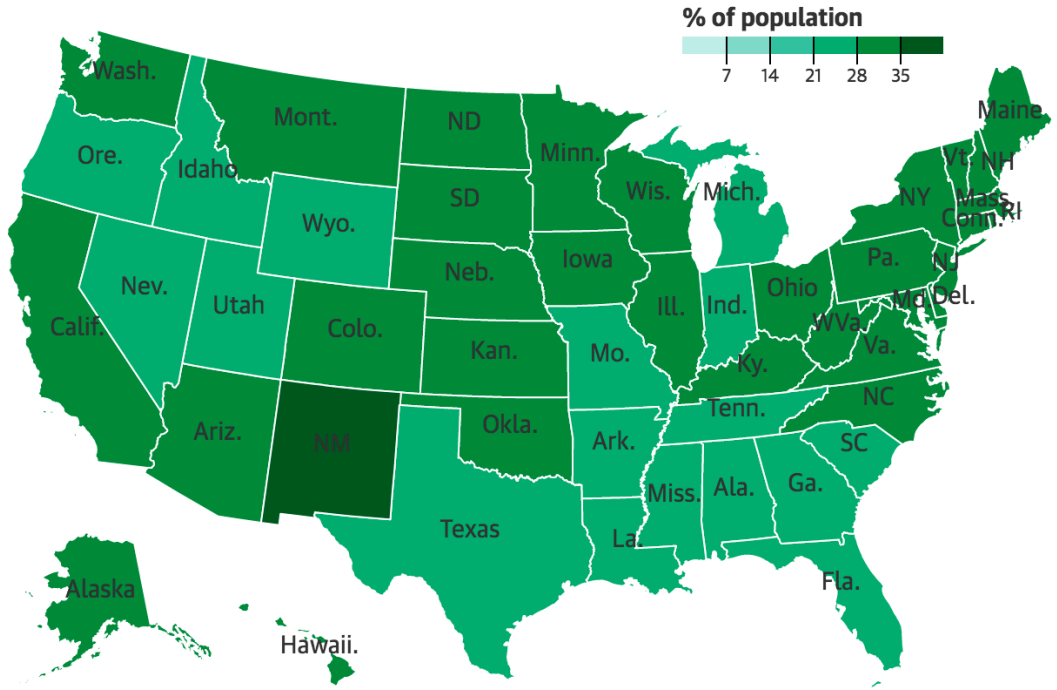
Source: The Guardian

Vaccination rates across the US

Fully vaccinated (nationwide)
15.8%

Received 1 or more doses (nationwide)
28.6%

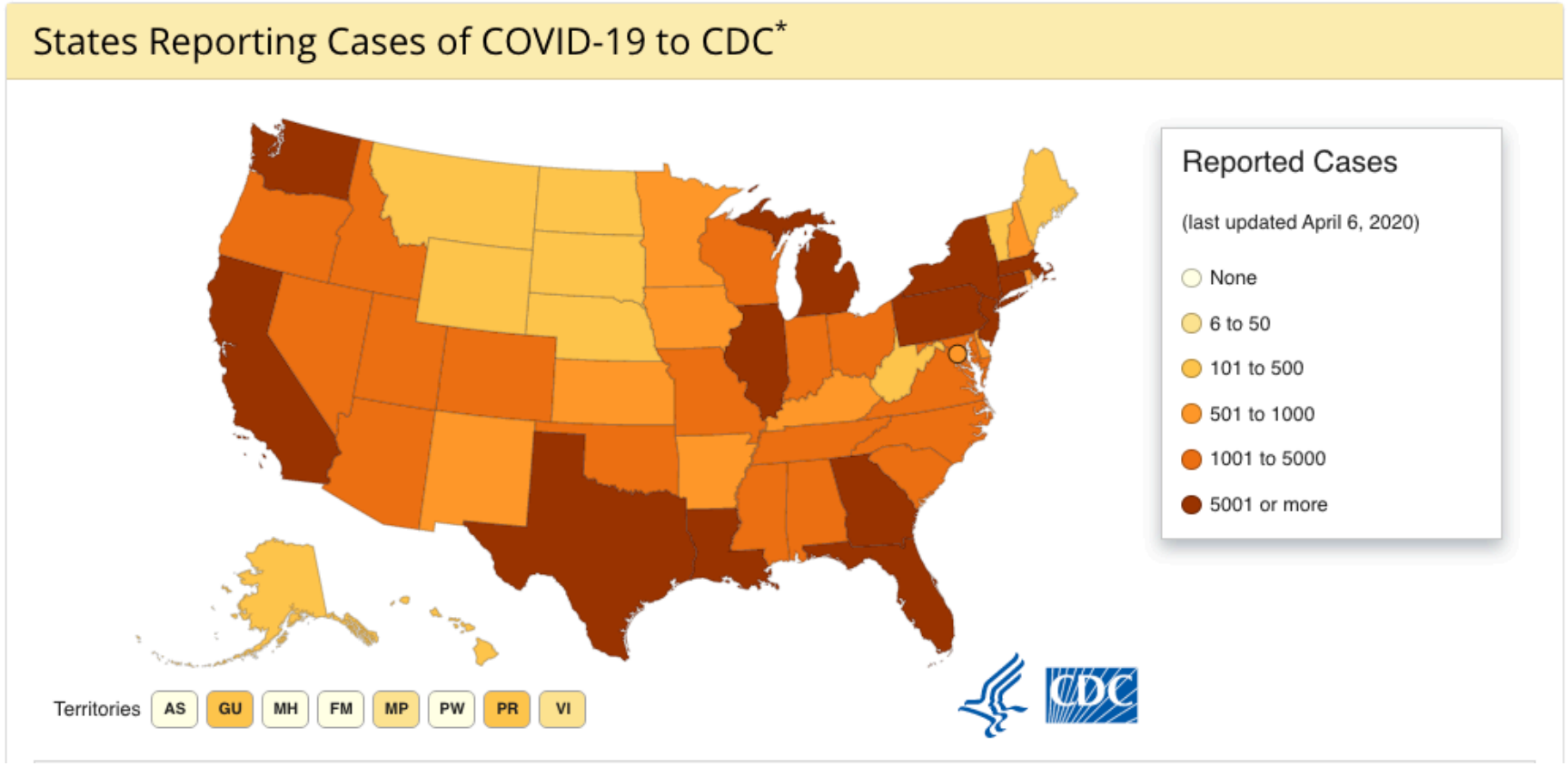
Fully vaccinated **Received 1 or more doses**



Guardian graphic | Source: Centers for Disease Control. Last updated on 2021-03-29.

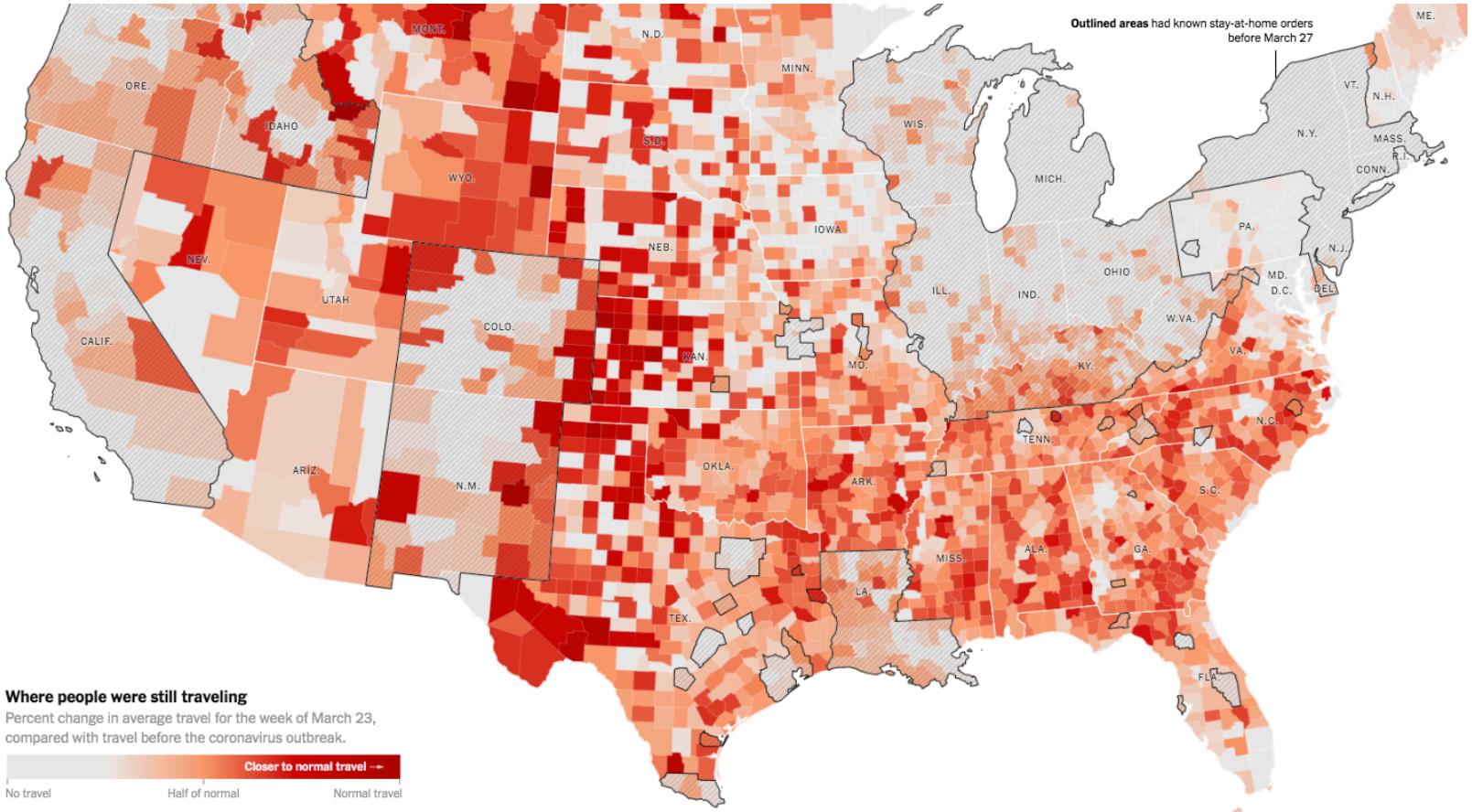
Note: The percentages reported here represent a proportion of the total state population, including children. However, the Pfizer vaccine is only available to those 16 years and older, whereas the Moderna vaccine is available

