

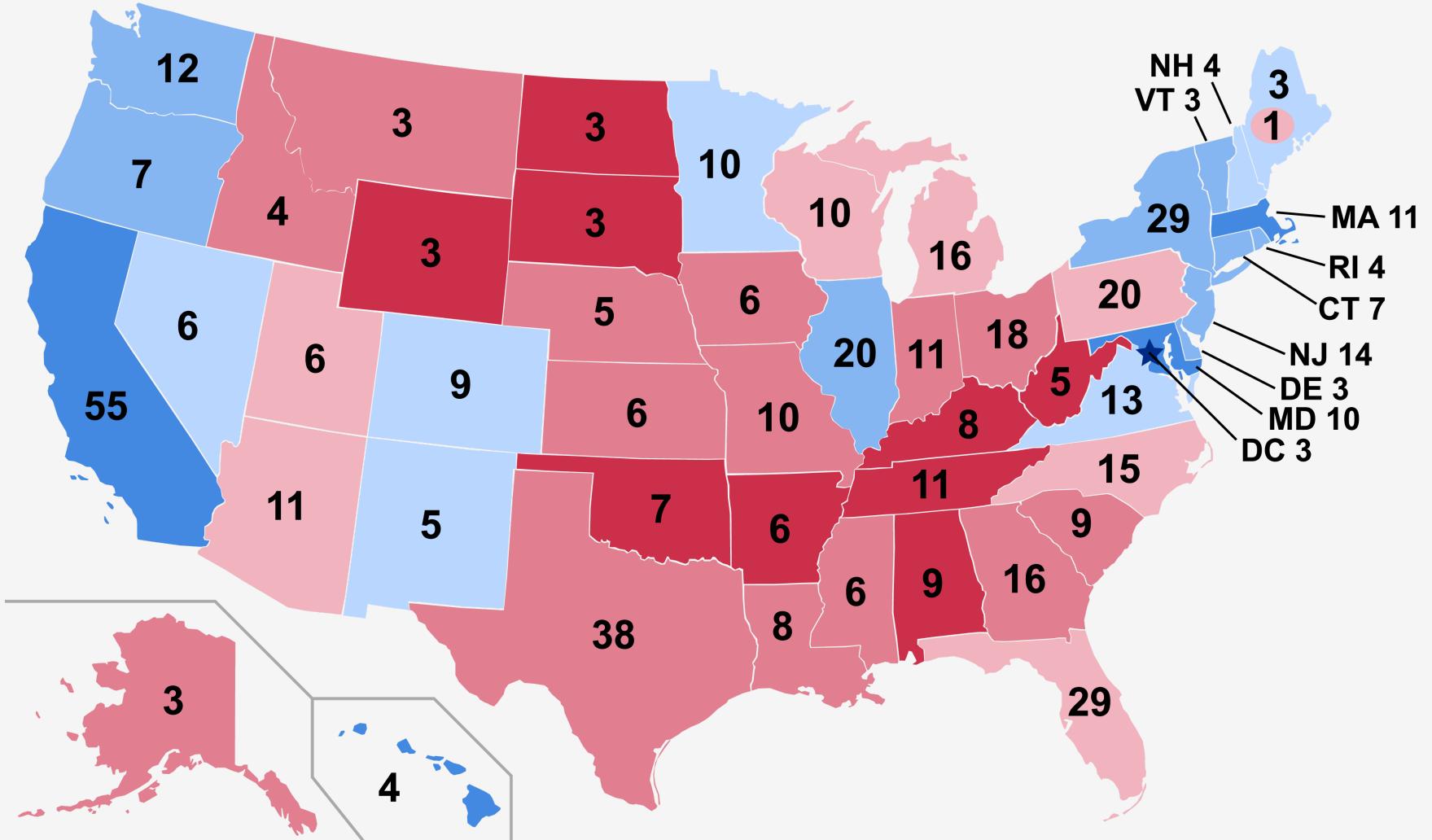
Making Maps

Data Visualization: Session 7
Kieran Healy
Code Horizons, May 2022

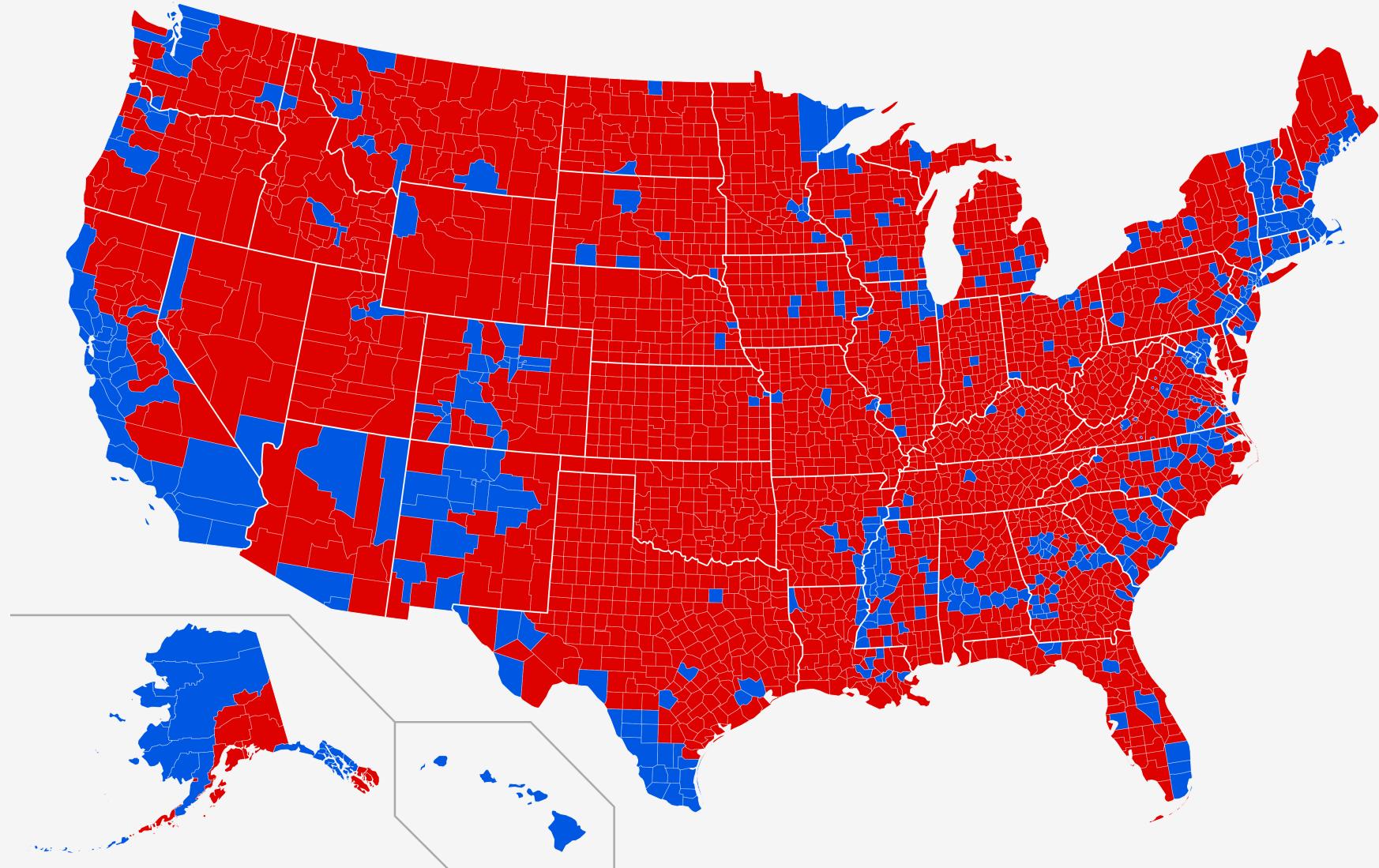
Load our libraries

```
library(here)      # manage file paths  
library(socviz)    # data and some useful functions  
library(tidyverse) # your friend and mine  
library(maps)      # Some basic maps  
library(sf)        # Simple Features Geometries and geom_sf()  
library(ggforce)   # Useful enhancements to ggplot
```

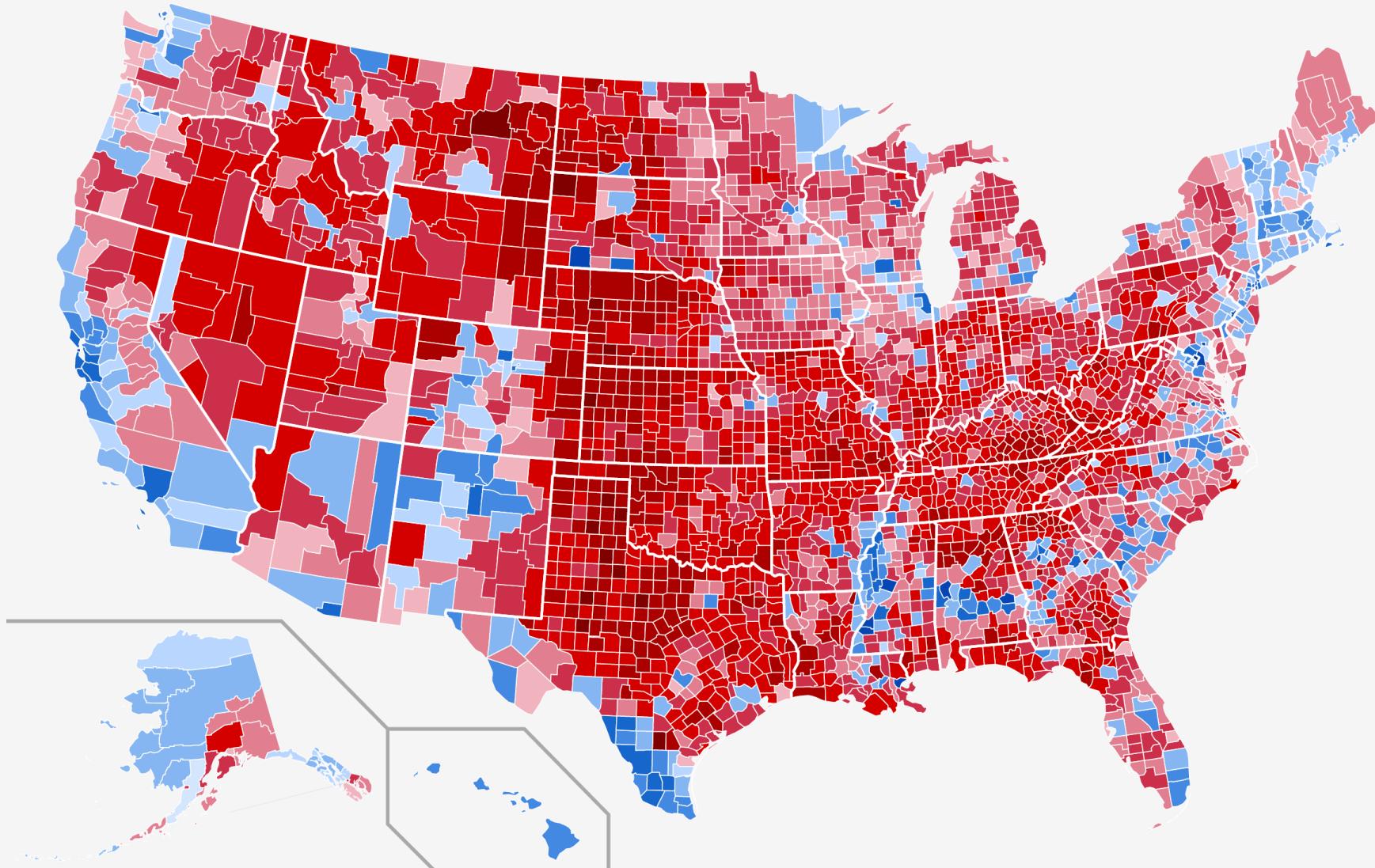
Choropleths



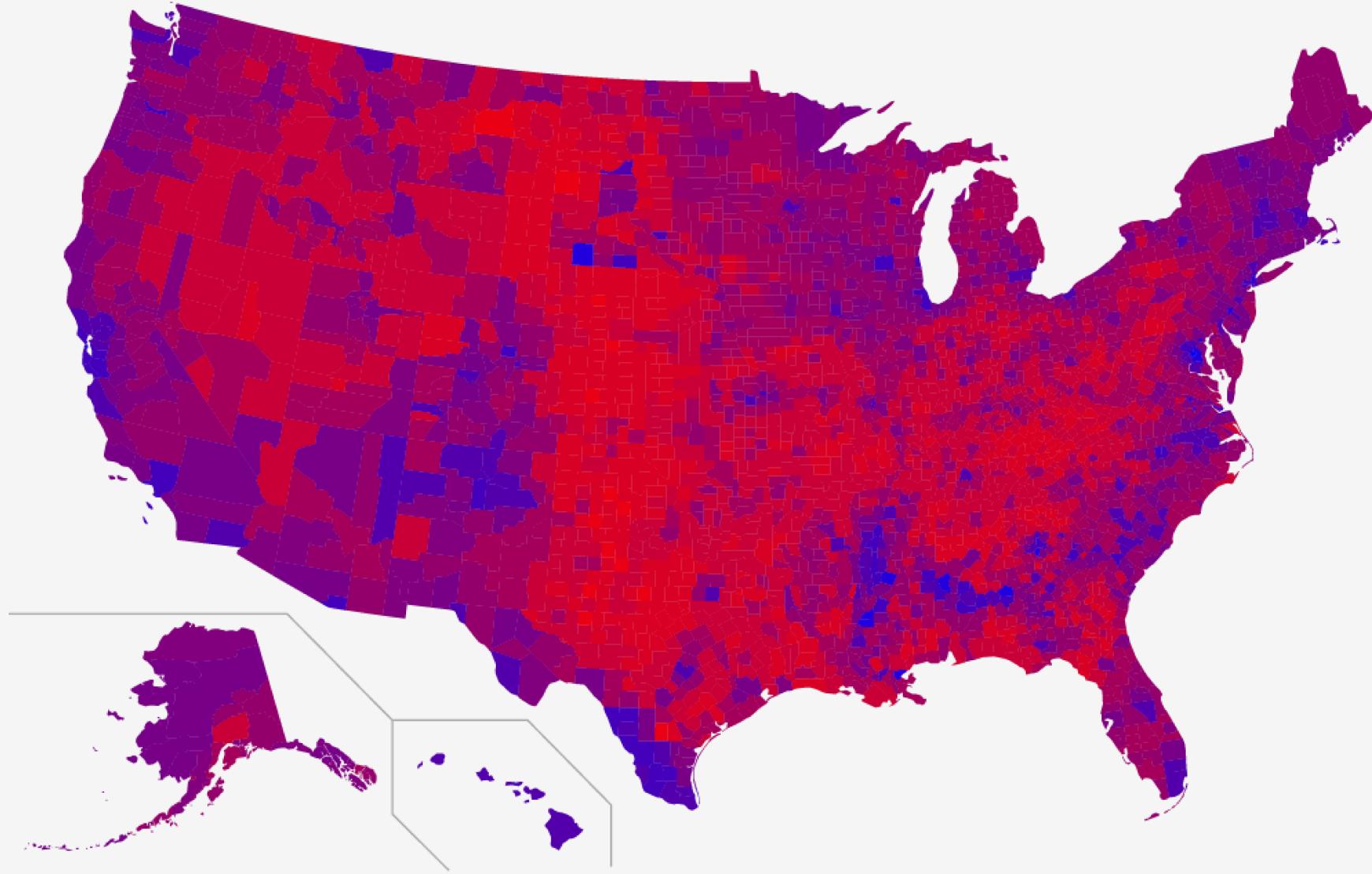
State-level; vote share; diverging; binned into four categories.



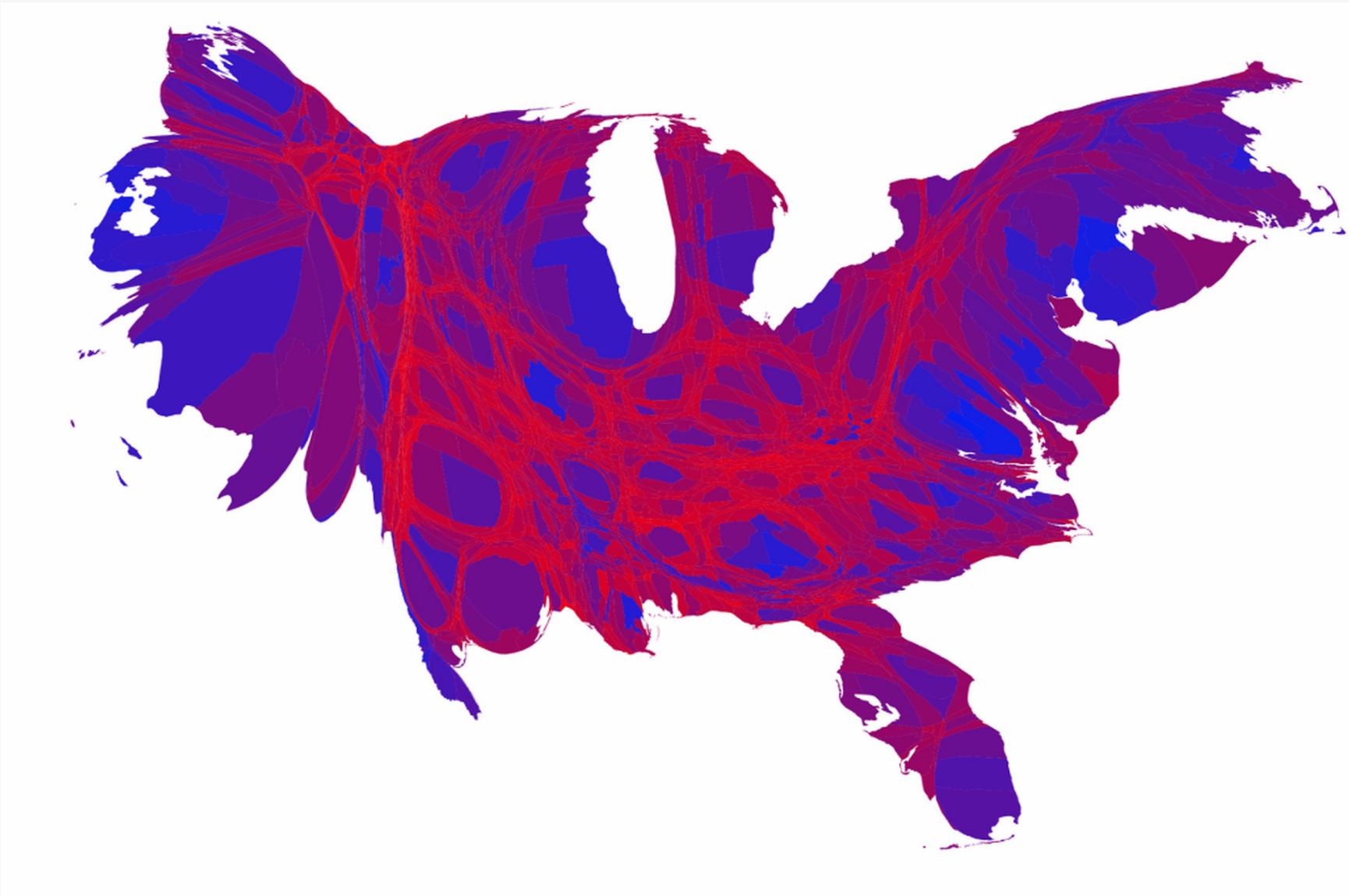
County level; winner only



County level vote share; diverging; binned into six categories

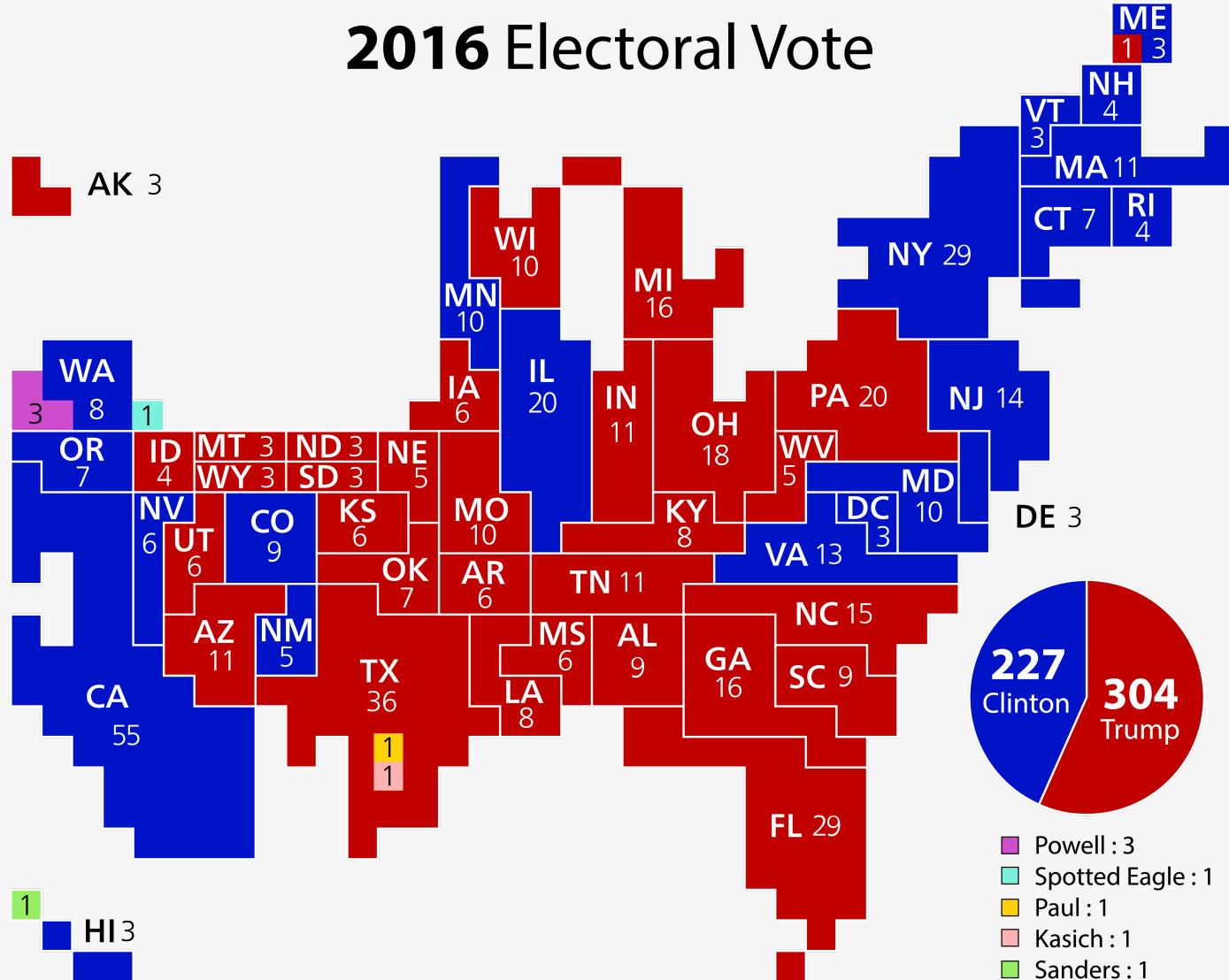


County level vote share; diverging continuous; purple midpoint



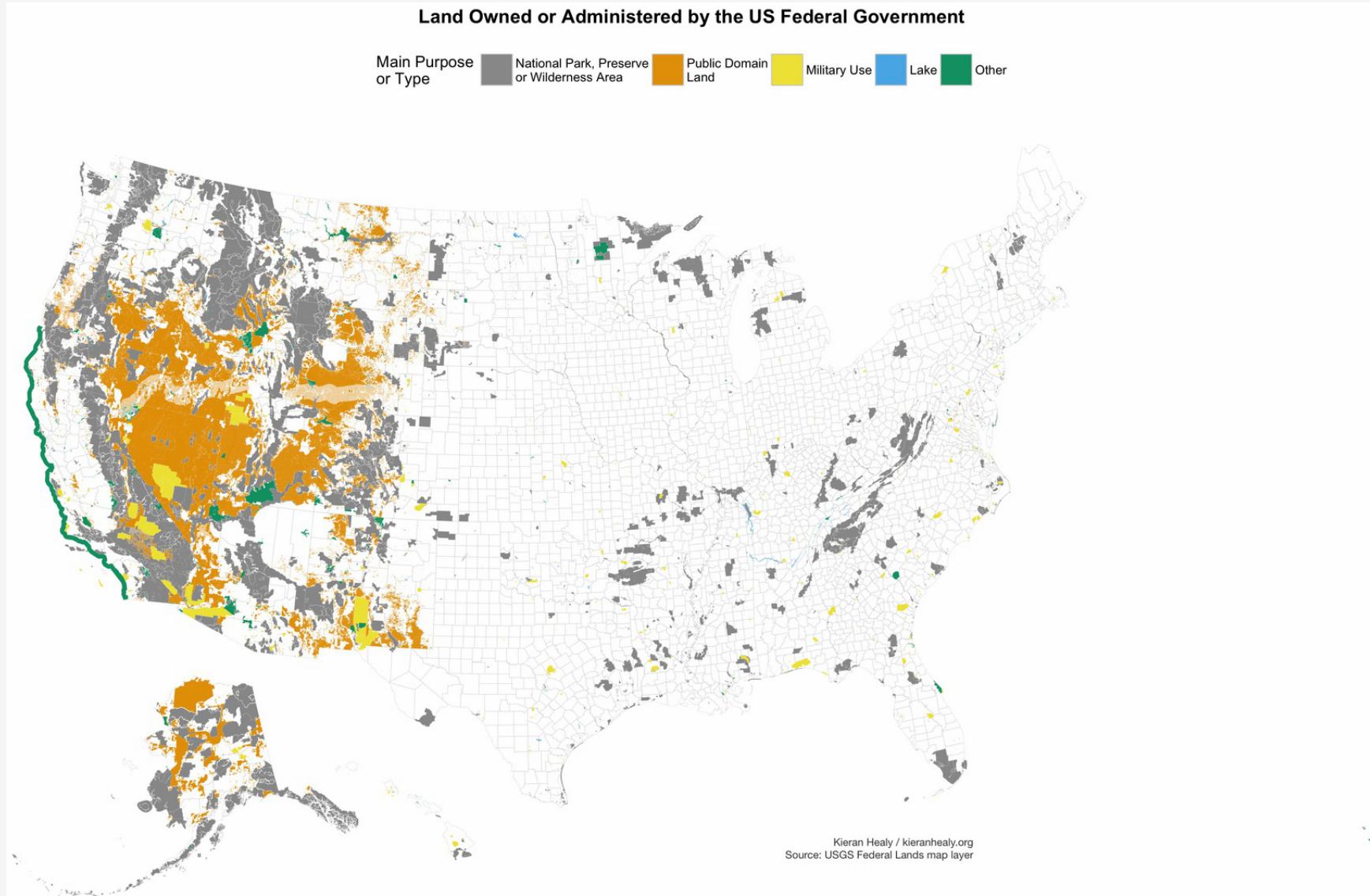
County level vote share; purple midpoint; county area deformed in proportion to population. By Mark Newman

2016 Electoral Vote

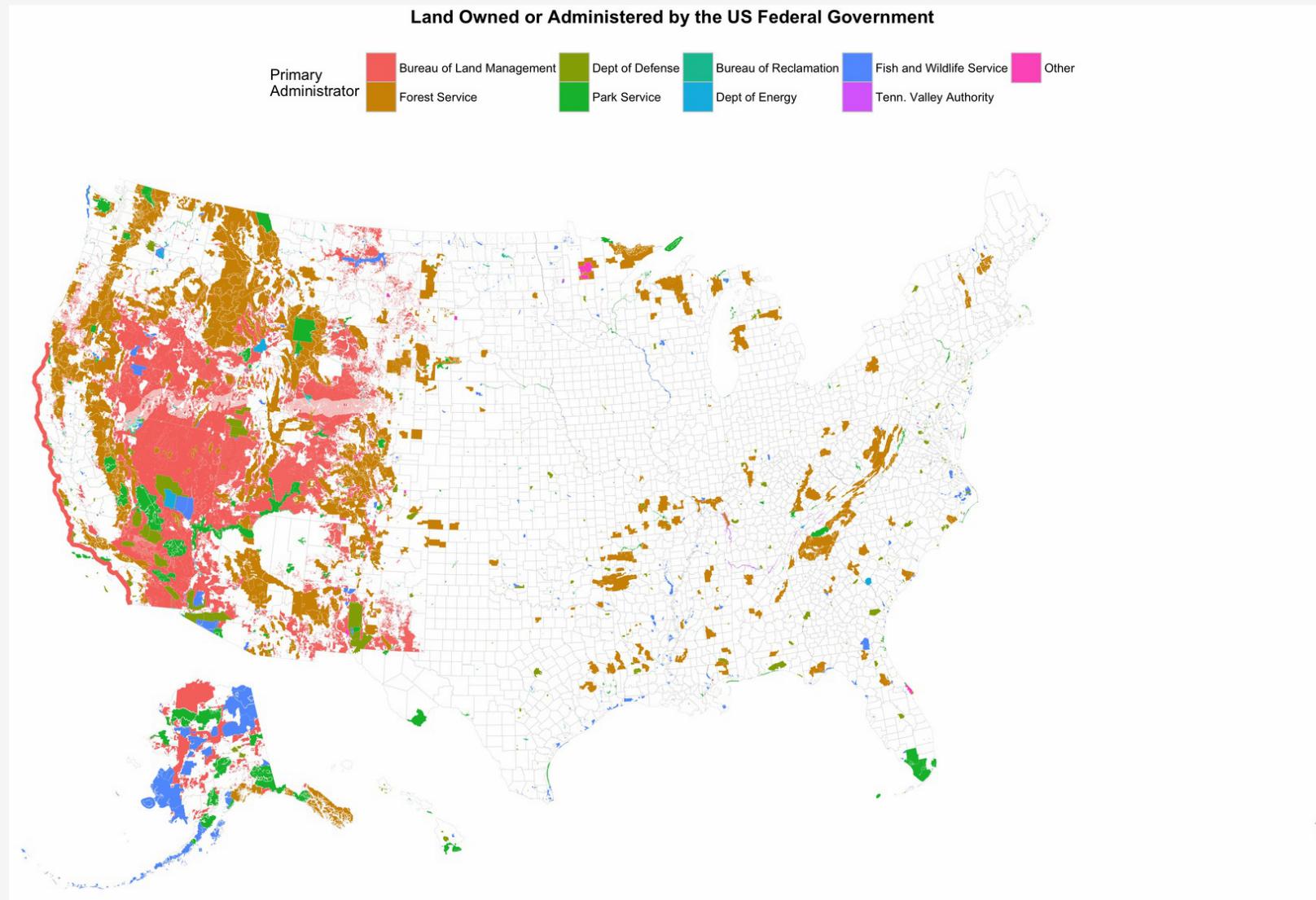


Electoral college cartogram

Problems showing Non-Spatial Data in choropleth maps

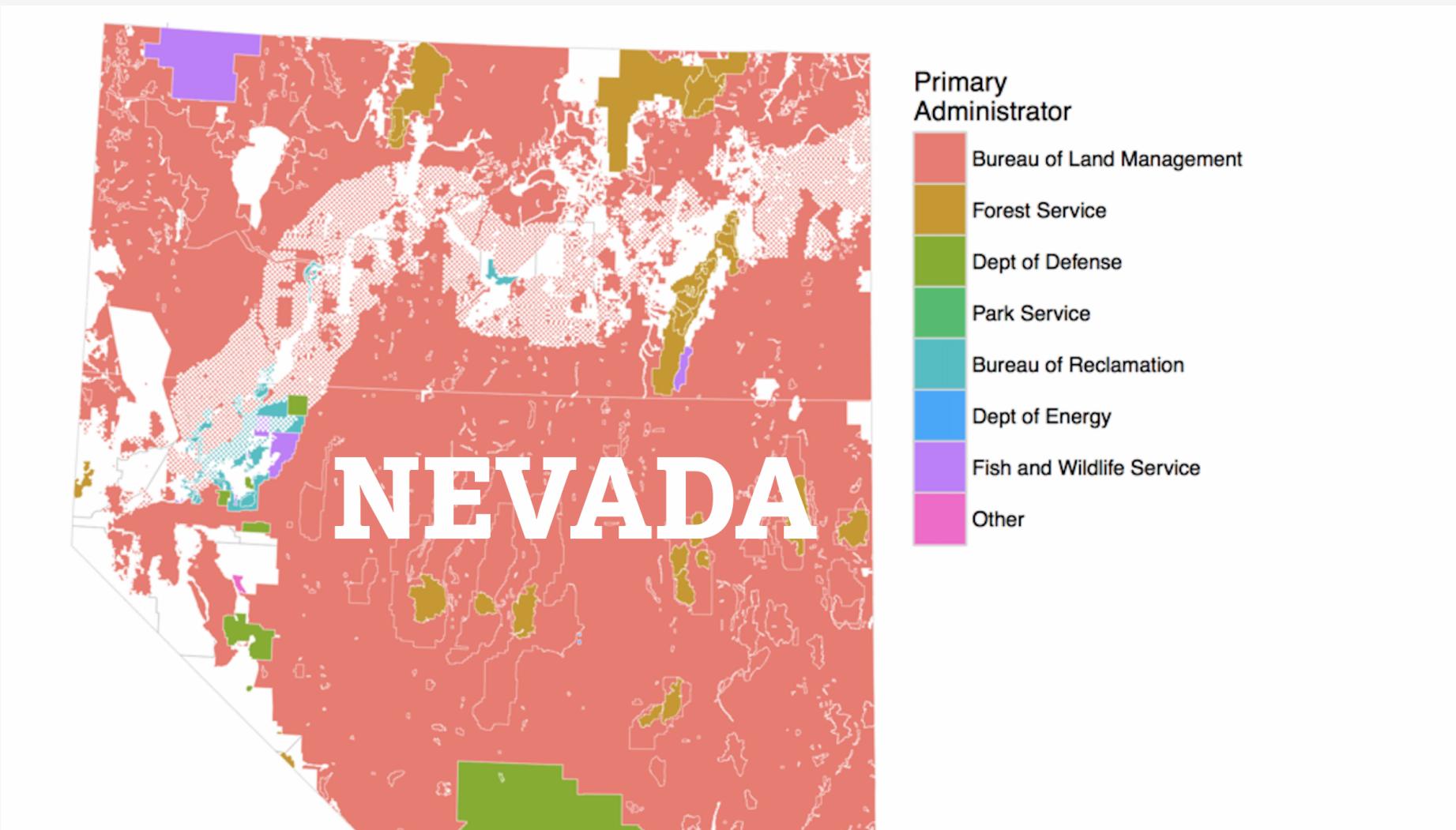


Pretty, Big, and Pretty Empty



Pretty, Big, and Pretty Empty

Aside: What the hell's that?

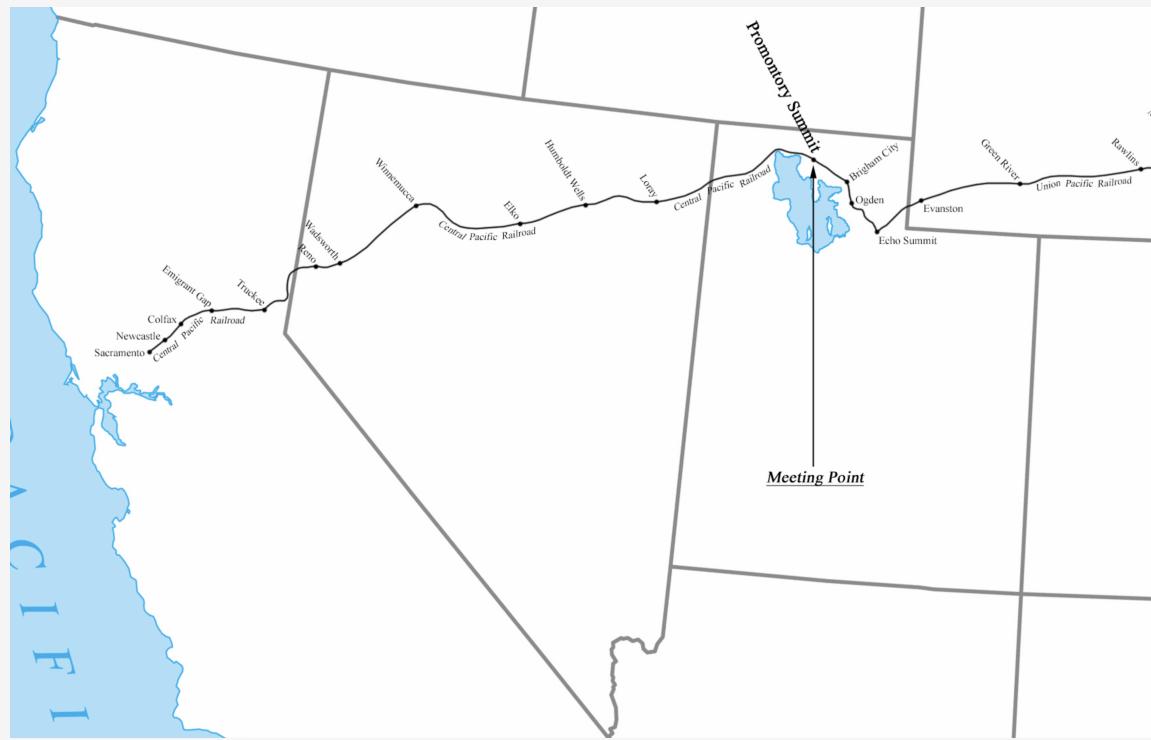


Zoom and Enhance



Surprisingly, not a coding error on my part.

It's the Transcontinental Railroad



Making its way through the **Great Basin**, America's largest **endorheic watershed**. The checkerboard is a deliberate assignation of property rights along the borders of the railway line.

Still with us, too



Not identical, as Interstate 80 was able to go through some parts the railroad had to go around. OK, now back to scheduled programming.

U.S. State-Level Election Data

Set up the data

```
## Hex color codes for Democratic Blue and Republican Red
party_colors <- c("#2E74C0", "#CB454A")

election |>
  select(state, total_vote, r_points, pct_trump, party, census)

## # A tibble: 51 × 6
##   state      total_vote  r_points  pct_trump party    census
##   <chr>        <dbl>     <dbl>      <dbl> <chr>    <chr>
## 1 Alabama      2123372    27.7      62.1  Republican South
## 2 Alaska       318608     14.7      51.3  Republican West
## 3 Arizona      2604657     3.5      48.1  Republican West
## 4 Arkansas     1130635     26.9      60.6  Republican South
## 5 California   14237893   -30.0      31.5  Democratic West
## 6 Colorado      2780247    -4.91     43.2  Democratic West
## 7 Connecticut   1644920    -13.6     40.9  Democratic Northeast
## 8 Delaware      443814    -11.4      41.7  Democratic South
## 9 District of Columbia 311268   -86.8      4.09 Democratic South
## 10 Florida      9502747     1.19      48.6  Republican South
## # ... with 41 more rows
```

Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")
```

Look before Mapping

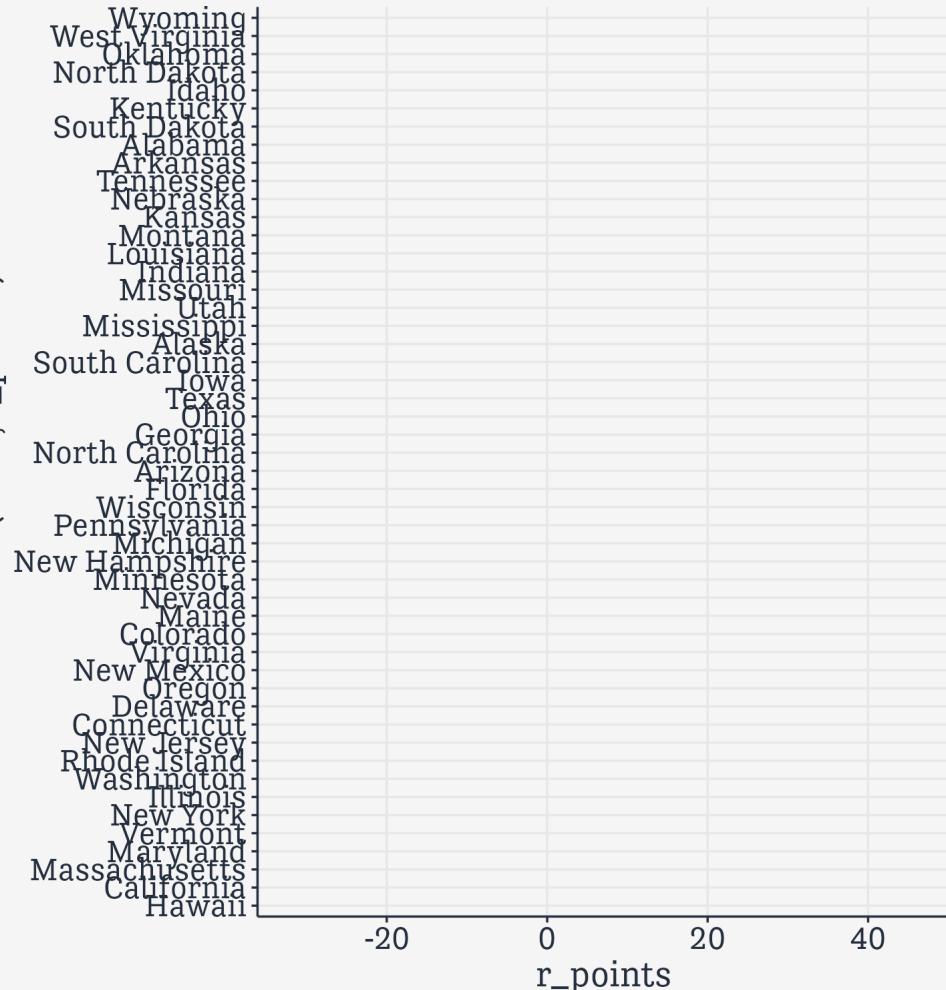
```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election  
## # A tibble: 51 × 22  
##   state    st     fips total_vote vote_margin winner party pct_marg  
##   <chr>    <chr> <dbl>      <dbl>       <dbl> <chr> <chr> <dbl>  
##   1 Alabama  AL      1  2123372     588708 Trump  Repu...  0.27  
##   2 Alaska   AK      2   318608      46933 Trump  Repu...  0.14  
##   3 Arizona  AZ      4   2604657     91234 Trump  Repu...  0.03  
##   4 Arkansas AR      5   1130635     304378 Trump  Repu...  0.26  
##   5 Californ... CA      6  14237893    4269978 Clint... Demo...  0.30  
##   6 Colorado CO      8   2780247     136386 Clint... Demo...  0.04  
##   7 Connecti... CT      9   1644920     224357 Clint... Demo...  0.13  
##   8 Delaware DE     10    443814      50476 Clint... Demo...  0.11  
##   9 District... DC     11    311268     270107 Clint... Demo...  0.86  
##  10 Florida  FL     12   9502747     112911 Trump  Repu...  0.01  
## # ... with 41 more rows, and 13 more variables: d_points <dbl>,  
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>, pct_other <dbl>,  
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>, other_vote <dbl>,  
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC")  
  
## # A tibble: 50 × 22  
##   state    st    fips total_vote vote_margin winner party pct_marg  
##   <chr>    <chr> <dbl>     <dbl>      <dbl> <chr> <chr> <dbl>  
##   1 Alabama AL      1  2123372    588708 Trump  Repu...  0.27  
##   2 Alaska AK      2   318608    46933  Trump  Repu...  0.14  
##   3 Arizona AZ      4  2604657    91234  Trump  Repu...  0.03  
##   4 Arkansas AR      5  1130635   304378 Trump  Repu...  0.26  
##   5 California CA      6  14237893  4269978 Clint... Demo...  0.30  
##   6 Colorado CO      8   2780247  136386 Clint... Demo...  0.04  
##   7 Connecticut CT      9  1644920  224357 Clint... Demo...  0.13  
##   8 Delaware DE     10   443814   50476  Clint... Demo...  0.11  
##   9 Florida FL     12   9502747  112911 Trump  Repu...  0.01  
##  10 Georgia GA     13   4141447  211141 Trump  Repu...  0.05  
## # ... with 40 more rows, and 13 more variables: d_points <dbl>,  
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>, pct_other <dbl>,  
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>, other_vote <dbl>,  
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

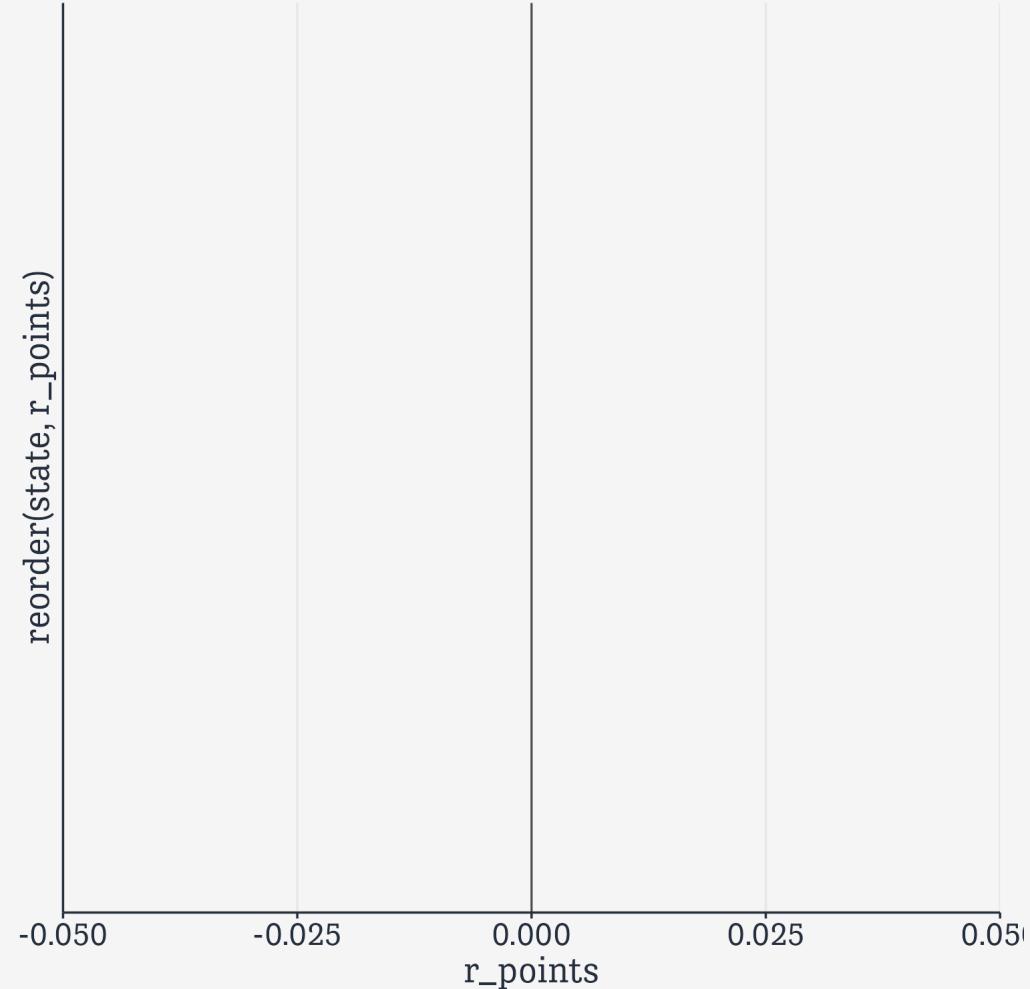
Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party))
```



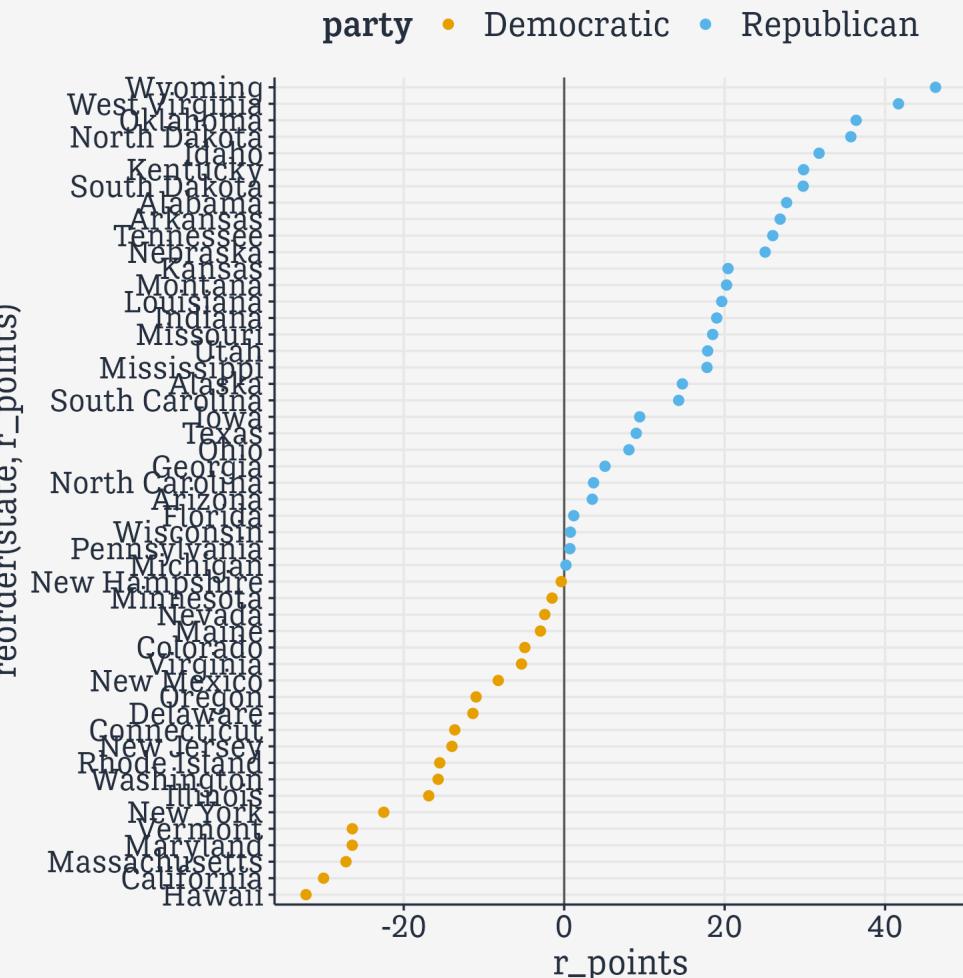
Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
    geom_vline(xintercept = 0,  
               color = "gray30")
```



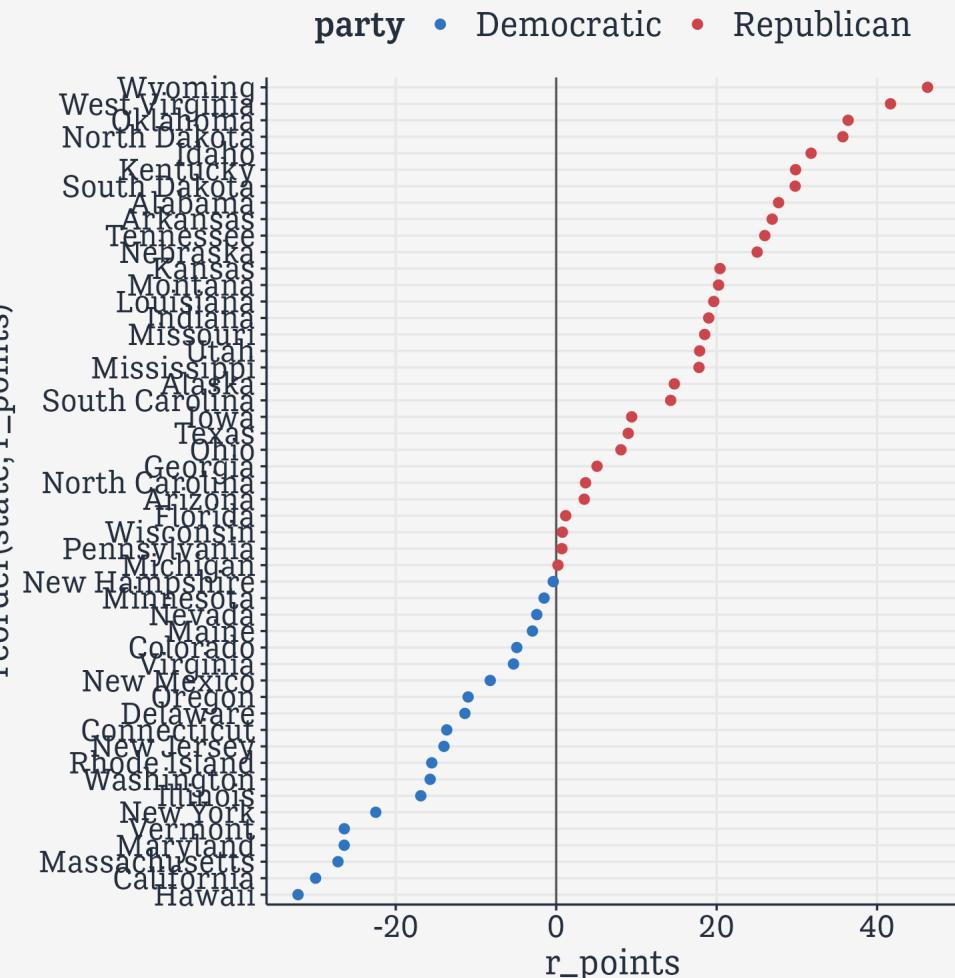
Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
    geom_vline(xintercept = 0,  
               color = "gray30") +  
    geom_point(size = 2)
```



Look before Mapping

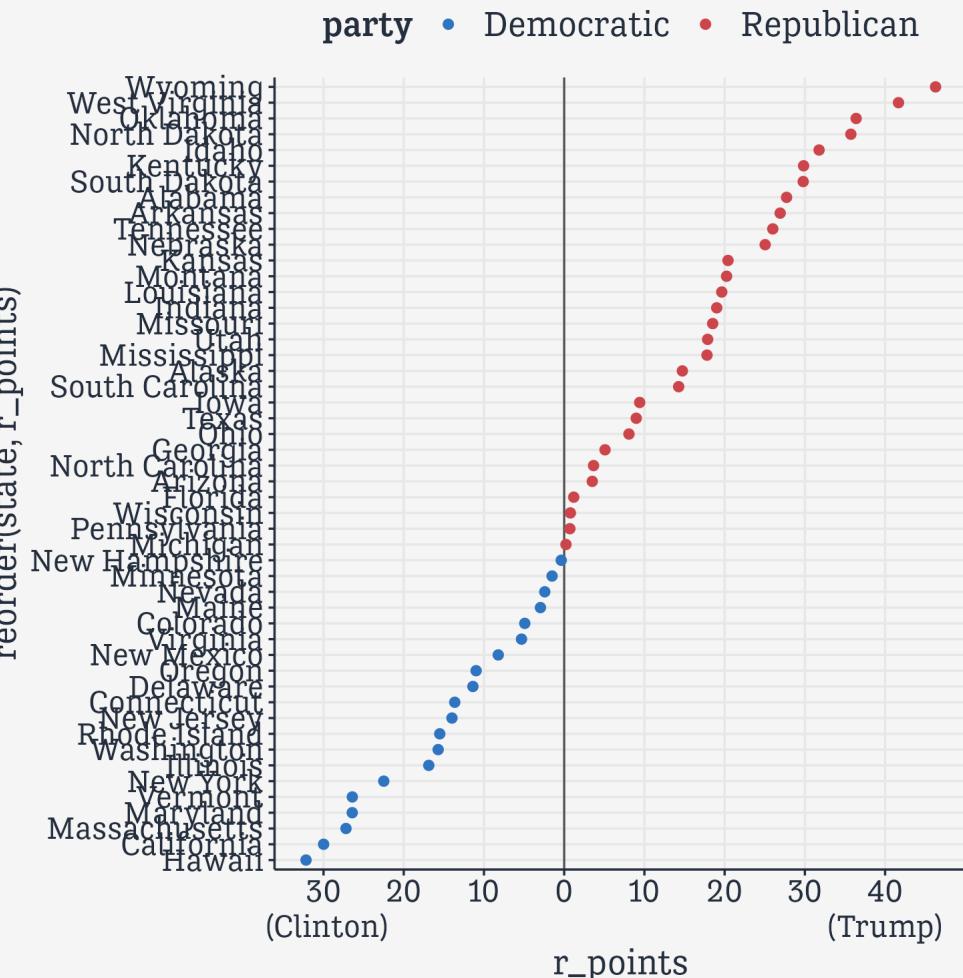
```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
    geom_vline(xintercept = 0,  
               color = "gray30") +  
    geom_point(size = 2) +  
    scale_color_manual(values = party_colors)
```



Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Red
party_colors <- c("#2E74C0", "#CB454A")

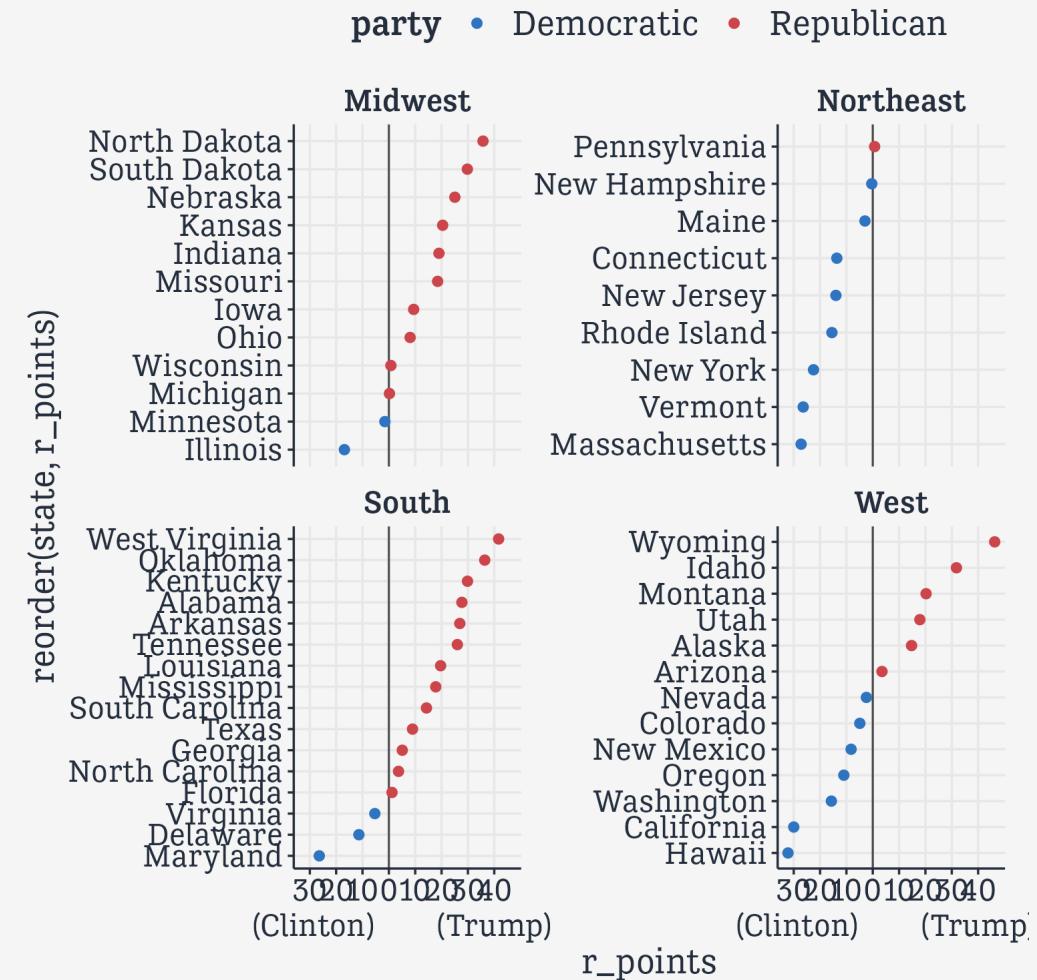
election |>
  filter(st %in% "DC") |>
  ggplot(mapping = aes(x = r_points,
                        y = reorder(state, r_points),
                        color = party)) +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2) +
  scale_color_manual(values = party_colors) +
  scale_x_continuous(breaks = c(-30, -20, -10, 0,
                                10, 20, 30, 40),
                     labels = c("30\n(Clinton)",
                               "20", "10", "0",
                               "10", "20", "30",
                               "40\n(Trump)"))
```



Look before Mapping

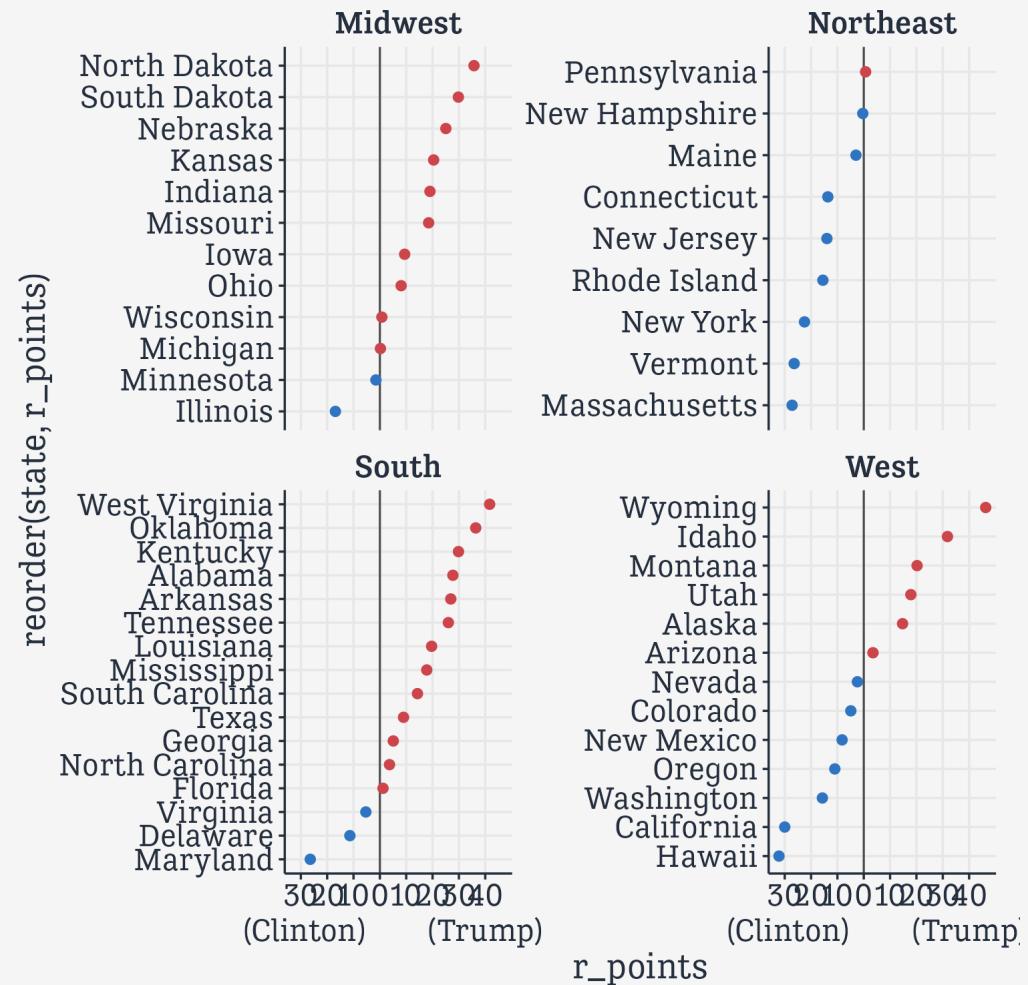
```
## Hex color codes for Democratic Blue and Republican Red
party_colors <- c("#2E74C0", "#CB454A")

election |>
  filter(st %in% "DC") |>
  ggplot(mapping = aes(x = r_points,
                        y = reorder(state, r_points),
                        color = party)) +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2) +
  scale_color_manual(values = party_colors) +
  scale_x_continuous(breaks = c(-30, -20, -10, 0,
                                10, 20, 30, 40),
                     labels = c("30\n(Clinton)",
                               "20", "10", "0",
                               "10", "20", "30",
                               "40\n(Trump)")) +
  facet_wrap(~ census, ncol=2,
             scales="free_y")
```



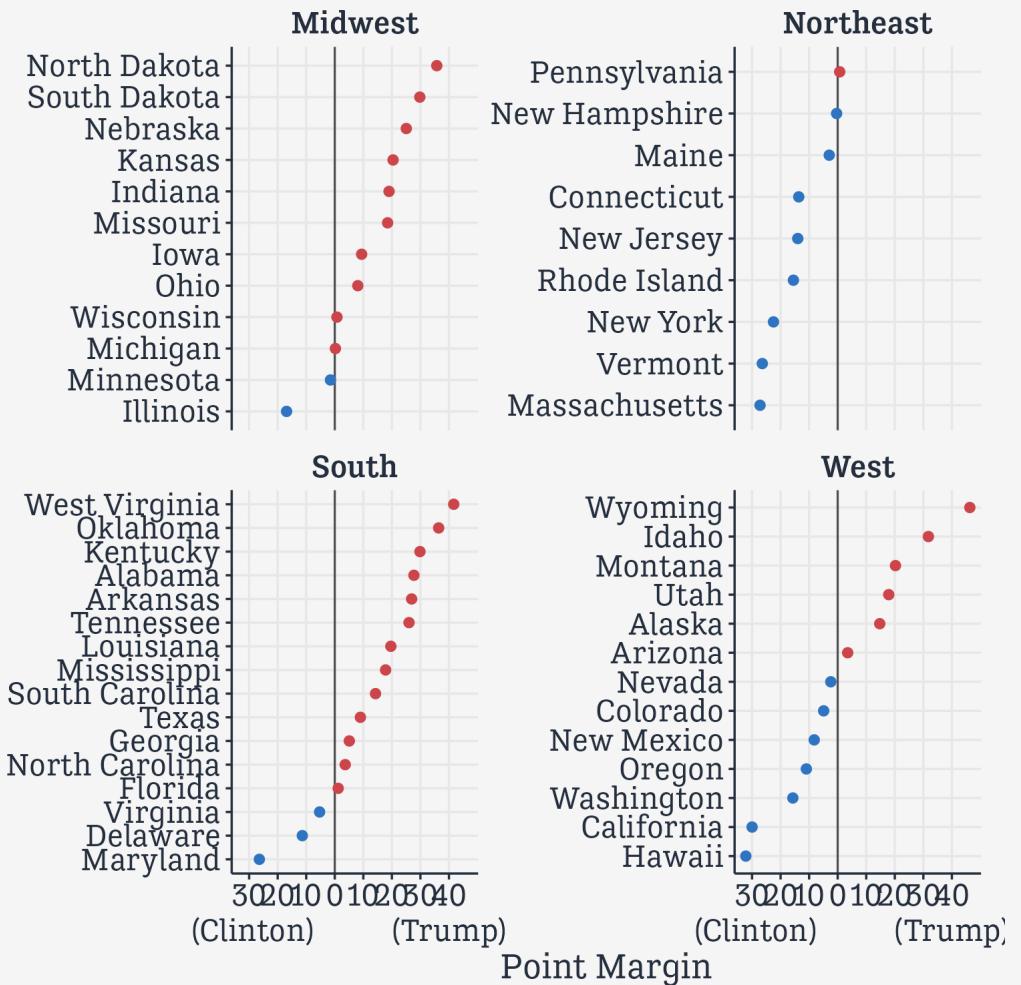
Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
    geom_vline(xintercept = 0,  
               color = "gray30") +  
    geom_point(size = 2) +  
    scale_color_manual(values = party_colors) +  
    scale_x_continuous(breaks = c(-30, -20, -10, 0,  
                                 10, 20, 30, 40),  
                       labels = c("30\n(Clinton)",  
                                 "20", "10", "0",  
                                 "10", "20", "30",  
                                 "40\n(Trump)")) +  
    facet_wrap(~ census, ncol=2,  
              scales="free_y") +  
    guides(color = "none")
```



Look before Mapping

```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
    geom_vline(xintercept = 0,  
               color = "gray30") +  
    geom_point(size = 2) +  
    scale_color_manual(values = party_colors) +  
    scale_x_continuous(breaks = c(-30, -20, -10, 0,  
                                 10, 20, 30, 40),  
                       labels = c("30\n(Clinton)",  
                                 "20", "10", "0",  
                                 "10", "20", "30",  
                                 "40\n(Trump)")) +  
    facet_wrap(~ census, ncol=2,  
              scales="free_y") +  
    guides(color = "none") +  
    labs(x = "Point Margin", y = NULL)
```



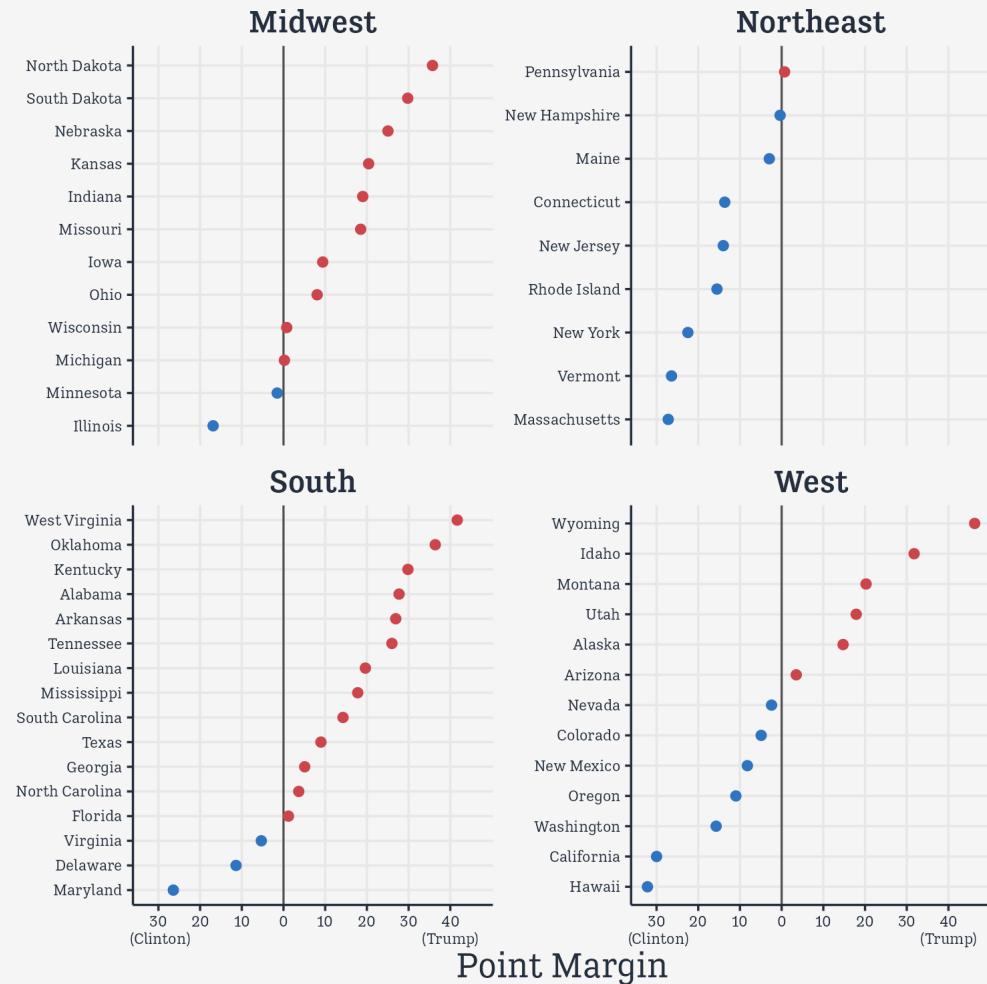
Look before Mapping