Choropleth Maps



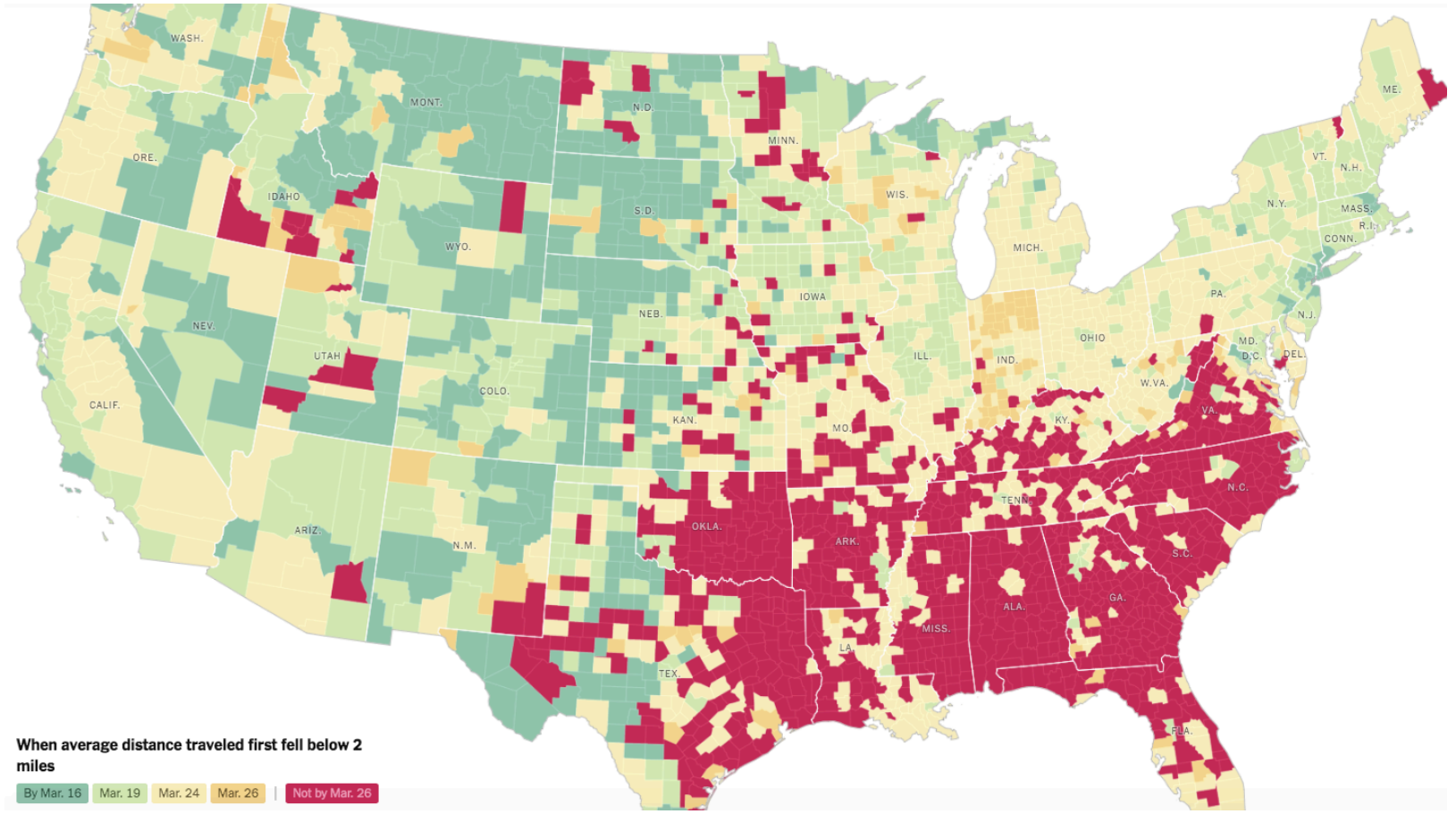
Source: CDC

Choropleth Maps



Source: NYtimes.com

Choropleth Maps

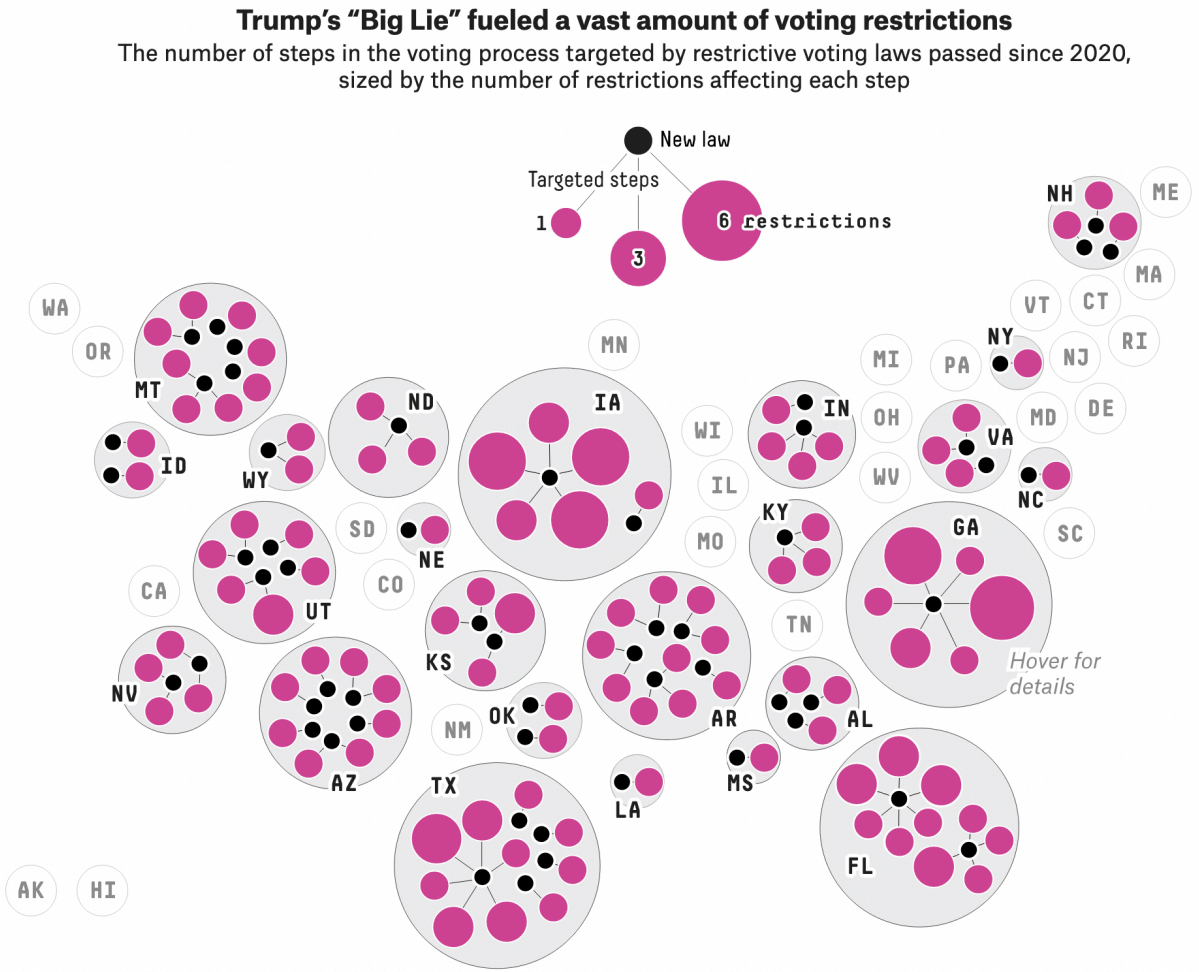


Source: NYtimes.com

Cartogram

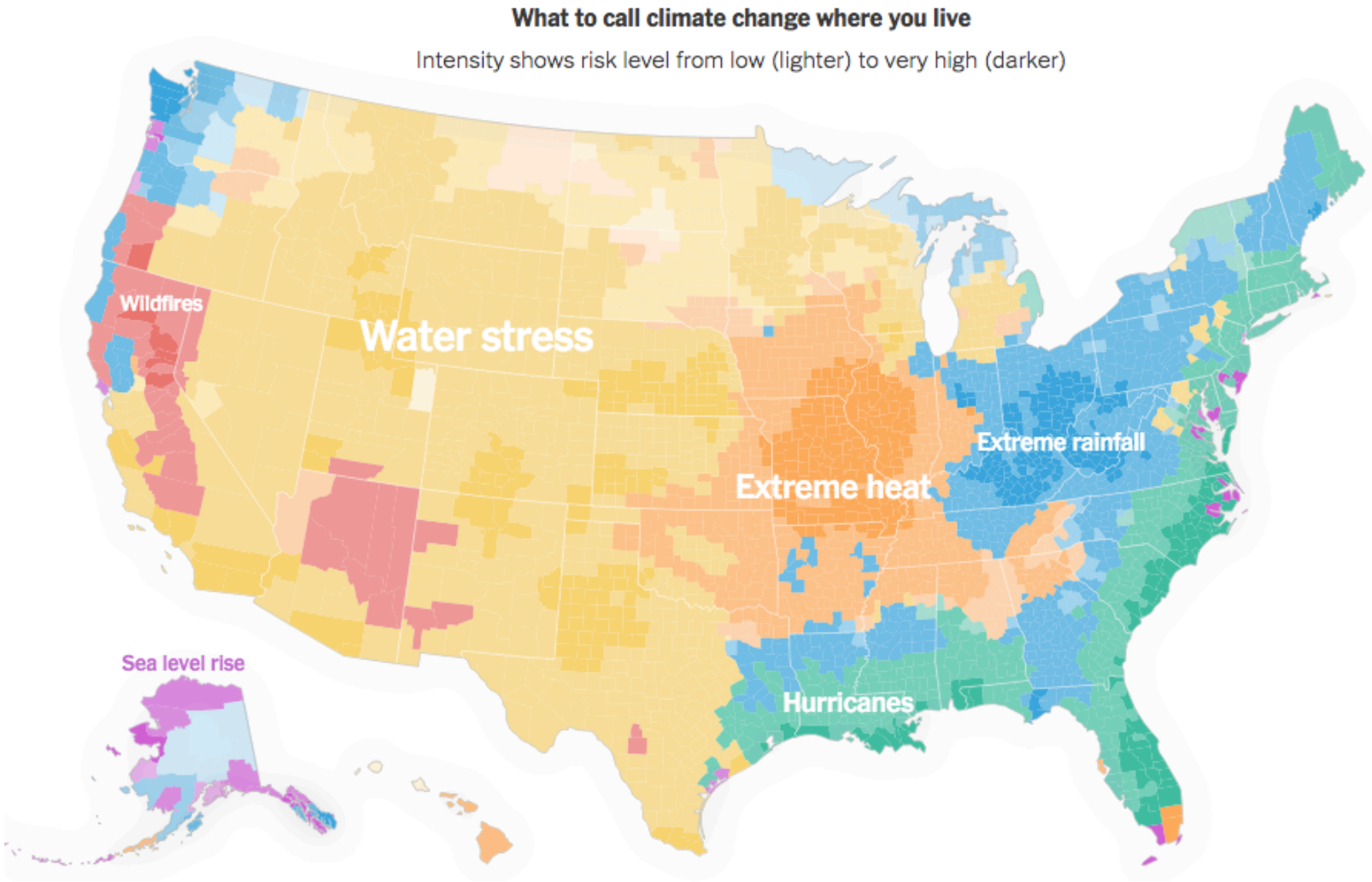
Source: Our World in Data

“Hex”bin Maps



Source: FiveThirtyEight

Interactive Maps



Source: NYTimes

Spatial Data Structures

- Often stored in special data structures
  - Ex: Shapefile(s)
  - Consist of geometric objects like points, lines, and polygons
- Visualizing Spatial Data
  - Need to draw the map
  - Add metadata on top of the map
    - Color in regions
    - Dots

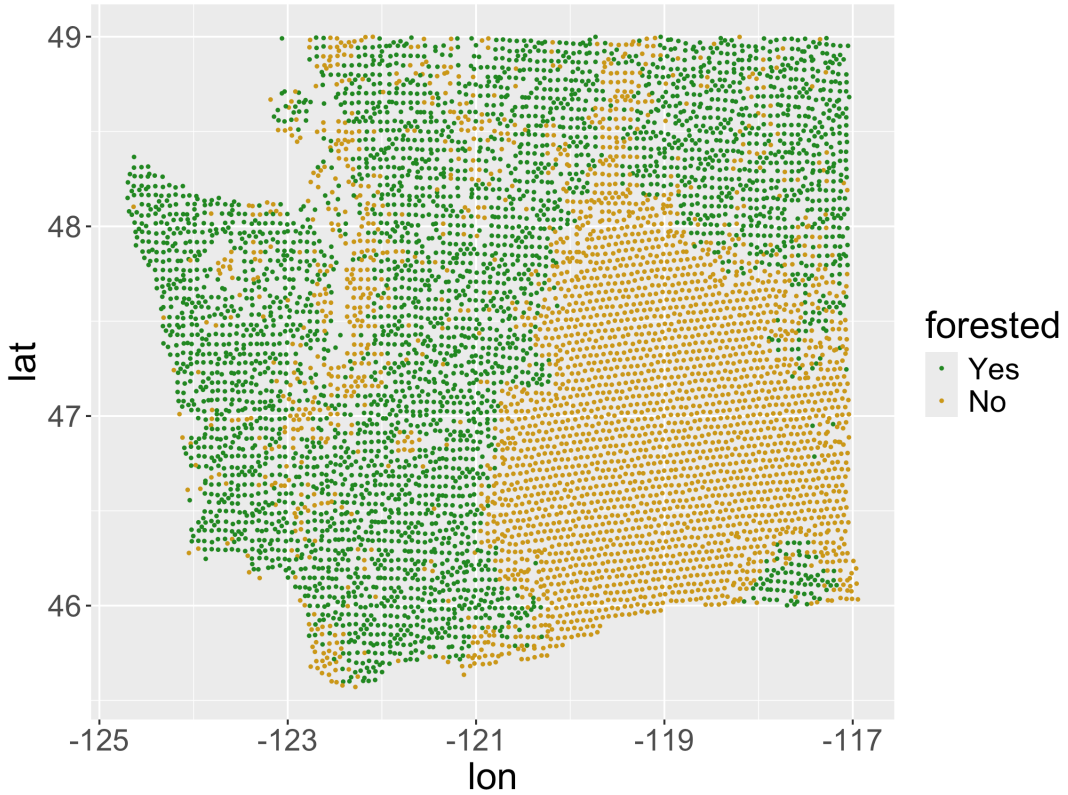
Let's begin with data from the `forested` R package

```
1 library(forested)
2 data(forested)
3 glimpse(forested)
```

```
Rows: 7,107
Columns: 20
$ forested         <fct> Yes, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes,…
$ year             <dbl> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005,…
$ elevation        <dbl> 881, 113, 164, 299, 806, 736, 636, 224, 52, 2240, 104…
$ eastness         <dbl> 90, -25, -84, 93, 47, -27, -48, -65, -62, -67, 96, -4…
$ northness        <dbl> 43, 96, 53, 34, -88, -96, 87, -75, 78, -74, -26, 86, …
$ roughness        <dbl> 63, 30, 13, 6, 35, 53, 3, 9, 42, 99, 51, 190, 95, 212…
$ tree_no_tree     <fct> Tree, Tree, Tree, No tree, Tree, Tree, No tree, Tree,…
$ dew_temp         <dbl> 0.04, 6.40, 6.06, 4.43, 1.06, 1.35, 1.42, 6.39, 6.50,…
$ precip_annual    <dbl> 466, 1710, 1297, 2545, 609, 539, 702, 1195, 1312, 103…
$ temp_annual_mean <dbl> 6.42, 10.64, 10.07, 9.86, 7.72, 7.89, 7.61, 10.45, 10…
$ temp_annual_min  <dbl> -8.32, 1.40, 0.19, -1.20, -5.98, -6.00, -5.76, 1.11, …
$ temp_annual_max  <dbl> 12.91, 15.84, 14.42, 15.78, 13.84, 14.66, 14.23, 15.3…
```

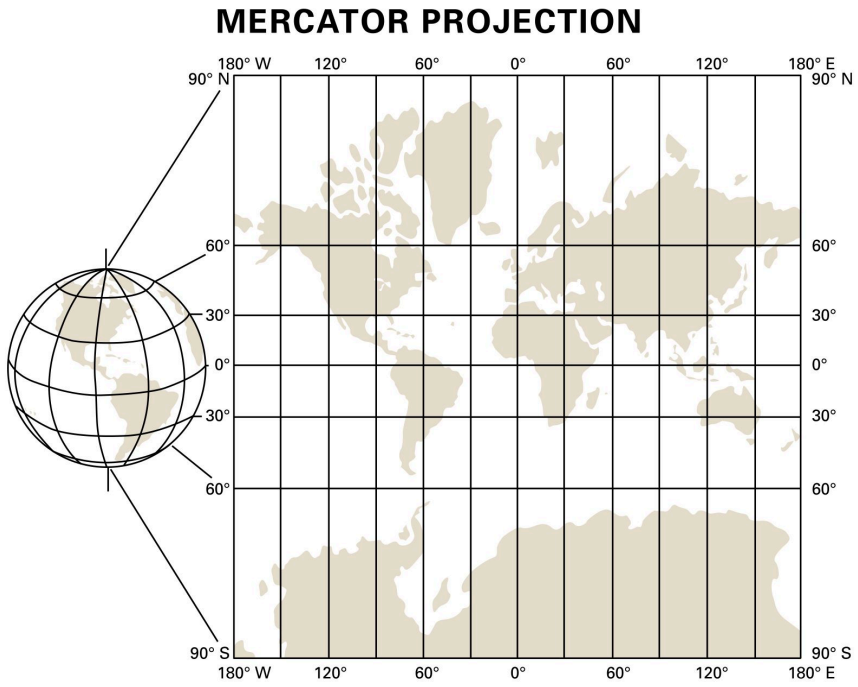
What we've done so far:

```
ggplot(forested, aes(x = lon,
                     y = lat,
                     color = forested)) +
  geom_point(size = 0.5) +
  scale_color_manual(
    values = c("forestgreen", "goldenrod3")
    )
```



- x-axis: longitude, y-axis: latitude
- No projection
- No base layer
- Let's improve it!

Projections



- We live on a globe.
- We like to create maps on flat pages.
- We have to decide how to map things on a globe to a flat page.
- Enter, projections.

Simple Features Maps

- We'd often like to take spatial data that comes in the lat-long format given in the `forested` package and convert it to a format that accounts for the projection.
- Often use the "simple features" standard produced by the Open Geospatial Consortium
  - R package: `sf` handles this type of data
  - Within `ggplot2`, `geom_sf()` and `coord_sf()` work with `sf`
- In reality, this is the lat-long data, but encoded in a way that preserves projection information.

Making `forested` spatial

`sf` R package: **s**imple **f**eatures.

`sf` objects contain projection information and allow us to plot spatial data more accurately.

```
class(forested)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

```
library(sf)
forested_sf <- forested %>%
  st_as_sf(coords = c("lon", "lat"), # columns in df that correspond to x and y coordinates
           crs = "EPSG:4269") # the projection the coordinates are in
class(forested_sf)
```

```
[1] "sf"         "tbl_df"     "tbl"        "data.frame"
```

Making `forested` spatial

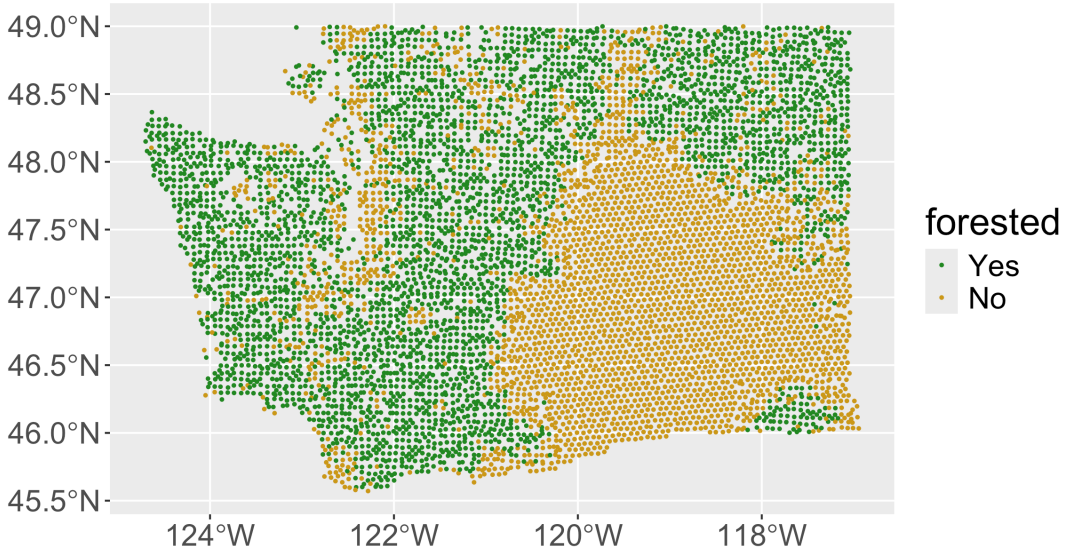
```
1 glimpse(forested_sf)
```

```
Rows: 7,107
Columns: 19
$ forested         <fct> Yes, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes,…
$ year             <dbl> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005,…
$ elevation        <dbl> 881, 113, 164, 299, 806, 736, 636, 224, 52, 2240, 104…
$ eastness         <dbl> 90, -25, -84, 93, 47, -27, -48, -65, -62, -67, 96, -4…
$ northness        <dbl> 43, 96, 53, 34, -88, -96, 87, -75, 78, -74, -26, 86, …
$ roughness        <dbl> 63, 30, 13, 6, 35, 53, 3, 9, 42, 99, 51, 190, 95, 212…
$ tree_no_tree     <fct> Tree, Tree, Tree, No tree, Tree, Tree, No tree, Tree,…
$ dew_temp         <dbl> 0.04, 6.40, 6.06, 4.43, 1.06, 1.35, 1.42, 6.39, 6.50,…
$ precip_annual    <dbl> 466, 1710, 1297, 2545, 609, 539, 702, 1195, 1312, 103…
$ temp_annual_mean <dbl> 6.42, 10.64, 10.07, 9.86, 7.72, 7.89, 7.61, 10.45, 10…
$ temp_annual_min  <dbl> -8.32, 1.40, 0.19, -1.20, -5.98, -6.00, -5.76, 1.11, …
$ temp_annual_max  <dbl> 12.91, 15.84, 14.42, 15.78, 13.84, 14.66, 14.23, 15.3…
```

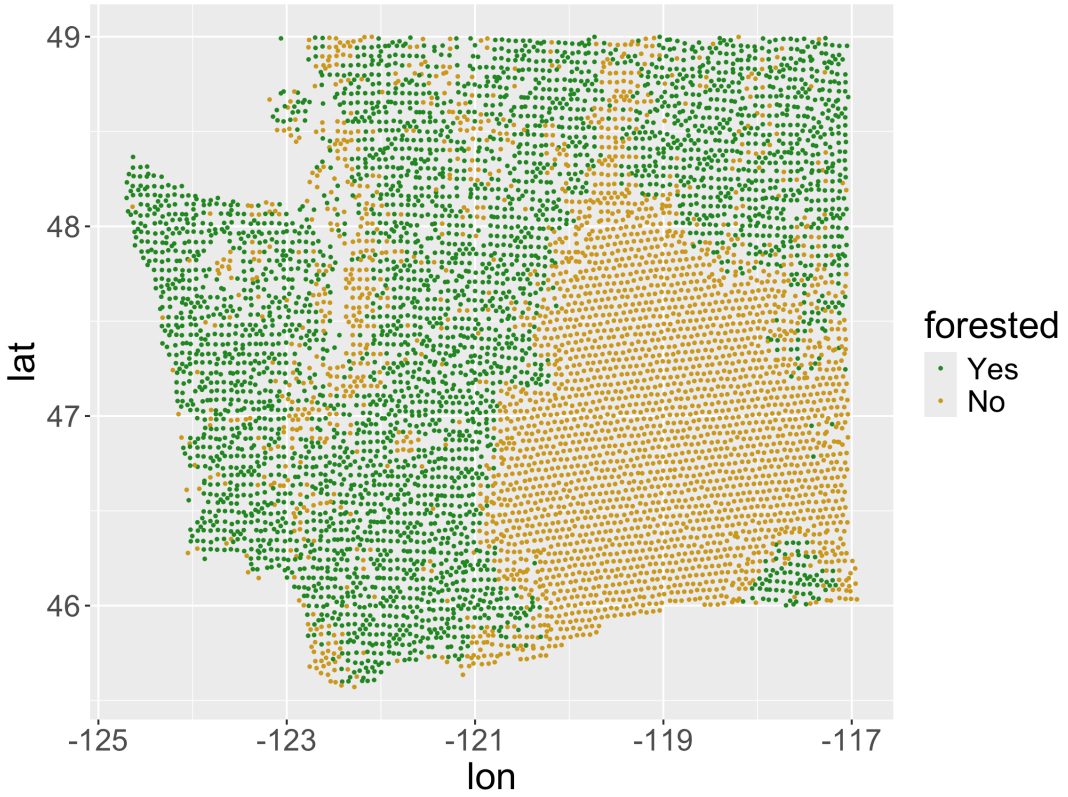
- No longer has `lon` or `lat` columns.
- Has a `geometry` column
  - This is a special type of column. Contains location and projection information.

Plotting `forested_sf`

```
1 ggplot(forested_sf, aes(color = forested)) +
2   geom_sf(size = 0.5) +
3   scale_color_manual(values = c("forestgreen", "goldenr
```



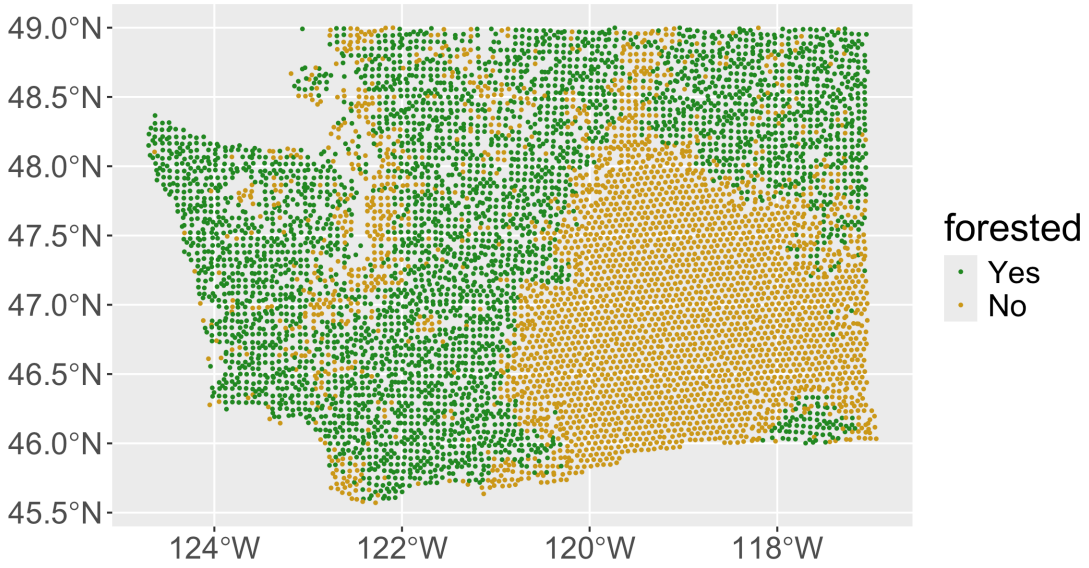

```
1 ggplot(forested, aes(x = lon, y = lat, color = forested
2   geom_point(size = 0.5) +
3   scale_color_manual(values = c("forestgreen", "goldenr
```



- Preserves relative distances, rather than just plotting on the x/y plane.

Plotting `forested_sf`

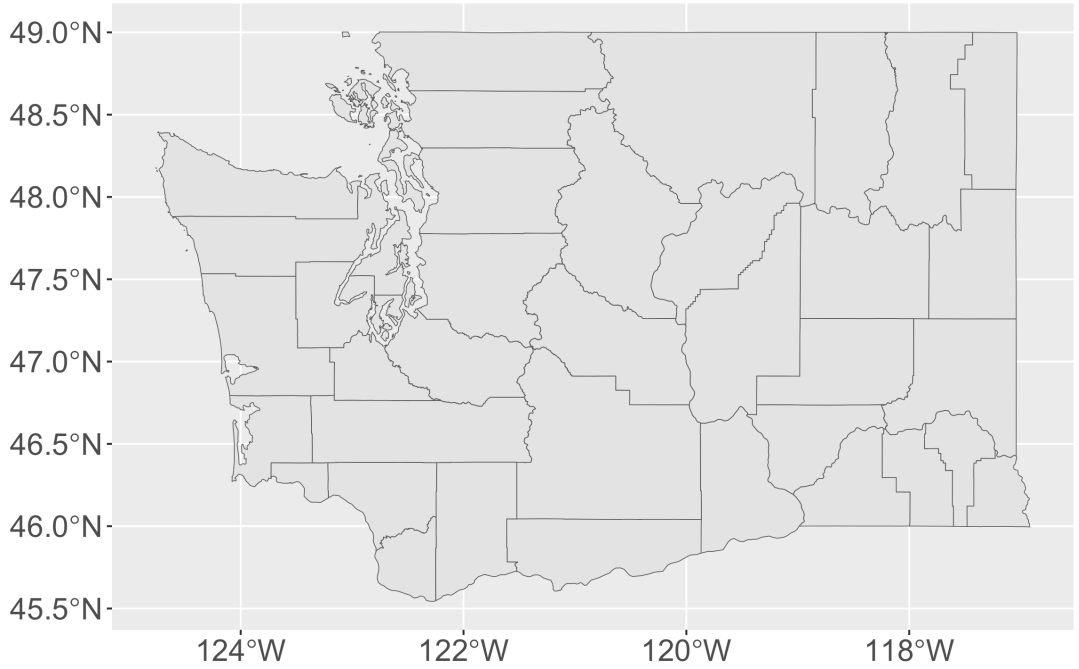
```
ggplot(forested_sf, aes(color = forested)) +
  geom_sf(size = 0.5) +
  scale_color_manual(values = c("forestgreen", "goldenrod3"))
```



- Implicitly, `geom_sf()` looks for a geometry column to set its "geometry" aesthetic to.
- How can we add a base layer?

Polygon Maps

- Want to draw various boundaries
  - EXs: Countries, states, counties, etc
- polygon = closed area including the boundaries making up the area



tidycensus

- Data from the US Census or American Community Survey
  - Comes in simple features format (using `tigris`)
  - Need to obtain an API key

```
api_key <- "insert key"
```

```
library(tidycensus)
options(tigris_use_cache = TRUE)
wa <- get_acs(state = "WA",  geography = "county",
              variables = "B19013_001",
              geometry = TRUE, key = api_key)
```

What is the structure of `wa`?

```
1 class(wa)
```

```
[1] "sf"         "data.frame"
```

```
1 head(wa$geometry)
```

```
Geometry set for 6 features
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -123.3727 ymin: 46.38346 xmax: -117.43 ymax: 49.00084
Geodetic CRS:  NAD83
First 5 geometries:
```

- Row for each polygon
- `geometry` contains the polygon object (borders of the county)

Compared to `forested_sf`

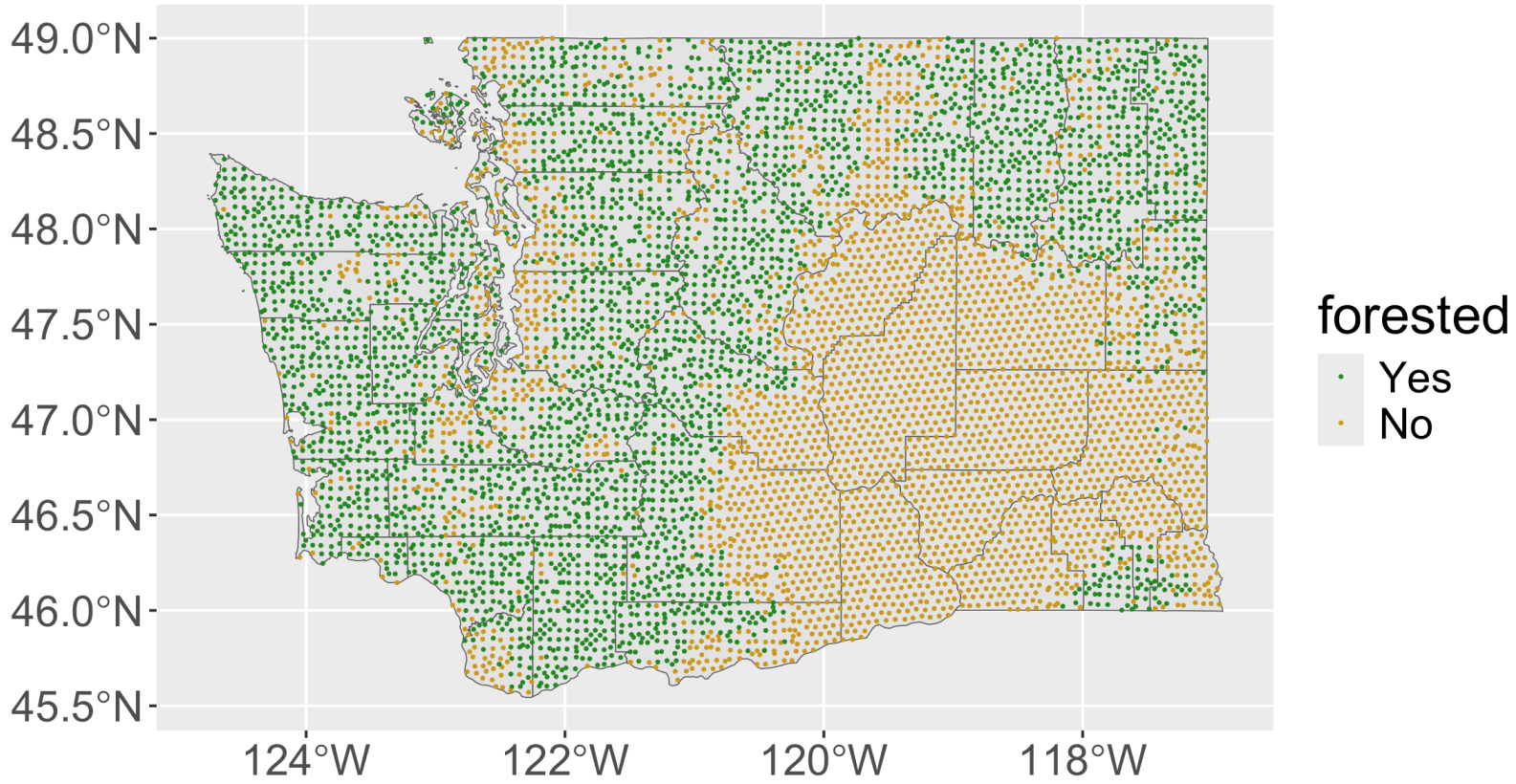
```
1 head(forested_sf$geometry)
```

```
Geometry set for 6 features
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -123.0825 ymin: 45.80776 xmax: -117.7224 ymax: 48.77132
Geodetic CRS:  NAD83
First 5 geometries:
```

- Row for each point, `geometry` contains the point object (center of the plot)

Adding a base layer to our original map

```
ggplot() +
  geom_sf(data = wa) +
  geom_sf(data = forested_sf, aes(color = forested), size = 0.2) +
  scale_color_manual(values = c("forestgreen", "goldenrod3"))
```



What if we'd like to map the proportion of forested plots in each county in WA?

```
1 forested_props <- forested %>%
2   group_by(county) %>%
3   summarize(prop_forested = mean(forested == "Yes"))
4
5 forested_props
```

```
# A tibble: 39 × 2
   county   prop_forested
   <fct>            <dbl>
 1 Adams            0
 2 Asotin           0.239
 3 Benton           0
 4 Chelan           0.741
 5 Clallam          0.928
 6 Clark            0.535
 7 Columbia         0.347
 8 Cowlitz          0.887
 9 Douglas          0.0109
10 Ferry            0.908
# ℹ 29 more rows
```

- Want to make a **chloropleth** map.

Need to join with wa... 🤔

```
1 head(forested_props)
```

```
# A tibble: 6 × 2
  county  prop_forested
  <fct>           <dbl>
1 Adams           0
2 Asotin          0.239
3 Benton          0
4 Chelan          0.741
5 Clallam         0.928
6 Clark           0.535
```

```
1 head(wa)
```

```
Simple feature collection with 6 features and 5 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -123.3727 ymin: 46.38346 xmax: -117.43 ymax: 49.00084
Geodetic CRS:  NAD83
  GEOID                         NAME   variable estimate  moe
1 53067 Thurston County, Washington B19013_001    93985 1815
2 53029   Island County, Washington B19013_001    88358 2918
3 53041    Lewis County, Washington B19013_001    69690 3237
4 53065  Stevens County, Washington B19013_001    67405 3514
5 53017  Douglas County, Washington B19013_001    80374 4122
6 53055 San Juan County, Washington B19013_001    83682 4942
                       geometry
1 MULTIPOLYGON (((-123.2009 4...
```

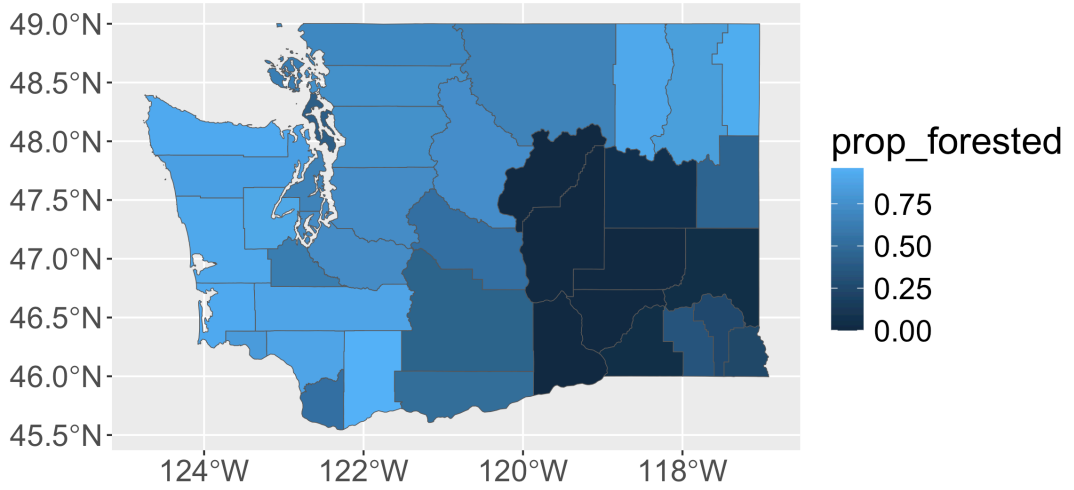
- Want `by = c("county" = "NAME")`. Why won't this work??

Solution: use the `stringr` package!

```
wa <- wa %>%
  mutate(short_name = str_replace_all(NAME,
                                      pattern = " County, Washington",
                                      replacement = ""))
head(wa$short_name)
```

```
[1] "Thurston" "Island"   "Lewis"    "Stevens"  "Douglas"  "San Juan"
```

```
# join
left_join(forested_props,
          wa,
          by = c("county" = "short_name")) %>%
  # make R remember this is an sf object
  # (R keeps the class of the primary dataset)
  st_as_sf() %>%
  ggplot(aes(fill = prop_forested)) +
  geom_sf()
```

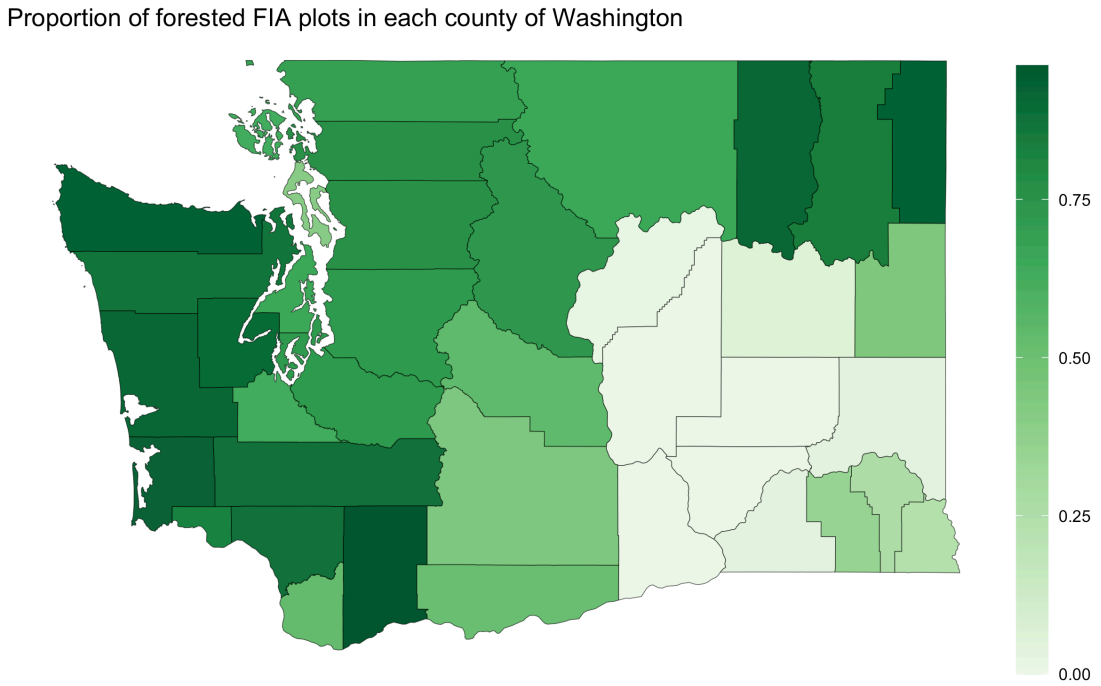


Some mapping suggestions