```

## Hex color codes for Democratic Blue and Republican Re
party_colors <- c("#2E74C0", "#CB454A")

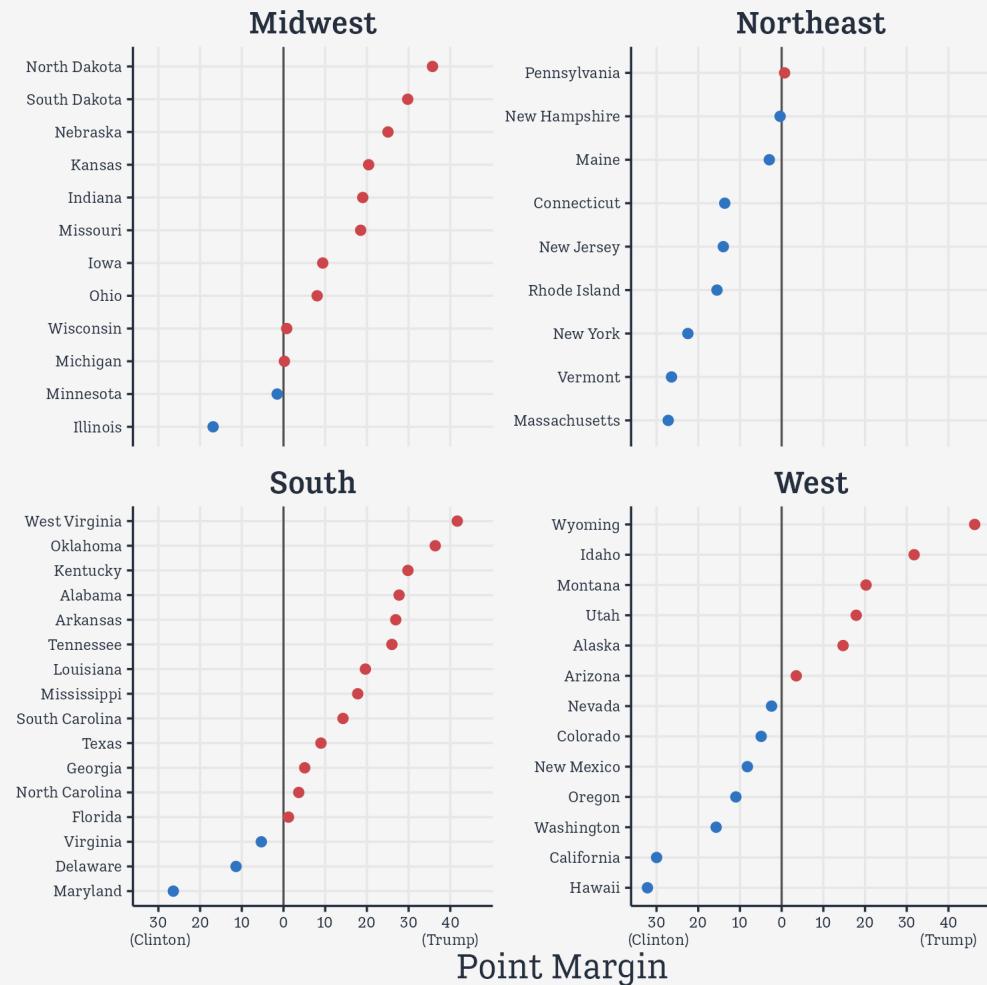
election |>
  filter(st %in% "DC") |>
  ggplot(mapping = aes(x = r_points,
                        y = reorder(state, r_points),
                        color = party)) +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2) +
  scale_color_manual(values = party_colors) +
  scale_x_continuous(breaks = c(-30, -20, -10, 0,
                                10, 20, 30, 40),
                     labels = c("30\n(Clinton)",
                               "20", "10", "0",
                               "10", "20", "30",
                               "40\n(Trump)")) +
  facet_wrap(~ census, ncol=2,
             scales="free_y") +
  guides(color = "none") +
  labs(x = "Point Margin", y = NULL) +
  theme(axis.text=element_text(size=8))

```

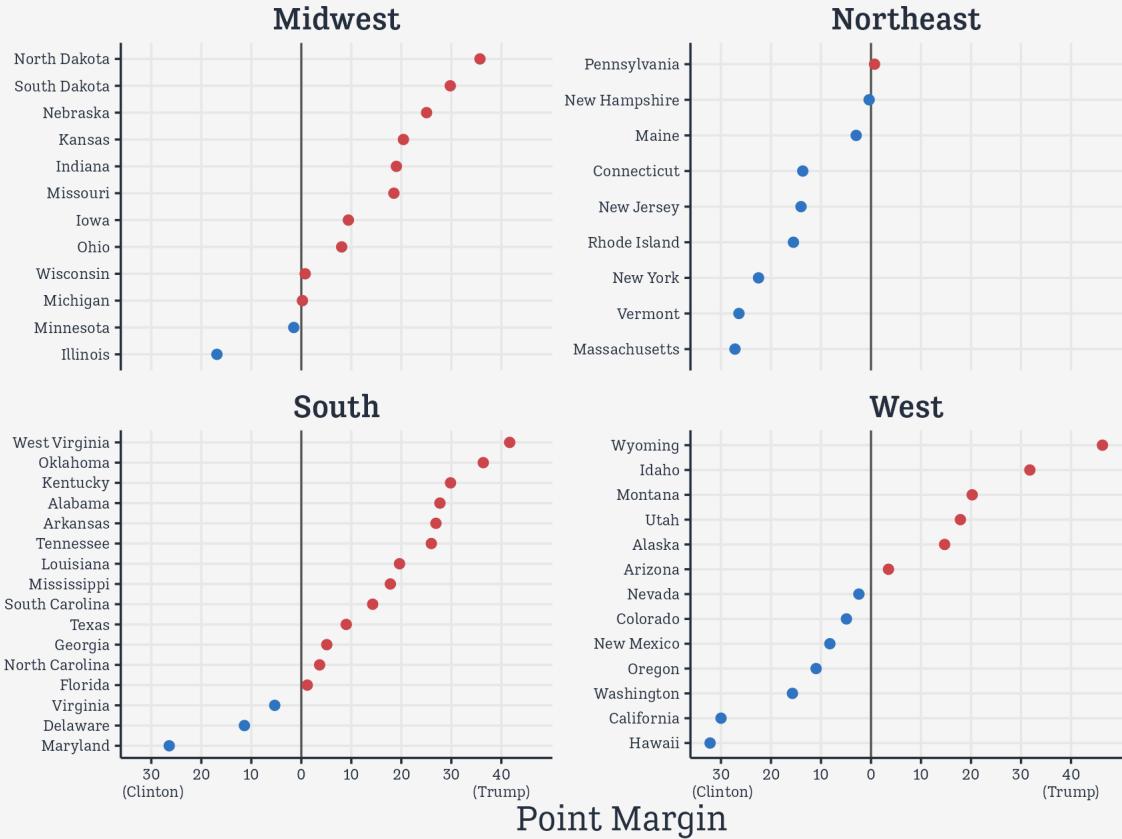


Look before Mapping

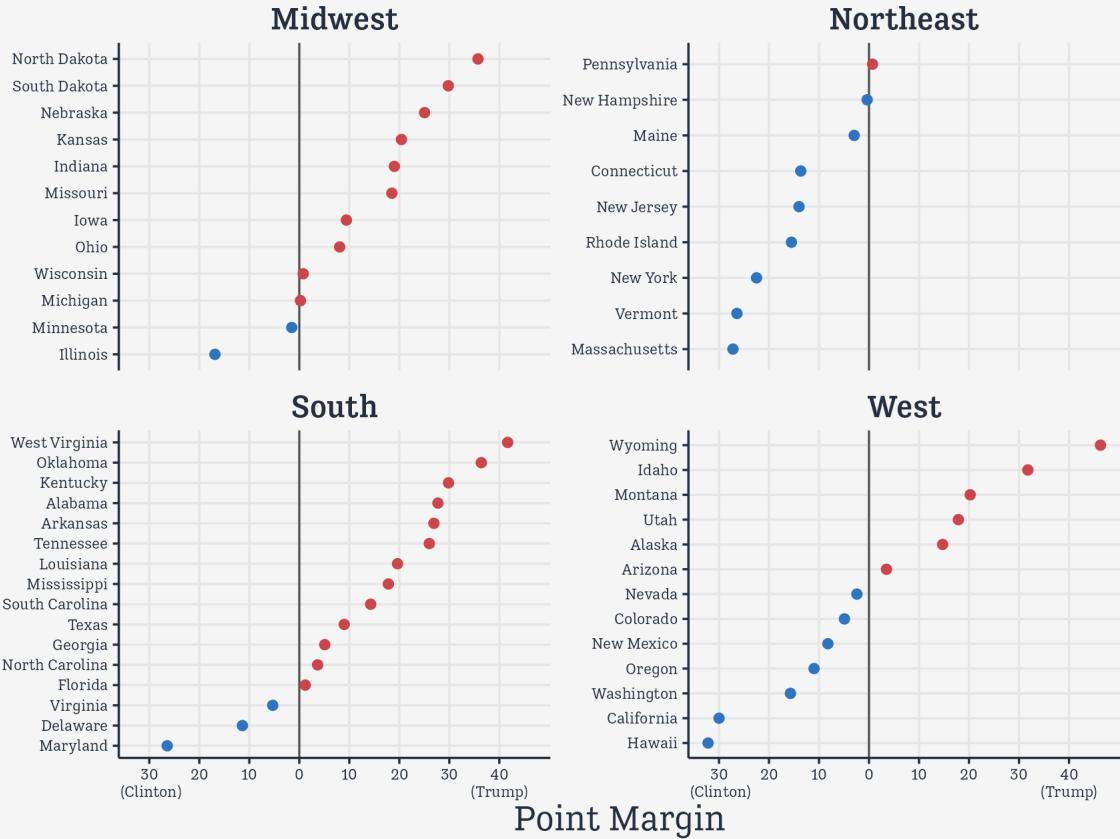
```
## Hex color codes for Democratic Blue and Republican Re  
party_colors <- c("#2E74C0", "#CB454A")  
  
election |>  
  filter(st %in% "DC") |>  
  ggplot(mapping = aes(x = r_points,  
                        y = reorder(state, r_points),  
                        color = party)) +  
  geom_vline(xintercept = 0,  
             color = "gray30") +  
  geom_point(size = 2) +  
  scale_color_manual(values = party_colors) +  
  scale_x_continuous(breaks = c(-30, -20, -10, 0,  
                                10, 20, 30, 40),  
                     labels = c("30\n(Clinton)",  
                               "20", "10", "0",  
                               "10", "20", "30",  
                               "40\n(Trump)")) +  
  facet_wrap(~ census, ncol=2,  
            scales="free_y") +  
  guides(color = "none") +  
  labs(x = "Point Margin", y = NULL) +  
  theme(axis.text=element_text(size=8))
```



With a bit more room



With a bit more room



See how the panels are unbalanced, even with scales = "free_y"?

This happens because we have unequal number of states per region.

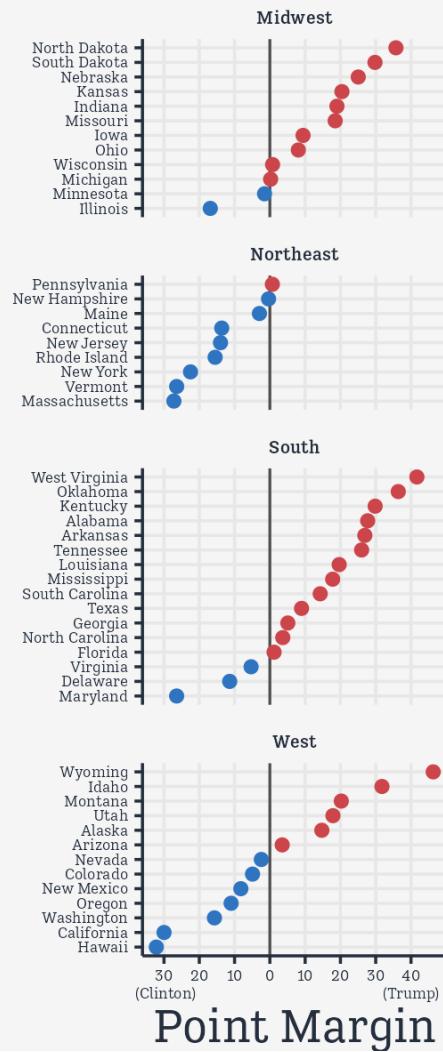
We can use `facet_col()` from `ggforce`

```
p_out <- election |>
  filter(st %nin% "DC") |>
  ggplot(mapping = aes(x = r_points,
                        y = reorder(state, r_points),
                        color = party)) +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2) +
  scale_color_manual(values = party_colors) +
  scale_x_continuous(breaks = c(-30, -20, -10, 0,
                                10, 20, 30, 40),
                     labels = c("30\n(Clinton)",
                               "20", "10", "0",
                               "10", "20", "30",
                               "40\n(Trump)")) +
  facet_col(~ census,
            scales="free_y",
            space = "free") +
  guides(color = "none") +
  labs(x = "Point Margin", y = NULL) +
  theme(axis.text=element_text(size=6),
        strip.text = element_text(size = rel(0.6)))
```

p_out

We can use `facet_col()` from `ggforce`

```
p_out <- election |>
  filter(st %in% "DC") |>
  ggplot(mapping = aes(x = r_points,
                        y = reorder(state, r_points),
                        color = party)) +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2) +
  scale_color_manual(values = party_colors) +
  scale_x_continuous(breaks = c(-30, -20, -10, 0,
                                10, 20, 30, 40),
                     labels = c("30\n(Clinton)",
                               "20", "10", "0",
                               "10", "20", "30",
                               "40\n(Trump)")) +
  facet_col(~ census,
            scales="free_y",
            space = "free") +
  guides(color = "none") +
  labs(x = "Point Margin", y = NULL) +
  theme(axis.text=element_text(size=6),
        strip.text = element_text(size = rel(0.6)))
p_out
```



Basic Maps

Maps as polygons

Take a look at this data

```
## This is from the map library
# library(maps)

us_states <- map_data("state")

dim(us_states)

## [1] 15537      6

## Making it a tibble prevents crashes
## in the slide rendering later on
us_states <- as_tibble(us_states)

us_states

## # A tibble: 15,537 × 6
##       long     lat group order region   subregion
##       <dbl> <dbl> <dbl> <int> <chr>    <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

What is this, at root?

us_states

```
## # A tibble: 15,537 × 6
##   long     lat group order region  subregion
##   <dbl> <dbl> <dbl> <int> <chr>    <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

It's a series of rows defining x and y coordinates on a plane.

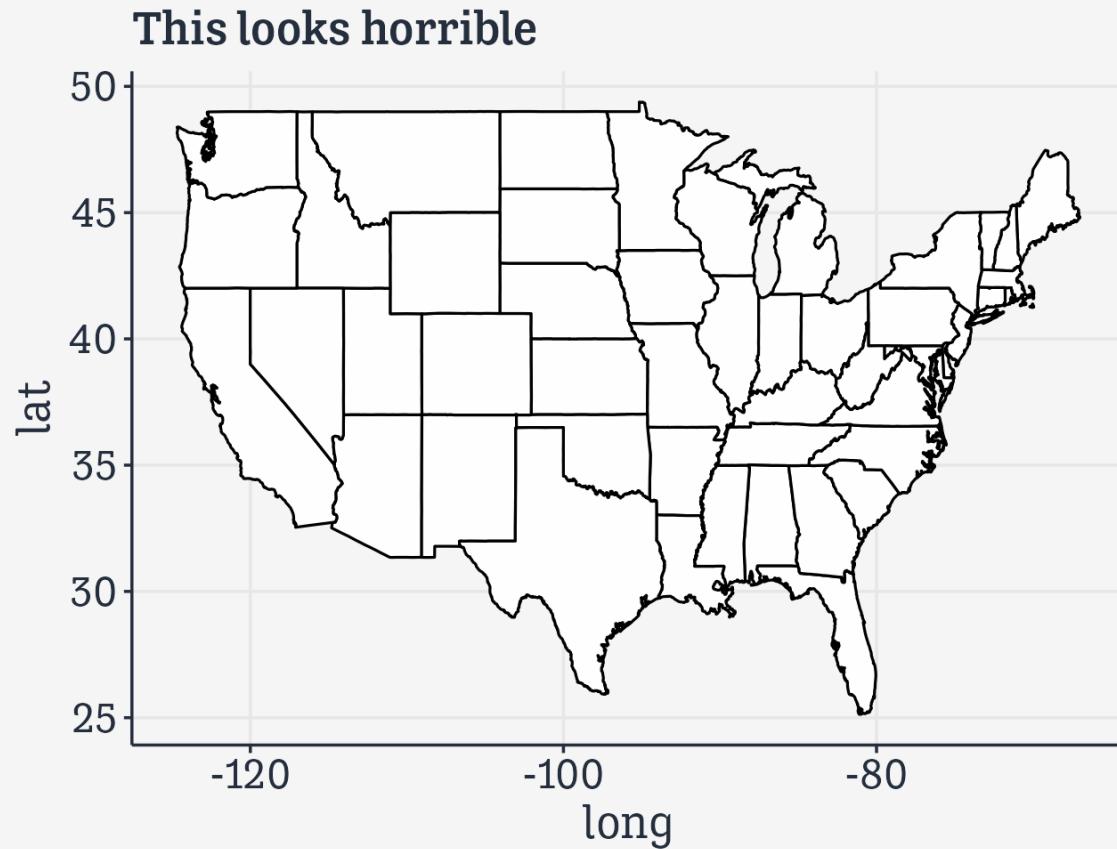
If we join those points up as lines while respecting their group (i.e. so ggplot knows when to "lift the pen", as with the gapminder line plot), we will get an outline map of states in the U.S.

Like this, with geom_polygon()

```
us_states |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        group = group)) +  
  geom_polygon(fill = "white",  
               color = "black") +  
  labs(title = "This looks horrible")
```

Like this, with `geom_polygon()`

```
us_states |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        group = group)) +  
  geom_polygon(fill = "white",  
               color = "black") +  
  labs(title = "This looks horrible")
```

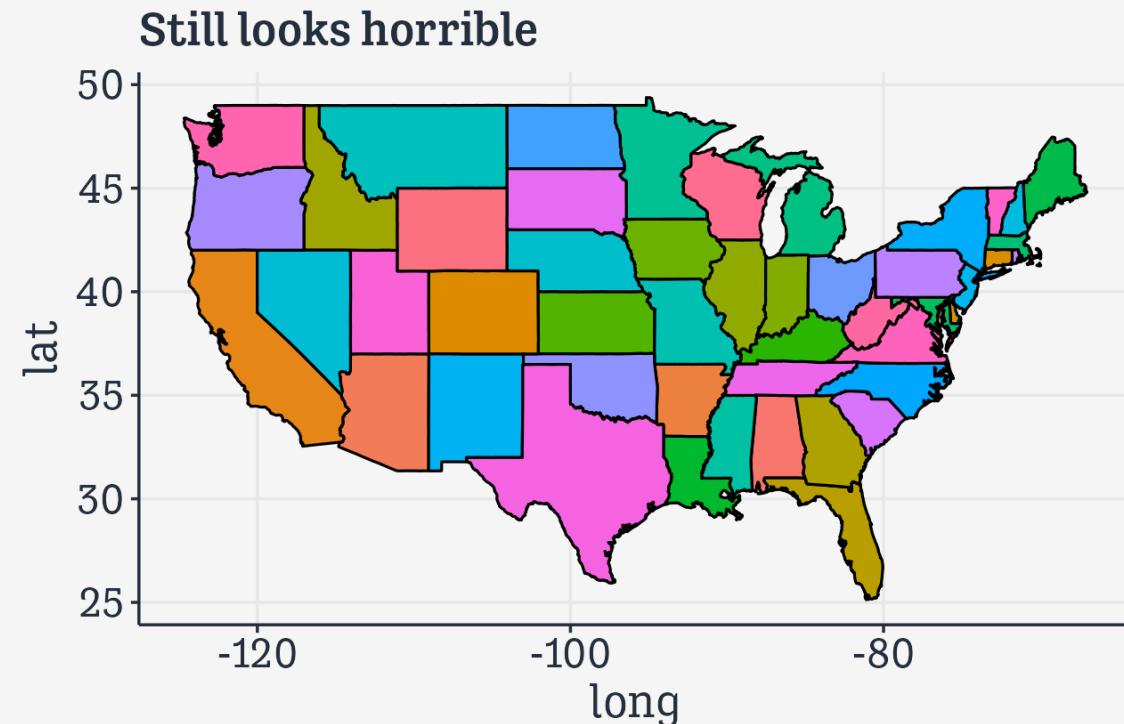


We can represent a fill, too, like any geom

```
us_states |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = region,  
                        group = group)) +  
  geom_polygon(color = "black") +  
  guides(fill = "none") +  
  labs(title = "Still looks horrible",  
       caption = "Set fill = none  
                  to stop ggplot from  
                  producing a key  
                  with 50 entries")
```

We can represent a fill, too, like any geom

```
us_states |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = region,  
                        group = group)) +  
  geom_polygon(color = "black") +  
  guides(fill = "none") +  
  labs(title = "Still looks horrible",  
       caption = "Set fill = none  
                  to stop ggplot from  
                  producing a key  
                  with 50 entries")
```



Set fill = none
to stop ggplot from
producing a key
with 50 entries

We need to do two things

1: Fix the map projection

2: Add some data to fill with.

For now, we'll do it the direct way

To make explicit what's happening, and to emphasize how *it's all just points and lines made from tables* we'll first do it at the level of the `ggplot` grammar with a geom that just draws shapes, `geom_polygon()`. After that, we'll introduce a new package, `sf` and a new geom, `geom_sf()` that will handle this for us, and more.

Fix the projection

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

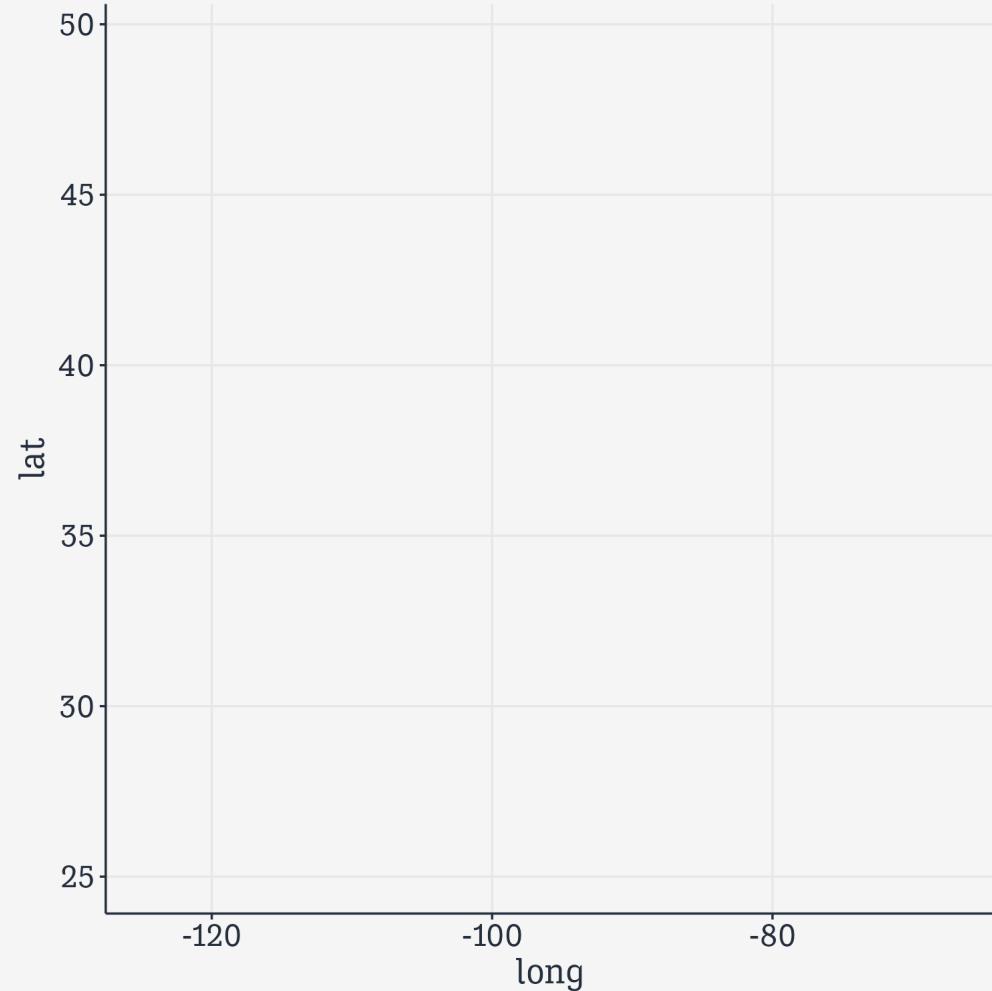
Fix the projection

```
us_states <- as_tibble(map_data("state"))
us_states
## # A tibble: 15,537 × 6
##       long     lat group order region subregion
##       <dbl>   <dbl> <dbl> <int> <chr>   <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

Fix the projection

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

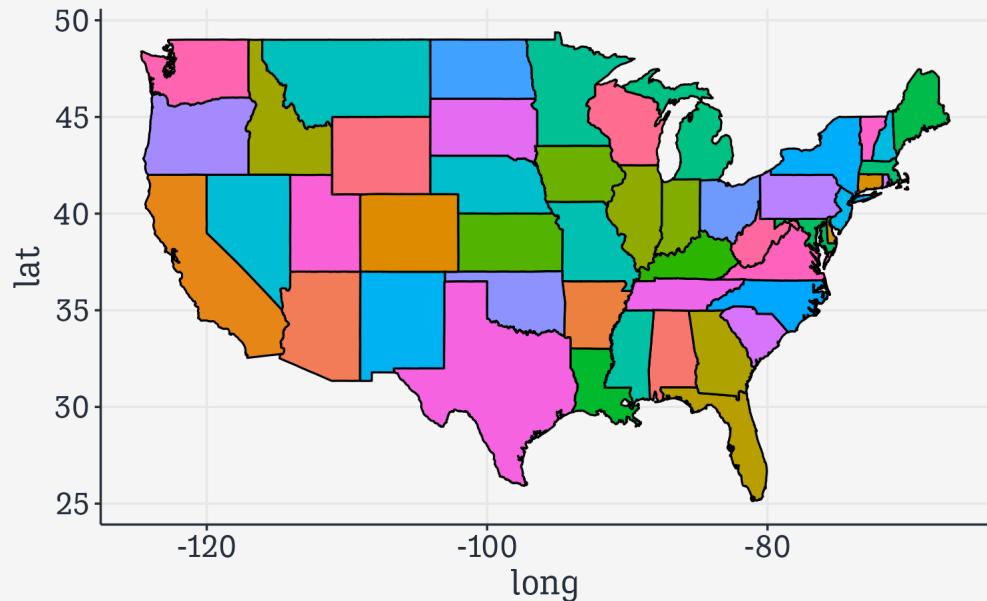
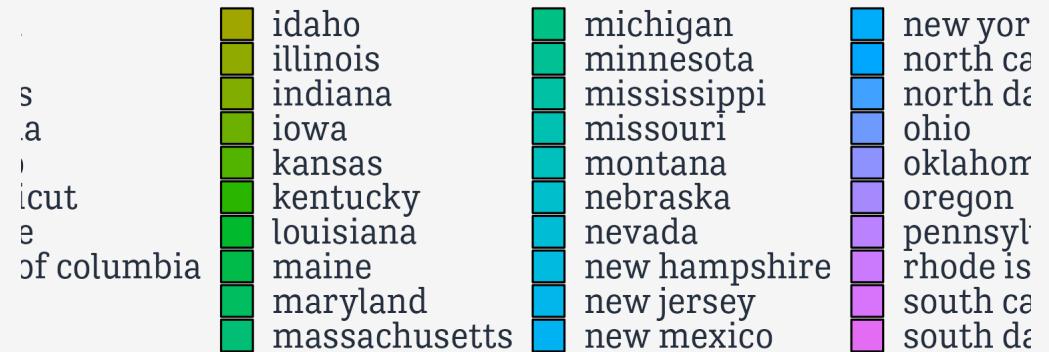
us_states |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = region,
                        group = group))
```



Fix the projection

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

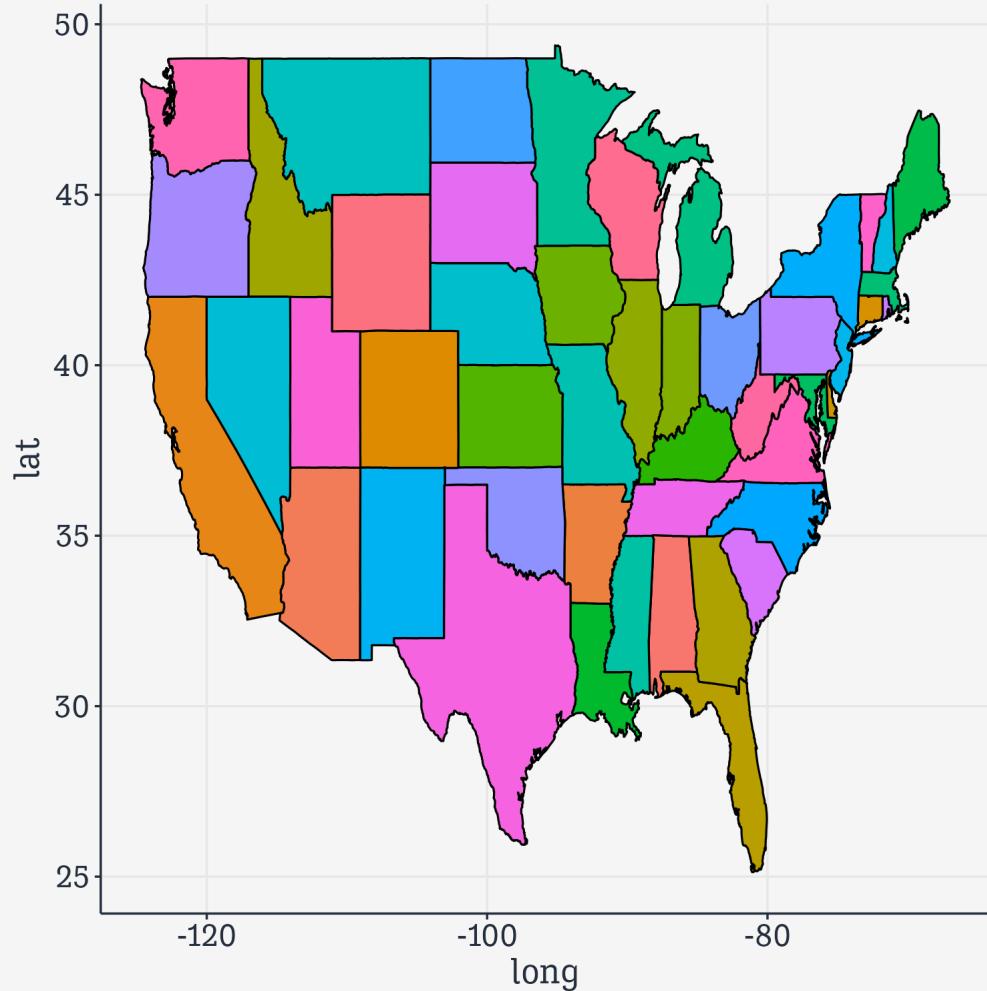
us_states |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = region,
                        group = group)) +
  geom_polygon(color = "black")
```



Fix the projection

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

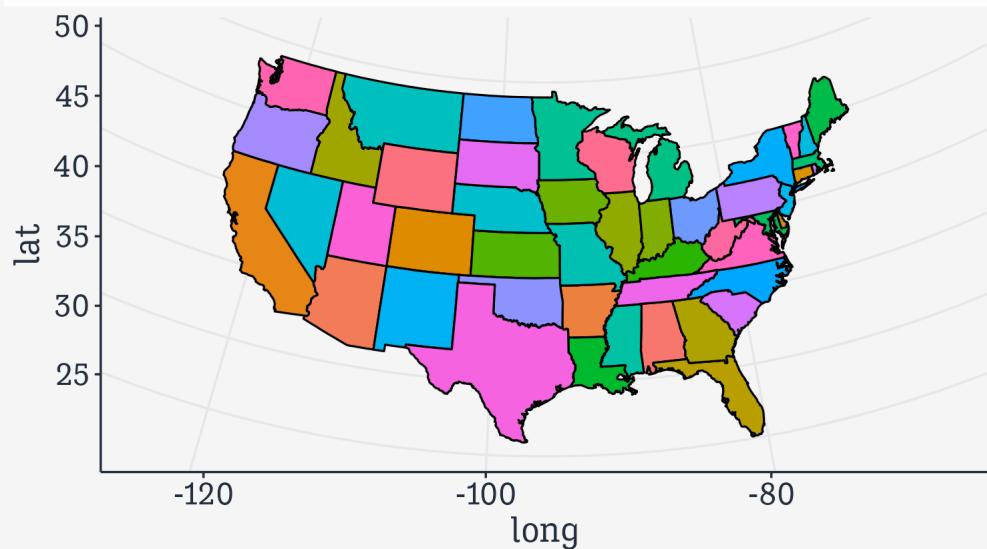
us_states |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = region,
                        group = group)) +
  geom_polygon(color = "black") +
  guides(fill = "none")
```



Fix the projection

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

us_states |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = region,
                        group = group)) +
  geom_polygon(color = "black") +
  guides(fill = "none") +
  coord_map(projection = "albers",
            lat0 = 39,
            lat1 = 45)
```