```r
wa <- wa %>%
  mutate(short_name = str_replace_all(NAME,
                                      pattern = " County, Washington",
                                      replacement = ""))
head(wa$short_name)
```

```
[1] "Thurston" "Island"   "Lewis"    "Stevens"  "Douglas"  "San Juan"
```

```r
# join
left_join(forested_props,
          wa,
          by = c("county" = "short_name")) %>%
  # make R remember this is an sf object
  # (R keeps the class of the primary dataset)
  st_as_sf() %>%
  ggplot(aes(fill = prop_forested)) +
  geom_sf() +
  scale_fill_distiller(type = "seq",
                       palette = "Greens",
                       direction = 1) +
  labs(title = "Proportion of forested FIA plots in eac
       fill = "") +
  theme_void() +
  theme(legend.position = "right",
        legend.key.height = unit(0.9, "in"))
```



Back to `tidycensus`

Recall our call to get county polygons:

```
api_key <- "insert key"
```

```
library(tidycensus)
options(tigris_use_cache = TRUE)
wa <- get_acs(state = "WA",  geography = "county",
              variables = "B19013_001",
              geometry = TRUE, key = api_key)
```

- What is going on with `variables = "B19013_001"`?

Back to tidycensus

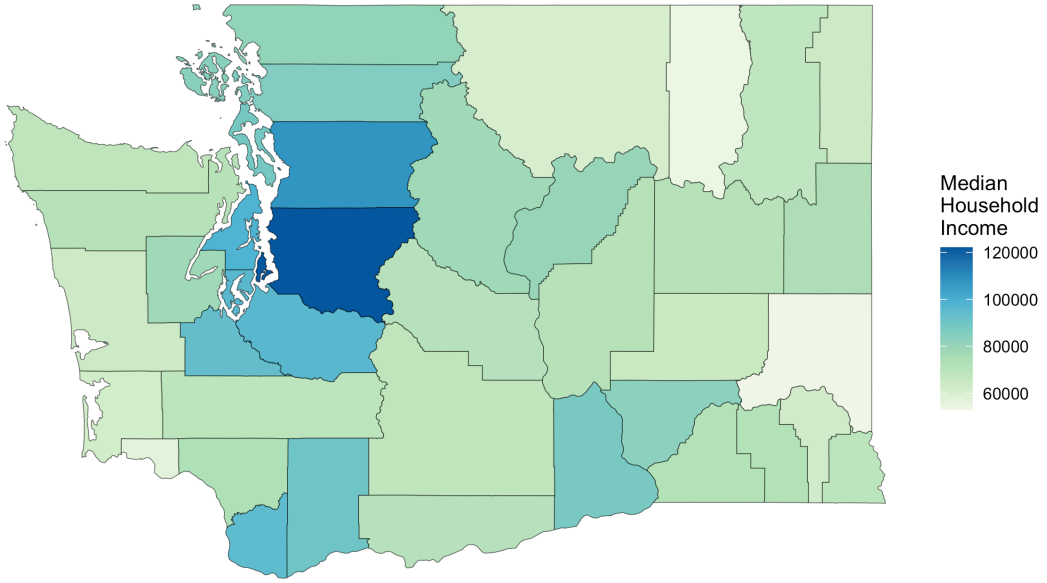
B19013_001 = Median Household Income

```
vars <- load_variables(2021, "acs5", cache = FALSE)
vars
```

```
# A tibble: 27,886 × 4
   name        label                                      concept          geography
   <chr>       <chr>                                      <chr>            <chr>
 1 B01001A_001 Estimate!!Total:                           SEX BY AGE (W… tract
 2 B01001A_002 Estimate!!Total:!!Male:                    SEX BY AGE (W… tract
 3 B01001A_003 Estimate!!Total:!!Male:!!Under 5 years     SEX BY AGE (W… tract
 4 B01001A_004 Estimate!!Total:!!Male:!!5 to 9 years      SEX BY AGE (W… tract
 5 B01001A_005 Estimate!!Total:!!Male:!!10 to 14 years    SEX BY AGE (W… tract
 6 B01001A_006 Estimate!!Total:!!Male:!!15 to 17 years    SEX BY AGE (W… tract
 7 B01001A_007 Estimate!!Total:!!Male:!!18 and 19 years   SEX BY AGE (W… tract
 8 B01001A_008 Estimate!!Total:!!Male:!!20 to 24 years    SEX BY AGE (W… tract
 9 B01001A_009 Estimate!!Total:!!Male:!!25 to 29 years    SEX BY AGE (W… tract
10 B01001A_010 Estimate!!Total:!!Male:!!30 to 34 years    SEX BY AGE (W… tract
# ℹ 27,876 more rows
```

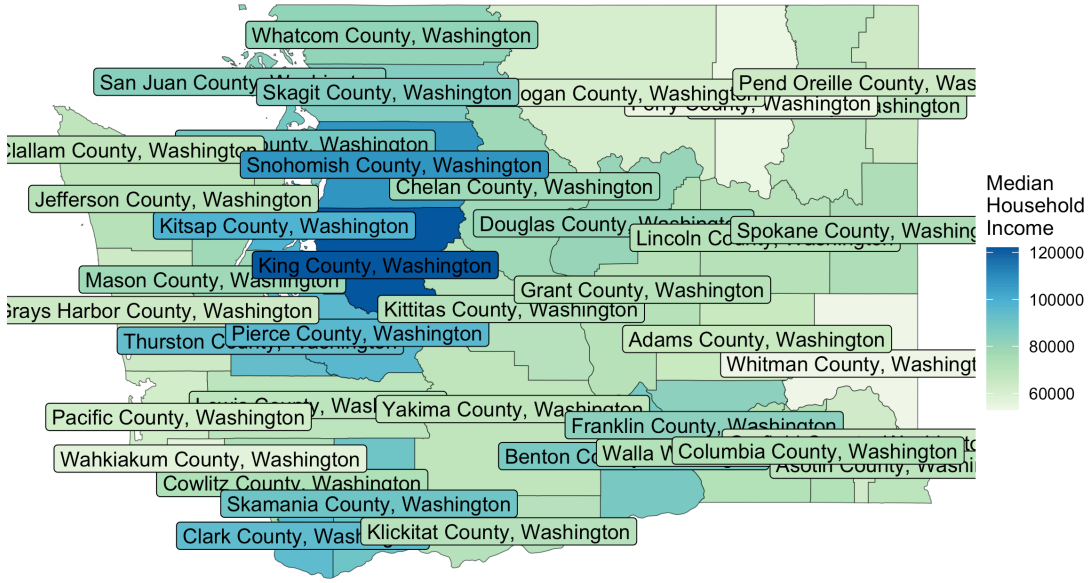
Choropleth Map Showing Median Household Income

```
ggplot(data = wa,
       mapping =
         aes(geometry = geometry,
             fill = estimate)) +
  geom_sf() +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



Choropleth Map

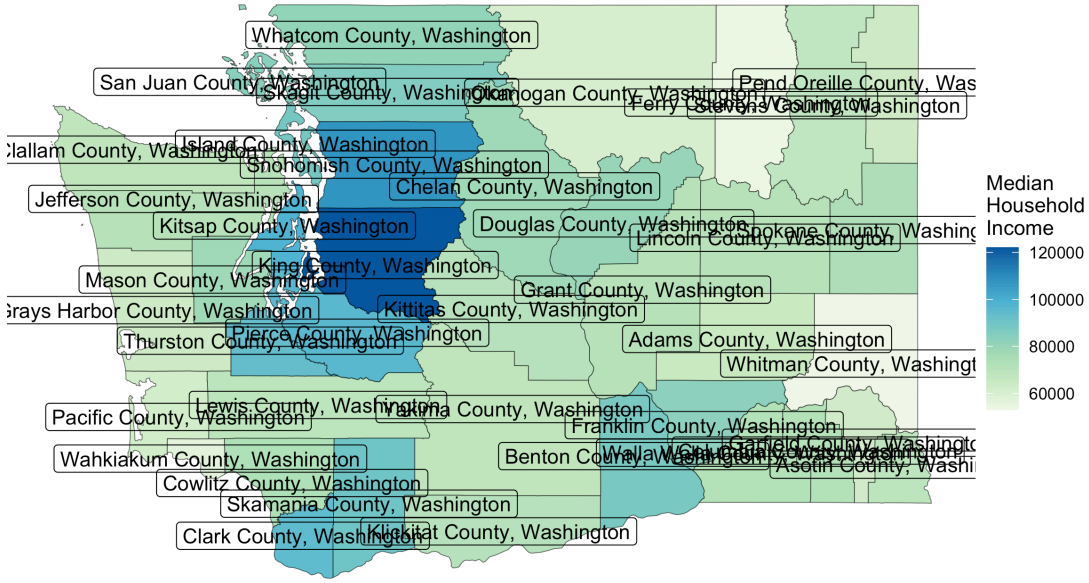
```
ggplot(data = wa,
       mapping =
         aes(geometry = geometry,
             fill = estimate)) +
  geom_sf() +
  geom_sf_label(aes(label = NAME)) +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



- How should we modify the code if we don't want the labels to be colored differently?

Choropleth Map

```
ggplot(data = wa,
       mapping =
         aes(geometry = geometry)) +
  geom_sf(mapping =
            aes(fill = estimate)) +
  geom_sf_label(aes(label = NAME)) +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



- Fixes for crowding?

Again, use the `stringr` Package

This time, for fun, a different function for the same result:

Last time:

```
wa <- wa %>%
  mutate(short_name =
           str_replace_all(
             NAME,
             pattern = " County, Washington",
             replacement = ""
            )
         )
head(wa$short_name)
```

```
[1] "Thurston" "Island"   "Lewis"    "Stevens"  "Douglas"
"San Juan"
```

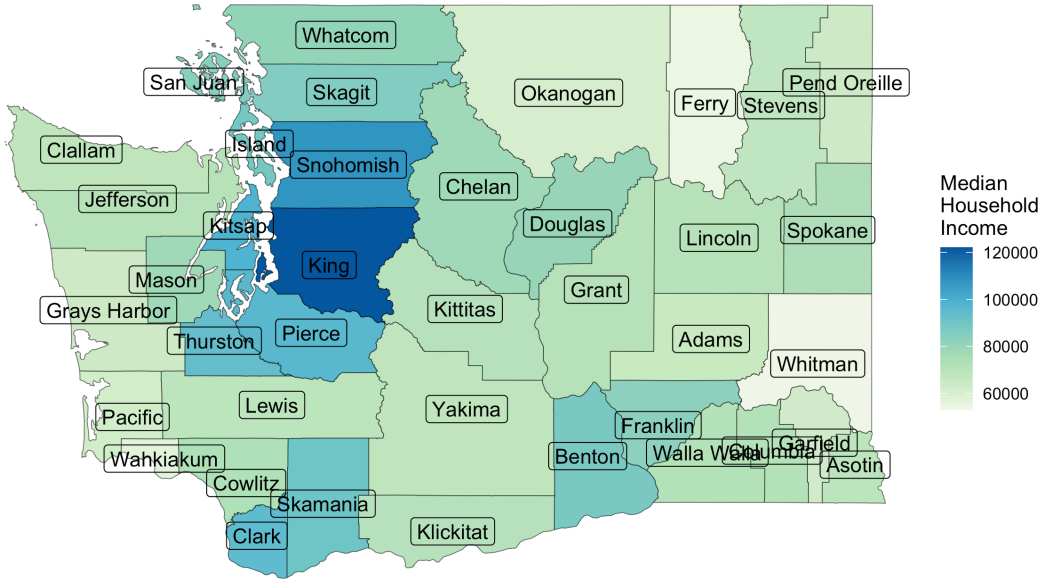
This time:

```
wa <- wa %>%
  mutate(short_name =
           str_sub(NAME,
                   start = 1,
                   end = -20))
head(wa$short_name)
```

```
[1] "Thurston" "Island"   "Lewis"    "Stevens"  "Douglas"
"San Juan"
```

Choropleth Map

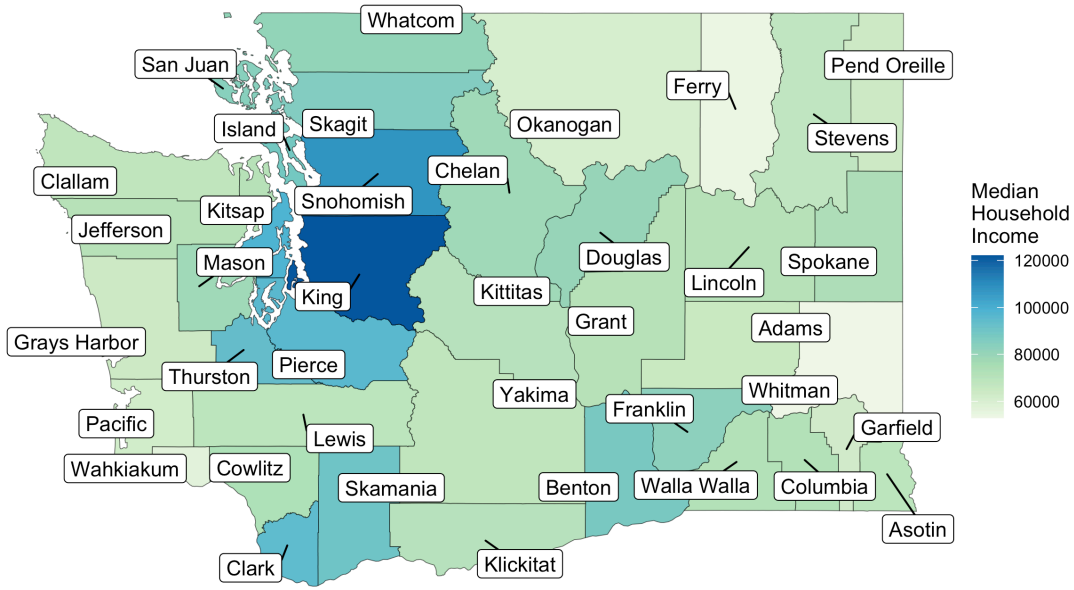
```
ggplot(data = wa,
       mapping =
         aes(geometry = geometry)) +
  geom_sf(mapping =
            aes(fill = estimate)) +
  geom_sf_label(aes(label = short_name)) +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



- Fixes for crowding?

Choropleth Map

```
#devtools::install_github("yutannihilation/ggsflabel")
library(ggsflabel)
ggplot(data = wa,
       mapping =
         aes(geometry = geometry)) +
  geom_sf(mapping =
            aes(fill = estimate)) +
  geom_sf_label_repel(aes(label = short_name),
                      force = 40) +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



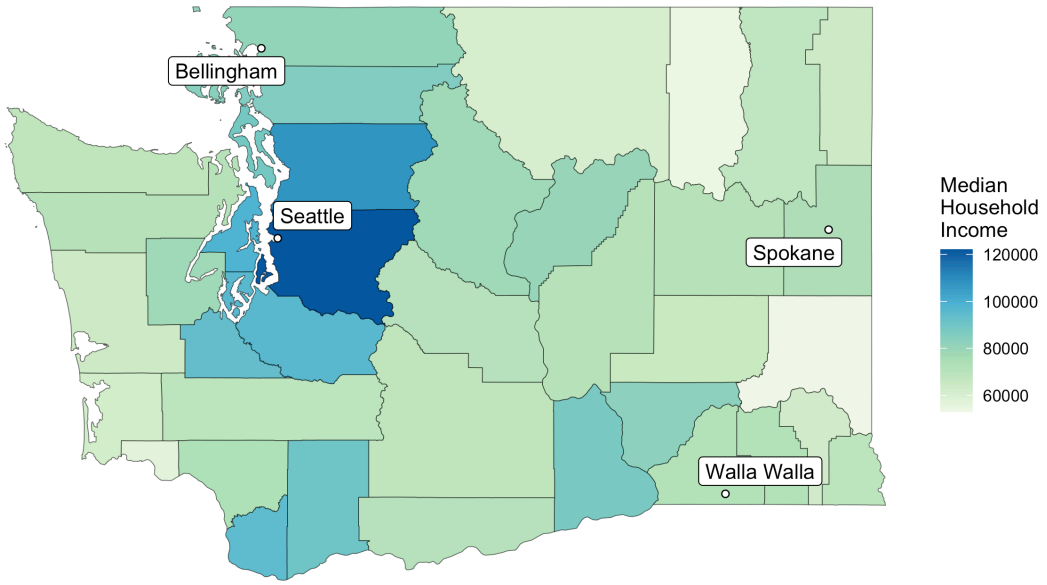
Can Add Information from Another Dataset

```r
wa_cities <- data_frame(city = c("Seattle", "Bellingham",
                                 "Walla Walla", "Spokane"),
                        lat = c(47.6061, 48.7519, 46.0646, 47.6580),
                        long = c(-122.3328, -122.4787,
                                 -118.3430, -117.4235))
wa_cities
```

```
# A tibble: 4 × 3
  city          lat  long
  <chr>       <dbl> <dbl>
1 Seattle      47.6 -122.
2 Bellingham   48.8 -122.
3 Walla Walla  46.1 -118.
4 Spokane      47.7 -117.
```

Can Add Information from Another Dataset

```
library(ggrepel)
ggplot() +
  geom_sf(data = wa,
          mapping =
            aes(geometry = geometry,
                fill = estimate)) +
  geom_point(data = wa_cities,
             mapping = aes(x = long,
                           y = lat),
             shape = 21,
             fill = "white",
             color = "black") +
  geom_label_repel(data = wa_cities,
                   mapping =
                     aes(x = long,
                         y = lat,
                         label = city)) +
  scale_fill_distiller(
```



Coordinate Reference System

Census defaults to NAD 1983 (EPSG: 4269) for its coordinate reference system

```
1 wa
```

```
Simple feature collection with 39 features and 6 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -124.7631 ymin: 45.54354 xmax: -116.916 ymax: 49.00249
Geodetic CRS:  NAD83
First 10 features:
   GEOID                         NAME   variable estimate  moe
1  53067 Thurston County, Washington B19013_001    93985 1815
2  53029   Island County, Washington B19013_001    88358 2918
3  53041    Lewis County, Washington B19013_001    69690 3237
4  53065  Stevens County, Washington B19013_001    67405 3514
5  53017  Douglas County, Washington B19013_001    80374 4122
6  53055 San Juan County, Washington B19013_001    83682 4942
7  53035   Kitsap County, Washington B19013_001    98546 2118
```

Coordinate Reference System

Census defaults to NAD 1983 (EPSG: 4269) for its coordinate reference system

```
1 library(sf)
2 st_crs(wa)
```

```
Coordinate Reference System:
  User input: NAD83
  wkt:
GEOGCRS["NAD83",
    DATUM["North American Datum 1983",
        ELLIPSOID["GRS 1980",6378137,298.257222101,
            LENGTHUNIT["metre",1]]],
    PRIMEM["Greenwich",0,
        ANGLEUNIT["degree",0.0174532925199433]],
    CS[ellipsoidal,2],
        AXIS["latitude",north,
            ORDER[1],
            ANGLEUNIT["degree",0.0174532925199433]],
        AXIS["longitude",east,
```

Coordinate Reference System

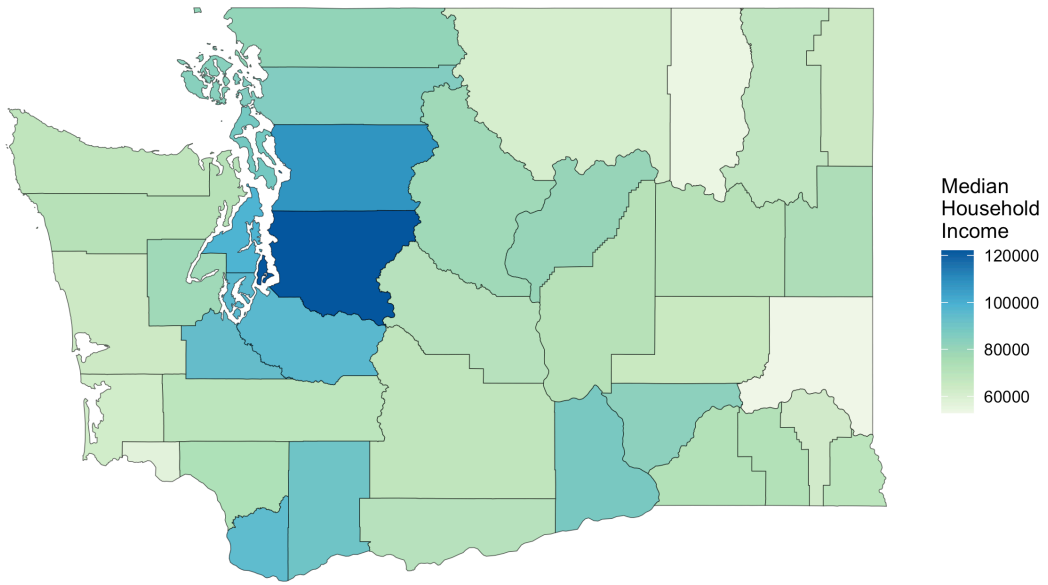
- You can choose a more local coordinate reference system depending on your map extent.
- Projection EPSG:32610 = UTM zone 10N, good for the PNW.

```
1 wa_transf <- sf::st_transform(wa, "EPSG:32610")
2 wa_transf
```

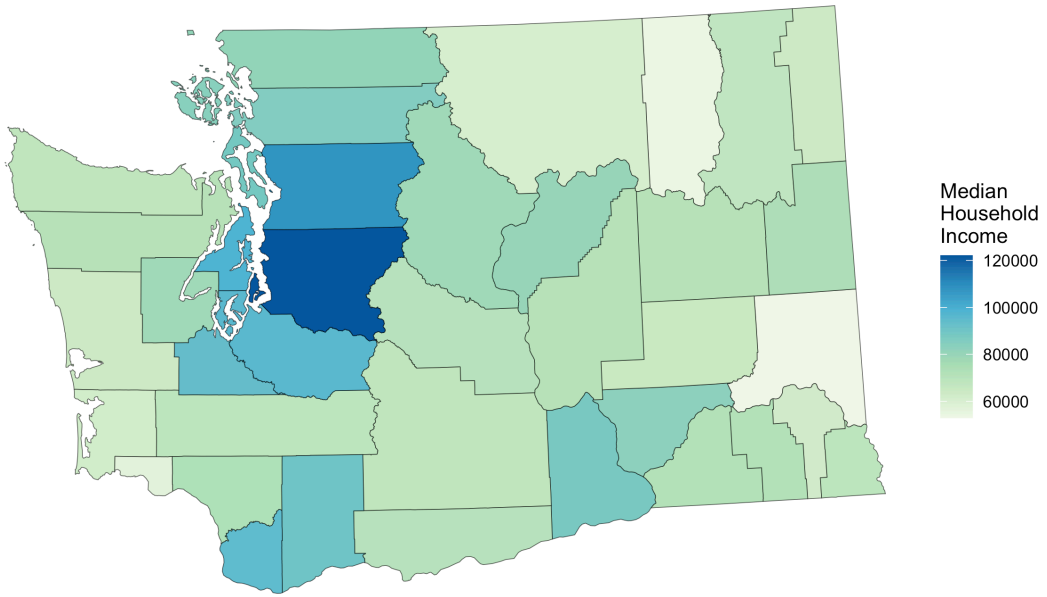
```
Simple feature collection with 39 features and 6 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 368934 ymin: 5043576 xmax: 971107.1 ymax: 5444546
Projected CRS: WGS 84 / UTM zone 10N
First 10 features:
   GEOID                       NAME   variable estimate  moe
1  53067 Thurston County, Washington B19013_001    93985 1815
2  53029   Island County, Washington B19013_001    88358 2918
3  53041    Lewis County, Washington B19013_001    69690 3237
4  53065  Stevens County, Washington B19013_001    67405 3514
5  53017  Douglas County, Washington B19013_001    80374 4122
6  53055 San Juan County, Washington B19013_001    83682 4942
7  53035   Kitsap County, Washington B19013_001    98546 2118
```

Different CRS -> Different Looking Map

```
ggplot() +
  geom_sf(data = wa,
          mapping =
            aes(geometry = geometry,
                fill = estimate)) +
  coord_sf() +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



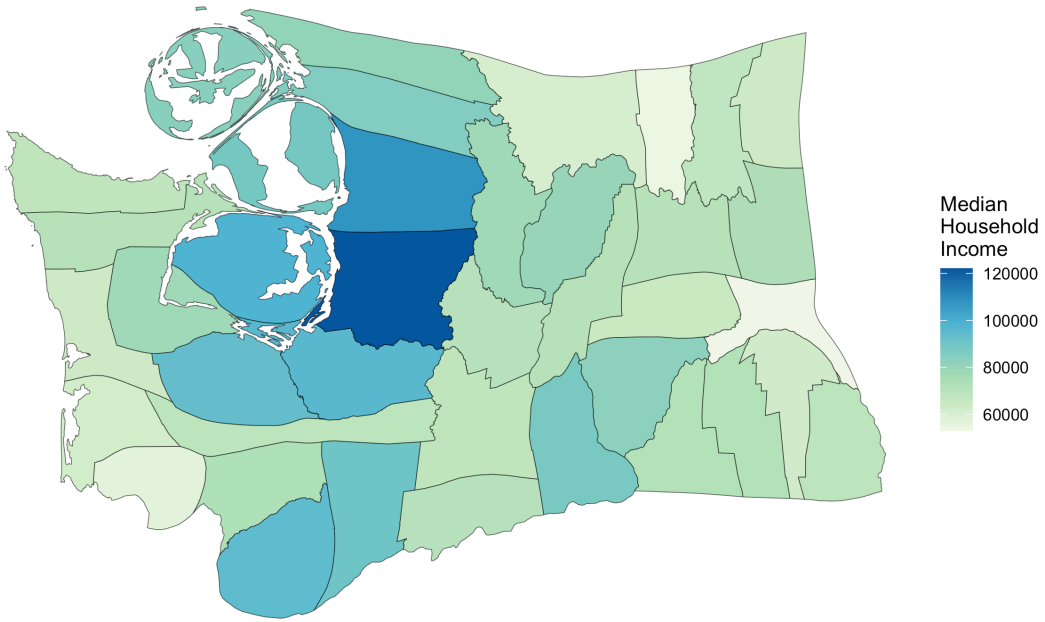

```
ggplot() +
  geom_sf(data = wa_transf,
          mapping =
            aes(geometry = geometry,
                fill = estimate)) +
  coord_sf() +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



Cartograms

Scale polygons by a variable.

```
library(cartogram)
wa_carto <- cartogram_cont(wa_transf,
                           weight = "estimate")

ggplot() +
  geom_sf(data = wa_carto,
          mapping =
            aes(geometry = geometry,
                fill = estimate)) +
  coord_sf() +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
  theme_void()
```



Cartograms

Often scaled by a population size type variable.

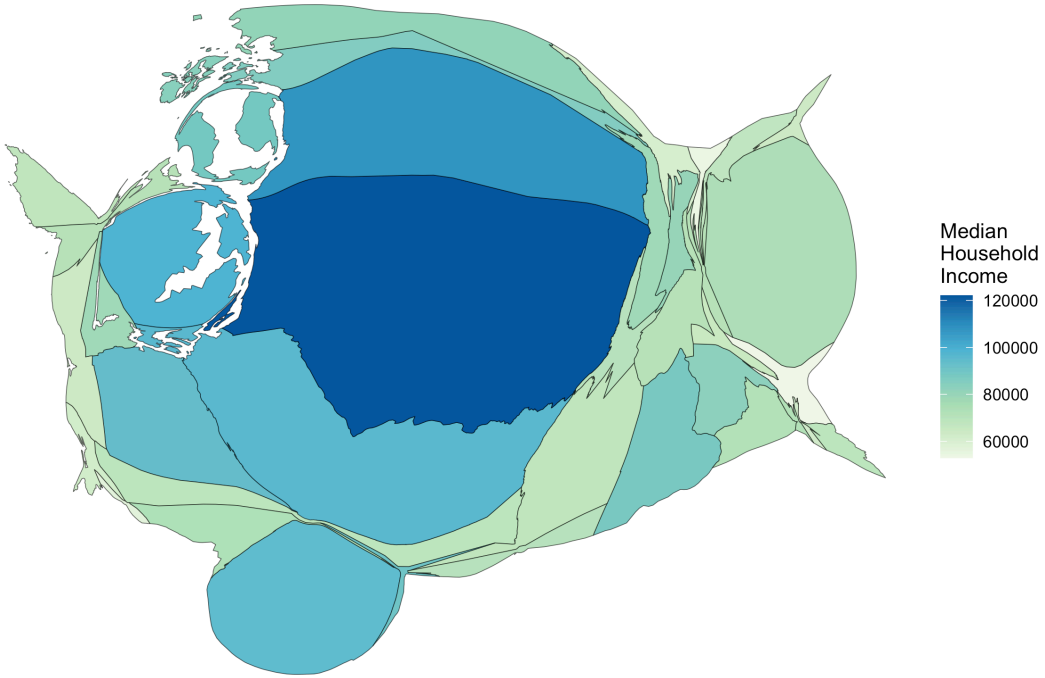
```
wa_pop <- get_decennial(state = "WA", geography = "county",
                        variables = "P001001", year = 2010, api_key = api_key)

wa_transf_pop <- inner_join(wa_transf, wa_pop, by = c("GEOID" = "GEOID"))
wa_transf_pop
```

```
Simple feature collection with 39 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 368934 ymin: 5043576 xmax: 971107.1 ymax: 5444546
Projected CRS: WGS 84 / UTM zone 10N
First 10 features:
   GEOID                      NAME.x variable.x estimate  moe short_name
1  53067 Thurston County, Washington B19013_001    93985 1815   Thurston
2  53029   Island County, Washington B19013_001    88358 2918     Island
3  53041    Lewis County, Washington B19013_001    69690 3237      Lewis
4  53065  Stevens County, Washington B19013_001    67405 3514    Stevens
5  53017  Douglas County, Washington B19013_001    80374 4122    Douglas
6  53055 San Juan County, Washington B19013_001    83682 4942   San Juan
7  53035   Kitsap County, Washington B19013_001    98546 2118     Kitsap
```

Cartograms

Often scaled by a population size type variable.

```
wa_transf_pop <- wa_transf_pop %>%
  rename(population_size = value,
         median_income = estimate)

wa_transf_pop <- cartogram_cont(
  wa_transf_pop,
  weight = "population_size"
  )

ggplot() +
  geom_sf(data = wa_transf_pop,
          mapping =
            aes(geometry = geometry,
                fill = median_income)) +
  coord_sf() +
  scale_fill_distiller(
   name = "Median \nHousehold \nIncome",
   direction = 1, type = "seq", palette = 4) +
```



New Dataset

State Level Information

```
1 library(Lock5Data)
2 glimpse(USStates)
```

```
Rows: 50
Columns: 22
$ State           <fct> Alabama, Alaska, Arizona, Arkansas, California, Color…
$ HouseholdIncome <dbl> 46.472, 76.114, 53.510, 43.813, 67.169, 65.458, 73.78…
$ Region          <fct> S, W, W, S, W, W, NE, NE, S, S, W, W, MW, MW, MW, MW,…
$ Population      <dbl> 4.875, 0.740, 7.016, 3.004, 39.537, 5.607, 3.588, 0.9…
$ EighthGradeMath <dbl> 268.7, 274.3, 279.9, 274.4, 275.6, 284.7, 286.2, 276.…
$ HighSchool      <dbl> 87.1, 92.8, 87.1, 89.1, 87.4, 91.5, 92.4, 89.2, 89.4,…
$ College         <dbl> 26.0, 26.5, 27.4, 24.7, 34.5, 39.6, 42.7, 33.4, 28.8,…
$ IQ              <dbl> 95.7, 99.0, 97.4, 97.5, 95.5, 101.6, 103.1, 100.4, 98…
$ GSP             <dbl> 40.279, 70.936, 43.096, 38.467, 67.698, 59.057, 67.78…
$ Vegetables      <dbl> 80.7, 81.0, 79.2, 80.7, 78.6, 82.6, 83.1, 82.8, 80.6,…
$ Fruit           <dbl> 55.1, 63.1, 62.8, 55.3, 67.5, 67.0, 68.5, 64.6, 65.6,…
$ Smokers         <dbl> 20.9, 21.0, 15.6, 22.3, 11.3, 14.6, 12.7, 17.0, 16.1,…
```

Hexbin Maps

```
#install.packages("statebins")
library(statebins)
statebins(state_data = USStates,
          state_col = "State",
          value_col = "Insured",
          ggplot2_scale_function =
            ggplot2::scale_fill_distiller,
          direction = 1, type = "seq", palette = 4,
          round = TRUE) +
  theme_statebins()
```

```
Error in `nchar()`:
! 'nchar()' requires a character vector
```

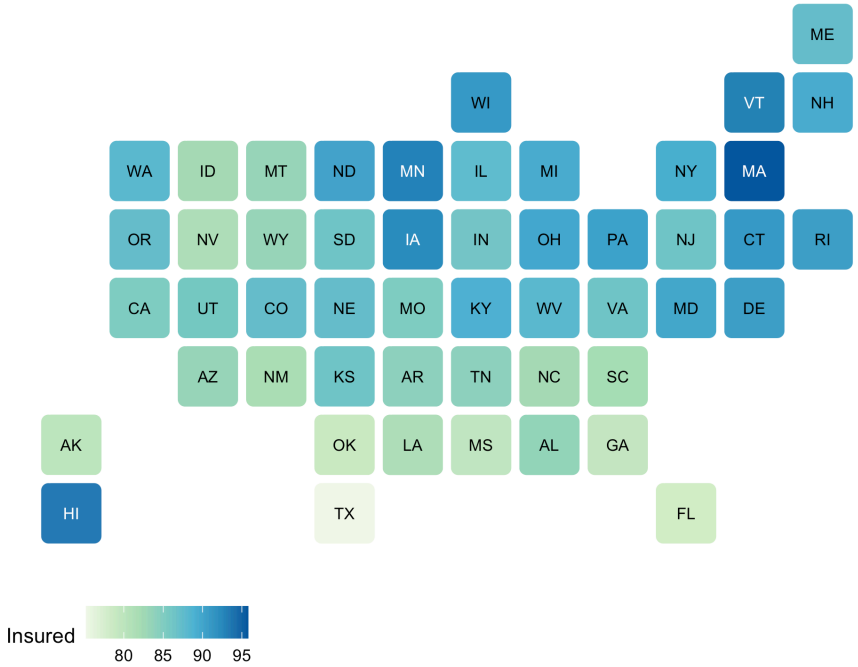
```
class(USStates$State)
```

```
[1] "factor"
```

```
USStates <- mutate(USStates,
                   State = as.character(State))
```

Hexbin Maps

```
library(statebins)
statebins(state_data = USStates,
          state_col = "State",
          value_col = "Insured",
          ggplot2_scale_function =
            ggplot2::scale_fill_distiller,
          direction = 1, type = "seq", palette = 4,
          round = TRUE) +
  theme_statebins()
```