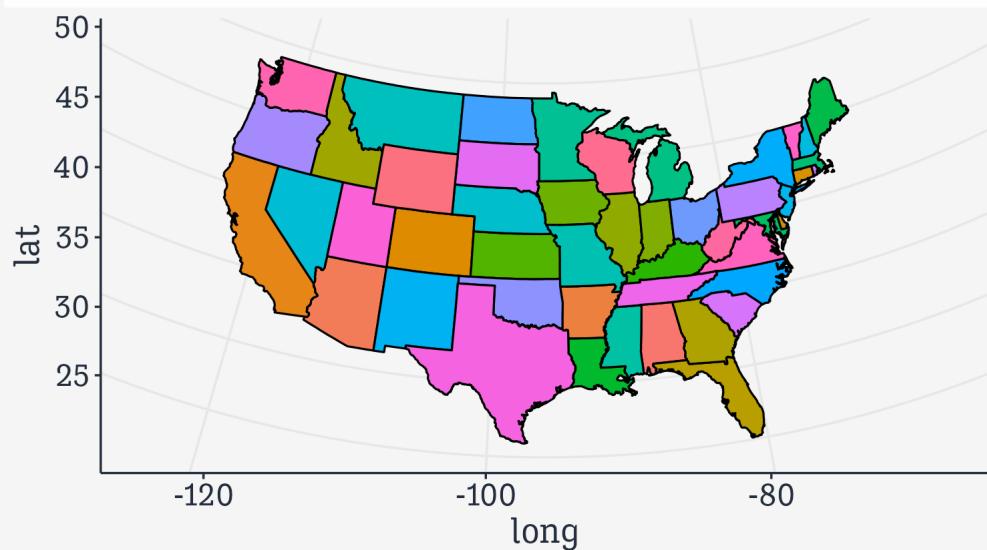


Fix the projection

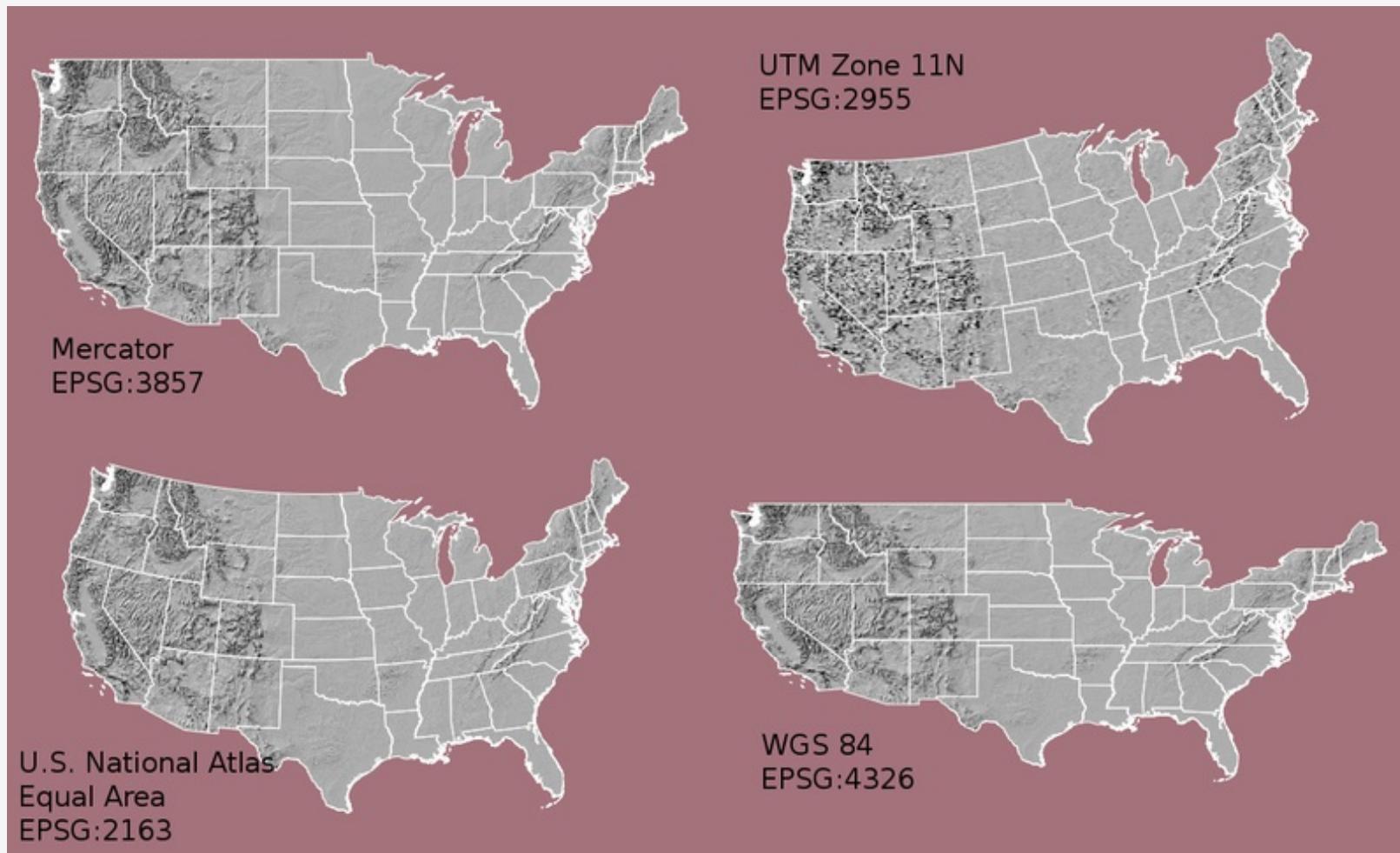
```
us_states <- as_tibble(map_data("state"))

us_states |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = region,
                        group = group)) +
  geom_polygon(color = "black") +
  guides(fill = "none") +
  coord_map(projection = "albers",
            lat0 = 39,
            lat1 = 45)

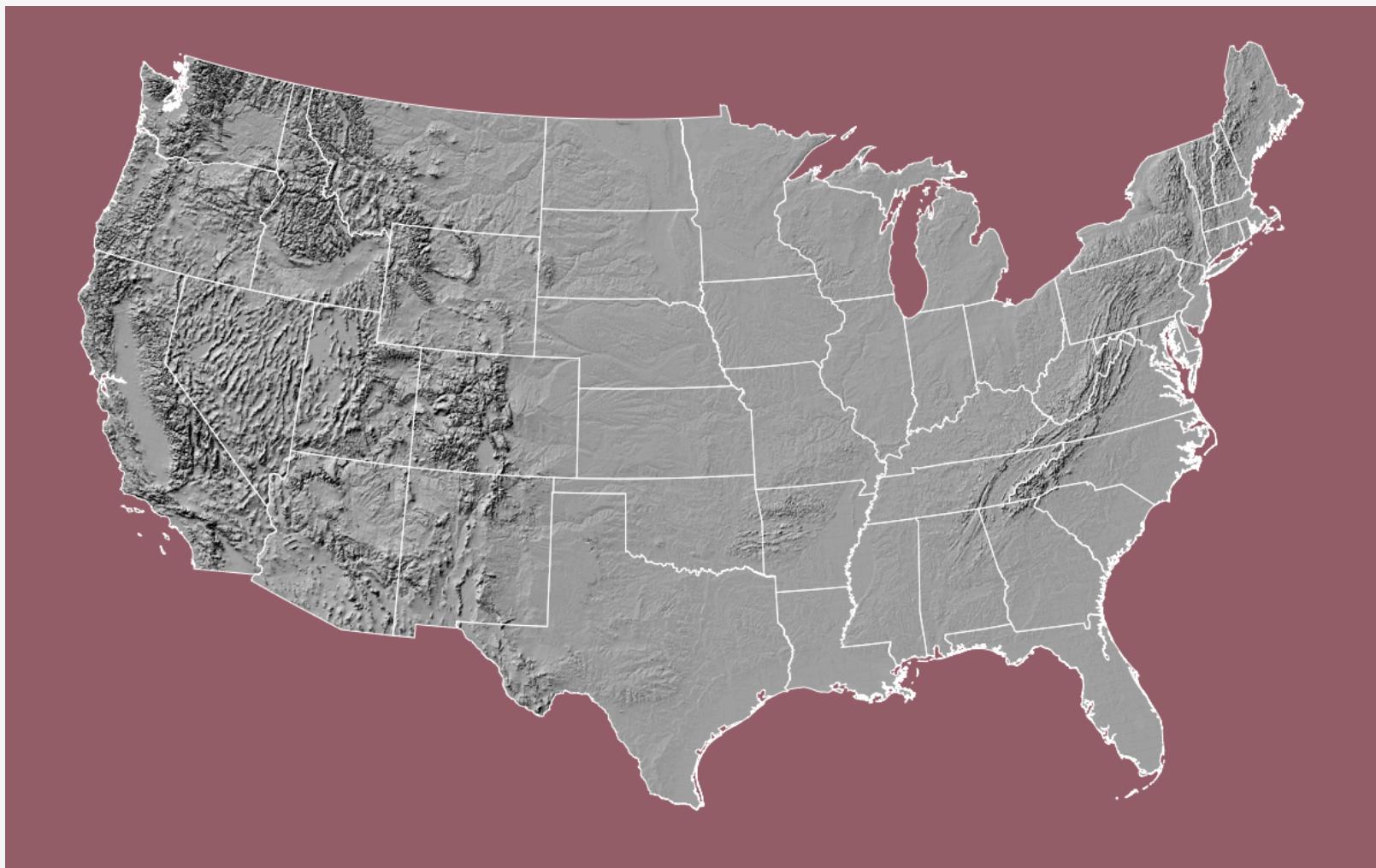
## A coordinate transformation!
```

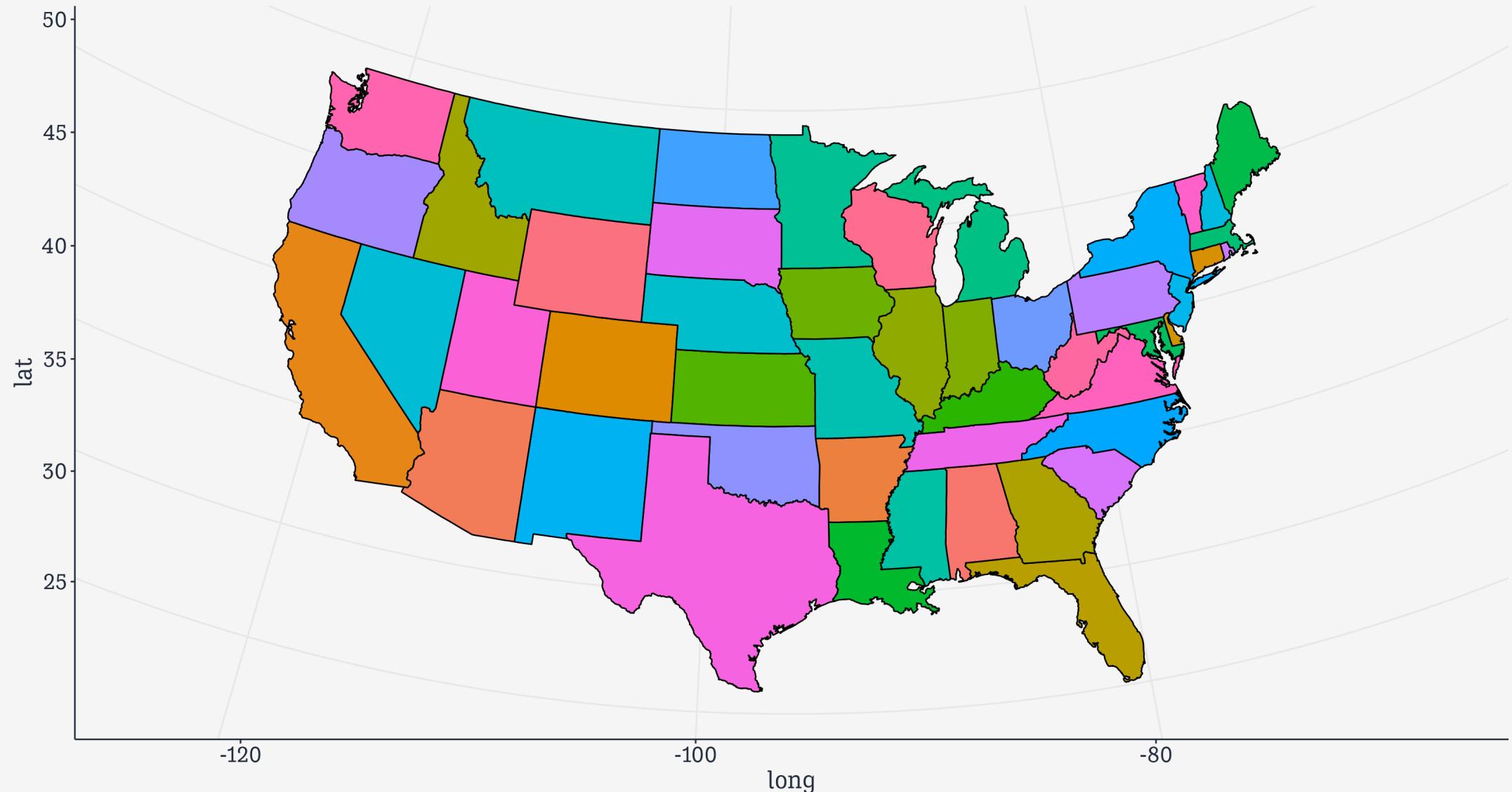


U.S. Map Projections



U.S. Map Projections





Our U.S. Map again, now transformed

Next, some data

We can merge our state-level election data with the `us_states` table, but we need to do a little work.

`us_states`

```
## # A tibble: 15,537 × 6
##   long    lat group order region subregion
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

`election`

```
## # A tibble: 51 × 22
##   state      st      fips total_vote vote_margin winner party
##   <chr>     <chr>   <dbl>    <dbl>       <dbl> <chr>  <chr>
## 1 Alabama    AL      1  2123372  588708 Trump   Rep
## 2 Alaska     AK      2  318608   46933 Trump   Rep
## 3 Arizona    AZ      4  2604657  91234 Trump   Rep
## 4 Arkansas   AR      5  1130635  304378 Trump   Rep
## 5 Californi... CA      6  14237893 4269978 Clint... Dem
## 6 Colorado   CO      8  2780247  136386 Clint... Dem
## 7 Connecti... CT      9  1644920  224357 Clint... Dem
## 8 Delaware   DE     10  443814   50476 Clint... Dem
## 9 District... DC     11  311268  270107 Clint... Dem
## 10 Florida   FL     12  9502747  112911 Trump   Rep
## # ... with 41 more rows, and 13 more variables: d_points <dbl>,
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>,
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>,
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

Next, some data

We can merge our state-level election data with the `us_states` table, but we need to do a little work.

`us_states`

```
## # A tibble: 15,537 × 6
##   long    lat group order region subregion
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

`election`

```
## # A tibble: 51 × 22
##   state      st     fips total_vote vote_margin winner party
##   <chr>     <chr>  <dbl>    <dbl>       <dbl> <chr>  <chr>
## 1 Alabama    AL     1  2123372    588708 Trump   Rep
## 2 Alaska     AK     2  318608     46933  Trump   Rep
## 3 Arizona    AZ     4  2604657    91234  Trump   Rep
## 4 Arkansas   AR     5  1130635    304378 Trump   Rep
## 5 Californi... CA     6  14237893  4269978 Clint... Dem
## 6 Colorado   CO     8  2780247    136386 Clint... Dem
## 7 Connecti... CT     9  1644920    224357 Clint... Dem
## 8 Delaware   DE    10  443814     50476  Clint... Dem
## 9 District... DC    11  311268    270107 Clint... Dem
## 10 Florida   FL    12  9502747   112911 Trump   Rep
## # ... with 41 more rows, and 13 more variables: d_points <dbl>,
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>,
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>,
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

To merge, or *join* these tables, they need to have a column in common to act as a key.

Recode to make a key

```
election <- election |>
  mutate(region = tolower(state)) |>
  relocate(region)

election

## # A tibble: 51 × 23
##   region      state st     fips total_vote vote_margin winner party pct_margin
##   <chr>        <chr> <chr> <dbl>    <dbl>      <dbl> <chr>  <chr>      <dbl>
## 1 alabama    Alab... AL      1    2123372    588708 Trump   Repu...    0.277
## 2 alaska      Alas... AK      2    318608     46933 Trump   Repu...    0.147
## 3 arizona     Ariz... AZ      4    2604657    91234 Trump   Repu...    0.035
## 4 arkansas    Arka... AR      5    1130635    304378 Trump   Repu...    0.269
## 5 california  Cali... CA      6    14237893   4269978 Clint... Demo...    0.300
## 6 colorado    Colo... CO      8    2780247    136386 Clint... Demo...    0.0491
## 7 connecticut Conn... CT      9    1644920    224357 Clint... Demo...    0.136
## 8 delaware    Dela... DE     10    443814     50476 Clint... Demo...    0.114
## 9 district of... Dist... DC     11    311268    270107 Clint... Demo...    0.868
## 10 florida    Flor... FL     12    9502747   112911 Trump   Repu...    0.0119
## # ... with 41 more rows, and 14 more variables: r_points <dbl>, d_points <dbl>,
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>, pct_other <dbl>,
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>, other_vote <dbl>,
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

Now we can join them

us_states

```
## # A tibble: 15,537 × 6
##   long     lat group order region subregion
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>
## 1 -87.5  30.4     1     1 alabama <NA>
## 2 -87.5  30.4     1     2 alabama <NA>
## 3 -87.5  30.4     1     3 alabama <NA>
## 4 -87.5  30.3     1     4 alabama <NA>
## 5 -87.6  30.3     1     5 alabama <NA>
## 6 -87.6  30.3     1     6 alabama <NA>
## 7 -87.6  30.3     1     7 alabama <NA>
## 8 -87.6  30.3     1     8 alabama <NA>
## 9 -87.7  30.3     1     9 alabama <NA>
## 10 -87.8 30.3     1    10 alabama <NA>
## # ... with 15,527 more rows
```

election

```
## # A tibble: 51 × 23
##   region      state st     fips total_vote vote_margin w
##   <chr>        <chr> <chr> <dbl>    <dbl>       <dbl> <
## 1 alabama     Alab... AL      1  2123372  588708 T
## 2 alaska       Alas... AK      2  318608   46933 T
## 3 arizona      Ariz... AZ      4  2604657  91234 T
## 4 arkansas     Arka... AR      5  1130635  304378 T
## 5 california   Cali... CA      6  14237893 4269978 C
## 6 colorado     Colo... CO      8  2780247  136386 C
## 7 connecticut  Conn... CT      9  1644920  224357 C
## 8 delaware     Dela... DE     10   443814  50476 C
## 9 district of... Dist... DC     11   311268  270107 C
## 10 florida     Flor... FL     12   9502747 112911 T
## # ... with 41 more rows, and 14 more variables: r_points <db...
## #   pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>,
## #   clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <db...
## #   ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

This is a *left join*

```
us_states_elec <- left_join(us_states, election, by = "region")

us_states_elec

## # A tibble: 15,537 × 28
##   long     lat group order region subregion state st     fips total_vote
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>    <chr> <dbl> <dbl>
## 1 -87.5  30.4     1     1 alabama <NA> Alabama AL     1  2123372
## 2 -87.5  30.4     1     2 alabama <NA> Alabama AL     1  2123372
## 3 -87.5  30.4     1     3 alabama <NA> Alabama AL     1  2123372
## 4 -87.5  30.3     1     4 alabama <NA> Alabama AL     1  2123372
## 5 -87.6  30.3     1     5 alabama <NA> Alabama AL     1  2123372
## 6 -87.6  30.3     1     6 alabama <NA> Alabama AL     1  2123372
## 7 -87.6  30.3     1     7 alabama <NA> Alabama AL     1  2123372
## 8 -87.6  30.3     1     8 alabama <NA> Alabama AL     1  2123372
## 9 -87.7  30.3     1     9 alabama <NA> Alabama AL     1  2123372
## 10 -87.8 30.3     1    10 alabama <NA> Alabama AL    1  2123372
## # ... with 15,527 more rows, and 18 more variables: vote_margin <dbl>,
## #   winner <chr>, party <chr>, pct_margin <dbl>, r_points <dbl>,
## #   d_points <dbl>, pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>,
## #   pct_other <dbl>, clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>,
## #   other_vote <dbl>, ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

This is a *left join*

```
us_states_elec <- left_join(us_states, election, by = "region")

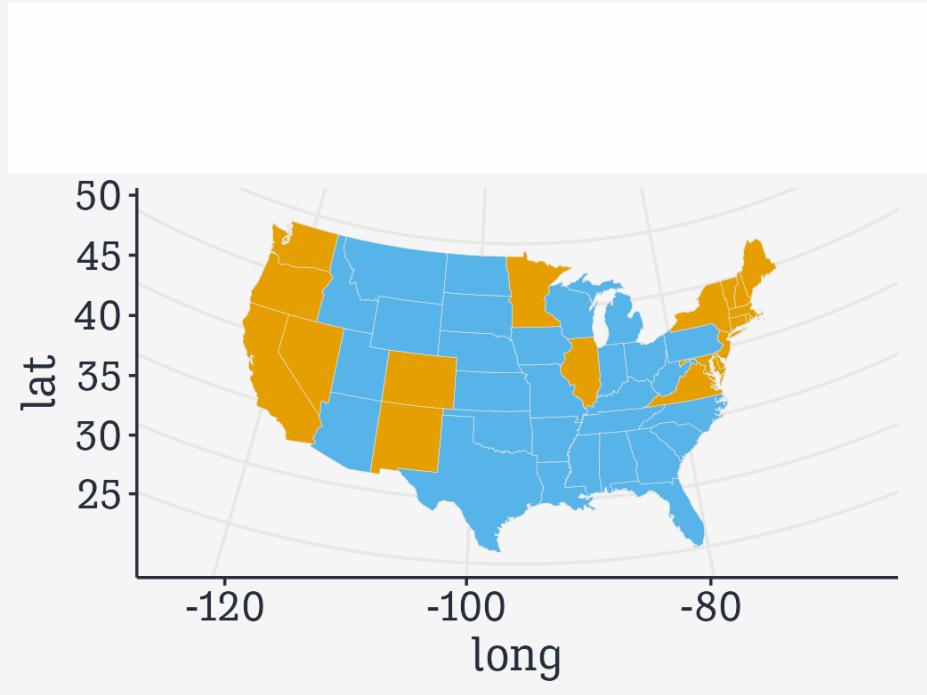
us_states_elec

## # A tibble: 15,537 × 28
##   long    lat group order region subregion state st     fips total_vote
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>    <chr> <dbl> <dbl>
## 1 -87.5  30.4    1      1 alabama <NA> Alabama AL     1  2123372
## 2 -87.5  30.4    1      2 alabama <NA> Alabama AL     1  2123372
## 3 -87.5  30.4    1      3 alabama <NA> Alabama AL     1  2123372
## 4 -87.5  30.3    1      4 alabama <NA> Alabama AL     1  2123372
## 5 -87.6  30.3    1      5 alabama <NA> Alabama AL     1  2123372
## 6 -87.6  30.3    1      6 alabama <NA> Alabama AL     1  2123372
## 7 -87.6  30.3    1      7 alabama <NA> Alabama AL     1  2123372
## 8 -87.6  30.3    1      8 alabama <NA> Alabama AL     1  2123372
## 9 -87.7  30.3    1      9 alabama <NA> Alabama AL     1  2123372
## 10 -87.8 30.3    1     10 alabama <NA> Alabama AL    1  2123372
## # ... with 15,527 more rows, and 18 more variables: vote_margin <dbl>,
## #   winner <chr>, party <chr>, pct_margin <dbl>, r_points <dbl>,
## #   d_points <dbl>, pct_clinton <dbl>, pct_trump <dbl>, pct_johnson <dbl>,
## #   pct_other <dbl>, clinton_vote <dbl>, trump_vote <dbl>, johnson_vote <dbl>,
## #   other_vote <dbl>, ev_dem <dbl>, ev_rep <dbl>, ev_oth <dbl>, census <chr>
```

Now our `us_states_elec` table has both the line-drawing information and (very redundantly) the election data merged in, with rows repeated as necessary.

Now we can start drawing choropleths

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                      y = lat,  
                      fill = party,  
                      group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  guides(fill = "none")
```



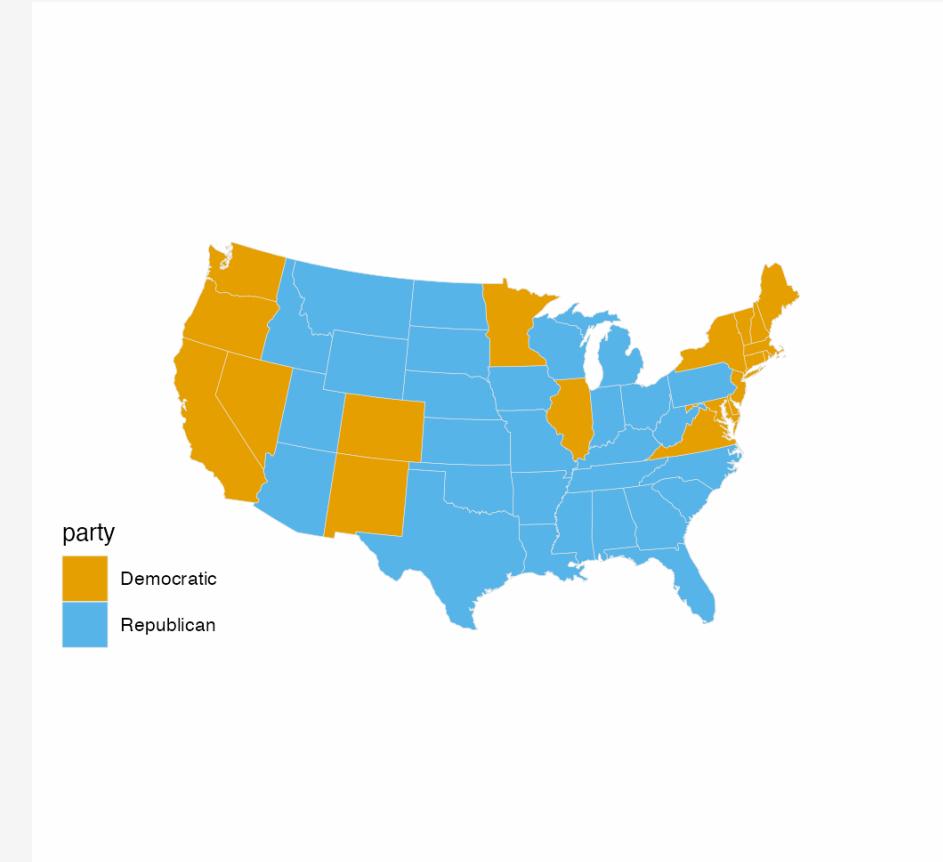
Let's turn off the gridlines

This is a *theme function*.

```
theme_map <- function(base_size=9, base_family="") {  
  require(grid)  
  theme_bw(base_size=base_size, base_family=base_family) %>%  
    theme(axis.line=element_blank(),  
          axis.text=element_blank(),  
          axis.ticks=element_blank(),  
          axis.title=element_blank(),  
          panel.background=element_blank(),  
          panel.border=element_blank(),  
          panel.grid=element_blank(),  
          panel.spacing=unit(0, "lines"),  
          plot.background=element_blank(),  
          legend.justification = c(0,0),  
          legend.position = c(0,0)  
    )  
}
```

Add the theme function at the end

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = party,  
                        group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  theme_map()
```



Fix the Party Colors

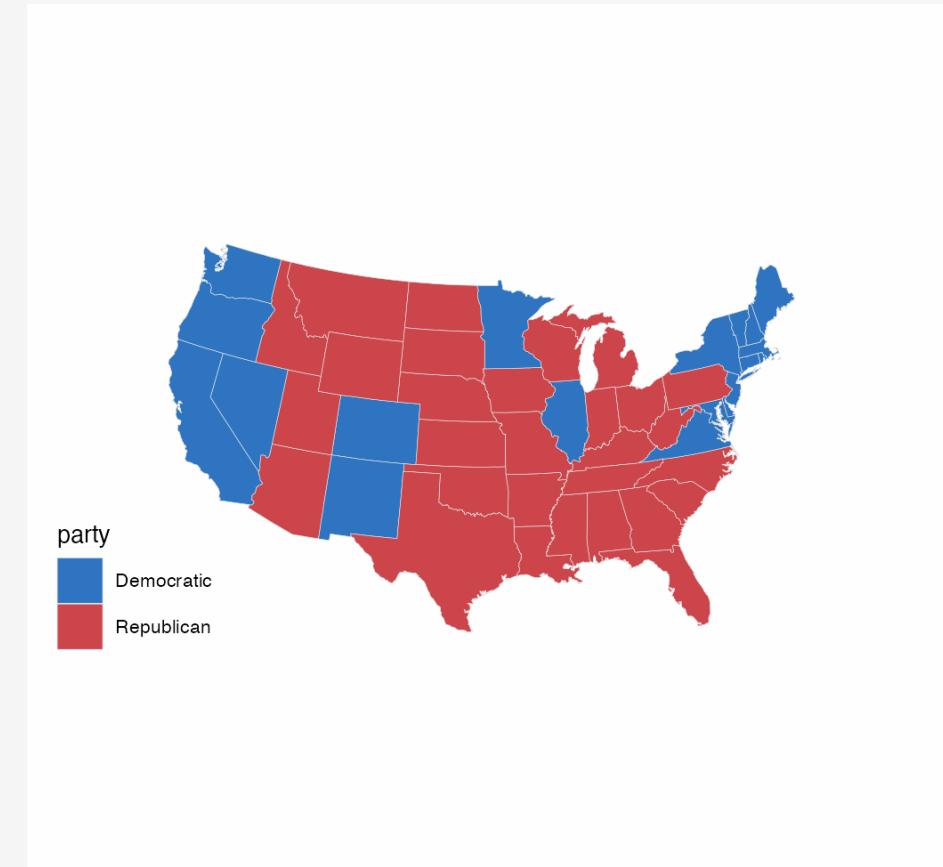
```
## Hex color codes for Democratic Blue and Republican Red
party_colors <- c("#2E74C0", "#CB454A")

us_states_elec |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = party,
                        group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_manual(values = party_colors) +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  theme_map()
```

Fix the Party Colors

```
## Hex color codes for Democratic Blue and Republican Red
party_colors <- c("#2E74C0", "#CB454A")

us_states_elec |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = party,
                        group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_manual(values = party_colors) +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  theme_map()
```

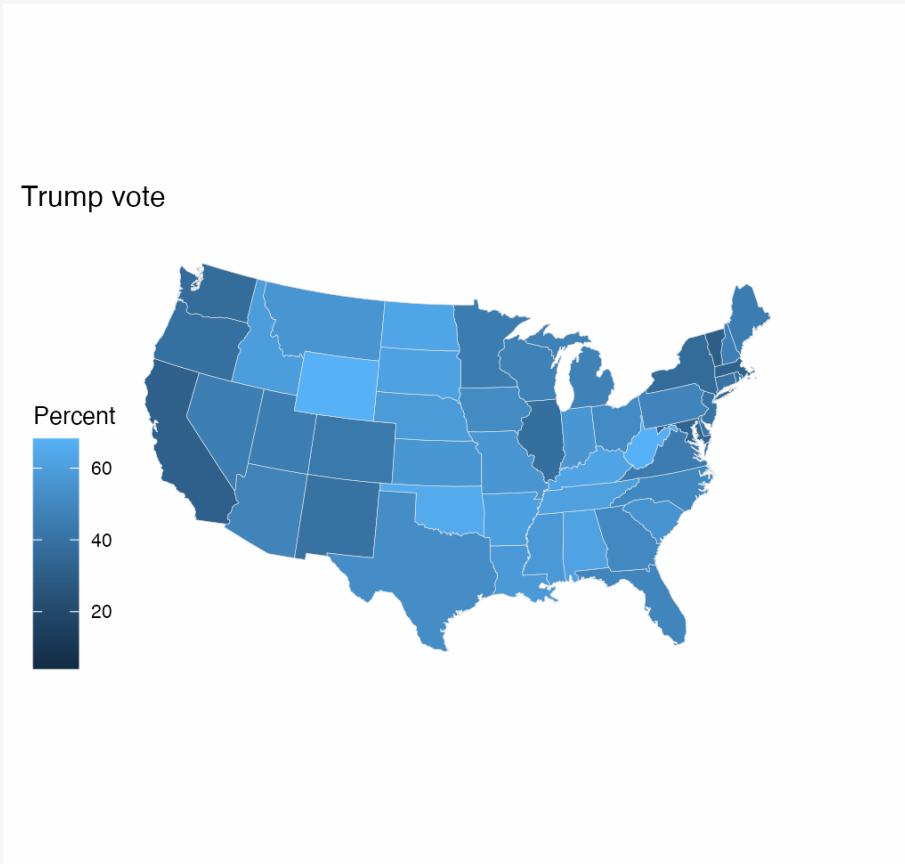


On maps, continuous measures are *gradients*

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = pct_trump,  
                        group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  labs(title = "Trump vote",  
       fill = "Percent") +  
  theme_map()
```

On maps, continuous measures are *gradients*

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = pct_trump,  
                        group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  labs(title = "Trump vote",  
       fill = "Percent") +  
  theme_map()
```

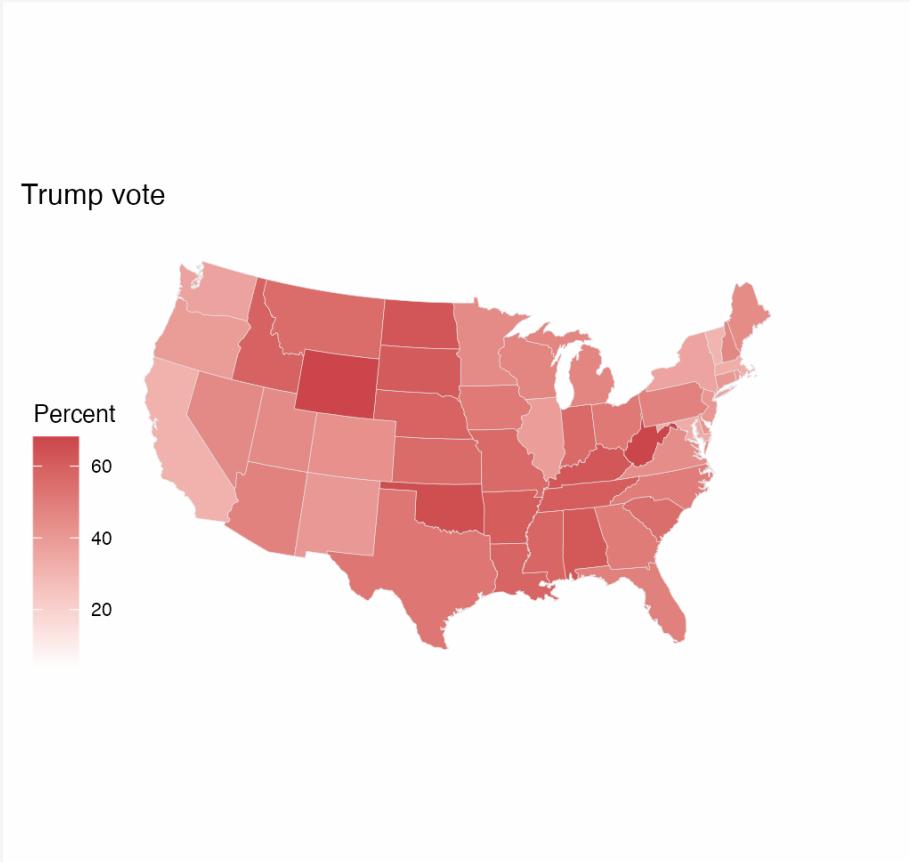


Fix the gradient scale with its **scale** function

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = pct_trump,  
                        group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  scale_fill_gradient(low = "white",  
                      high = "#CB454A") +  
  labs(title = "Trump vote") +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  labs(title = "Trump vote",  
       fill = "Percent") +  
  theme_map()
```

Fix the gradient scale with its **scale** function

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                        y = lat,  
                        fill = pct_trump,  
                        group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  scale_fill_gradient(low = "white",  
                      high = "#CB454A") +  
  labs(title = "Trump vote") +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  labs(title = "Trump vote",  
       fill = "Percent") +  
  theme_map()
```

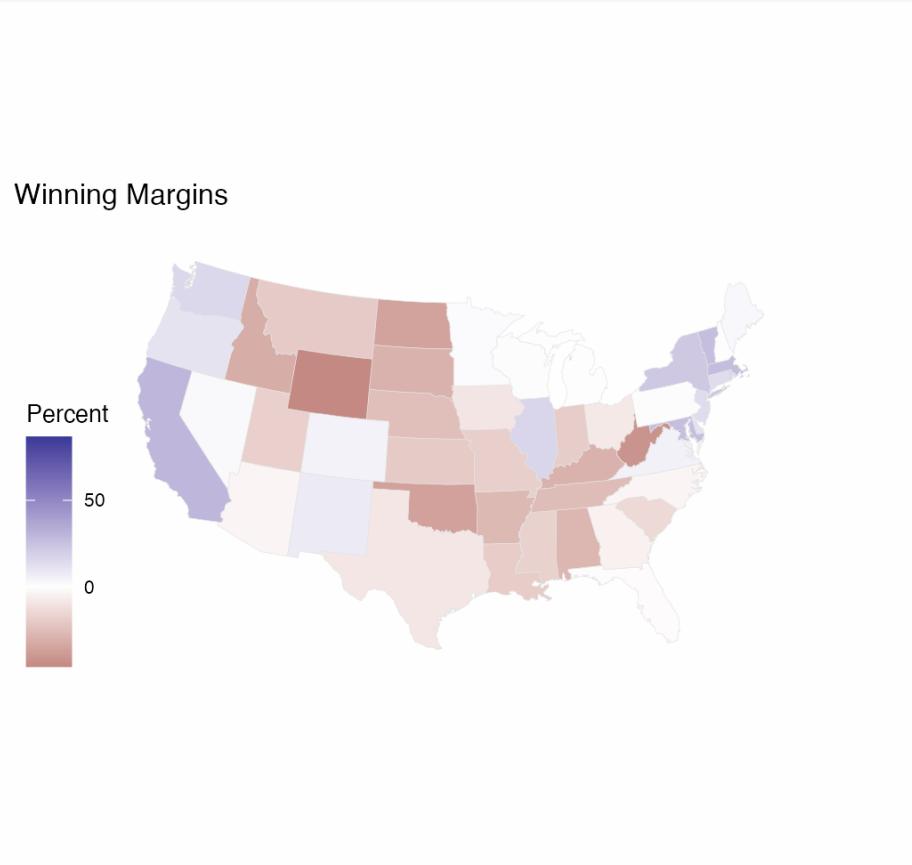


Some gradients are *diverging*

```
us_states_elec |>
  ggplot(mapping = aes(x = long,
                       y = lat,
                       fill = d_points,
                       group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_gradient2() +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  labs(title = "Winning Margins",
       fill = "Percent") +
  theme_map()
```

Some gradients are *diverging*