New data: volcano eruptions

```
Eruptions <- read_csv("data/GVP_Eruption_Results.csv")
select(Eruptions, VolcanoName, StartYear, Latitude, Longitude) %>%
  glimpse()
```

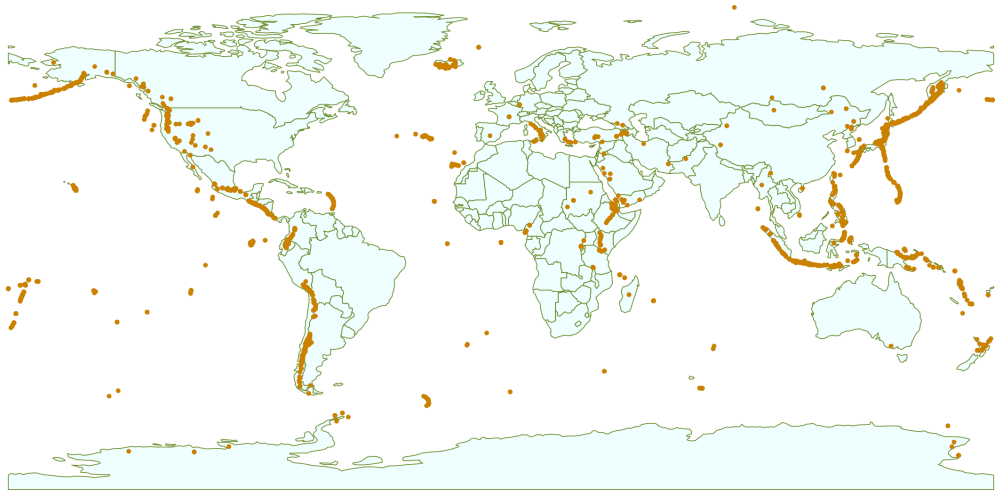
```
Rows: 11,019
Columns: 4
$ VolcanoName <chr> "Bulusan", "Alaid", "Turrialba", "San Miguel", "Chill\xe0n…
$ StartYear   <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2015, 2015, 2015, 2015…
$ Latitude    <dbl> 12.770, 50.861, 10.025, 13.434, -36.863, 1.112, 11.984, 37…
$ Longitude   <dbl> 124.050, 155.565, -83.767, -88.269, -71.377, 124.737, -86.…
```

```
library(rnaturalearth)
world <- rnaturalearth::countries110
world
```

```
Simple feature collection with 177 features and 168 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -180 ymin: -90 xmax: 180 ymax: 83.64513
Geodetic CRS:  WGS 84
First 10 features:
        featurecla scalerank LABELRANK                  SOVEREIGNT SOV_A3
1  Admin-0 country         1         6                        Fiji    FJI
2  Admin-0 country         1         3 United Republic of Tanzania    TZA
3  Admin-0 country         1         7              Western Sahara    SAH
4  Admin-0 country         1         2                      Canada    CAN
5  Admin-0 country         1         2   United States of America    US1
6  Admin-0 country         1         3                  Kazakhstan    KA1
7  Admin-0 country         1         3                  Uzbekistan    UZB
```

Static Maps

```
Eruptions_sf <- Eruptions %>%
  st_as_sf(coords = c("Longitude",
                      "Latitude"),
           crs = "EPSG:4269")

ggplot(data = world) +
  geom_sf(fill = "azure",
          color = "olivedrab4") +
  geom_sf(data = Eruptions_sf,
          color = "orange3",
          size = 0.5) +
  theme_void()
```



- What if we want to zoom in?

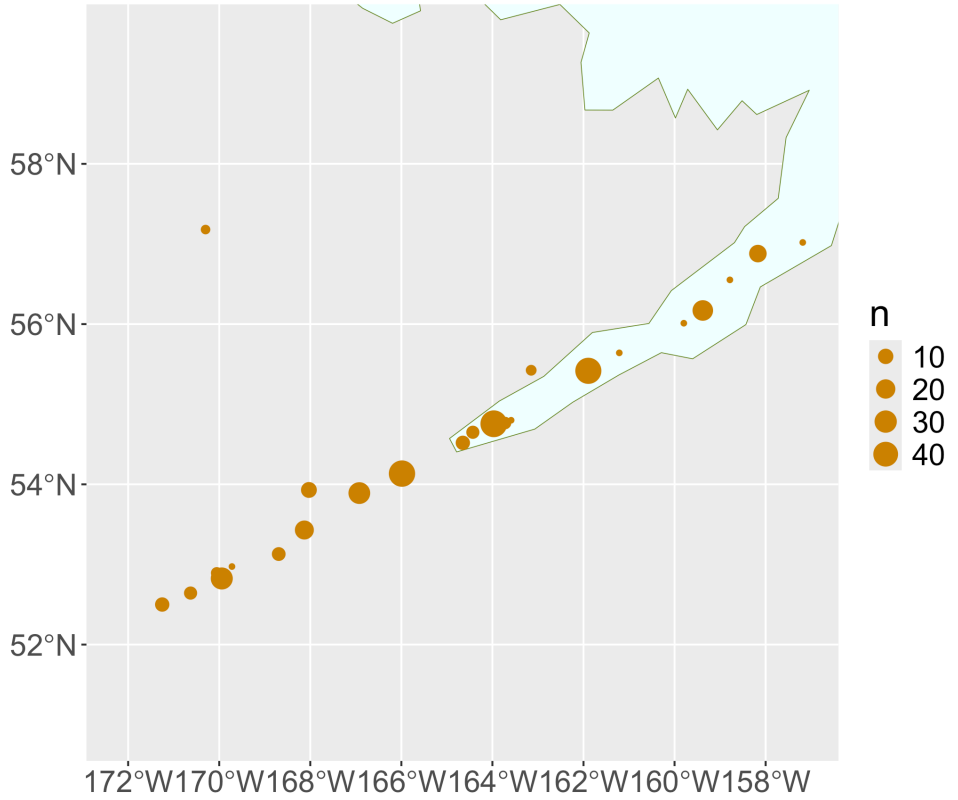
Alluetian Islands

```
eruption_count <- count(Eruptions,
                        VolcanoName,
                        Latitude,
                        Longitude) %>%
  filter(Longitude > -172.164,
         Longitude < -157.1507,
         Latitude > 50.977,
         Latitude < 59.5617) %>%
  arrange(desc(n)) %>%
  st_as_sf(coords = c("Longitude",
                      "Latitude"),
           crs = "EPSG:4269")
eruption_count
```

```
Simple feature collection with 24 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -171.252 ymin: 52.5 xmax: -157.185 ymax: 57.18
Geodetic CRS:  NAD83
# A tibble: 24 × 3
   VolcanoName     n          geometry
 * <chr>       <int>       <POINT [°]>
 1 Shishaldin     49  (-163.97 54.756)
 2 Akutan         47 (-165.986 54.134)
 3 Pavlof         46 (-161.894 55.417)
 4 Cleveland      29 (-169.944 52.825)
 5 Makushin       28 (-166.923 53.891)
 6 Veniaminof     24   (-159.38 56.17)
```

Alluetian Islands

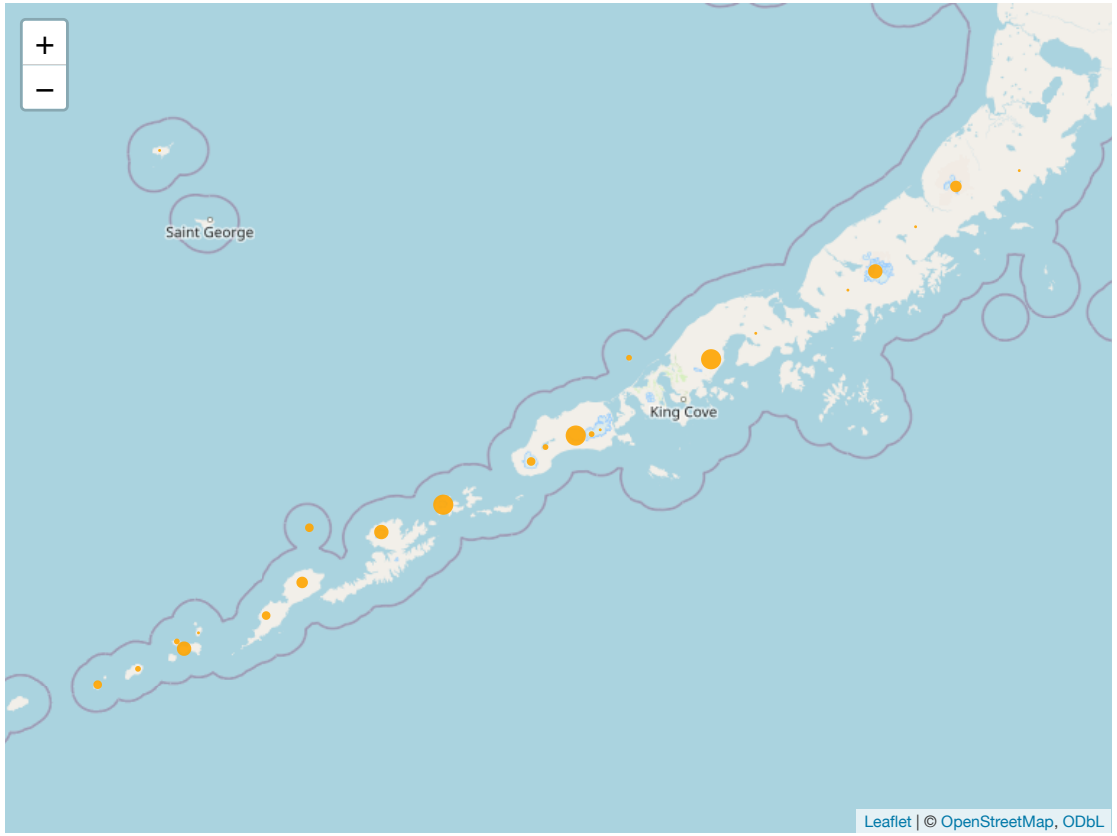
```
ggplot() +
  geom_sf(data = world,
          fill = "azure",
          color = "olivedrab4") +
  geom_sf(data = eruption_count,
          mapping = aes(size = n),
          color = "orange3") +
  coord_sf(xlim = c(-172.164, -157.1507),
           ylim = c(50.977, 59.5617))
```



- Not a great base layer…
- What about dynamic zooming and clickable labels?

Interactive Maps with `leaflet`

- Syntax similar to `ggplot2` and `dplyr`
- Interactive zooming
- Embed interactive map into Quarto documents (HTML output)
- Can take lat/lon **or** geometry/`sf` objects



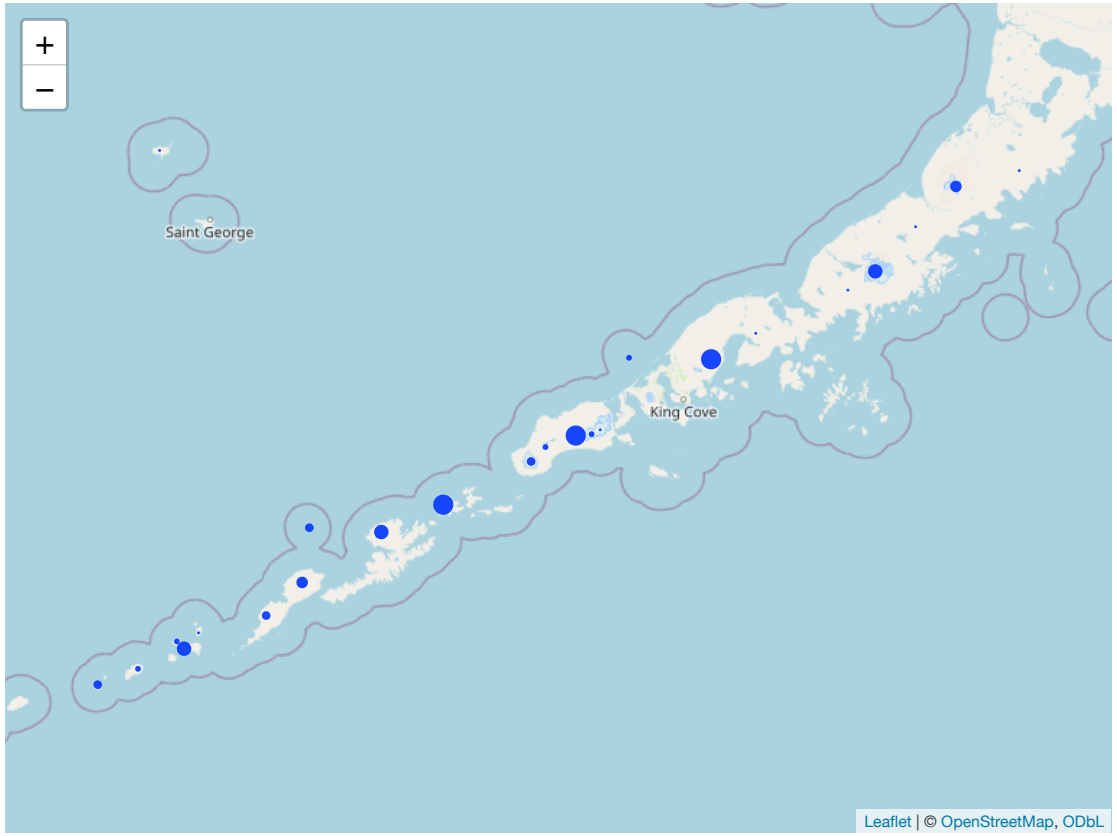
Interactive Maps with `leaflet`

```
1 library(leaflet)
2 leaflet() %>%
3   addTiles() %>%
4   addCircleMarkers(data = eruption_count,
5                    radius = ~sqrt(n),
6                    stroke = FALSE,
7                    fillOpacity = 0.9,
8                    color = "orange")
```

- First Layer: `leaflet()`
  - Creates the map widget
- Second layer: `addTiles()`
  - Creates a base map using OpenStreetMap
- Additional layers: `addMarkers()`, `addPolygons()`

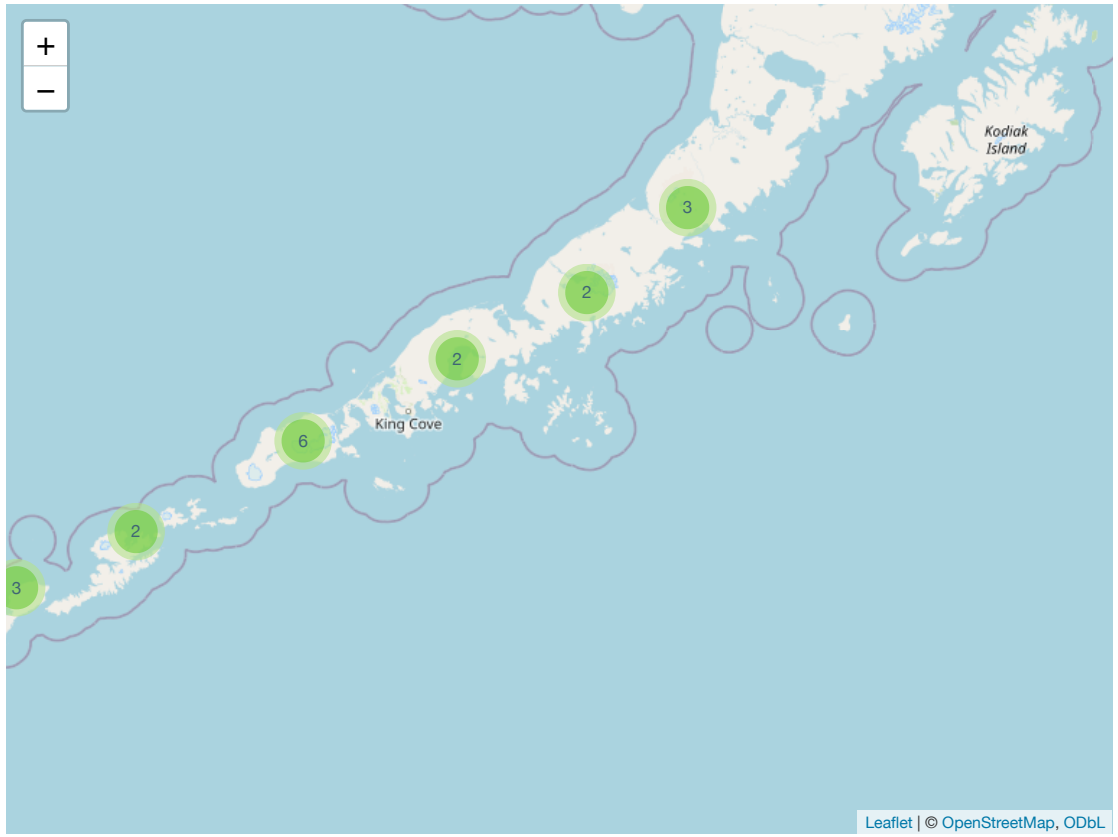
Interactive Maps with `leaflet`

```
1  library(leaflet)
2  leaflet() %>%
3    addTiles() %>%
4    addCircleMarkers(data = eruption_count,
5                     radius = ~sqrt(n),
6                     stroke = FALSE,
7                     fillOpacity = 0.9)
```



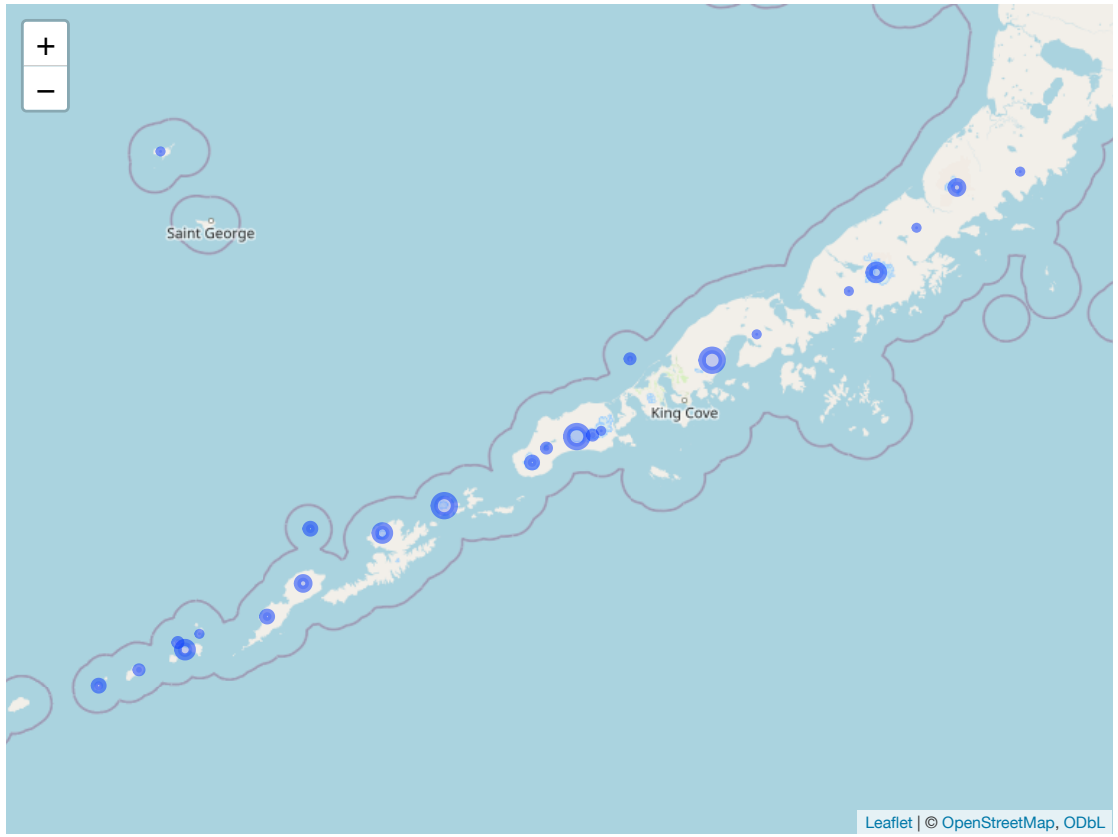
Clusters

```
leaflet() %>%
  setView(lng = -160, lat = 55, zoom = 6) %>%
  addTiles() %>%
  addCircleMarkers(data = eruption_count,
                   clusterOptions =
                     markerClusterOptions())
```



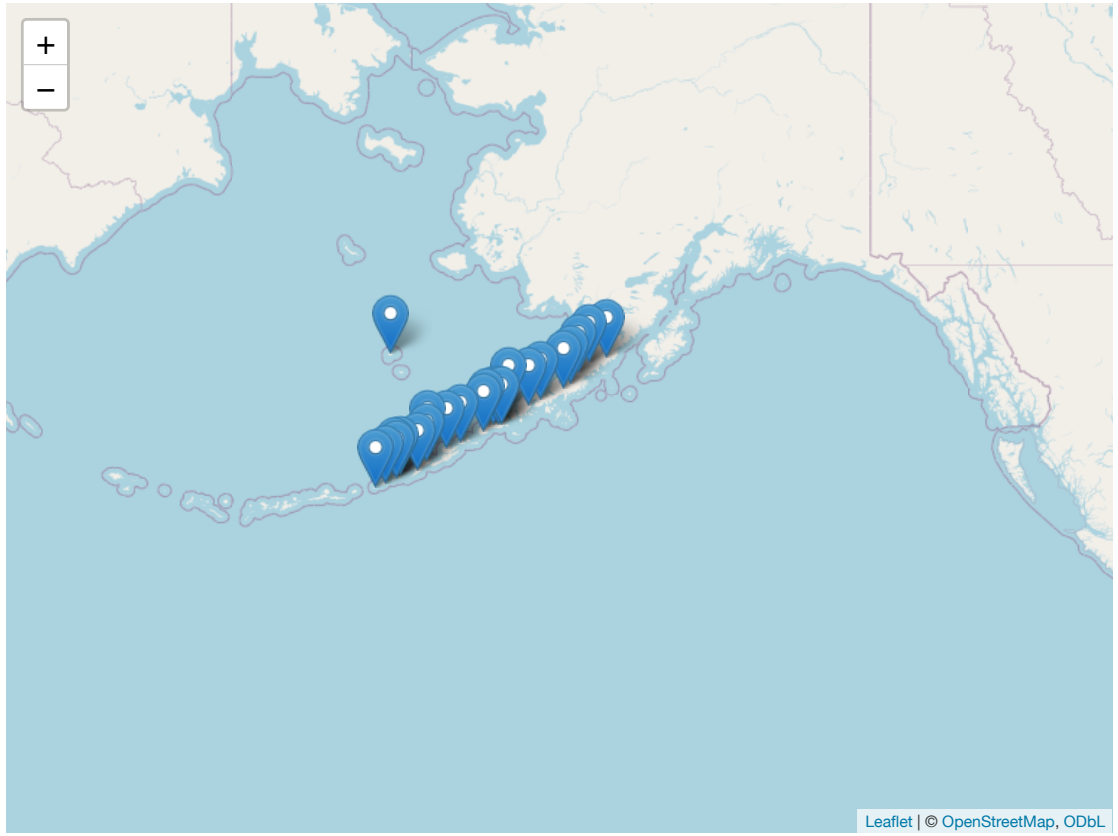
Zooming Constraints

```
leaflet(options =
          leafletOptions(minZoom = 3,
                         maxZoom = 7)) %>%
  addTiles() %>%
  addCircleMarkers(data = eruption_count,
                   radius = ~sqrt(n))
```



Other Markers

```
leaflet() %>%
  setView(lng = -160, lat = 55, zoom = 4) %>%
  addTiles() %>%
  addMarkers(data = eruption_count,
             label = ~as.character(n))
```



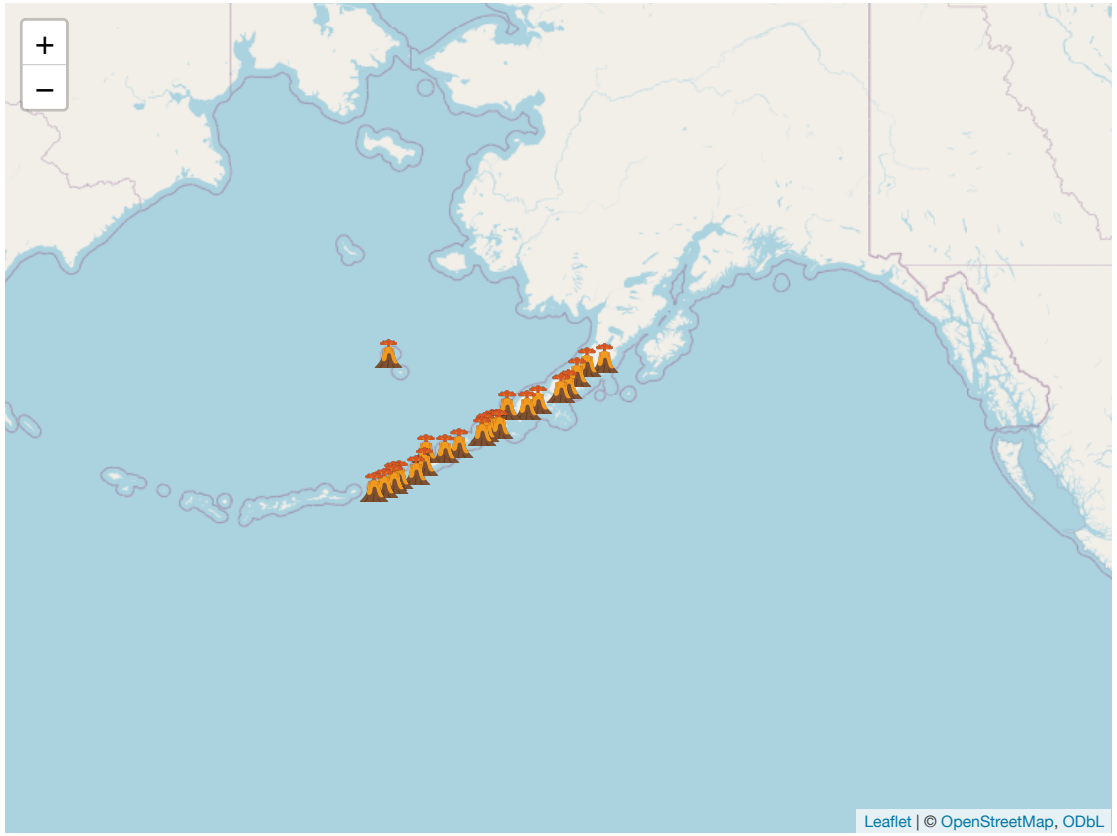
Custom Icons

```
volcano  <- makeIcon(
  iconUrl = "https://raw.githubusercontent.com/Reed-Data-Science/Reed-Data-Science.github.io/refs/heads/main/slides/
  iconWidth = 20, iconHeight = 20)
volcano
```

```
$iconUrl
[1] "https://raw.githubusercontent.com/Reed-Data-Science/Reed-Data-
Science.github.io/refs/heads/main/slides/img/volcano.png"

$iconWidth
[1] 20

$iconHeight
[1] 20

attr(,"class")
[1] "leaflet_icon"
```

Custom Icons

```
leaflet() %>%
  setView(lng = -160, lat = 55, zoom = 4) %>%
  addTiles() %>%
  addMarkers(data = eruption_count,
             label = ~as.character(n),
             icon = volcano)
```



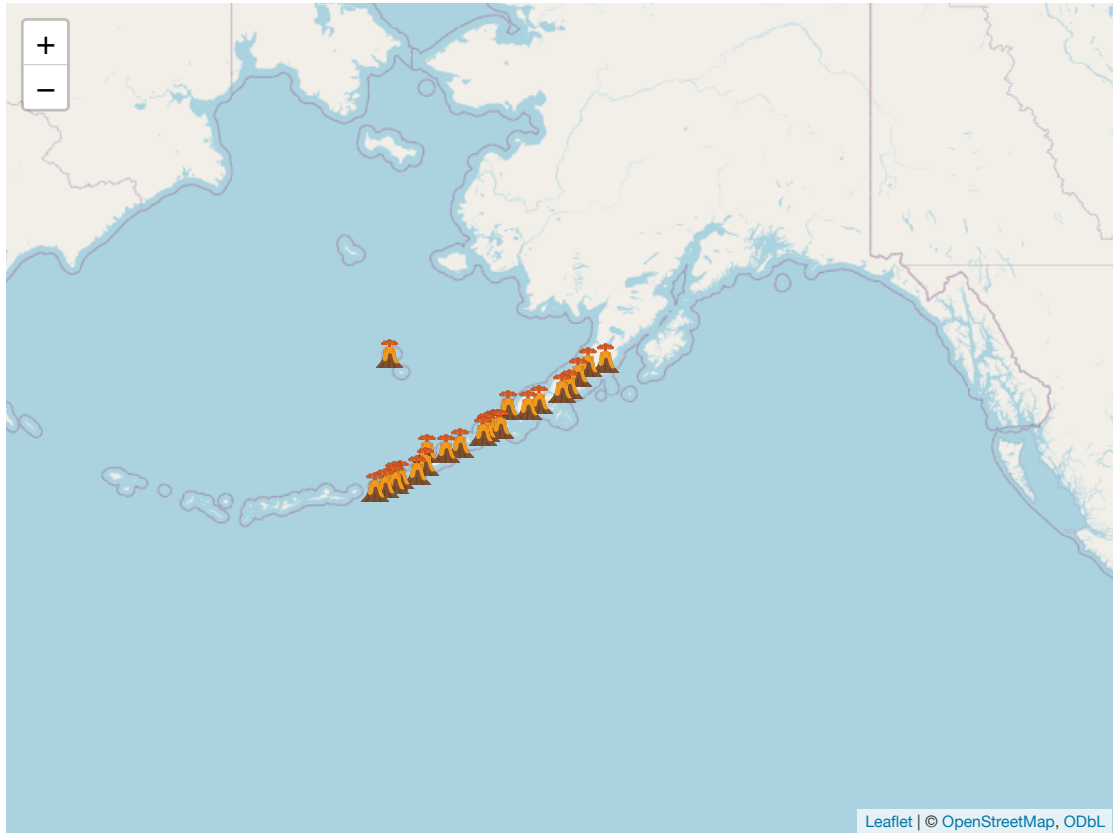
Pop-Ups

```
content <- paste("<b>",
                 eruption_count$VolcanoName,
                 "</b></br>",
                 "Number of eruptions:",
                 eruption_count$n)
content
```

```
 [1] "<b> Shishaldin </b></br> Number of eruptions: 49"
 [2] "<b> Akutan </b></br> Number of eruptions: 47"
 [3] "<b> Pavlof </b></br> Number of eruptions: 46"
 [4] "<b> Cleveland </b></br> Number of eruptions: 29"
 [5] "<b> Makushin </b></br> Number of eruptions: 28"
 [6] "<b> Veniaminof </b></br> Number of eruptions: 24"
 [7] "<b> Okmok </b></br> Number of eruptions: 19"
 [8] "<b> Aniakchak </b></br> Number of eruptions: 15"
 [9] "<b> Bogoslof </b></br> Number of eruptions: 11"
[10] "<b> Amukta </b></br> Number of eruptions: 8"
[11] "<b> Westdahl </b></br> Number of eruptions: 8"
[12] "<b> Vsevidof </b></br> Number of eruptions: 7"
[13] "<b> Fisher </b></br> Number of eruptions: 6"
[14] "<b> Yunaska </b></br> Number of eruptions: 6"
```

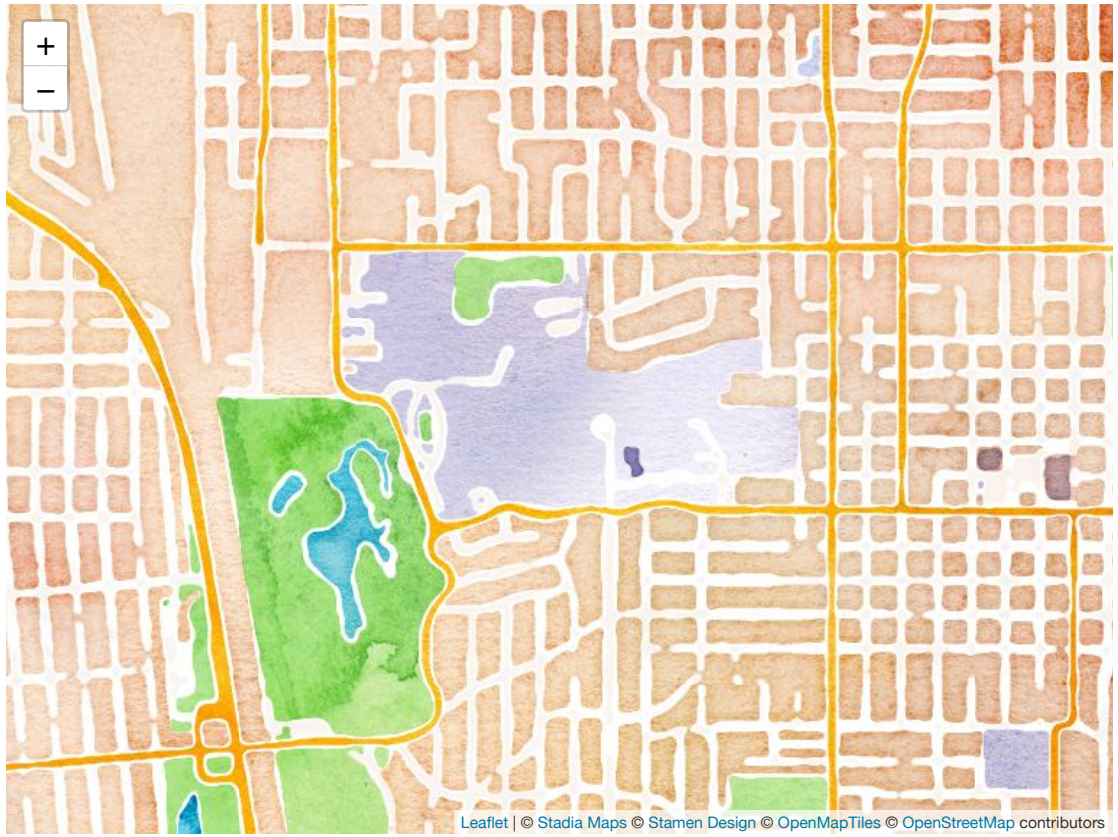
Pop-Ups

```
leaflet() %>%
  setView(lng = -160, lat = 55, zoom = 4) %>%
  addTiles() %>%
  addMarkers(data = eruption_count,
             popup = content,
             icon = volcano)
```