```
us_states_elec |>  
  ggplot(mapping = aes(x = long,  
                      y = lat,  
                      fill = d_points,  
                      group = group)) +  
  geom_polygon(color = "gray90",  
               size = 0.1) +  
  scale_fill_gradient2() +  
  coord_map(projection = "albers",  
            lat0 = 39, lat1 = 45) +  
  labs(title = "Winning Margins",  
       fill = "Percent") +  
  theme_map()
```

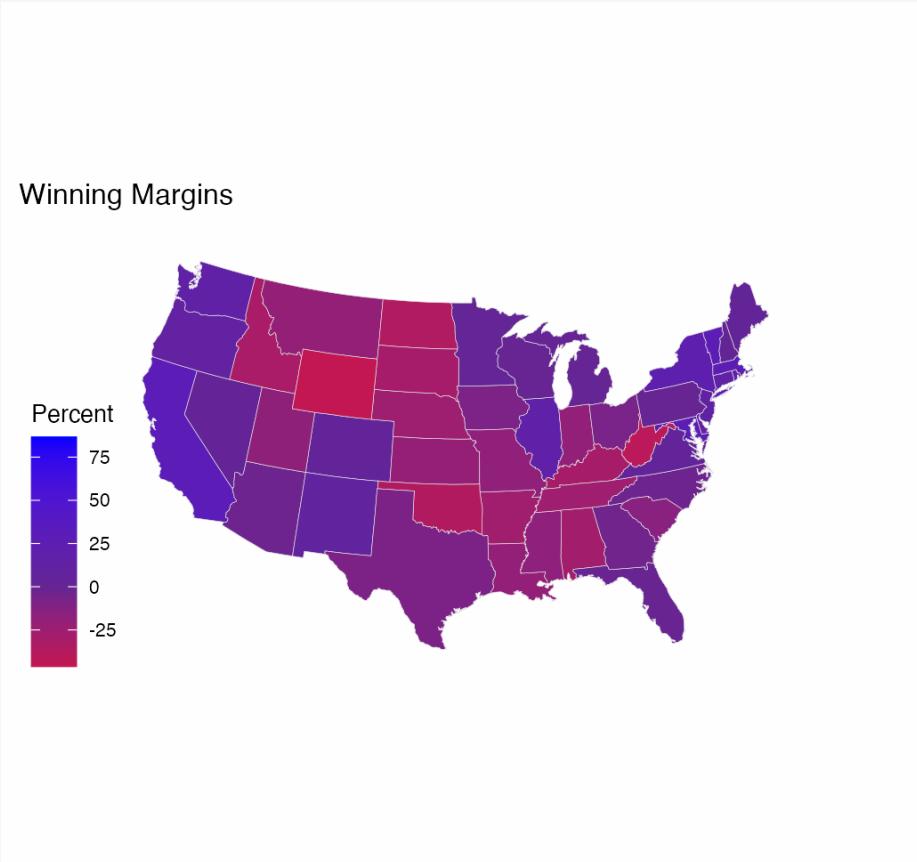


Purple America Map

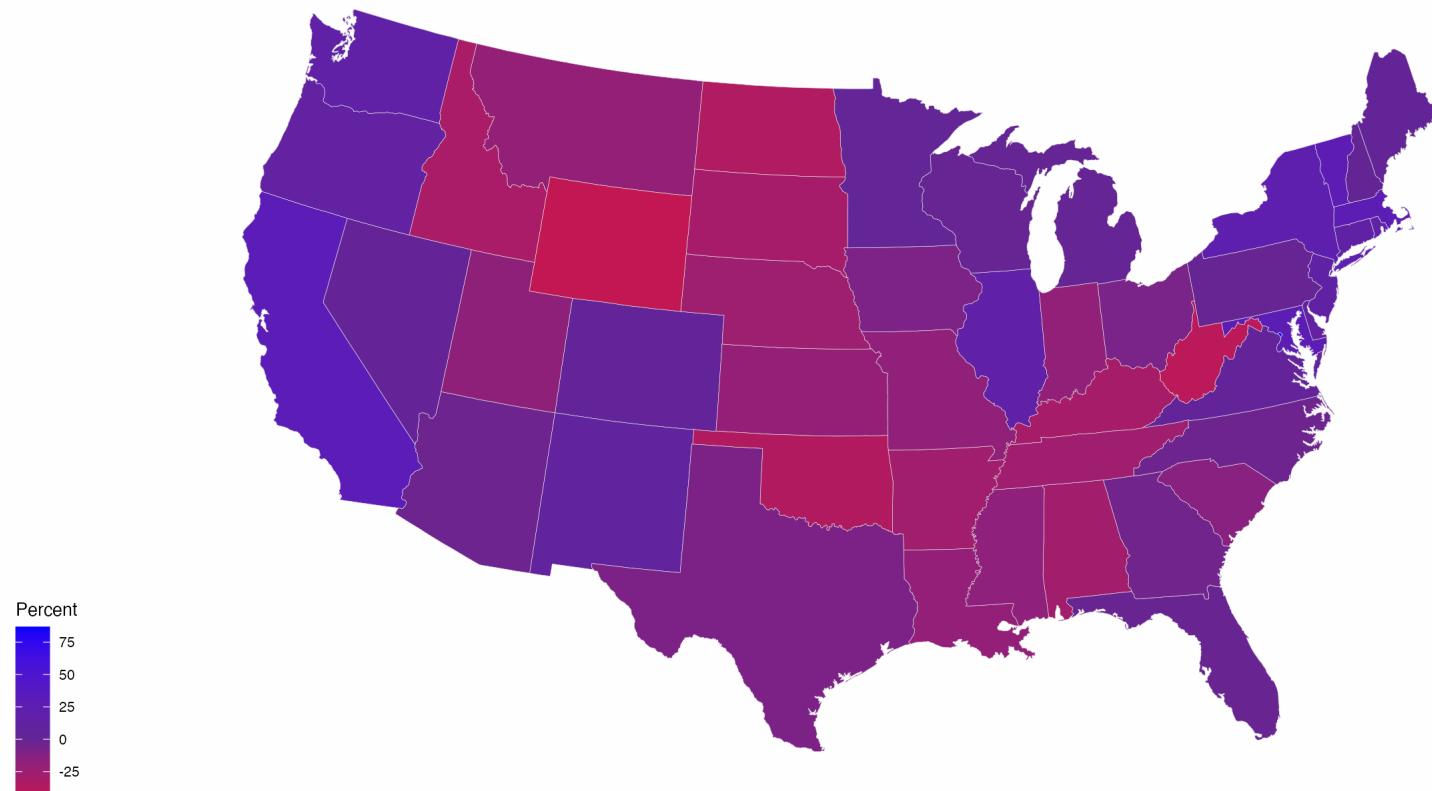
```
us_states_elec |>
  ggplot(mapping = aes(x = long,
                       y = lat,
                       fill = d_points,
                       group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_gradient2(low = "red",
                       mid = scales::muted("purple"),
                       high = "blue",
                       breaks = c(-25, 0, 25,
                                 50, 75)) +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  labs(title = "Winning Margins",
       fill = "Percent") +
  theme_map()
```

Purple America Map

```
us_states_elec |>
  ggplot(mapping = aes(x = long,
                       y = lat,
                       fill = d_points,
                       group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_gradient2(low = "red",
                       mid = scales::muted("purple"),
                       high = "blue",
                       breaks = c(-25, 0, 25,
                                 50, 75)) +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  labs(title = "Winning Margins",
       fill = "Percent") +
  theme_map()
```



Winning Margins



Take a closer look at this, though.



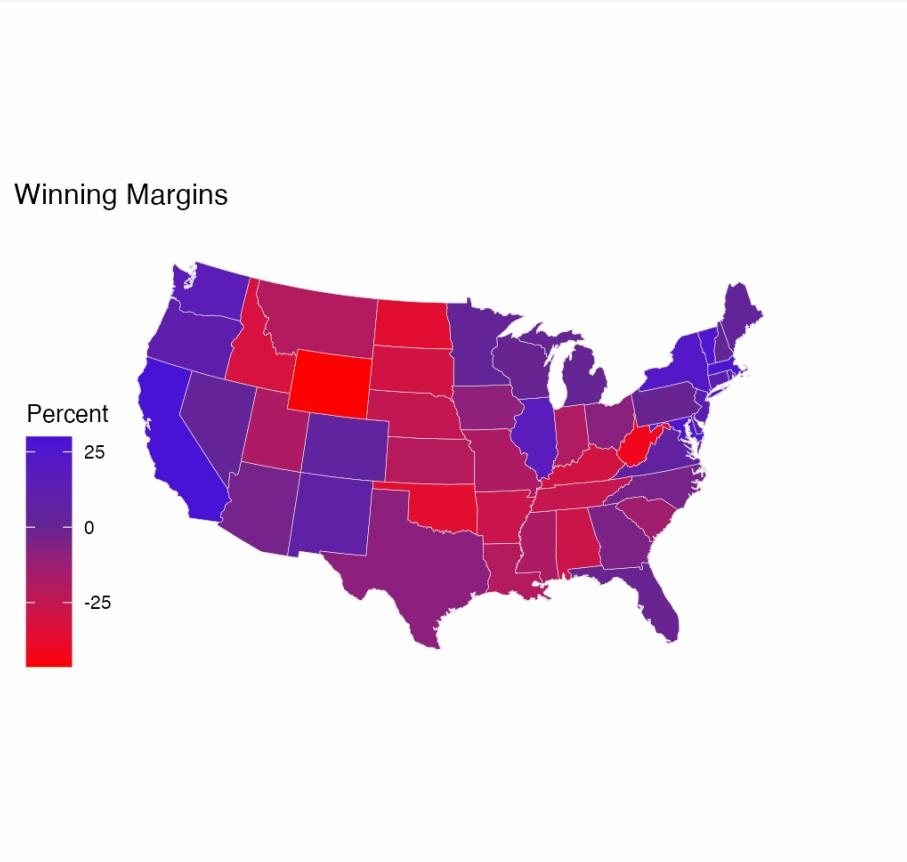
Washington, DC

Purple America Map, without DC

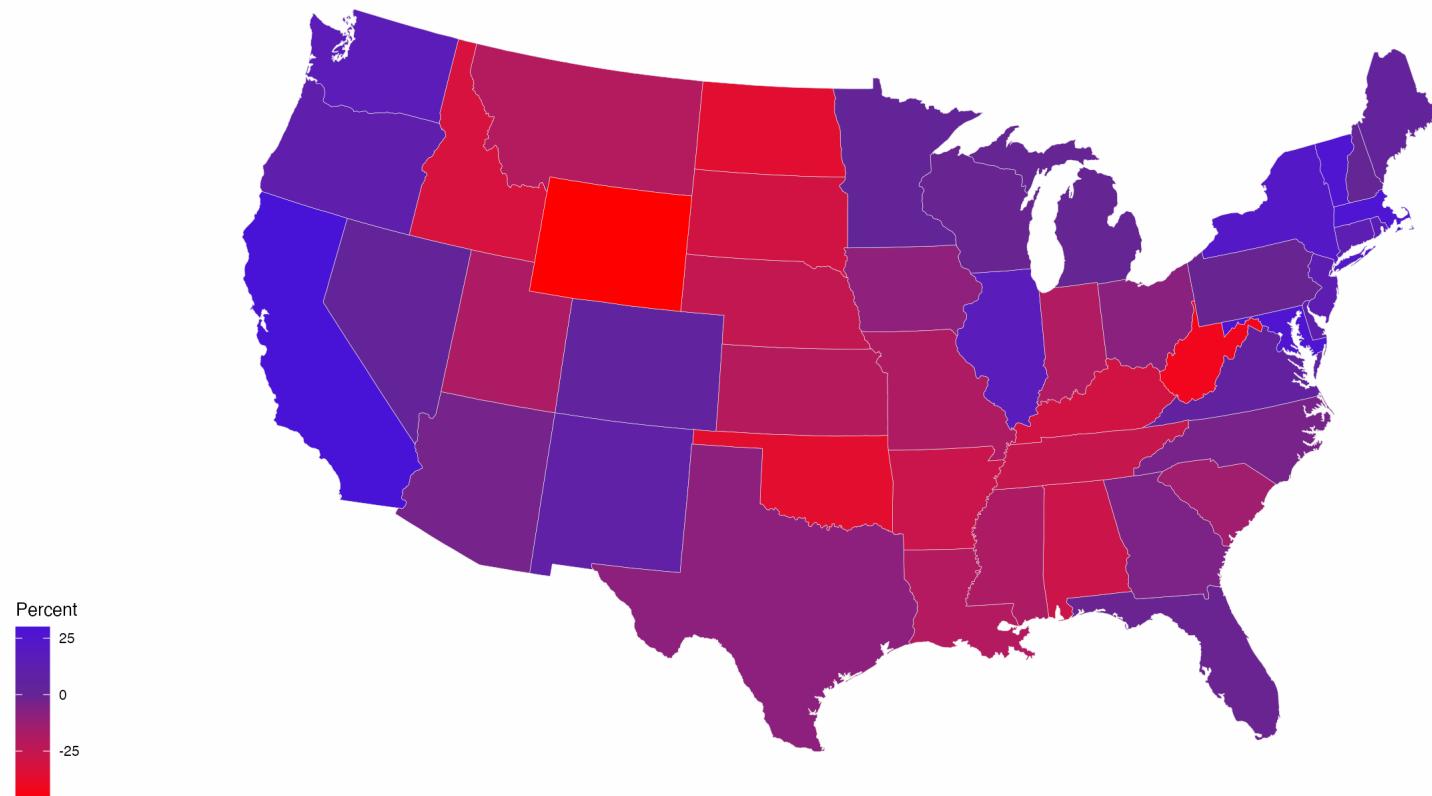
```
us_states_elec |>
  filter(region %nin% "district of columbia") |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = d_points,
                        group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_gradient2(low = "red",
                       mid = scales::muted("purple"),
                       high = "blue") +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  labs(title = "Winning Margins",
       fill = "Percent") +
  theme_map()
```

Purple America Map, without DC

```
us_states_elec |>
  filter(region %nin% "district of columbia") |>
  ggplot(mapping = aes(x = long,
                        y = lat,
                        fill = d_points,
                        group = group)) +
  geom_polygon(color = "gray90",
               size = 0.1) +
  scale_fill_gradient2(low = "red",
                       mid = scales::muted("purple"),
                       high = "blue") +
  coord_map(projection = "albers",
            lat0 = 39, lat1 = 45) +
  labs(title = "Winning Margins",
       fill = "Percent") +
  theme_map()
```



Winning Margins



More balanced.

America's Ur-Choropleths

County-level choropleths

Conceptually identical to state ones. The tables are just bigger, because there are way more lines to draw.

```
county_map <- as_tibble(county_map)
county_map

## # A tibble: 191,382 × 7
##       long      lat order  hole piece group      id
##       <dbl>    <dbl> <int> <lgl> <fct> <fct>    <chr>
## 1 1225889. -1275020.     1 FALSE  1 0500000US01001.1 01001
## 2 1235324. -1274008.     2 FALSE  1 0500000US01001.1 01001
## 3 1244873. -1272331.     3 FALSE  1 0500000US01001.1 01001
## 4 1244129. -1267515.     4 FALSE  1 0500000US01001.1 01001
## 5 1272010. -1262889.     5 FALSE  1 0500000US01001.1 01001
## 6 1276797. -1295514.     6 FALSE  1 0500000US01001.1 01001
## 7 1273832. -1297124.     7 FALSE  1 0500000US01001.1 01001
## 8 1272727. -1296631.     8 FALSE  1 0500000US01001.1 01001
## 9 1272513. -1299771.     9 FALSE  1 0500000US01001.1 01001
## 10 1269950. -1302038.    10 FALSE  1 0500000US01001.1 01001
## # ... with 191,372 more rows
```

191,000 or so rows

id here is the county FIPS code.

County-level choropleths

```
county_data <- as_tibble(county_data)
county_data

## # A tibble: 3,195 × 32
##   id      name state census_region pop_dens pop_dens4 pop_dens6 pct_black     pop
##   <chr> <chr> <fct> <fct>       <fct>    <fct>    <fct>    <int>
## 1 0      <NA>  <NA>  <NA>        [ 50,... [ 45, 1... [ 82, 2... [10.0,15... 3.19e8
## 2 01000 1     AL   South          [ 50,... [ 45, 1... [ 82, 2... [25.0,50... 4.85e6
## 3 01001 Auta... AL   South          [ 50,... [ 45, 1... [ 82, 2... [15.0,25... 5.54e4
## 4 01003 Bald... AL   South          [ 100,... [118,716... [ 82, 2... [ 5.0,10... 2.00e5
## 5 01005 Barb... AL   South          [ 10,... [ 17, ... [ 25, ... [25.0,50... 2.69e4
## 6 01007 Bibb... AL   South          [ 10,... [ 17, ... [ 25, ... [15.0,25... 2.25e4
## 7 01009 Blou... AL   South          [ 50,... [ 45, 1... [ 82, 2... [ 0.0, 2... 5.77e4
## 8 01011 Bull... AL   South          [ 10,... [ 17, ... [ 9, ... [50.0,85... 1.08e4
## 9 01013 Butl... AL   South          [ 10,... [ 17, ... [ 25, ... [25.0,50... 2.03e4
## 10 01015 Calh... AL   South         [ 100,... [118,716... [ 82, 2... [15.0,25... 1.16e5
## # ... with 3,185 more rows, and 23 more variables: female <dbl>, white <dbl>,
## #   black <dbl>, travel_time <dbl>, land_area <dbl>, hh_income <int>,
## #   su_gun4 <fct>, su_gun6 <fct>, fips <dbl>, votes_dem_2016 <int>,
## #   votes_gop_2016 <int>, total_votes_2016 <int>, per_dem_2016 <dbl>,
## #   per_gop_2016 <dbl>, diff_2016 <int>, per_dem_2012 <dbl>,
## #   per_gop_2012 <dbl>, diff_2012 <int>, winner <chr>, partywinner16 <chr>,
## #   winner12 <chr>, partywinner12 <chr>, flipped <chr>
```

County-level choropleths

3,195 entities, including states (FIPS id ends in four zeros)

And the US as a whole (FIPS id of 0)

Sample a few rows, with specific columns:

```
county_data |>
  select(id, name, state, pop_dens, pct_black) %>%
  sample_n(10)

## # A tibble: 10 × 5
##   id      name      state pop_dens     pct_black
##   <chr> <chr>    <fct> <fct>       <fct>
## 1 06000 5          CA     [ 100,  500) [ 5.0,10.0)
## 2 39029 Columbian County OH     [ 100,  500) [ 2.0, 5.0)
## 3 51520 Bristol city    VA     [ 1000, 5000) [ 5.0,10.0)
## 4 45085 Sumter County   SC     [ 100,  500) [25.0,50.0)
## 5 31023 Butler County   NE     [ 10,   50) [ 0.0, 2.0)
## 6 42125 Washington County PA     [ 100,  500) [ 2.0, 5.0)
## 7 48465 Val Verde County TX     [ 10,   50) [ 0.0, 2.0)
## 8 48075 Childress County   TX     [ 10,   50) [10.0,15.0)
## 9 13011 Banks County     GA     [ 50,  100) [ 2.0, 5.0)
## 10 38031 Foster County    ND     [ 0,   10) [ 0.0, 2.0)
```

Joined table

```
county_full <- left_join(county_map, county_data, by = "id")

county_full

## # A tibble: 191,382 × 38
##       long      lat order hole piece group     id    name state census_region
##       <dbl>    <dbl> <int> <lgl> <fct> <fct>   <chr> <chr> <fct> <fct>
## 1 1225889. -1275020.     1 FALSE 1 0500000... 01001 Auta... AL    South
## 2 1235324. -1274008.     2 FALSE 1 0500000... 01001 Auta... AL    South
## 3 1244873. -1272331.     3 FALSE 1 0500000... 01001 Auta... AL    South
## 4 1244129. -1267515.     4 FALSE 1 0500000... 01001 Auta... AL    South
## 5 1272010. -1262889.     5 FALSE 1 0500000... 01001 Auta... AL    South
## 6 1276797. -1295514.     6 FALSE 1 0500000... 01001 Auta... AL    South
## 7 1273832. -1297124.     7 FALSE 1 0500000... 01001 Auta... AL    South
## 8 1272727. -1296631.     8 FALSE 1 0500000... 01001 Auta... AL    South
## 9 1272513. -1299771.     9 FALSE 1 0500000... 01001 Auta... AL    South
## 10 1269950. -1302038.    10 FALSE 1 0500000... 01001 Auta... AL   South
## # ... with 191,372 more rows, and 28 more variables: pop_dens <fct>,
## #   pop_dens4 <fct>, pop_dens6 <fct>, pct_black <fct>, pop <int>, female <dbl>,
## #   white <dbl>, black <dbl>, travel_time <dbl>, land_area <dbl>,
## #   hh_income <int>, su_gun4 <fct>, su_gun6 <fct>, fips <dbl>,
## #   votes_dem_2016 <int>, votes_gop_2016 <int>, total_votes_2016 <int>,
## #   per_dem_2016 <dbl>, per_gop_2016 <dbl>, diff_2016 <int>,
## #   per_dem_2012 <dbl>, per_gop_2012 <dbl>, diff_2012 <int>, winner <chr>, ...
```

County Population Density

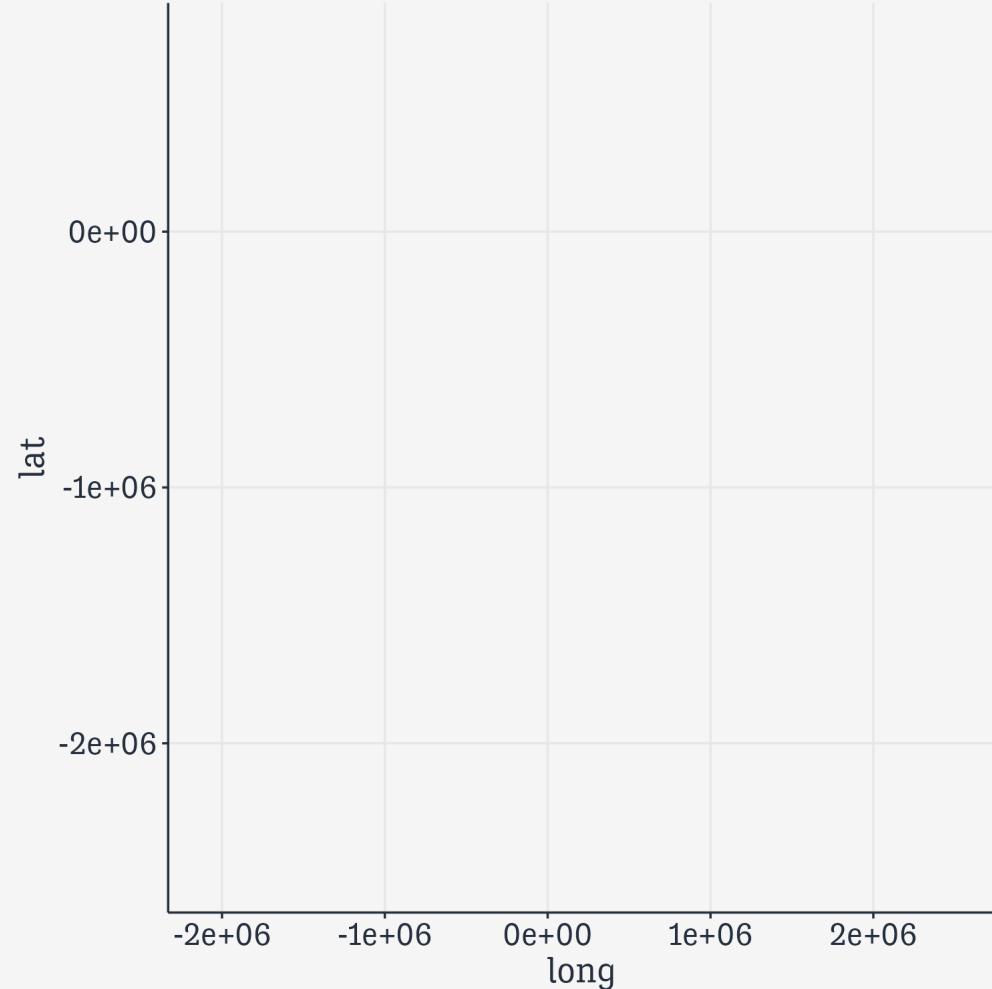
```
county_full <- left_join(county_map, county_data,
```

County Population Density

```
county_full <- left_join(county_map, county_data,  
                           by = c("id" = "id")  
                           )  
  
county_full  
## # A tibble: 191,382 × 38  
##       long     lat order hole piece group    id   name state census_re...  
##       <dbl>   <dbl> <int> <lgl> <fct> <fct>   <chr> <chr> <fct> <fct> ...  
## 1 1225889. -1275020.     1 FALSE 1 0500000... 01001 Auta... AL South  
## 2 1235324. -1274008.     2 FALSE 1 0500000... 01001 Auta... AL South  
## 3 1244873. -1272331.     3 FALSE 1 0500000... 01001 Auta... AL South  
## 4 1244129. -1267515.     4 FALSE 1 0500000... 01001 Auta... AL South  
## 5 1272010. -1262889.     5 FALSE 1 0500000... 01001 Auta... AL South  
## 6 1276797. -1295514.     6 FALSE 1 0500000... 01001 Auta... AL South  
## 7 1273832. -1297124.     7 FALSE 1 0500000... 01001 Auta... AL South  
## 8 1272727. -1296631.     8 FALSE 1 0500000... 01001 Auta... AL South  
## 9 1272513. -1299771.     9 FALSE 1 0500000... 01001 Auta... AL South  
## 10 1269950. -1302038.    10 FALSE 1 0500000... 01001 Auta... AL South  
## # ... with 191,372 more rows, and 28 more variables: pop_dens <fct>,...  
## #   pop_dens4 <fct>, pop_dens6 <fct>, pct_black <fct>, pop <int>, female <db...  
## #   white <dbl>, black <dbl>, travel_time <dbl>, land_area <dbl>,...  
## #   hh_income <int>, su_gun4 <fct>, su_gun6 <fct>, fips <dbl>,...  
## #   votes_dem_2016 <int>, votes_gop_2016 <int>, total_votes_2016 <int>,...  
## #   per_dem_2016 <dbl>, per_gop_2016 <dbl>, diff_2016 <int>,...  
## #   per_dem_2012 <dbl>, per_gop_2012 <dbl>, diff_2012 <int>, winner <chr>,...
```

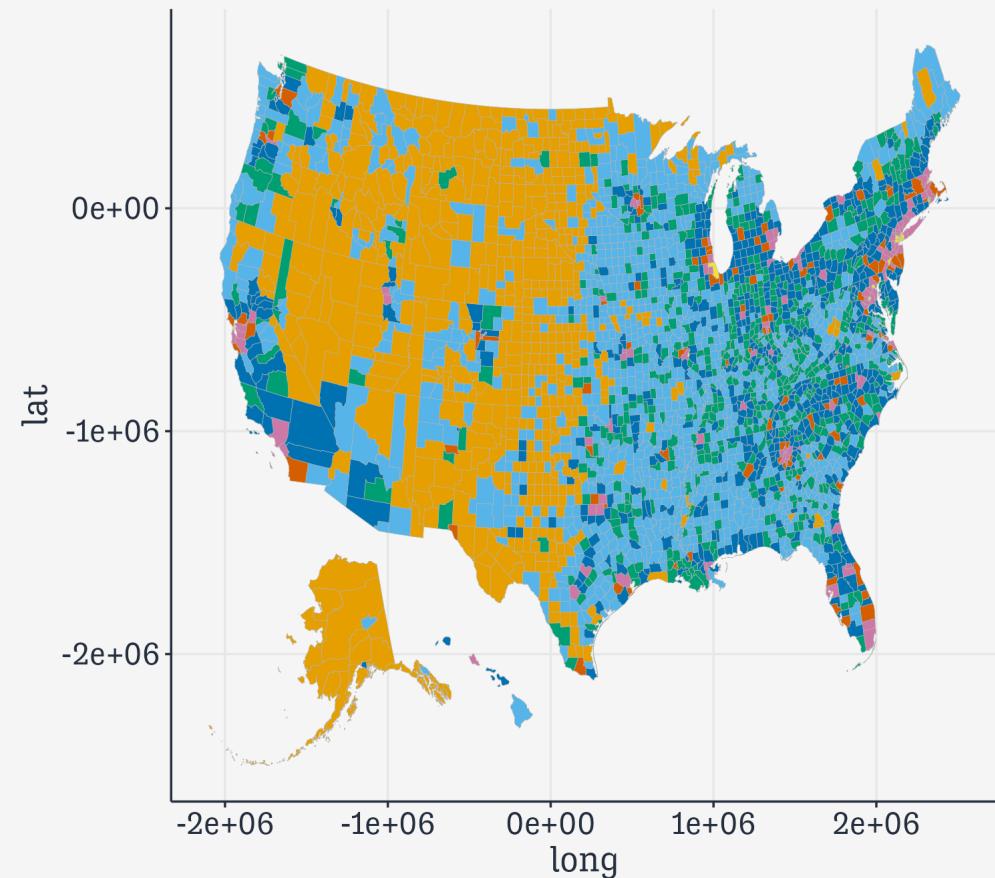
County Population Density

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group))
```



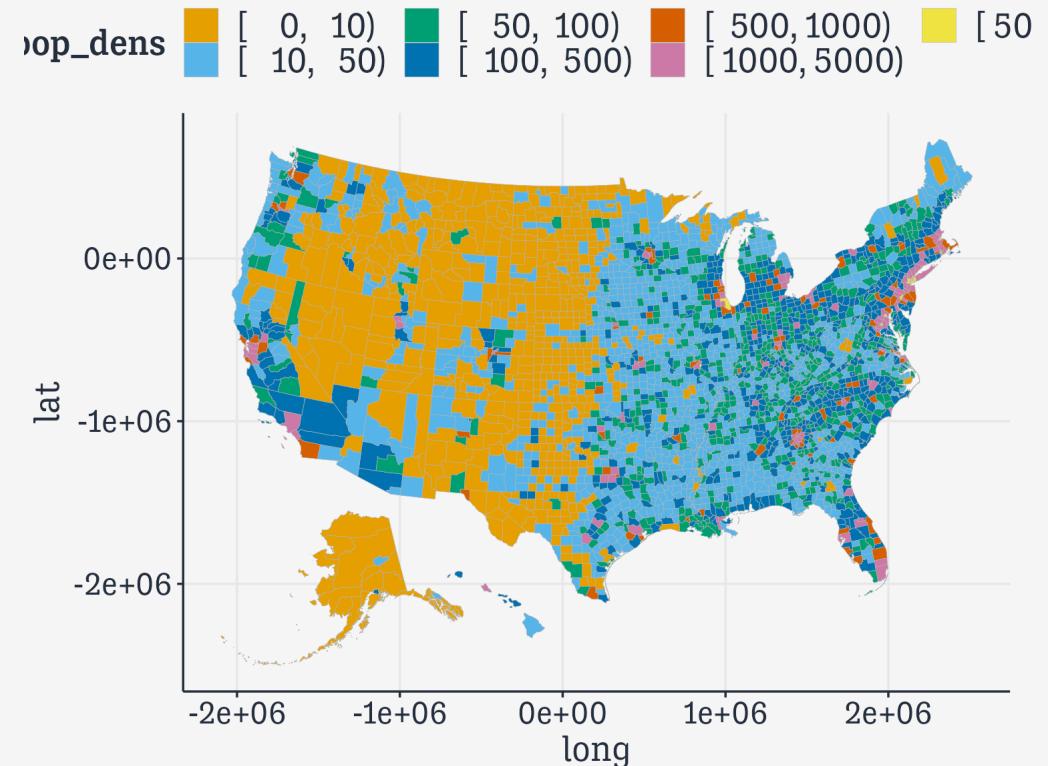
County Population Density

```
county_full <- left_join(county_map, county_data,  
  county_full |>  
    ggplot(mapping = aes(x = long, y = lat,  
                          fill = pop_dens,  
                          group = group)) +  
    geom_polygon(color = "gray70",  
                 size = 0.1)
```



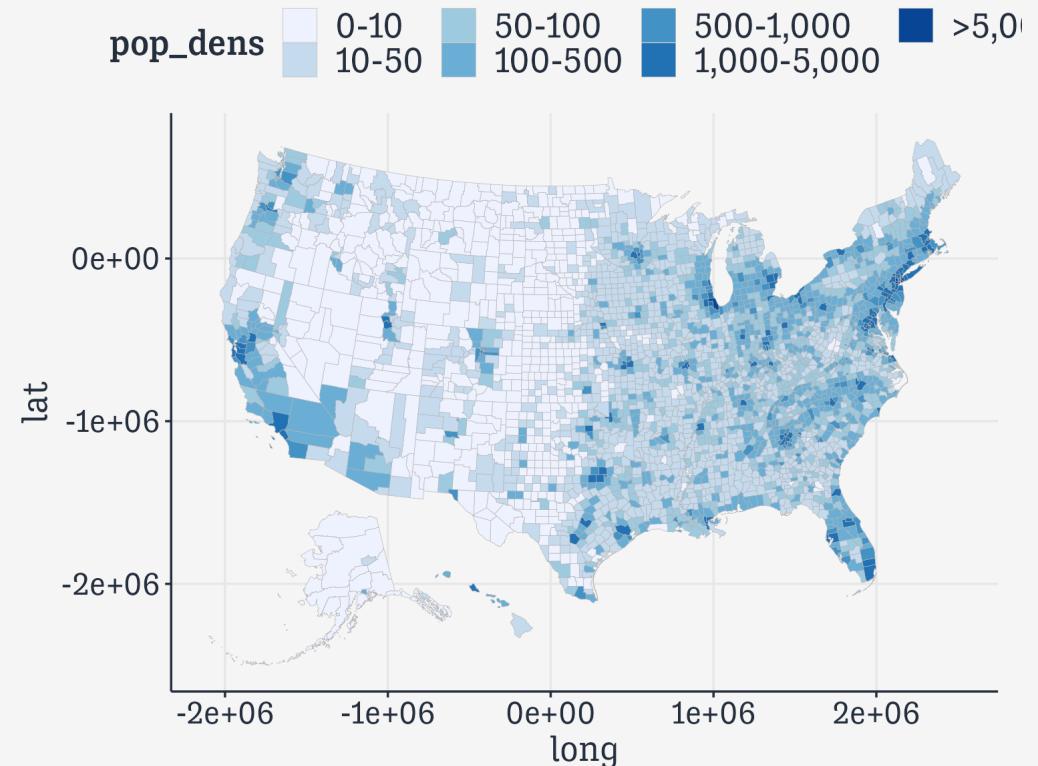
County Population Density

```
county_full <- left_join(county_map, county_data,  
  
  county_full |>  
    ggplot(mapping = aes(x = long, y = lat,  
                          fill = pop_dens,  
                          group = group)) +  
    geom_polygon(color = "gray70",  
                 size = 0.1) +  
    coord_fixed()
```



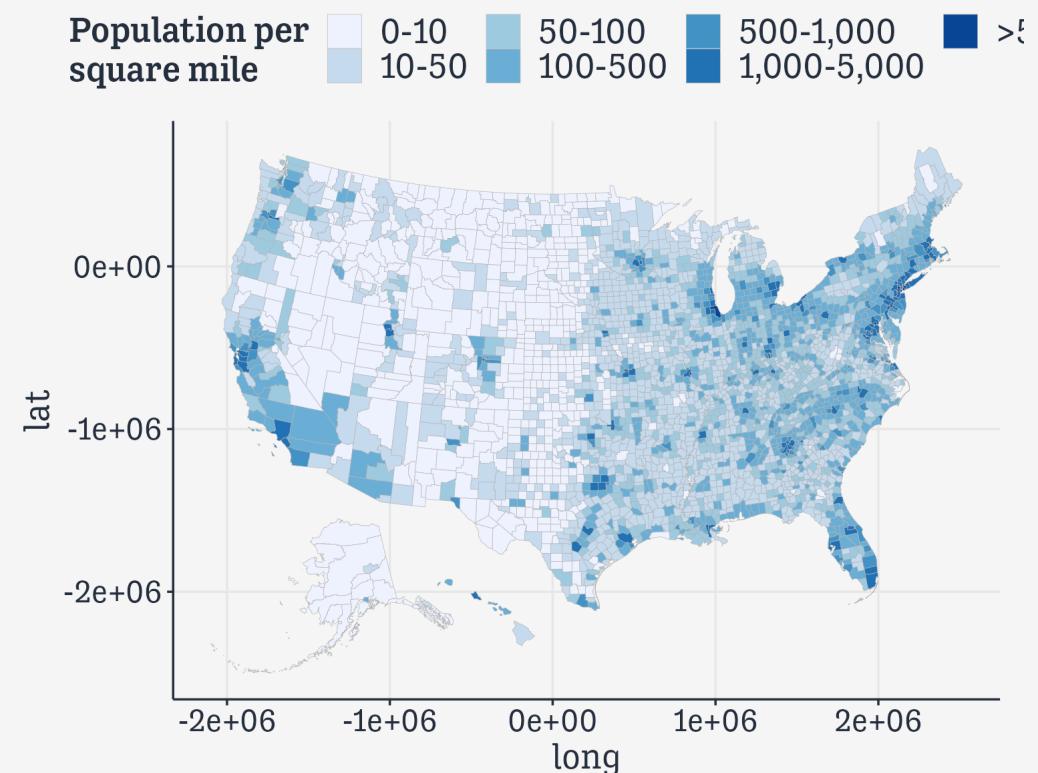
County Population Density

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Blues",  
                   labels = c("0-10", "10-50", "50-100",  
                           "100-500", "500-1,000",  
                           "1,000-5,000", ">5,000"))
```



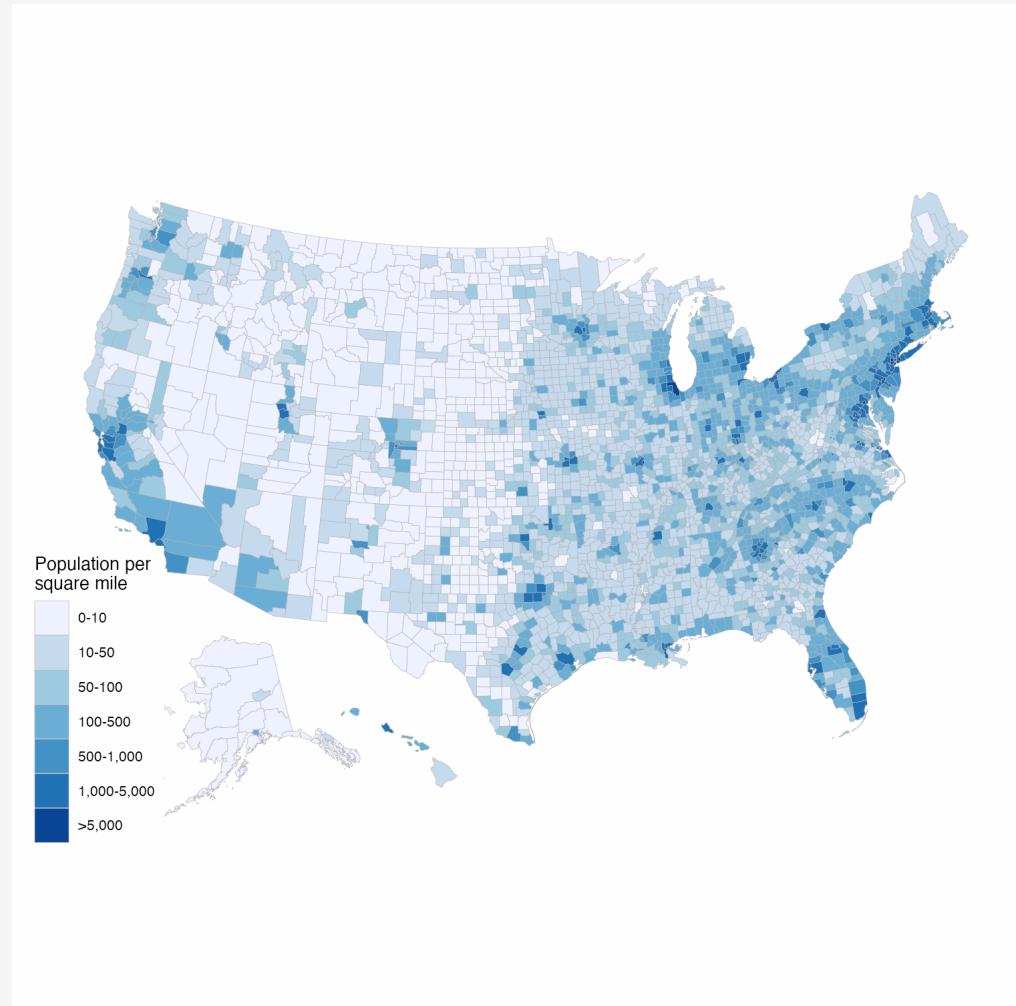
County Population Density

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Blues",  
                    labels = c("0-10", "10-50", "50-100",  
                            "100-500", "500-1,000",  
                            "1,000-5,000", ">5,000")) +  
  labs(fill = "Population per\nsquare mile")
```