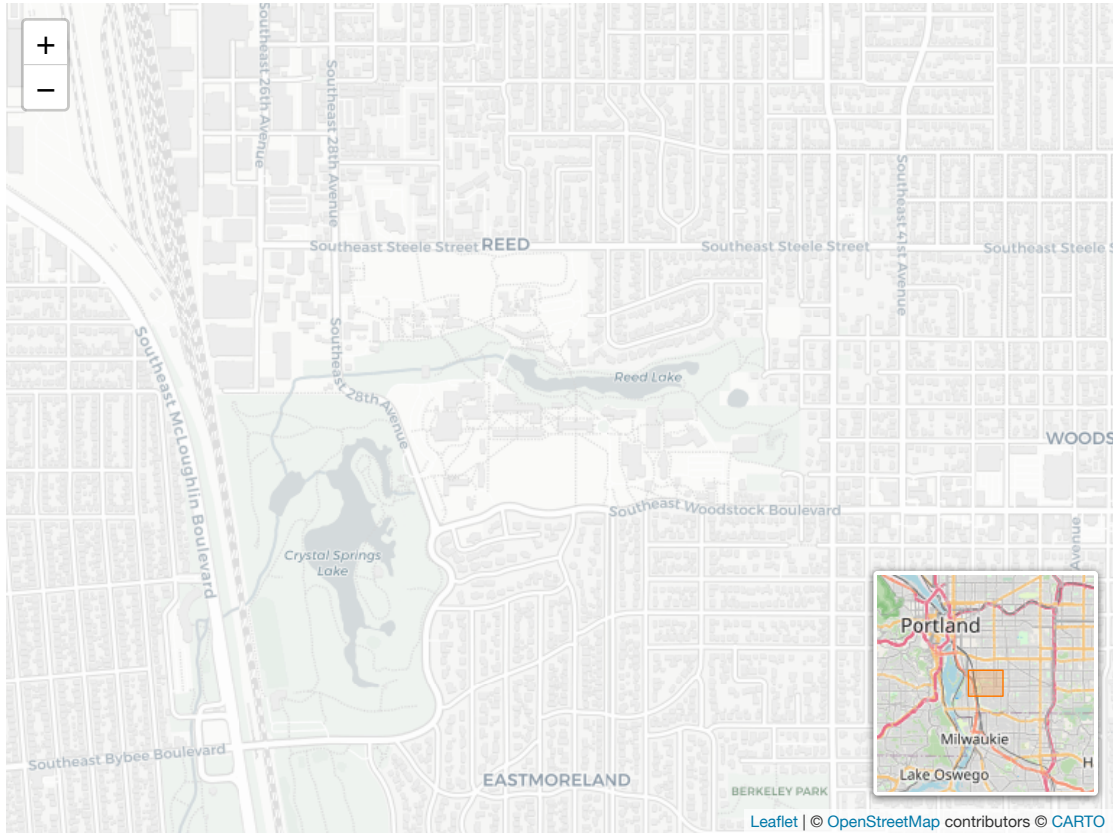
Other Tiles

```
library(leaflet.extras)
leaflet() %>%
  setView(lng = -122.6308,
          lat = 45.4811,
          zoom = 15) %>%
  addProviderTiles(providers$Stadia.StamenWatercolor)
```



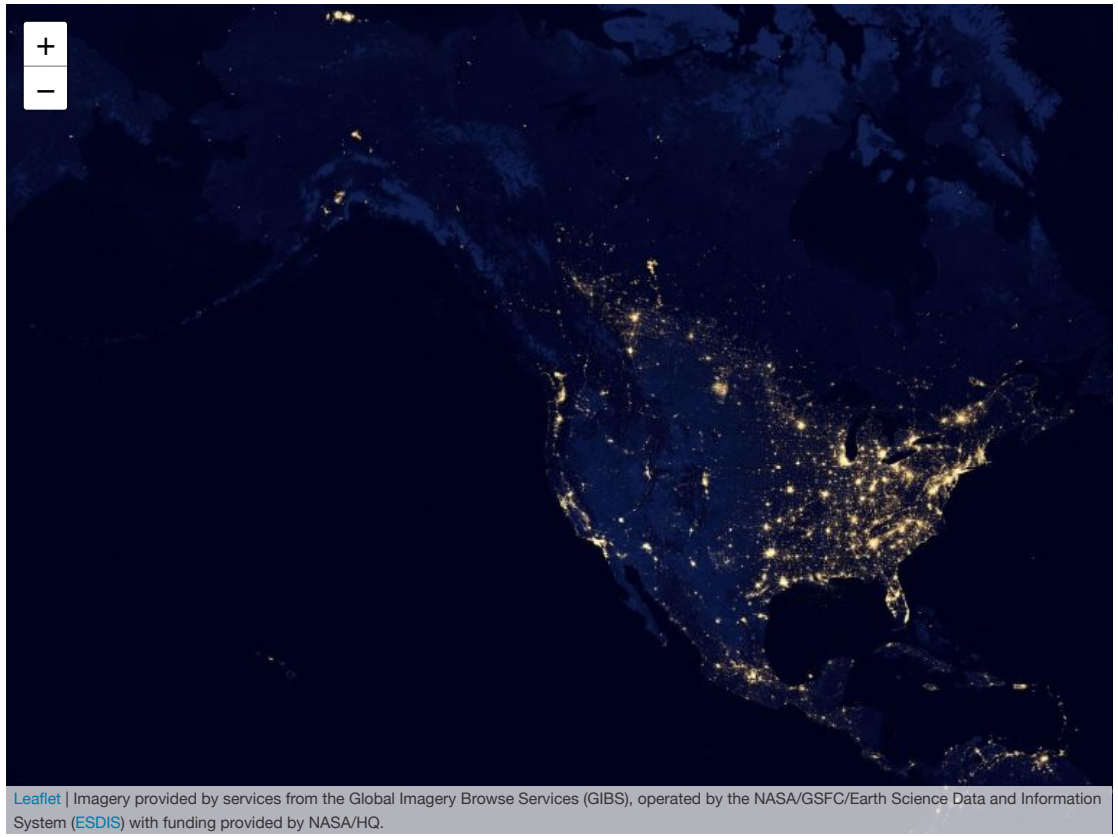
Other Tiles

```
leaflet() %>%
  setView(lng = -122.6308,
          lat = 45.4811,
          zoom = 15) %>%
 addProviderTiles(
   providers$CartoDB.Positron) %>%
  addMiniMap()
```



Other Tiles

```
leaflet() %>%
  setView(lng = -122.6308,
          lat = 45.4811,
          zoom = 3) %>%
  addProviderTiles(
    "NASAGIBS.ViirsEarthAtNight2012")
```



Interactive Choropleth Map

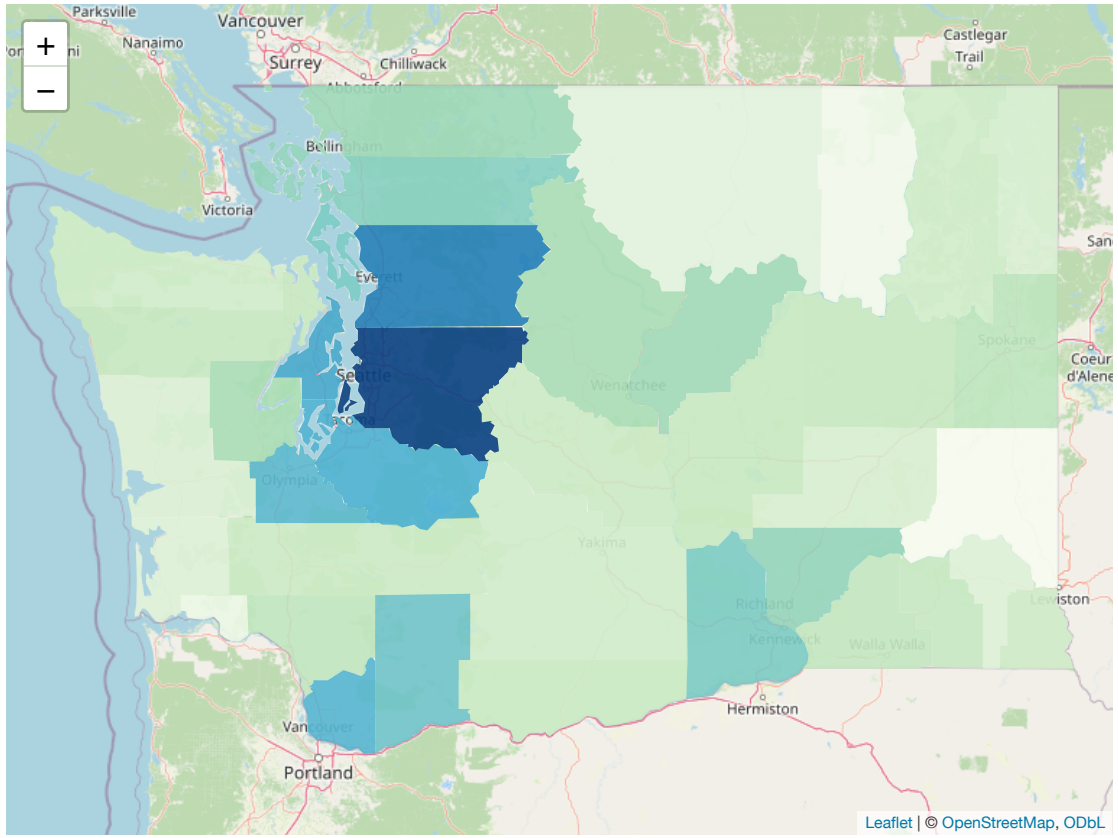
```
pal <- colorNumeric(palette = "GnBu",
                    domain = wa$estimate,
                    reverse = FALSE)
pal(wa$estimate)
```

```
 [1] "#5CB9CF" "#77CAC6" "#CDEBC6" "#D2EECC" "#A1DAB8" "#90D3BD" "#45A8CD"
 [8] "#56B6D1" "#AADEB6" "#4CB1D2" "#F7FCF0" "#F2FAEC" "#D1EDCB" "#E3F4DE"
[15] "#CDEBC6" "#AADEB6" "#C8E9C3" "#87D0C1" "#C8E9C3" "#DEF2D9" "#084081"
[22] "#227FB7" "#DBF1D6" "#7CCCC4" "#C1E7C0" "#C4E8C1" "#D1EDCB" "#95D5BC"
[29] "#9ED9B9" "#CBEBC5" "#ECF8E6" "#DEF2D9" "#CFECC8" "#DBF1D5" "#D8F0D2"
[36] "#BEE6BF" "#C7E9C3" "#C8EAC3" "#70C4C9"
```

Interactive Choropleth Map

```
wa %>%
  leaflet() %>%
  addTiles() %>%
  addPolygons(popup = ~NAME,
              color = ~pal(estimate),
              stroke = FALSE,
              fillOpacity = 0.9)
```

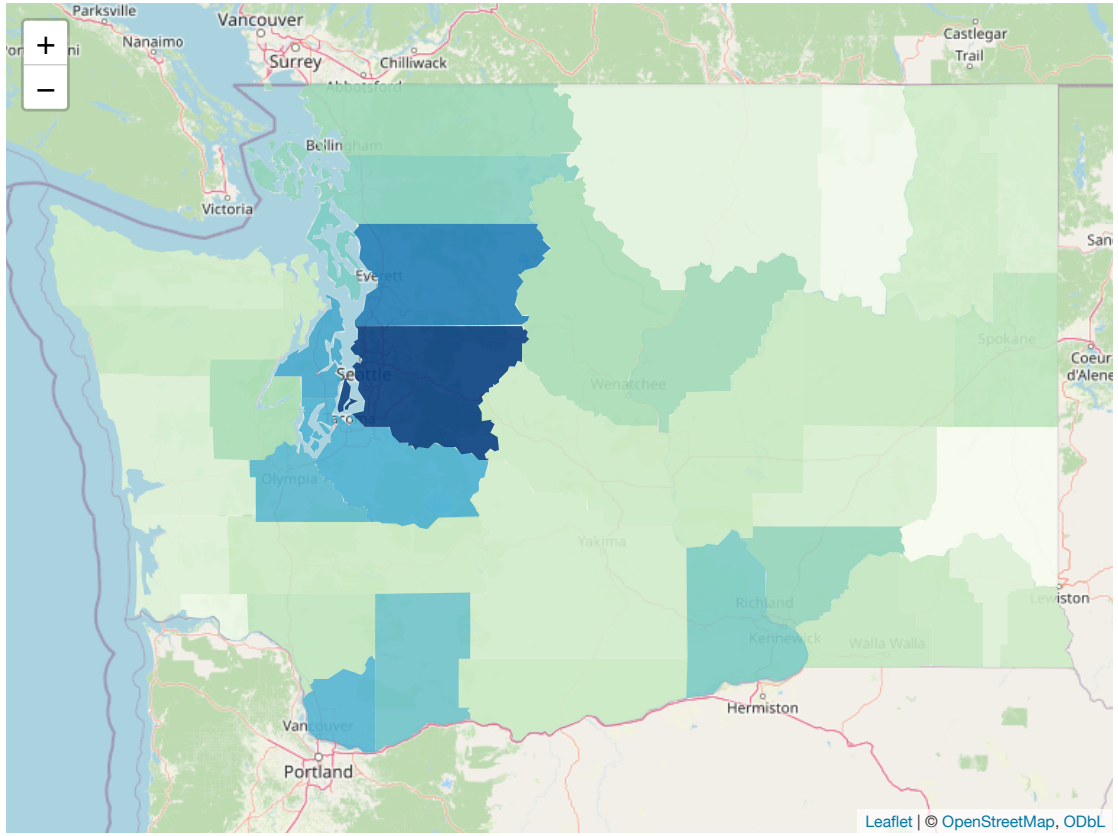
```
Warning: sf layer has inconsistent datum (+proj=longlat
+datum=NAD83 +no_defs).
Need '+proj=longlat +datum=WGS84'
```



- Throws a warning about the projection

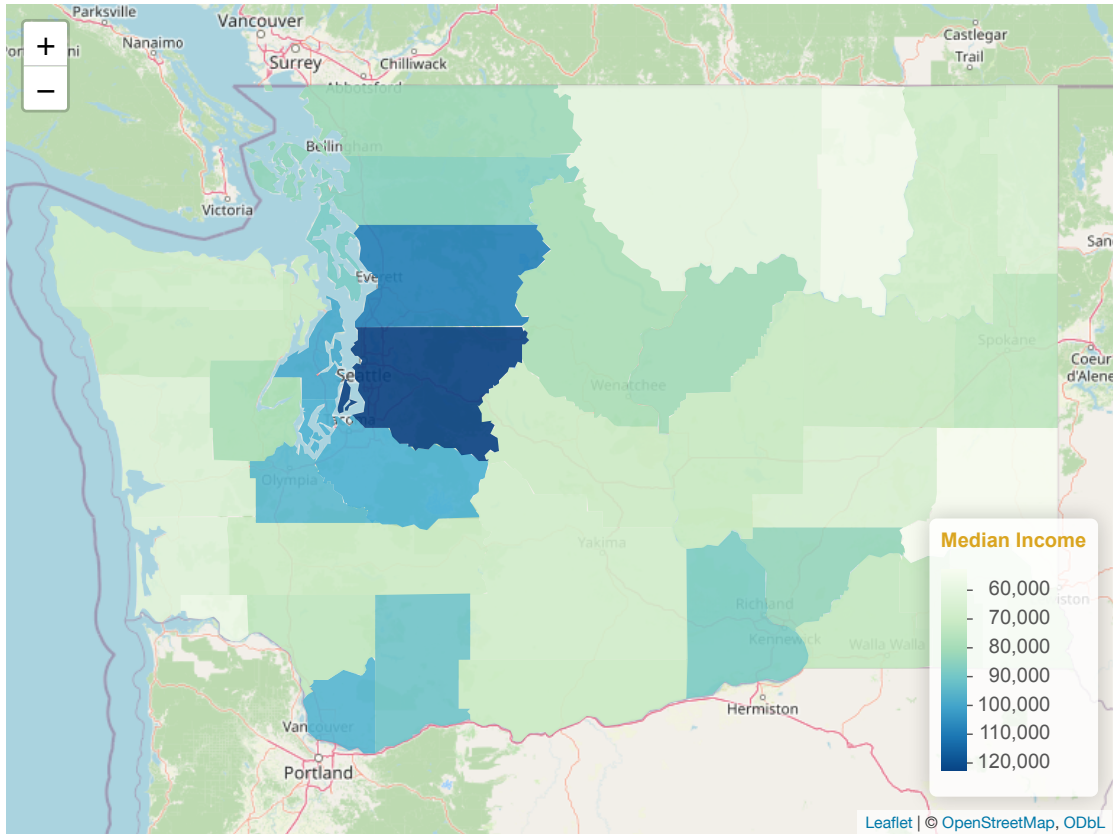
Interactive Choropleth Map

```
wa %>%
  sf::st_transform(crs = "EPSG:4326") %>%
  leaflet() %>%
  addTiles() %>%
  addPolygons(popup = ~NAME,
              color = ~pal(estimate),
              stroke = FALSE,
              fillOpacity = 0.9)
```



Interactive Choropleth Map

```
wa %>%
  sf::st_transform(crs = "EPSG:4326") %>%
  leaflet() %>%
  addTiles() %>%
  addPolygons(popup = ~NAME,
              color = ~pal(estimate),
              stroke = FALSE,
              fillOpacity = 0.9) %>%
  addLegend("bottomright", pal = pal,
            values = ~estimate,
            title = "Median Income",
            opacity = 1)
```



Next time

- More spatial data:
  - Spatial computations
  - Raster data

Next Week

- Learn to develop `shiny` dashboards
  - for increased user interactivity!
  - (and for Project 1)

Resources for learning more about spatial data

- Dig into wrangling spatial data with the `sf` package: https://github.com/rstudio/cheatsheets/blob/master/sf.pdf
- All things spatial data in R: https://geocompr.robinlovelace.net/index.html