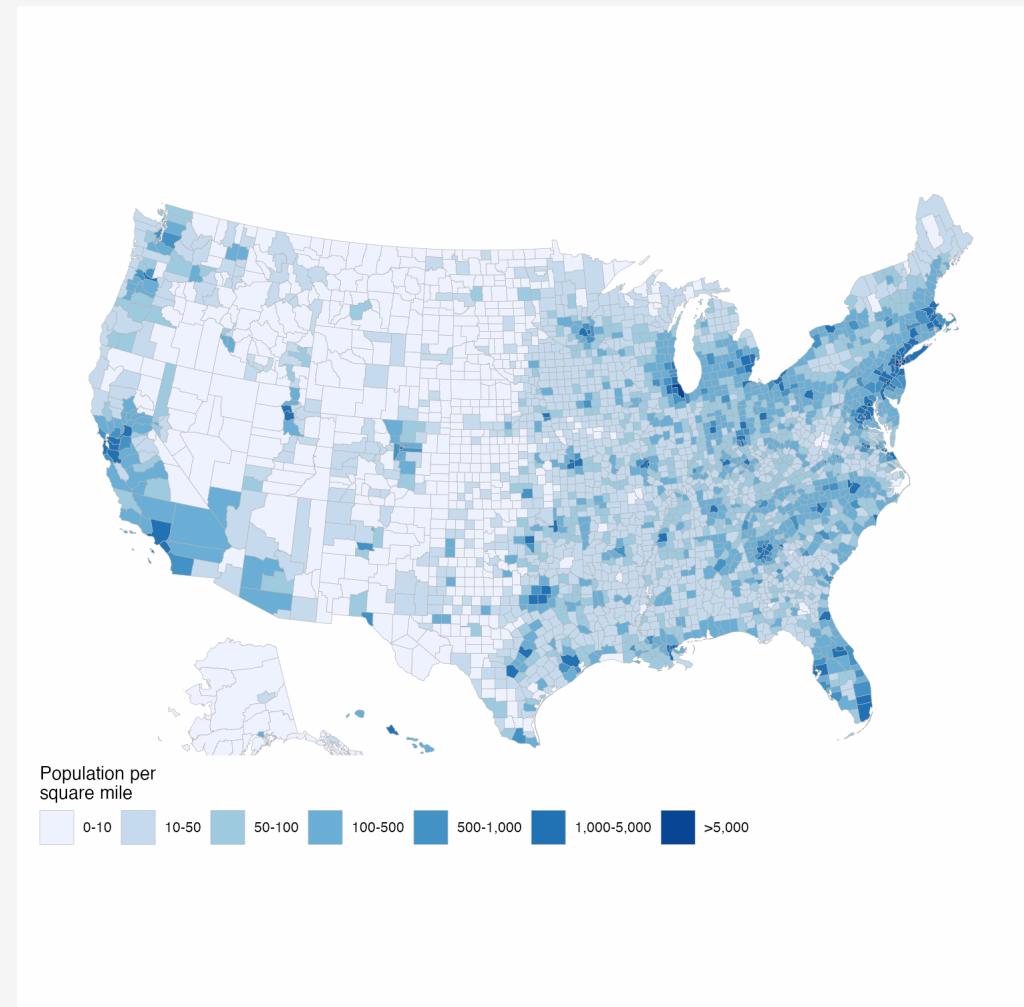
County Population Density

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Blues",  
                    labels = c("0-10", "10-50", "  
                    "100-500", "500-1,000",  
                    "1,000-5,000", ">5,000")) +  
  labs(fill = "Population per\nsquare mile") +  
  kjhslides::kjh_theme_map()
```



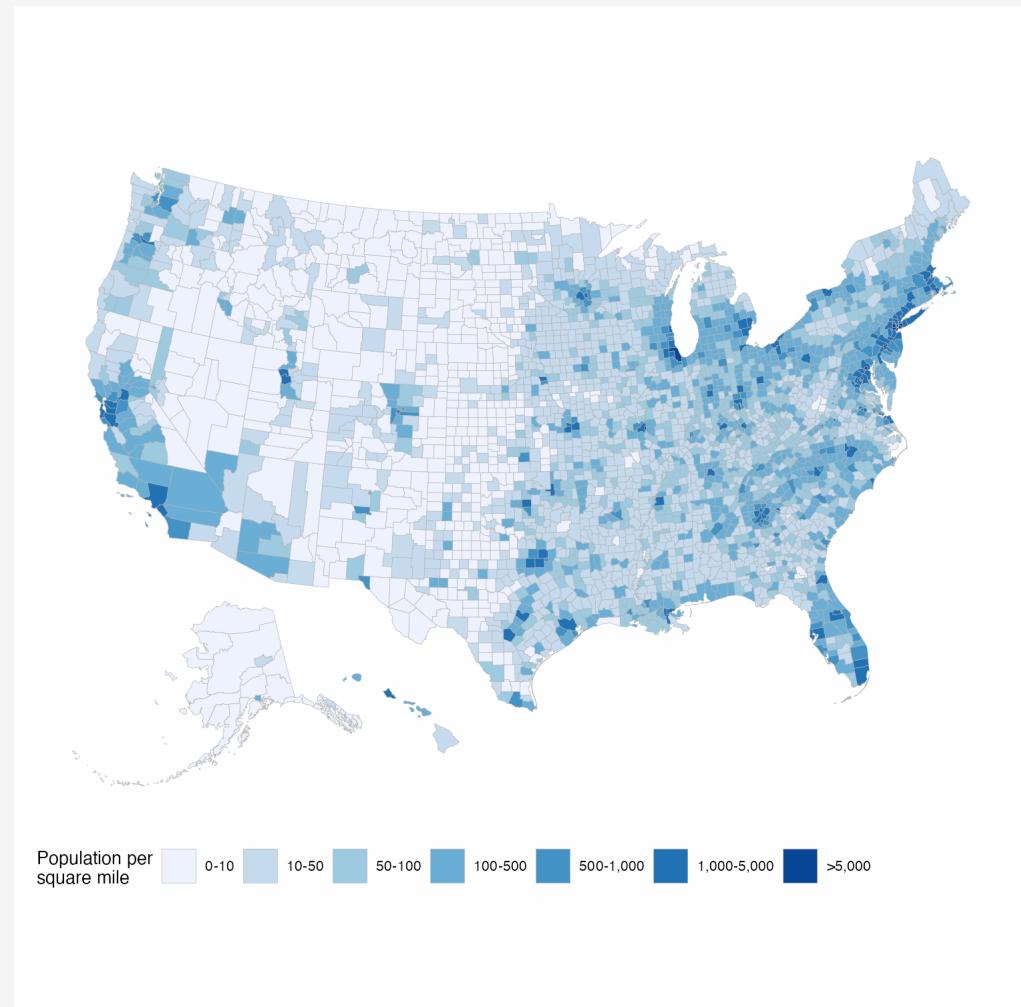
County Population Density

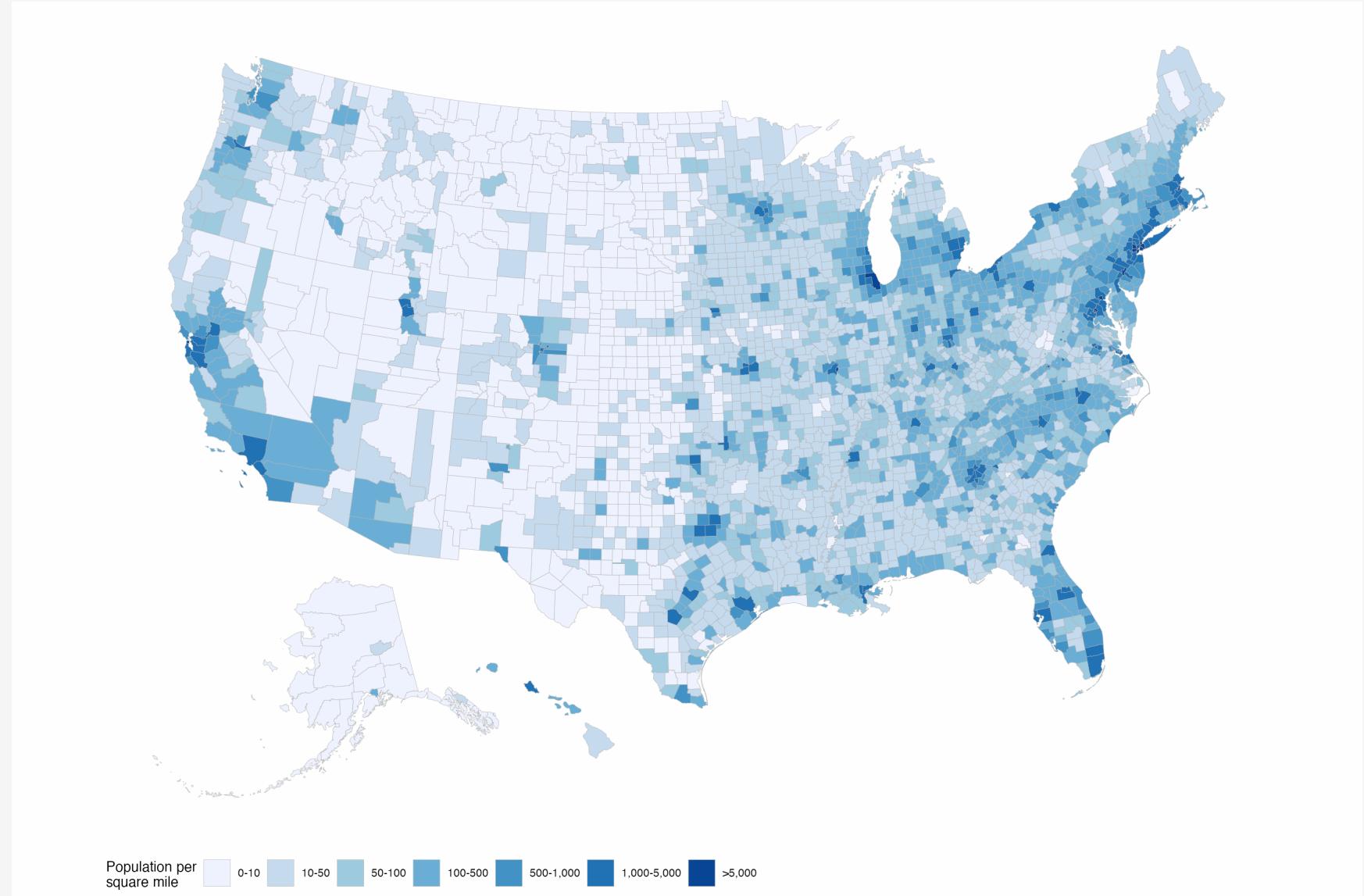
```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Blues",  
                    labels = c("0-10", "10-50", "50-100", "100-500", "500-1,000",  
                    "1,000-5,000", ">5,000")) +  
  labs(fill = "Population per\nsquare mile") +  
  kjhslides::kjh_theme_map() +  
  guides(fill = guide_legend(nrow = 1))
```



County Population Density

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pop_dens,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Blues",  
                    labels = c("0-10", "10-50", "5",  
                            "100-500", "500-1,000",  
                            "1,000-5,000", ">5,000")) +  
  labs(fill = "Population per\nsquare mile") +  
  kjhslides::kjh_theme_map() +  
  guides(fill = guide_legend(nrow = 1)) +  
  theme(legend.position = "bottom")
```





Population Density by County, binned

Same again for Percent Black

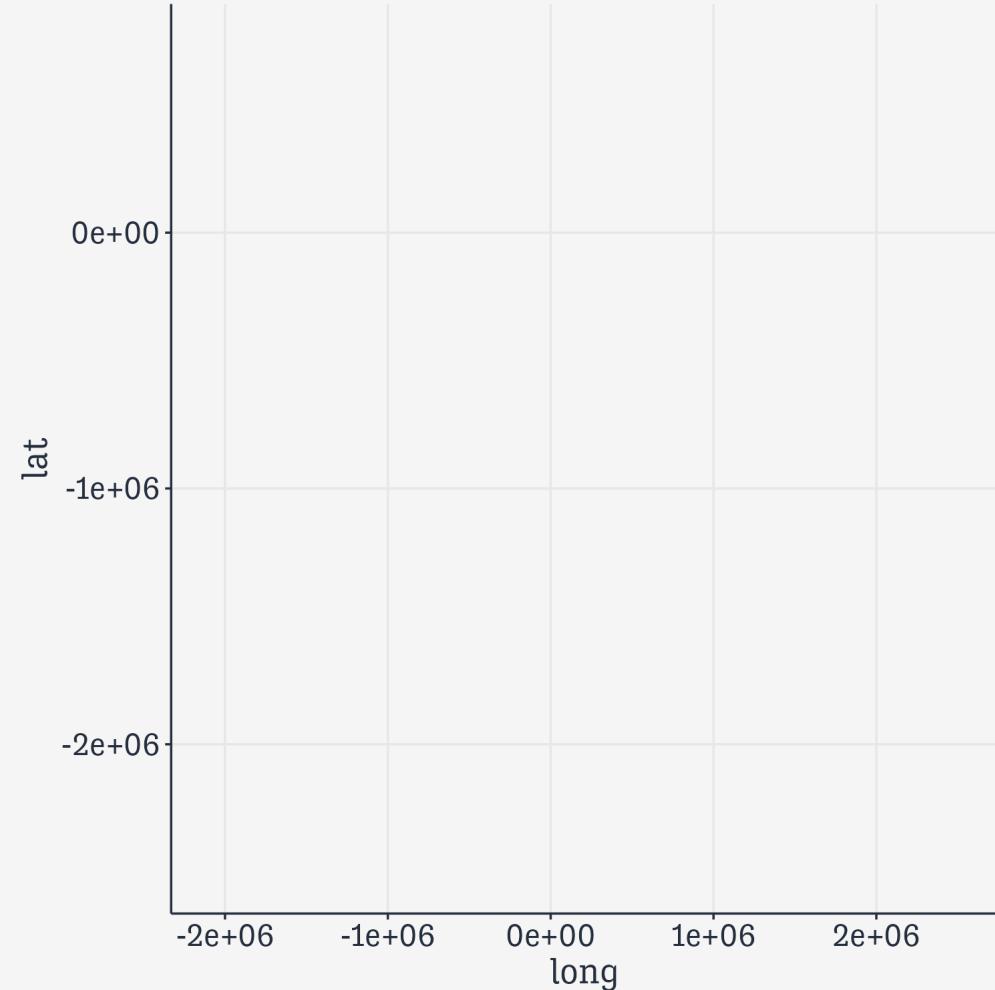
```
county_full <- left_join(county_map, county_data,
```

Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
  county_full  
  ## # A tibble: 191,382 × 38  
  ##       long      lat order hole piece group     id    name state census_re  
  ##   <dbl>    <dbl> <int> <lgl> <fct> <fct>   <chr> <chr> <fct> <fct>  
  ## 1 1225889. -1275020.     1 FALSE 1 0500000... 01001 Auta... AL South  
  ## 2 1235324. -1274008.     2 FALSE 1 0500000... 01001 Auta... AL South  
  ## 3 1244873. -1272331.     3 FALSE 1 0500000... 01001 Auta... AL South  
  ## 4 1244129. -1267515.     4 FALSE 1 0500000... 01001 Auta... AL South  
  ## 5 1272010. -1262889.     5 FALSE 1 0500000... 01001 Auta... AL South  
  ## 6 1276797. -1295514.     6 FALSE 1 0500000... 01001 Auta... AL South  
  ## 7 1273832. -1297124.     7 FALSE 1 0500000... 01001 Auta... AL South  
  ## 8 1272727. -1296631.     8 FALSE 1 0500000... 01001 Auta... AL South  
  ## 9 1272513. -1299771.     9 FALSE 1 0500000... 01001 Auta... AL South  
  ## 10 1269950. -1302038.    10 FALSE 1 0500000... 01001 Auta... AL South  
  ## # ... with 191,372 more rows, and 28 more variables: pop_dens <fct>,  
  ## #   pop_dens4 <fct>, pop_dens6 <fct>, pct_black <fct>, pop <int>, female <db  
  ## #   white <dbl>, black <dbl>, travel_time <dbl>, land_area <dbl>,  
  ## #   hh_income <int>, su_gun4 <fct>, su_gun6 <fct>, fips <dbl>,  
  ## #   votes_dem_2016 <int>, votes_gop_2016 <int>, total_votes_2016 <int>,  
  ## #   per_dem_2016 <dbl>, per_gop_2016 <dbl>, diff_2016 <int>,  
  ## #   per_dem_2012 <dbl>, per_gop_2012 <dbl>, diff_2012 <int>, winner <chr>,...
```

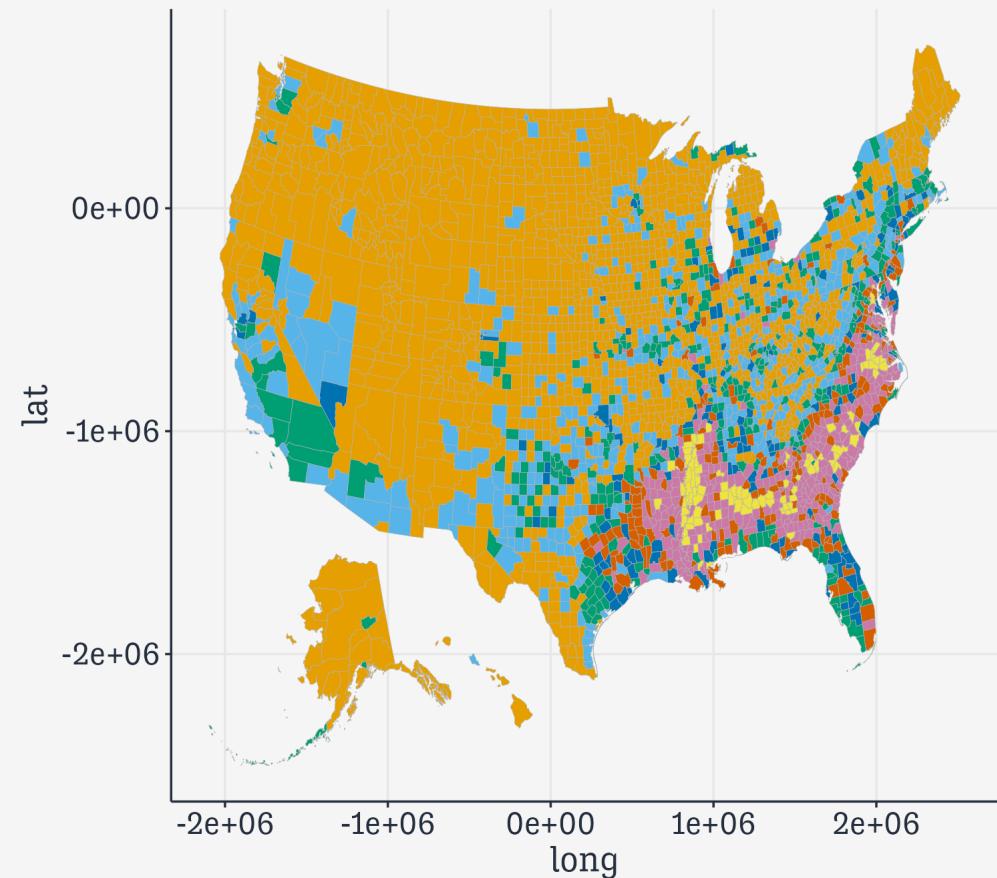
Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
  county_full |>  
    ggplot(mapping = aes(x = long, y = lat,  
                          fill = pct_black,  
                          group = group))
```



Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pct_black,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1)
```

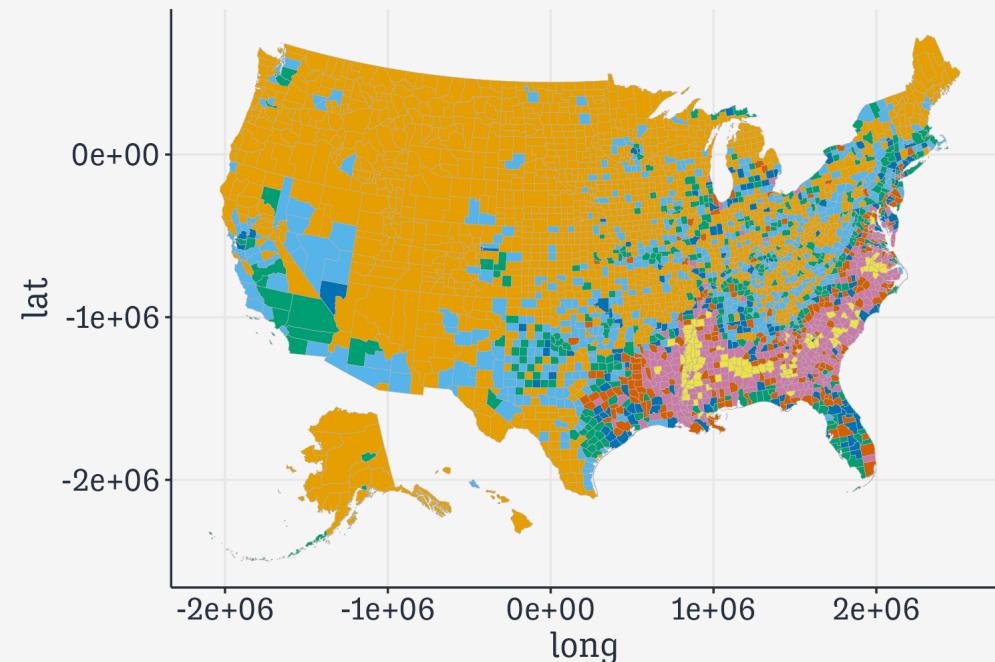


Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
  county_full |>  
    ggplot(mapping = aes(x = long, y = lat,  
                          fill = pct_black,  
                          group = group)) +  
    geom_polygon(color = "gray70",  
                 size = 0.1) +  
    coord_fixed()
```

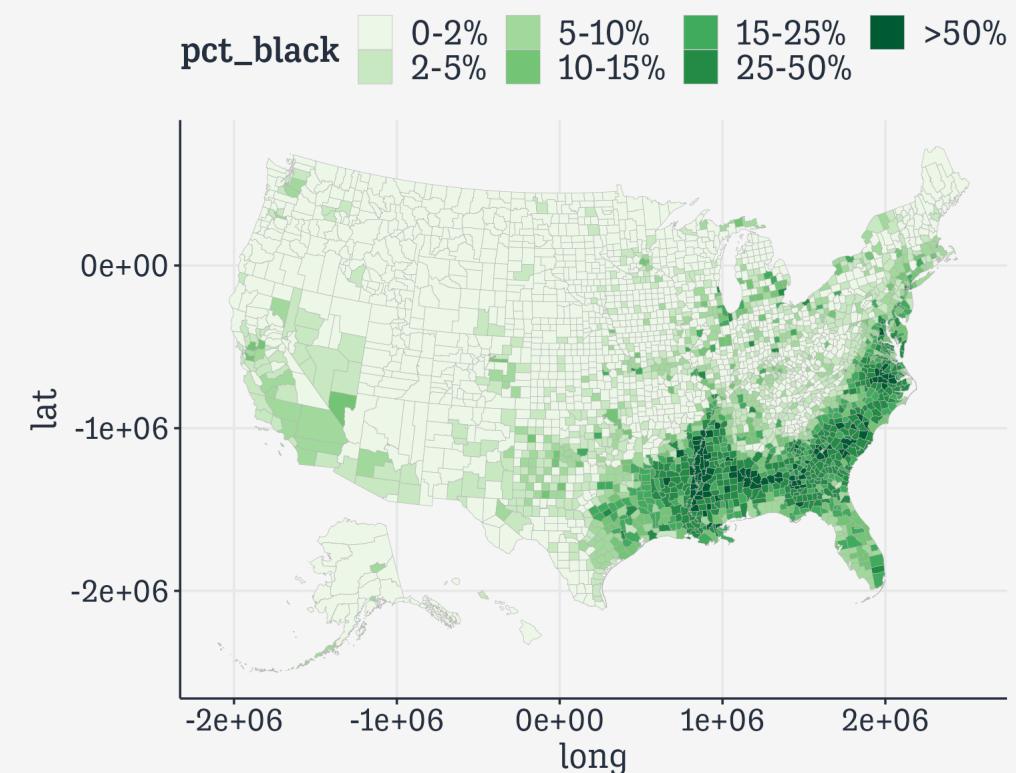
pct_black

[0.0, 2.0)	[5.0,10.0)	[15.0,25.0)	[50.0, 100.0)
[2.0, 5.0)	[10.0,15.0)	[25.0,50.0)	



Same again for Percent Black

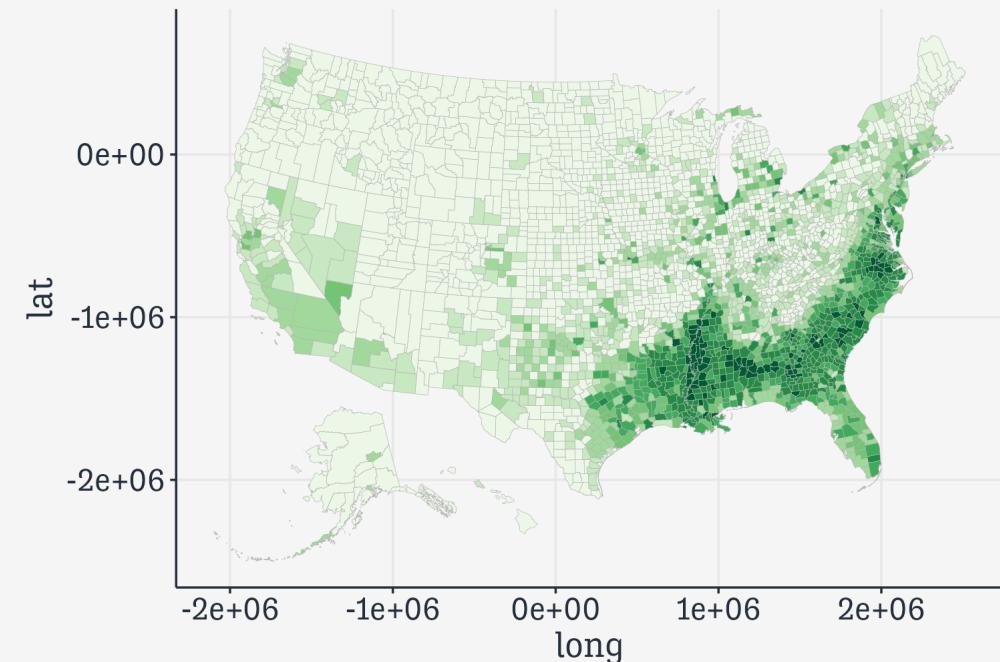
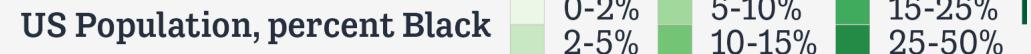
```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pct_black,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Greens",  
                    labels = c("0-2%", "2-5%", "5-  
                            "10-15%", "15-25%",  
                            "25-50%", ">50%"))
```



Same again for Percent Black

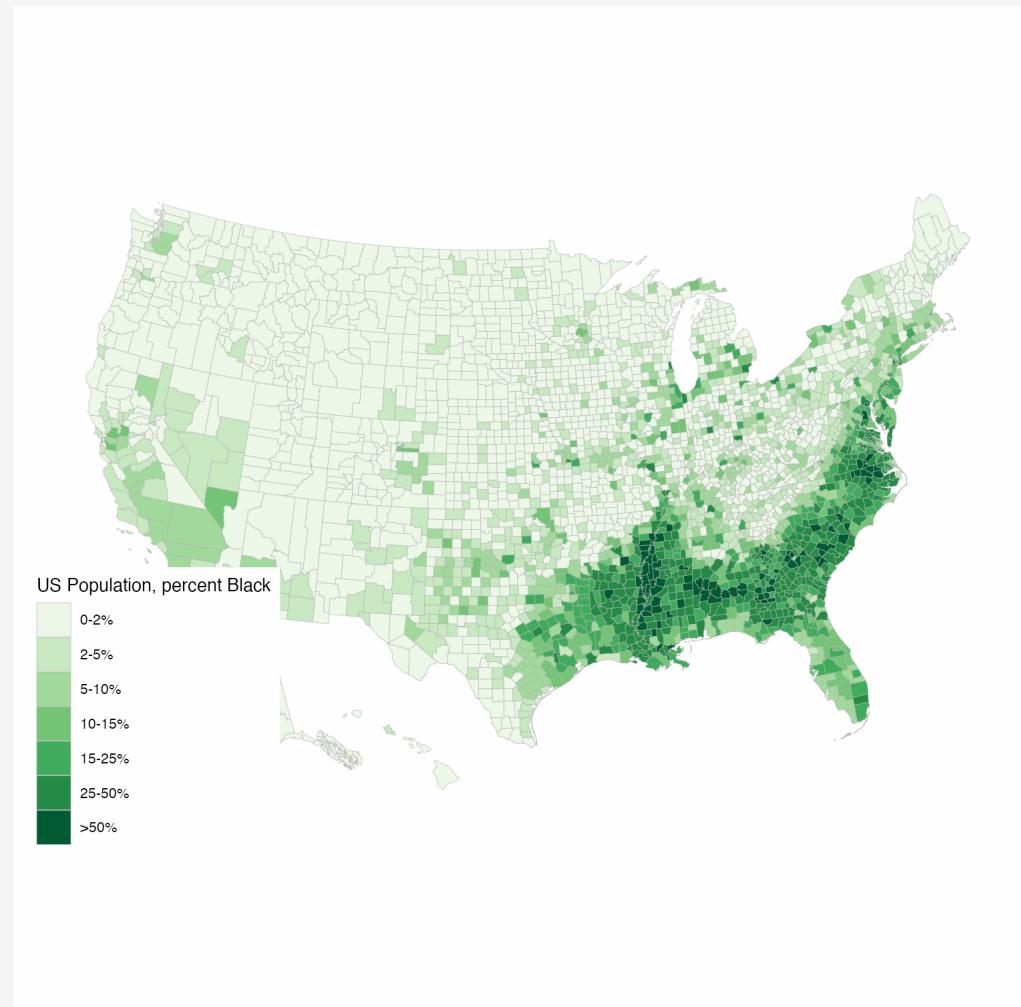
```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                        fill = pct_black,  
                        group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Greens",  
                    labels = c("0-2%", "2-5%", "5-  
                               "10-15%", "15-25%",  
                               "25-50%", ">50%"))  
  
  labs(fill = "US Population, percent Black")
```

US Population, percent Black



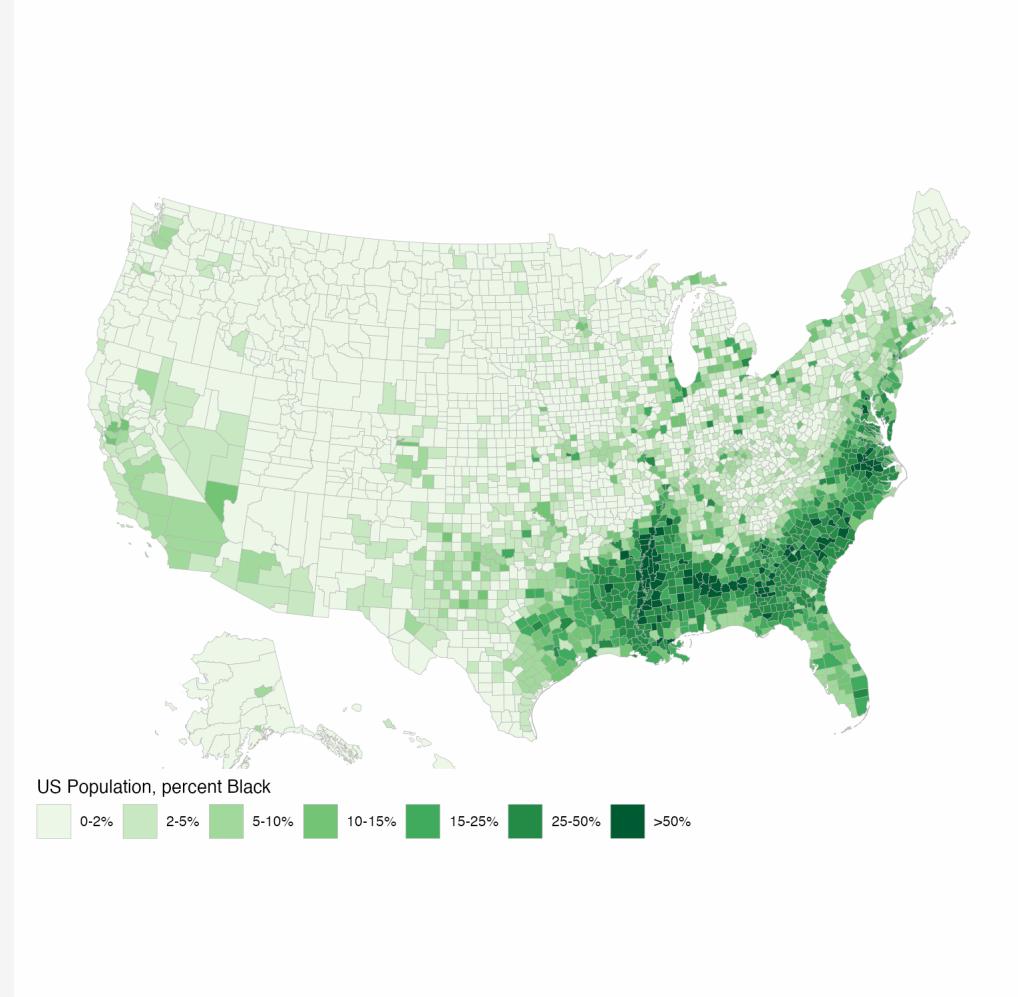
Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pct_black,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Greens",  
                    labels = c("0-2%", "2-5%", "5-  
                               "10-15%", "15-25%",  
                               "25-50%", ">50%"))  
  labs(fill = "US Population, percent Black") +  
  kjhslides::kjh_theme_map()
```



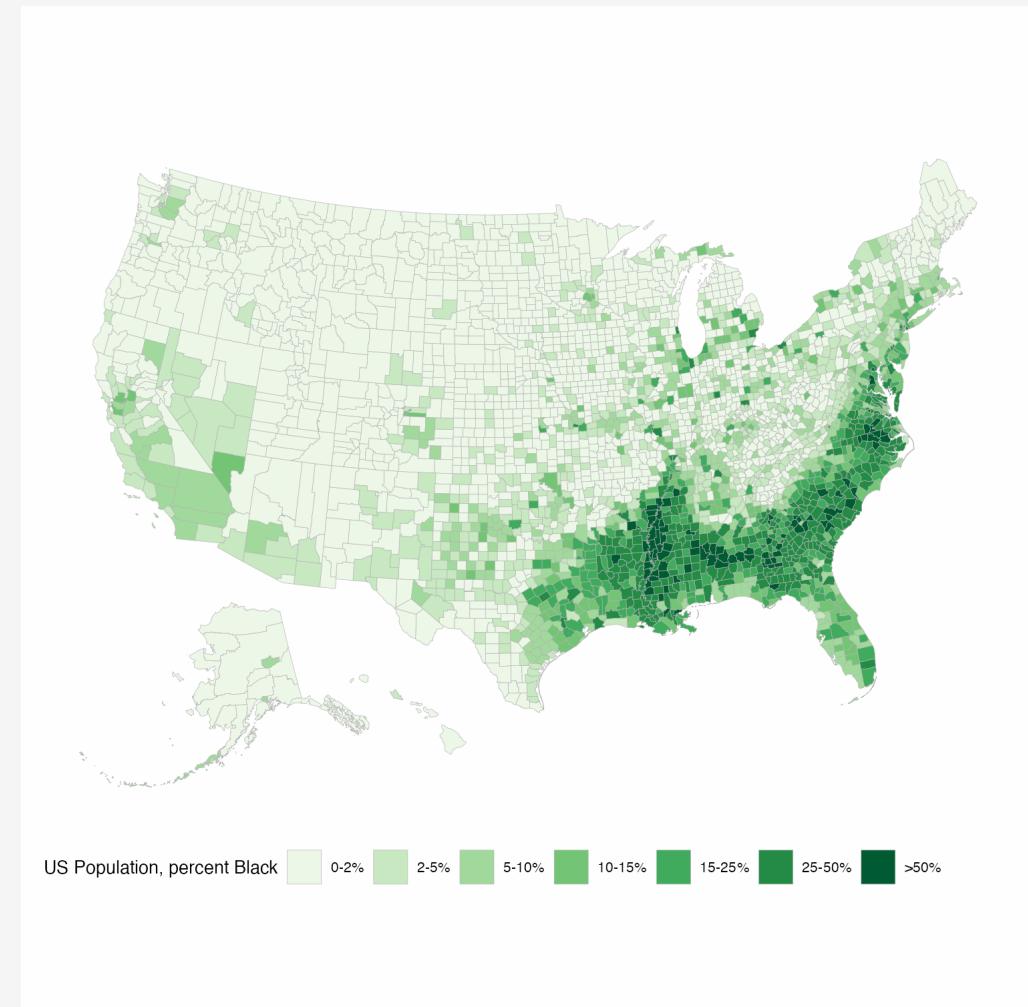
Same again for Percent Black

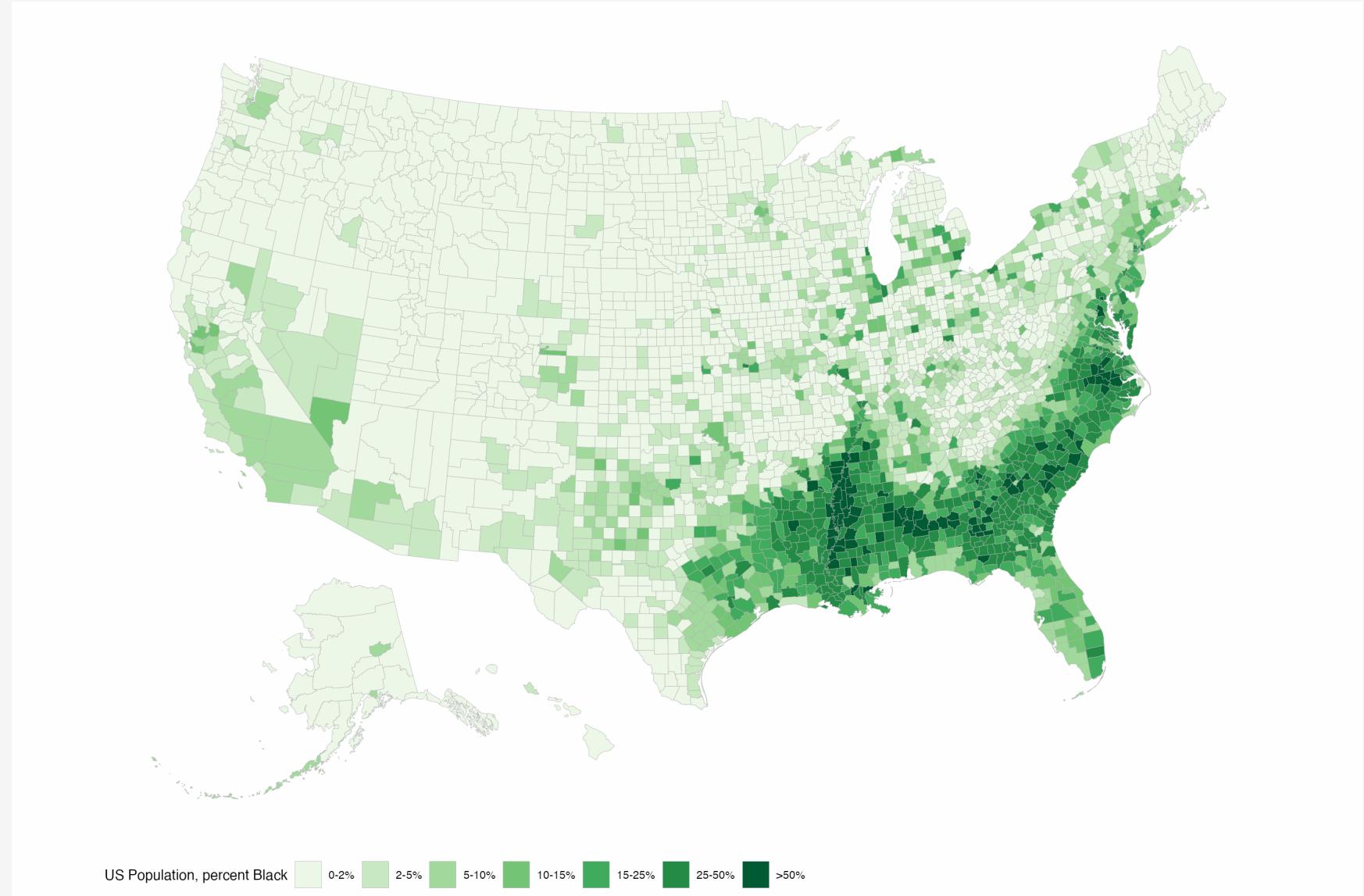
```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pct_black,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Greens",  
                    labels = c("0-2%", "2-5%", "5-  
                               "10-15%", "15-25%",  
                               "25-50%", ">50%"))  
  labs(fill = "US Population, percent Black") +  
  kjhslides::kjh_theme_map() +  
  guides(fill = guide_legend(nrow = 1))
```



Same again for Percent Black

```
county_full <- left_join(county_map, county_data,  
  
county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                      fill = pct_black,  
                      group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_brewer(palette="Greens",  
                    labels = c("0-2%", "2-5%", "5-  
                               "10-15%", "15-25%",  
                               "25-50%", ">50%"))  
  labs(fill = "US Population, percent Black") +  
  kjhslides::kjh_theme_map() +  
  guides(fill = guide_legend(nrow = 1)) +  
  theme(legend.position = "bottom")
```





Percent Black, by County, binned

**Big counties, small
populations, rare events**

Example: Reverse coding

```
orange_pal <- RColorBrewer::brewer.pal(n = 6,  
                                      name = "Oranges")  
orange_pal  
## [1] "#FEEEDDE" "#FDD0A2" "#FDAE6B" "#FD8D3C" "#E6550D" "#A63603"
```



Example: Reverse coding

```
orange_pal <- RColorBrewer::brewer.pal(n = 6,  
                                      name = "Oranges")  
orange_pal
```

```
## [1] "#FEEDDE" "#FDD0A2" "#FDAE6B" "#FD8D3C" "#E6550D" "#A63603"
```



```
# Just reverse it  
orange_rev <- rev(orange_pal)  
orange_rev
```

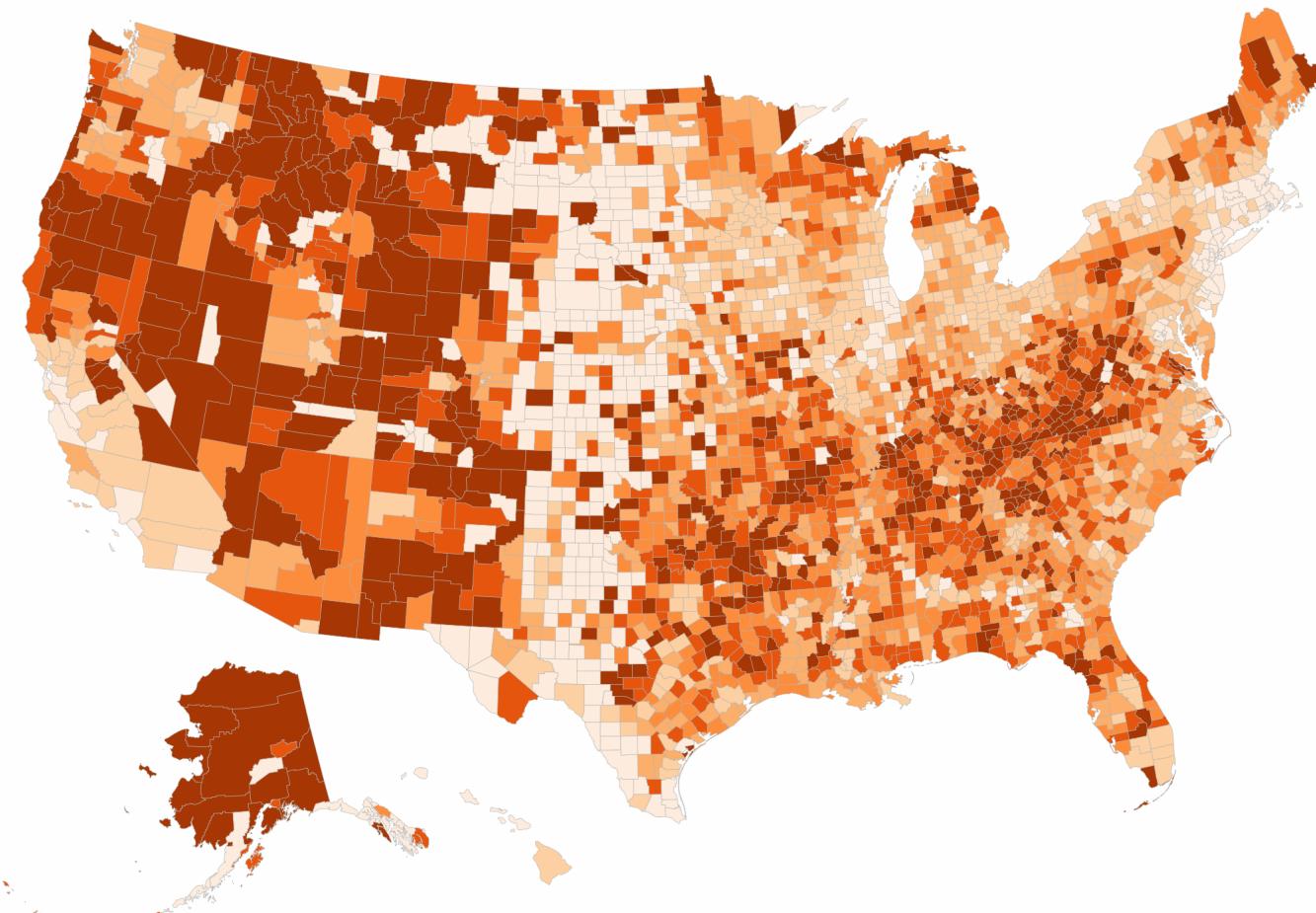
```
## [1] "#A63603" "#E6550D" "#FD8D3C" "#FDAE6B" "#FDD0A2" "#FE
```



Build a plot

```
p_g1 <- county_full |>  
  ggplot(mapping = aes(x = long, y = lat,  
                        fill = su_gun6,  
                        group = group)) +  
  geom_polygon(color = "gray70",  
               size = 0.1) +  
  coord_fixed() +  
  scale_fill_manual(values = orange_pal) +  
  labs(title = "Gun-Related Suicides, 1999-2015",  
       fill = "Rate per 100,000 pop.") +  
  theme_map() +  
  guides(fill = guide_legend(nrow = 1)) +  
  theme(legend.position = "bottom")
```

Gun-Related Suicides, 1999-2015

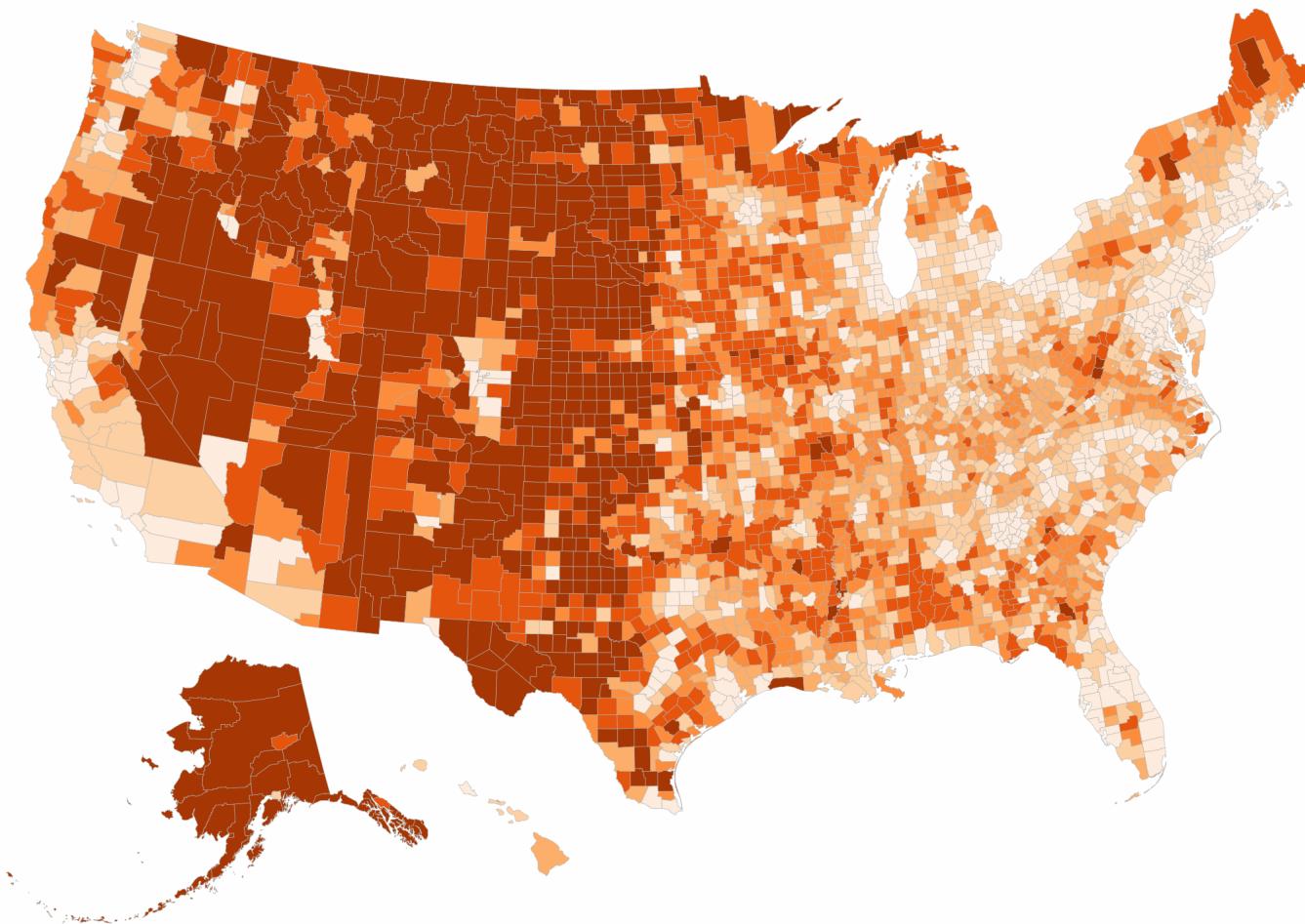


Regular palette

And another

```
p_g2 <- county_full |>
  ggplot(mapping = aes(x = long, y = lat,
                        fill = pop_dens6,
                        group = group)) +
  geom_polygon(color = "gray70",
               size = 0.1) +
  coord_fixed() +
  scale_fill_manual(values = orange_rev) +
  labs(title = "Reverse-coded Population Density",
       fill = "Persons per square mile") +
  theme_map() +
  guides(fill = guide_legend(nrow = 1)) +
  theme(legend.position = "bottom")
```

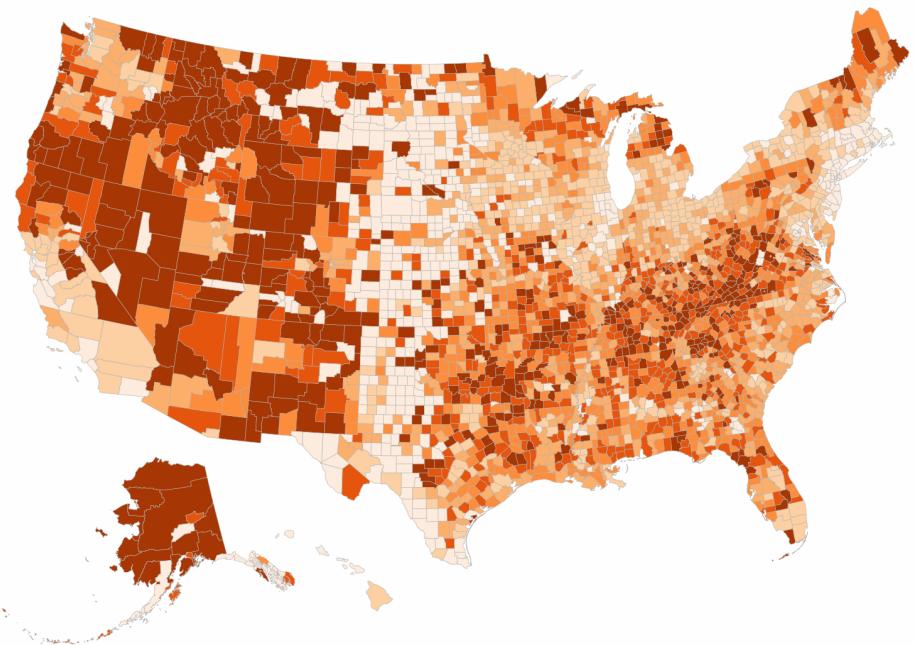
Reverse-coded Population Density



Persons per square mile [0, 9] [9, 25] [25, 45] [45, 82] [82, 215] [215, 71672]

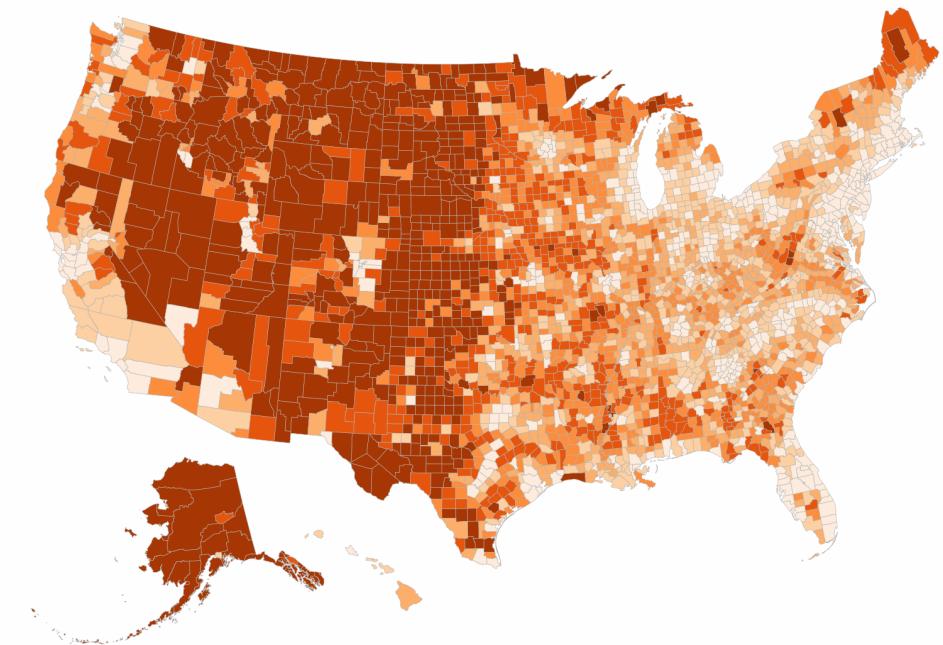
Reverse-coded density

Gun-Related Suicides, 1999-2015



Rate per 100,000 pop. [0, 4] [4, 7] [7, 8] [8, 10] [10, 12] [12, 54]

Reverse-coded Population Density



Persons per square mile [0, 9] [9, 25] [25, 45] [45, 82] [82, 215] [215, 71672]

Comparing the plots

Small multiples for maps

Opiate-related Mortality, 1999-2014

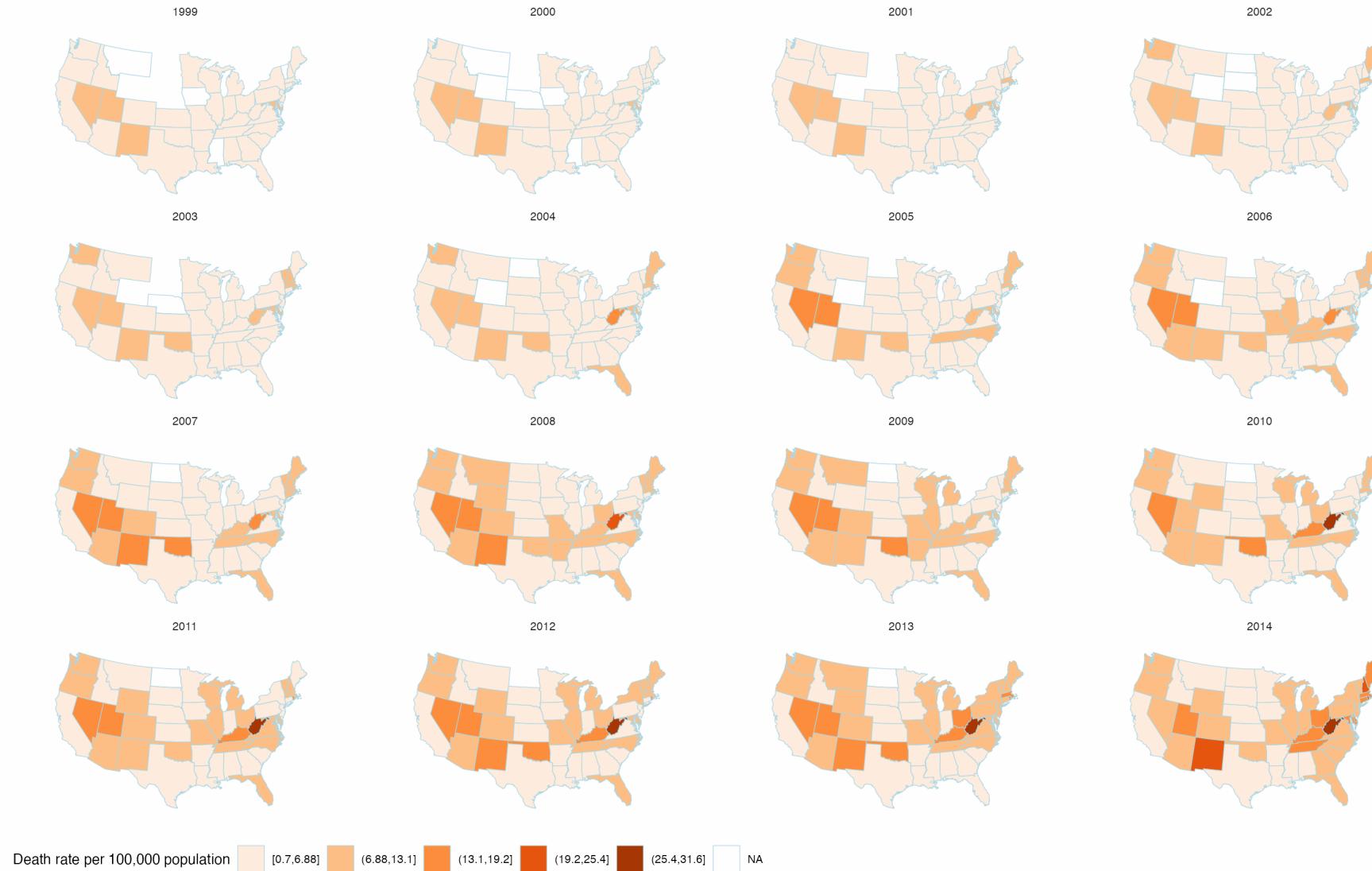
opiates

```
## # A tibble: 800 × 11
##   year state    fips deaths population crude adjusted adjusted_se region abbr
##   <int> <chr>   <int>   <int>      <dbl>    <dbl>     <dbl> <ord>   <chr>
## 1 1999 Alabama     1     37   4430141     0.8      0.8      0.1 South   AL
## 2 1999 Alaska      2     27   624779      4.3       4       0.8 West    AK
## 3 1999 Arizona     4    229   5023823     4.6      4.7      0.3 West    AZ
## 4 1999 Arkans...    5     28   2651860     1.1      1.1      0.2 South   AR
## 5 1999 Califo...    6    1474  33499204     4.4      4.5      0.1 West    CA
## 6 1999 Colora...    8    164   4226018     3.9      3.7      0.3 West    CO
## 7 1999 Connec...    9    151   3386401     4.5      4.4      0.4 North... CT
## 8 1999 Delawa...   10     32   774990      4.1      4.1      0.7 South   DE
## 9 1999 Distri...   11     28   570213      4.9      4.9      0.9 South   DC
## 10 1999 Florida    12    402  15759421     2.6      2.6      0.1 South   FL
## # ... with 790 more rows, and 1 more variable: division_name <chr>
```

```
opiates$state <- tolower(opiates$state)
us_states$state <- us_states$region
opiates_map <- left_join(us_states, opiates, by = "state")
```

```
p_out <- opiates_map |>
  ggplot(mapping = aes(x = long, y = lat,
                        group = group,
                        fill = cut_interval(adjusted, n = 5))) +
  geom_polygon(color = "lightblue", size = 0.2) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45) +
  scale_fill_brewer(type = "seq", palette = "Oranges") +
  kjhslides::kjh_theme_map() +
  facet_wrap(~ year, ncol = 4) +
  guides(fill = guide_legend(nrow = 1)) +
  theme(legend.position = "bottom",
        strip.background = element_blank()) +
  labs(fill = "Death rate per 100,000 population",
       title = "Opiate-Related Deaths by State, 1999-2014")
```

Opiate-Related Deaths by State, 1999-2014



Faceting works just as it would for any other kind of plot.

**Is your data
really spatial?**

The two leading states in each region in 2014

```
## Put this in an object called `st_top`  
opiates  
## # A tibble: 800 × 11  
##   year state    fips deaths population crude adjusted adjusted_se region a  
##   <int> <chr>    <int>  <int>      <int> <dbl>    <dbl>    <dbl>    <dbl> <ord> <  
## 1 1999 alabama     1     37    4430141    0.8     0.8     0.1 South A  
## 2 1999 alaska      2     27    624779     4.3     4       0.8 West A  
## 3 1999 arizona     4    229    5023823     4.6     4.7     0.3 West A  
## 4 1999 arkans...   5     28    2651860     1.1     1.1     0.2 South A  
## 5 1999 califo...   6    1474    33499204     4.4     4.5     0.1 West A  
## 6 1999 colora...   8     164    4226018     3.9     3.7     0.3 West A  
## 7 1999 connec...   9     151    3386401     4.5     4.4     0.4 North... A  
## 8 1999 delawa...  10     32    774990     4.1     4.1     0.7 South D  
## 9 1999 distri...  11     28    570213     4.9     4.9     0.9 South D  
## 10 1999 florida    12    402   15759421     2.6     2.6     0.1 South F  
## # ... with 790 more rows, and 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
## Put this in an object called `st_top`  
opiates |>  
  filter(year == max(year),  
         abbr != "DC")  
  
## # A tibble: 50 × 11  
##   year state    fips deaths population crude adjusted adjusted_se region a  
##   <int> <chr>    <int>  <int>      <int> <dbl>    <dbl>    <dbl> <dbl> <ord> <br>  
##   1 2014 alabama     1    270    4849377    5.6     5.6     0.3 South A  
##   2 2014 alaska      2     76    736732    10.3    10.6    1.2 West A  
##   3 2014 arizona     4    589    6731484    8.7     8.8    0.4 West A  
##   4 2014 arkans...    5    173    2966369    5.8     6.3    0.5 South A  
##   5 2014 califo...    6   2024    38802500    5.2     5     0.1 West O  
##   6 2014 colora...    8    517    5355866    9.7     9.4    0.4 West O  
##   7 2014 connec...    9    525    3596677   14.6    15.2    0.7 North... O  
##   8 2014 delawa...   10   124    935614   13.3    13.9    1.3 South D  
##   9 2014 florida    12  1399   19893297     7     7.2    0.2 South F  
##  10 2014 georgia    13   710   10097343     7     7    0.3 South S  
## # ... with 40 more rows, and 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
## Put this in an object called `st_top`  
opiates |>  
  filter(year == max(year),  
         abbr != "DC") |>  
  group_by(region)  
  
## # A tibble: 50 × 11  
## # Groups:   region [4]  
##       year state    fips deaths population crude adjusted adjusted_se region  
##       <int> <chr>   <int>  <int>     <dbl>   <dbl>   <dbl>   <dbl> <ord>  <chr>  
## 1 2014 alabama 1 270 4849377 5.6 5.6 0.3 South  
## 2 2014 alaska 2 76 736732 10.3 10.6 1.2 West  
## 3 2014 arizona 4 589 6731484 8.7 8.8 0.4 West  
## 4 2014 arkans... 5 173 2966369 5.8 6.3 0.5 South  
## 5 2014 califo... 6 2024 38802500 5.2 5 0.1 West  
## 6 2014 colora... 8 517 5355866 9.7 9.4 0.4 West  
## 7 2014 connec... 9 525 3596677 14.6 15.2 0.7 North...  
## 8 2014 delawa... 10 124 935614 13.3 13.9 1.3 South  
## 9 2014 florida 12 1399 19893297 7 7.2 0.2 South  
## 10 2014 georgia 13 710 10097343 7 7 0.3 South  
## # ... with 40 more rows, and 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
## Put this in an object called `st_top`  
opiates |>  
  filter(year == max(year),  
         abbr != "DC") |>  
  group_by(region) |>  
  slice_max(order_by = adjusted,  
            n = 2)  
  
## # A tibble: 8 × 11  
## # Groups:   region [4]  
##       year state     fips deaths population crude adjusted adjusted_se region  
##       <int> <chr>    <int>  <int>      <dbl>   <dbl>    <dbl>    <dbl> <ord> <chr>  
## 1 2014 new ham...     33    297    1326813  22.4    23.4    1.4 North... N  
## 2 2014 rhode i...     44    205    1055173  19.4    19.8    1.4 North... R  
## 3 2014 ohio          39    2106   11594163  18.2    19.1    0.4 Midwe... O  
## 4 2014 missouri      29    696    6063589  11.5     12     0.5 Midwe... M  
## 5 2014 new mex...     35    402    2085572  19.3    20.2     1 West    W  
## 6 2014 utah           49    455    2942902  15.5    16.8    0.8 West    U  
## 7 2014 west vi...     54    554    1850326  29.9    31.6    1.4 South   S  
## 8 2014 kentucky       21    729    4413457  16.5    16.8    0.6 South   K  
## # ... with 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
## Put this in an object called `st_top`  
opiates |>  
  filter(year == max(year),  
         abbr != "DC") |>  
  group_by(region) |>  
  slice_max(order_by = adjusted,  
            n = 2)  
  
## # A tibble: 8 × 11  
## # Groups:   region [4]  
##       year state      fips deaths population crude adjusted adjusted_se region  
##       <int> <chr>    <int>  <int>     <dbl>    <dbl>    <dbl>    <dbl>    <ord>  <chr>  
## 1 2014 new ham...     33    297  1326813  22.4    23.4    1.4 North...  
## 2 2014 rhode i...     44    205  1055173  19.4    19.8    1.4 North...  
## 3 2014 ohio          39   2106  11594163  18.2    19.1    0.4 Midwe...  
## 4 2014 missouri      29    696  6063589  11.5     12     0.5 Midwe...  
## 5 2014 new mex...     35    402  2085572  19.3    20.2     1 West  
## 6 2014 utah           49    455  2942902  15.5    16.8    0.8 West  
## 7 2014 west vi...     54    554  1850326  29.9    31.6    1.4 South  
## 8 2014 kentucky       21    729  4413457  16.5    16.8    0.6 South  
## # ... with 1 more variable: division_name <chr>
```

Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at
```

Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region))
```

Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)
```

Opiates Time Series plot

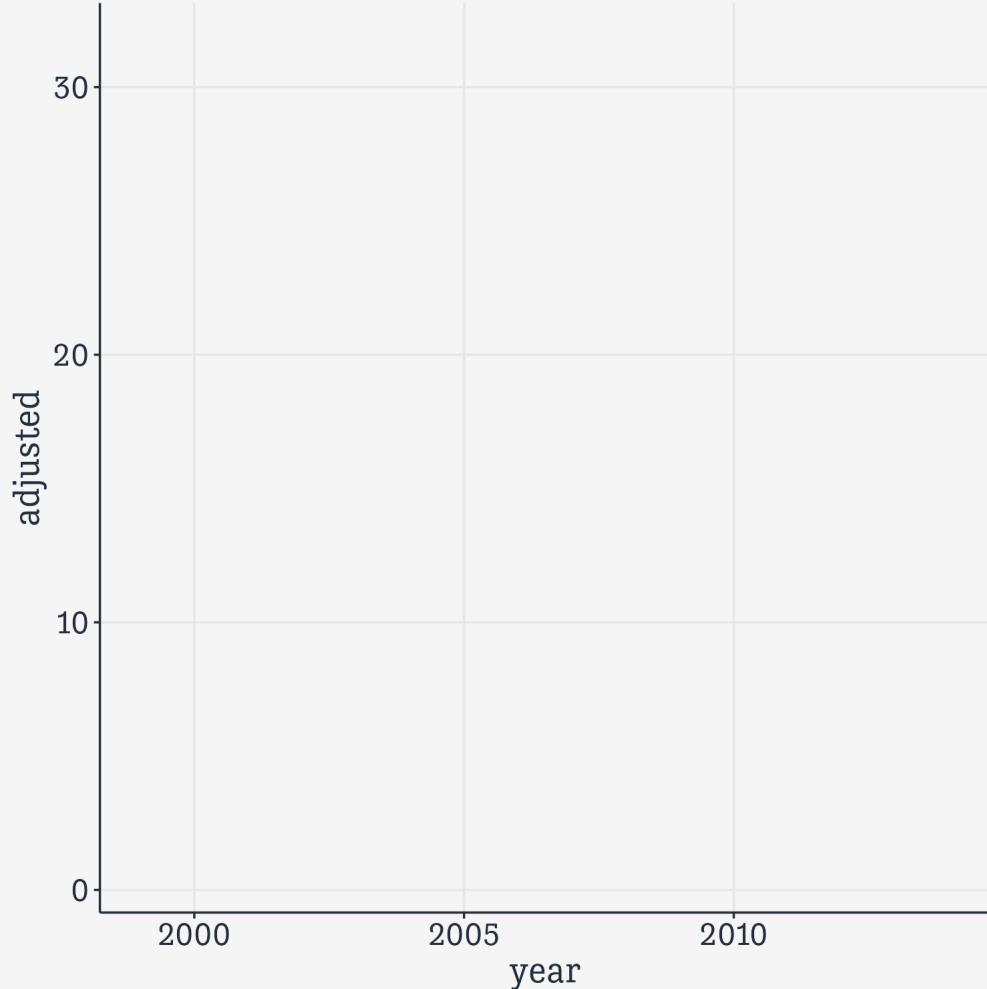
```

st_top <- opiates |> filter(year == max(year), at
  group_by(region) |>
  slice_max(order_by = adjusted, n = 2)
## # A tibble: 800 × 11
##   year state    fips deaths population crude adjusted adjusted_se region
##   <int> <chr>    <int>  <int>      <dbl>    <dbl>    <dbl>    <dbl> <ord>
## 1 1999 alabama 1 37 4430141 0.8 0.8 0.1 South
## 2 1999 alaska 2 27 624779 4.3 4 0.8 West
## 3 1999 arizona 4 229 5023823 4.6 4.7 0.3 West
## 4 1999 arkans... 5 28 2651860 1.1 1.1 0.2 South
## 5 1999 califo... 6 1474 33499204 4.4 4.5 0.1 West
## 6 1999 colora... 8 164 4226018 3.9 3.7 0.3 West
## 7 1999 connec... 9 151 3386401 4.5 4.4 0.4 North...
## 8 1999 delawa... 10 32 774990 4.1 4.1 0.7 South
## 9 1999 distri... 11 28 570213 4.9 4.9 0.9 South
## 10 1999 florida 12 402 15759421 2.6 2.6 0.1 South
## # ... with 790 more rows, and 1 more variable: division_name <chr>

```

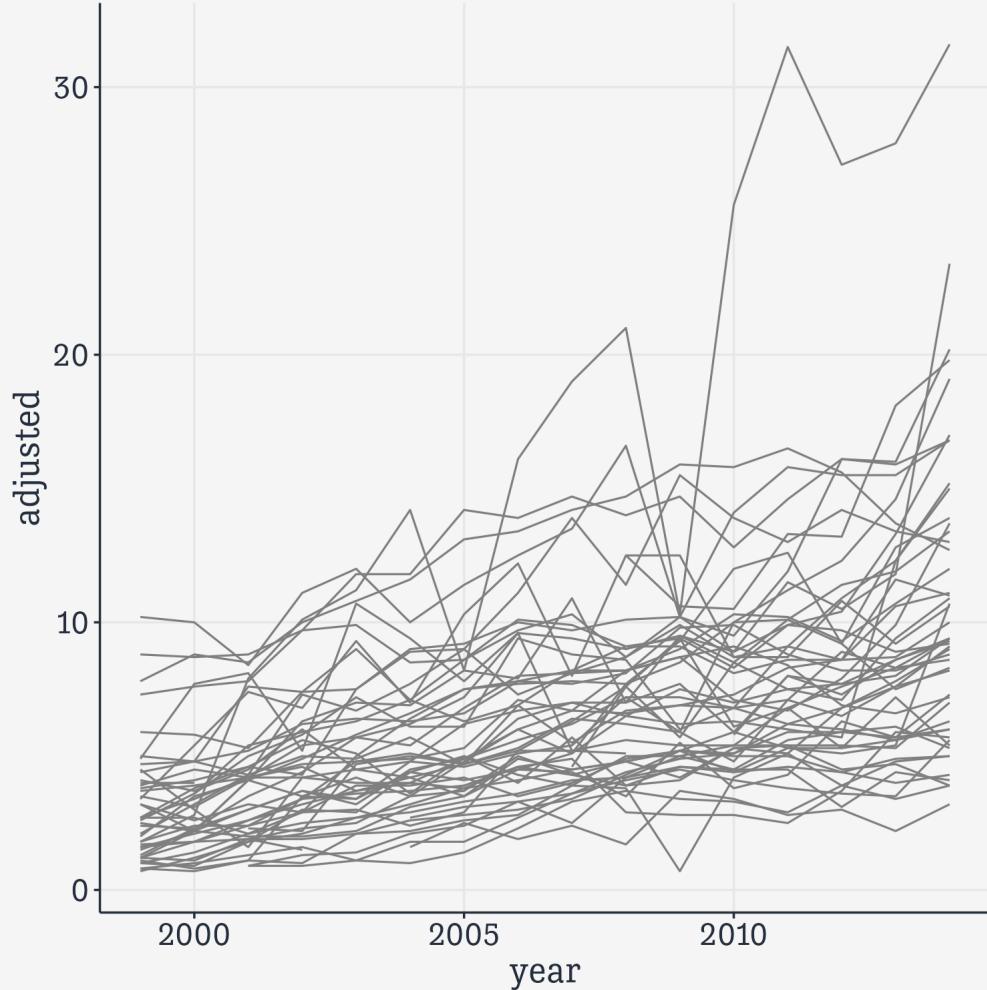
Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted))
```



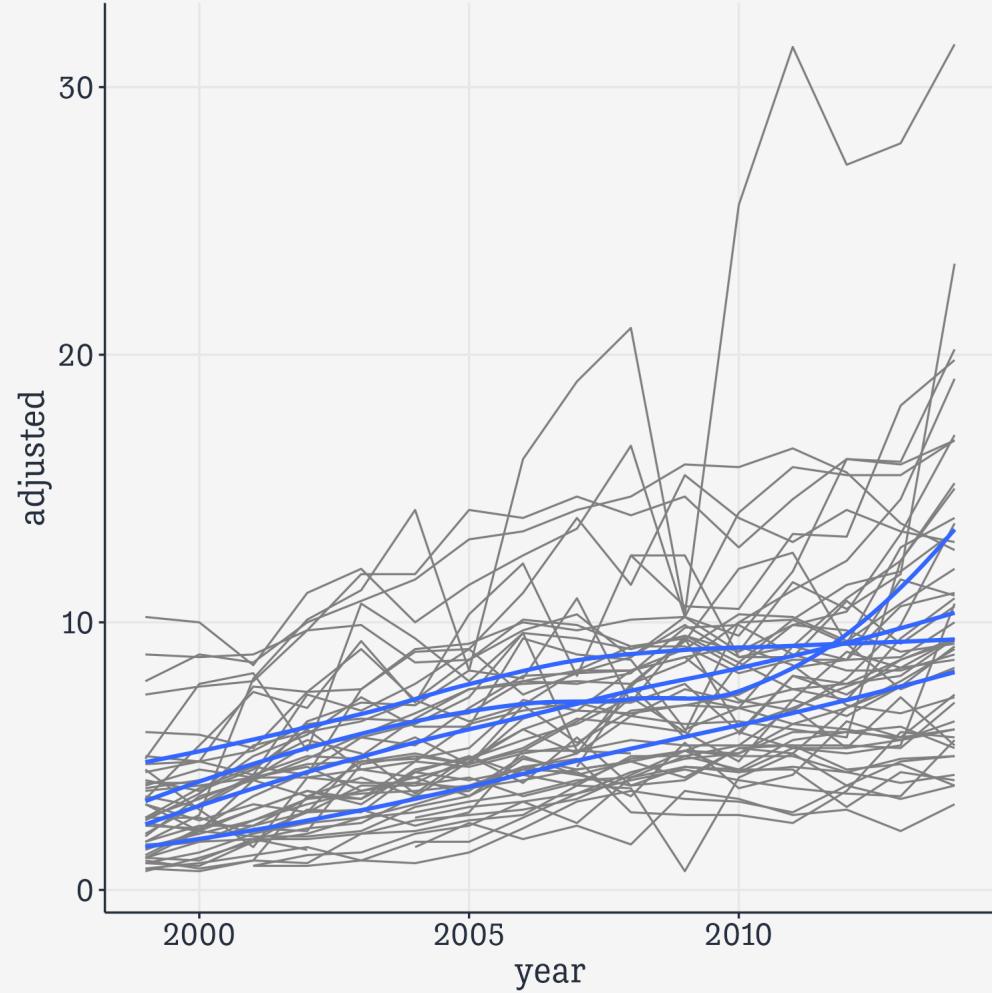
Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50")
```



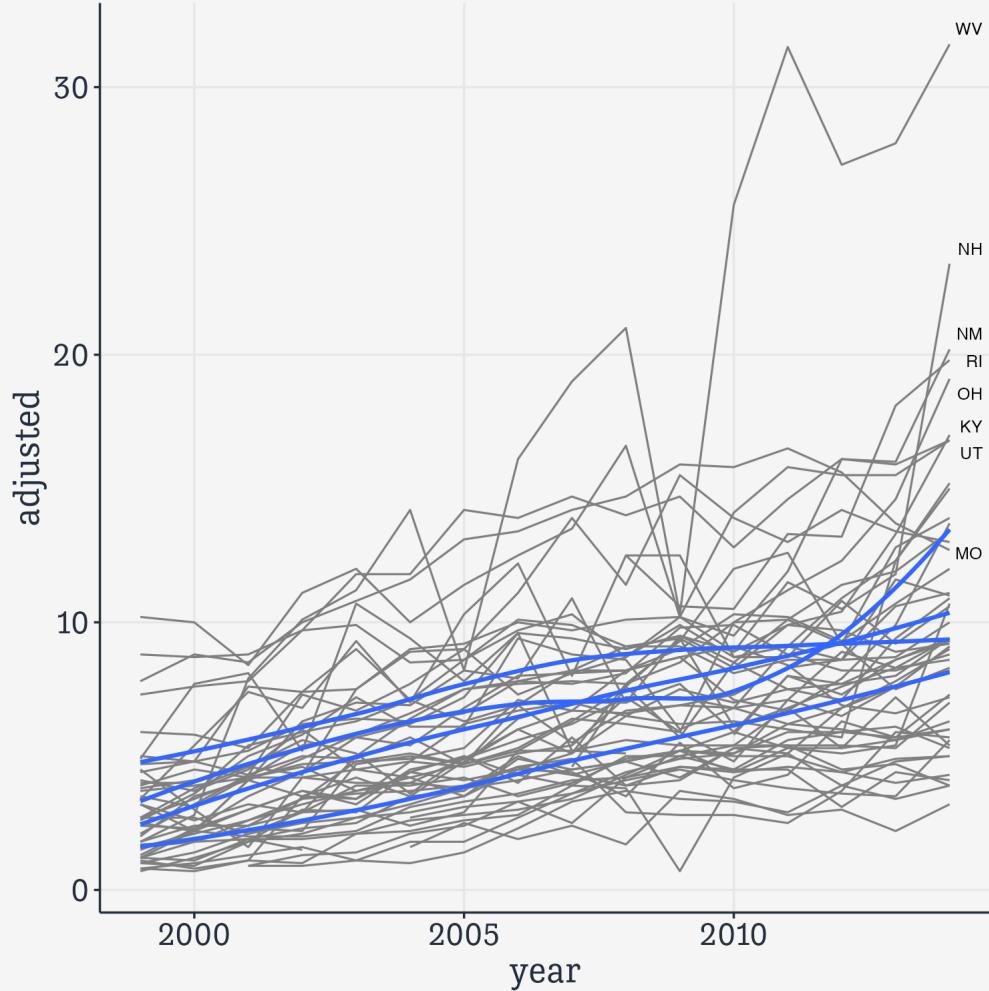
Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE)
```



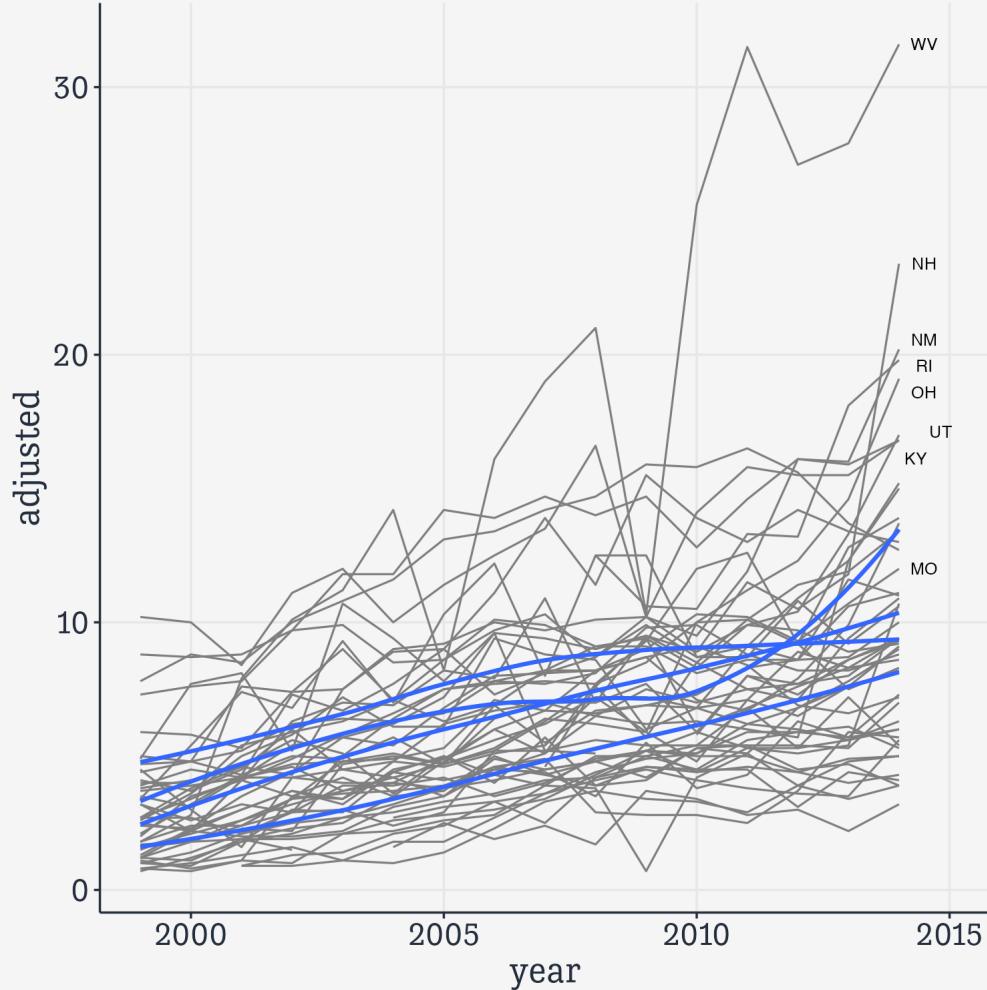
Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE) +  
  ggrepel::geom_text_repel(  
    data = st_top,  
    mapping = aes(x = year,  
                  y = adjusted,  
                  label = abbr),  
    size = 3,  
    segment.color = NA,  
    nudge_x = 0.5)
```



Opiates Time Series plot

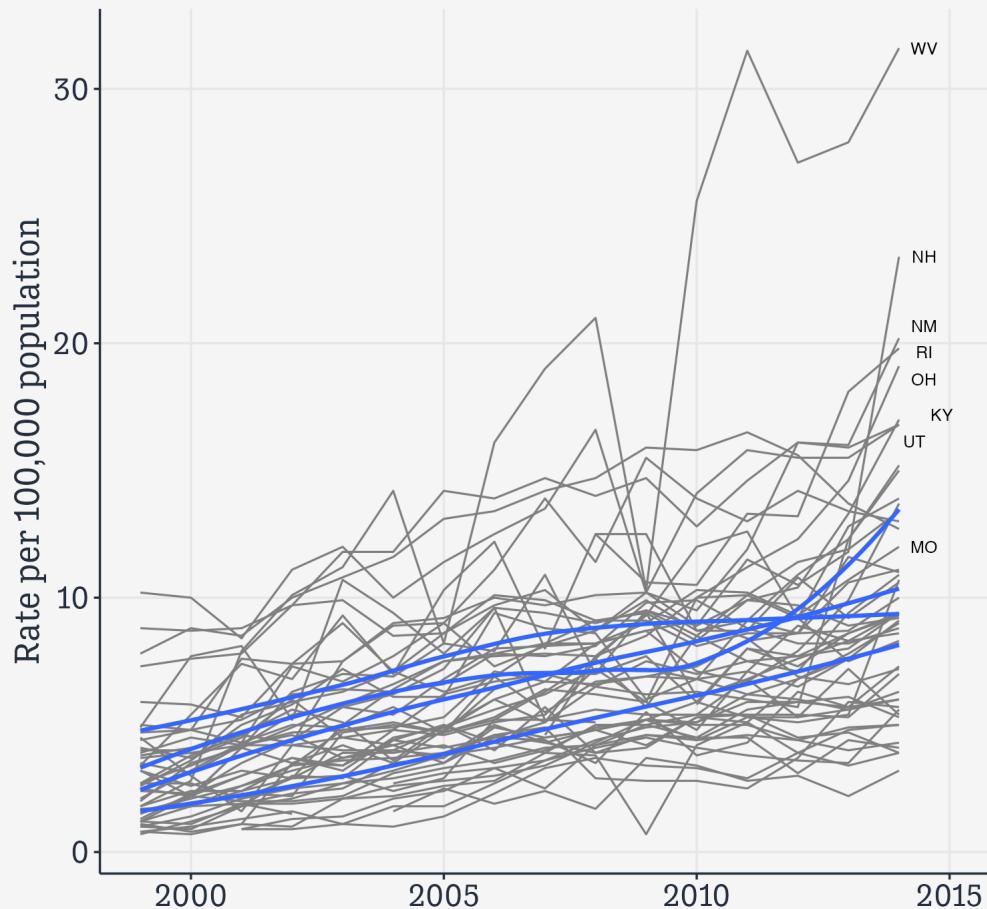
```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE) +  
  ggrepel::geom_text_repel(  
    data = st_top,  
    mapping = aes(x = year,  
                  y = adjusted,  
                  label = abbr),  
    size = 3,  
    segment.color = NA,  
    nudge_x = 0.5) +  
  coord_cartesian(c(min(opiates$year),  
                  max(opiates$year) + 1))
```



Opiates Time Series plot

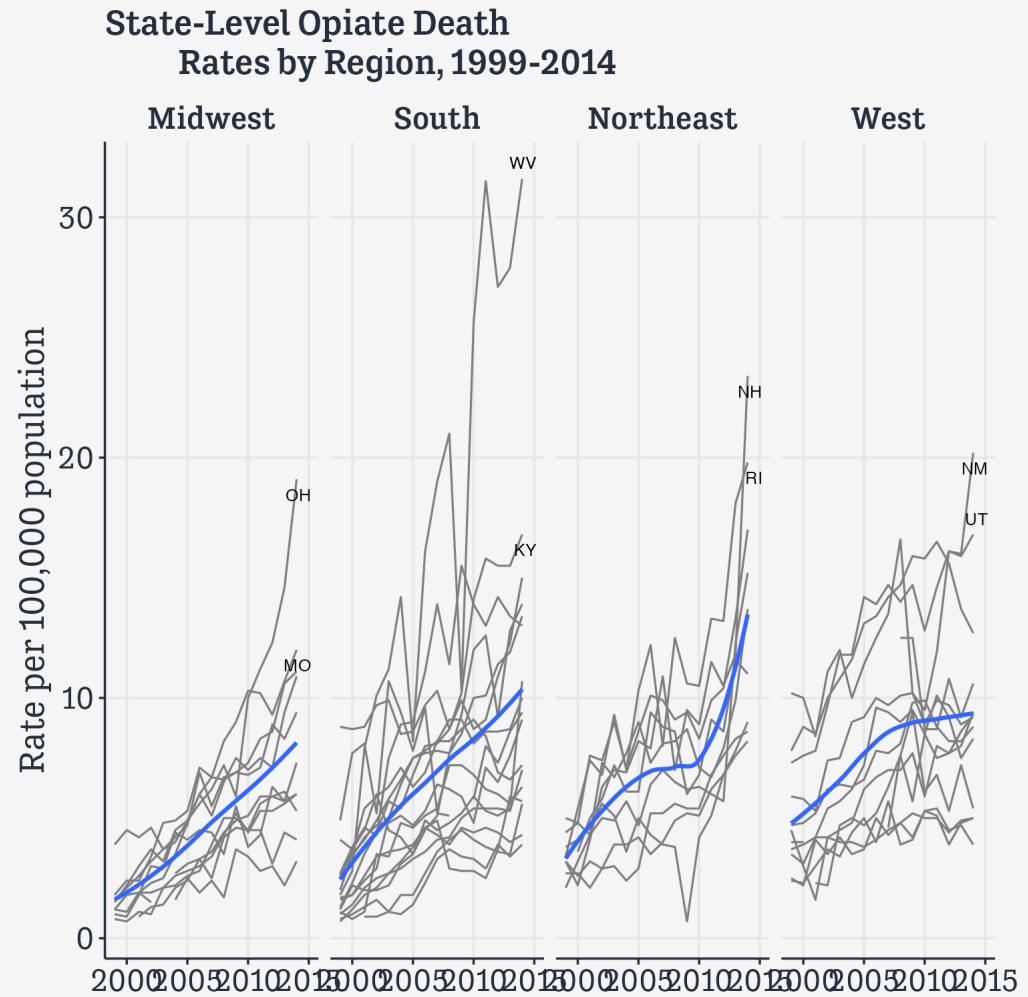
```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE) +  
  ggrepel::geom_text_repel(  
    data = st_top,  
    mapping = aes(x = year,  
                  y = adjusted,  
                  label = abbr),  
    size = 3,  
    segment.color = NA,  
    nudge_x = 0.5) +  
  coord_cartesian(c(min(opiates$year),  
                  max(opiates$year) + 1)) +  
  labs(x = NULL,  
       y = "Rate per 100,000 population",  
       title = "State-Level Opiate Death  
                 Rates by Region, 1999-2014")
```

State-Level Opiate Death #<<
Rates by Region, 1999-2014



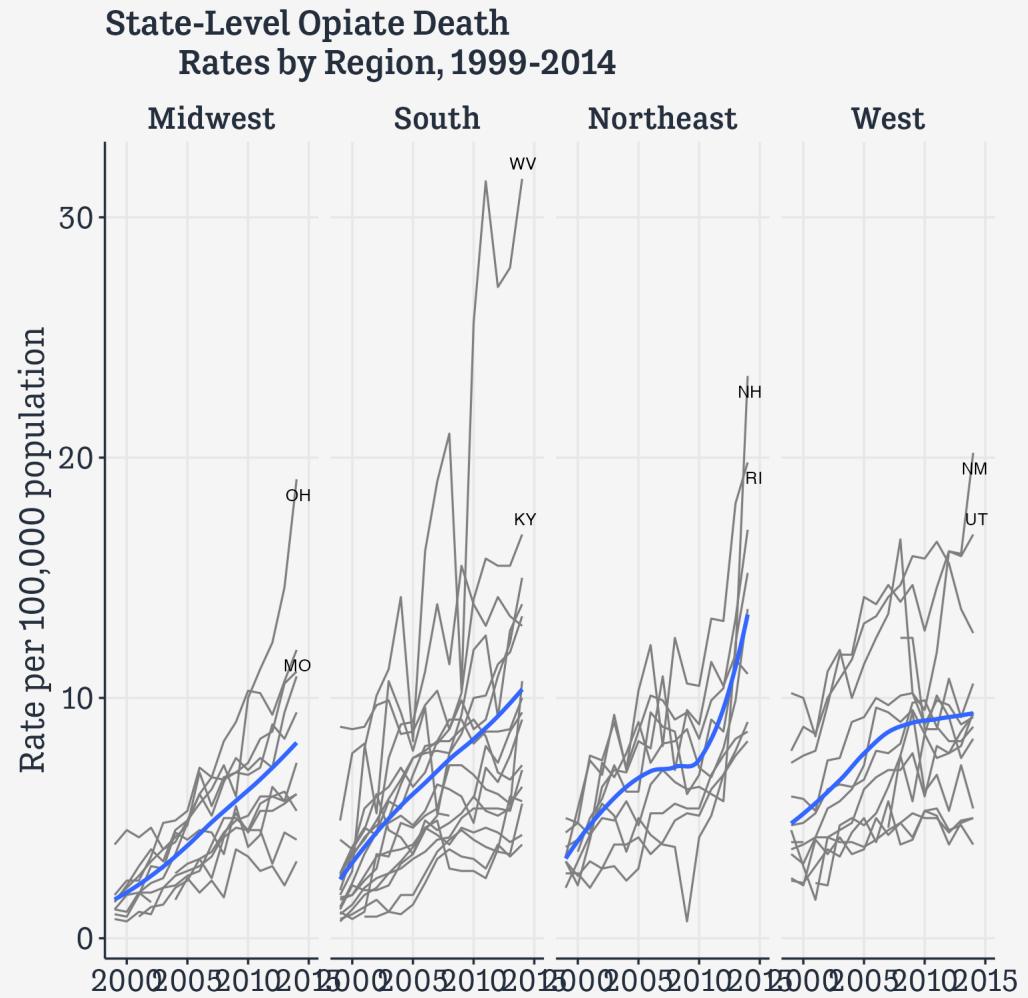
Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), at  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE) +  
  ggrepel::geom_text_repel(  
    data = st_top,  
    mapping = aes(x = year,  
                  y = adjusted,  
                  label = abbr),  
    size = 3,  
    segment.color = NA,  
    nudge_x = 0.5) +  
  coord_cartesian(c(min(opiates$year),  
                  max(opiates$year) + 1)) +  
  labs(x = NULL,  
       y = "Rate per 100,000 population",  
       title = "State-Level Opiate Death  
       Rates by Region, 1999-2014") +  
  facet_wrap(~ reorder(region, adjusted,  
                      na.rm = TRUE),  
            nrow = 1)
```

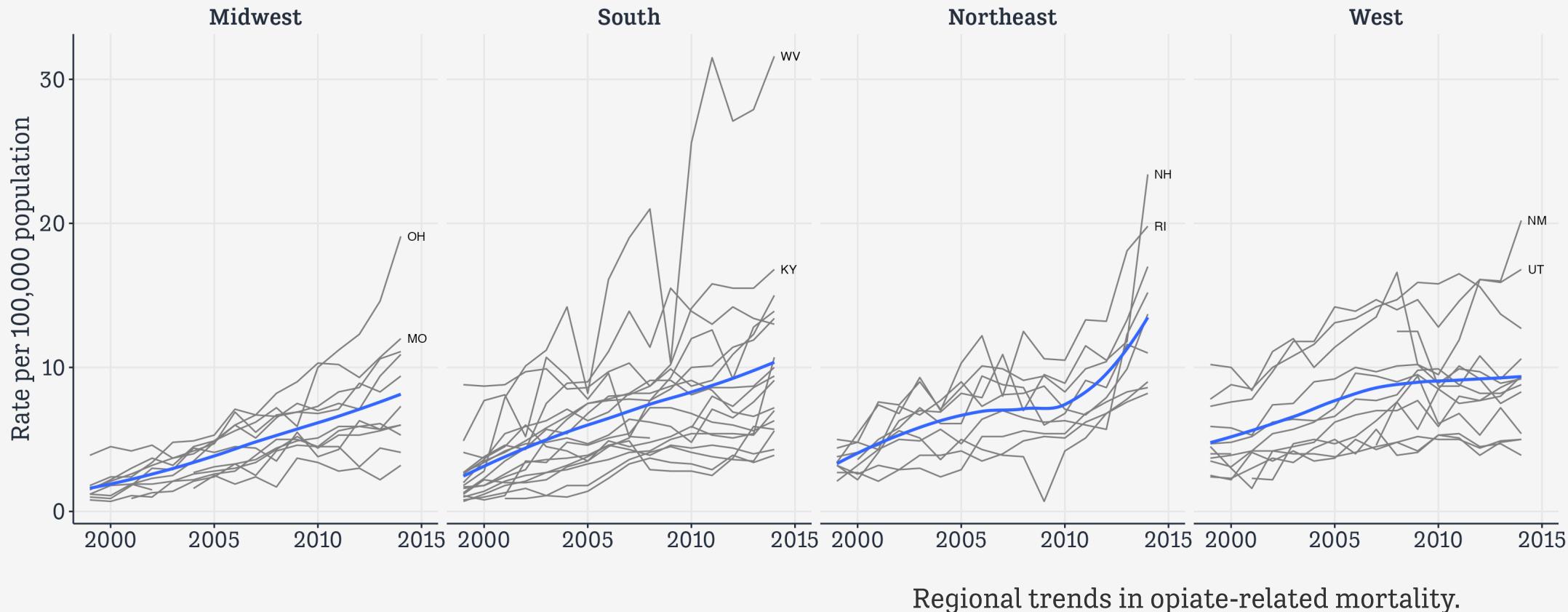


Opiates Time Series plot

```
st_top <- opiates |> filter(year == max(year), !  
group_by(region) |>  
slice_max(order_by = adjusted, n = 2)  
  
opiates |>  
  ggplot(aes(x = year,  
             y = adjusted)) +  
  geom_line(aes(group = state),  
            color = "gray50") +  
  geom_smooth(aes(group = region),  
              se = FALSE) +  
  ggrepel::geom_text_repel(  
    data = st_top,  
    mapping = aes(x = year,  
                  y = adjusted,  
                  label = abbr),  
    size = 3,  
    segment.color = NA,  
    nudge_x = 0.5) +  
  coord_cartesian(c(min(opiates$year),  
                  max(opiates$year) + 1)) +  
  labs(x = NULL,  
       y = "Rate per 100,000 population",  
       title = "State-Level Opiate Death  
          Rates by Region, 1999-2014") +  
  facet_wrap(~ reorder(region, adjusted,  
                      na.rm = TRUE),  
            nrow = 1)
```



State-Level Opiate Death Rates by Region, 1999-2014



Simple Features

`geom_polygon()` is limiting

It's very useful to have the intuition that, when drawing maps, we're just working with tables of x and y coordinates, and shapes represent quantities in our data, in a way that's essentially the same as any other geom. This makes it worth getting comfortable with what `geom_polygon()` and `coord_map()` are doing. But the business of having very large map tables and manually specifying projections is inefficient.

`geom_polygon()` is limiting

It's very useful to have the intuition that, when drawing maps, we're just working with tables of x and y coordinates, and shapes represent quantities in our data, in a way that's essentially the same as any other geom. This makes it worth getting comfortable with what `geom_polygon()` and `coord_map()` are doing. But the business of having very large map tables and manually specifying projections is inefficient.

In addition, sometimes our data *really is* properly spatial, at which point we need a more rigorous and consistent way of specifying those elements. There's a whole world of Geodesic standards and methods devoted to specifying these things for GIS applications. R is not a dedicated GIS, but we can take advantage of these tools.

`geom_polygon()` is limiting

It's very useful to have the intuition that, when drawing maps, we're just working with tables of x and y coordinates, and shapes represent quantities in our data, in a way that's essentially the same as any other geom. This makes it worth getting comfortable with what `geom_polygon()` and `coord_map()` are doing. But the business of having very large map tables and manually specifying projections is inefficient.

In addition, sometimes our data *really is* properly spatial, at which point we need a more rigorous and consistent way of specifying those elements. There's a whole world of Geodesic standards and methods devoted to specifying these things for GIS applications. R is not a dedicated GIS, but we can take advantage of these tools.

Enter simple features, the `sf` package, and `geom_sf()`

The Simple Features package

When we load `sf` it creates a way to use several standard GIS concepts and tools, such as the `GEOS` library for computational geometry, the `PROJ` software that transforms spatial coordinates from one reference system to another, as in map projections, and the Simple Features standard for specifying the elements of spatial attributes.

```
library(sf)  
## Linking to GEOS 3.9.1, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

The Simple Features package

When we load `sf` it creates a way to use several standard GIS concepts and tools, such as the `GEOS` library for computational geometry, the `PROJ` software that transforms spatial coordinates from one reference system to another, as in map projections, and the Simple Features standard for specifying the elements of spatial attributes.

```
library(sf)  
## Linking to GEOS 3.9.1, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

Let's see the main upshot for us as end-users.

The nycdogs package

```
library(nycdogs)
nyc_license

## # A tibble: 493,072 × 9
##   animal_name animal_gender animal_birth_year breed_rc      borough zip_code
##   <chr>        <chr>           <dbl> <chr>          <chr>     <int>
## 1 Paige         F              2014 Pit Bull (or Mi... Manhat...  10035
## 2 Yogi          M              2010 Boxer          Bronx       10465
## 3 Ali            M              2014 Basenji        Manhat...  10013
## 4 Queen         F              2013 Akita Crossbreed Manhat...  10013
## 5 Lola           F              2009 Maltese        Manhat...  10028
## 6 Ian            M              2006 Unknown        Manhat...  10013
## 7 Buddy          M              2008 Unknown        Manhat...  10025
## 8 Chewbacca     F              2012 Labrador (or Cr... Manhat...  10013
## 9 Heidi-Bo      F              2007 Dachshund Smoot... Brookl...  11215
## 10 Massimo       M              2009 Bull Dog, French Brookl... 11201
## # ... with 493,062 more rows, and 3 more variables: license_issued_date <date>,
## #   license_expired_date <date>, extract_year <dbl>
```



The nycdogs package

The metadata tells you this is not a regular tibble.

nyc_zips

```
## Simple feature collection with 262 features and 11 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 12
##   objectid zip_code po_name      state borough st_fips cty_fips bld_gpostal_code
##   <int>     <int> <chr>       <chr> <chr>    <chr>    <chr>           <int>
## 1       1     11372 Jackson He... NY    Queens    36      081                 0
## 2       2     11004 Glen Oaks    NY    Queens    36      081                 0
## 3       3     11040 New Hyde P... NY    Queens    36      081                 0
## 4       4     11426 Bellerose   NY    Queens    36      081                 0
## 5       5     11365 Fresh Mead... NY    Queens    36      081                 0
## 6       6     11373 Elmhurst    NY    Queens    36      081                 0
## 7       7     11001 Floral Park NY    Queens    36      081                 0
## 8       8     11375 Forest Hil... NY    Queens    36      081                 0
## 9       9     11427 Queens Vil... NY    Queens    36      081                 0
## 10     10     11374 Rego Park   NY    Queens    36      081                 0
## # ... with 252 more rows, and 4 more variables: shape_leng <dbl>,
## #   shape_area <dbl>, x_id <chr>, geometry <POLYGON [°]>
```

The nycdogs package

```
nyc_zips |>
  select(objectid:borough)

## Simple feature collection with 262 features and 5 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 6
##   objectid zip_code po_name      state borough          geometry
##       <int>    <int> <chr>      <chr> <chr>            <POLYGON [°]>
## 1         1     11372 Jackson Heights NY    Queens ((-73.86942 40.74916, -73.89...
## 2         2     11004 Glen Oaks        NY    Queens ((-73.71068 40.75004, -73.70...
## 3         3     11040 New Hyde Park     NY    Queens ((-73.70098 40.7389, -73.703...
## 4         4     11426 Belleroose       NY    Queens ((-73.7227 40.75373, -73.722...
## 5         5     11365 Fresh Meadows     NY    Queens ((-73.81089 40.72717, -73.81...
## 6         6     11373 Elmhurst         NY    Queens ((-73.88722 40.72753, -73.88...
## 7         7     11001 Floral Park       NY    Queens ((-73.70098 40.7389, -73.699...
## 8         8     11375 Forest Hills       NY    Queens ((-73.85625 40.73672, -73.85...
## 9         9     11427 Queens Village     NY    Queens ((-73.74169 40.73682, -73.73...
## 10        10    11374 Rego Park         NY    Queens ((-73.86451 40.73407, -73.85...
## # ... with 252 more rows
```

The **polygon** column is a list of lat/lon points that, when joined, draw the outline of the zip code area. This is *much* more compact than a big table where every row is a single point.

Let's make a summary table

Let's make a summary table

```
nyc_license
```

```
## # A tibble: 493,072 × 9
##   animal_name animal_gender animal_birth_year breed_rc      borough zip
##   <chr>        <chr>           <dbl> <chr>        <chr>
## 1 Paige         F              2014 Pit Bull (or Mi... Manhat...
## 2 Yogi          M              2010 Boxer          Bronx
## 3 Ali            M              2014 Basenji       Manhat...
## 4 Queen         F              2013 Akita Crossbreed Manhat...
## 5 Lola           F              2009 Maltese       Manhat...
## 6 Ian            M              2006 Unknown       Manhat...
## 7 Buddy          M              2008 Unknown       Manhat...
## 8 Chewbacca     F              2012 Labrador (or Cr... Manhat...
## 9 Heidi-Bo      F              2007 Dachshund Smoot... Brookl...
## 10 Massimo       M             2009 Bull Dog, French Brookl...
## # ... with 493,062 more rows, and 3 more variables: license_issued_date <date>
## #   license_expired_date <date>, extract_year <dbl>
```

Let's make a summary table

```
nyc_license |>  
filter(extract_year == 2018)  
  
## # A tibble: 117,371 × 9  
##   animal_name animal_gender animal_birth_year breed_rc  
##   <chr>        <chr>                  <dbl> <chr>  
## 1 Ali          M                   2014 Basenji  
## 2 Ian          M                   2006 Unknown  
## 3 Chewbacca    F                   2012 Labrador (or Cr...  
## 4 Lola          F                   2006 Miniature Pinsc...  
## 5 Lucy          F                   2014 Dachshund Smoot...  
## 6 June          F                   2010 Cavalier King C...  
## 7 Apple         M                   2013 Havanese  
## 8 Muneca        F                   2013 Beagle  
## 9 Benson        M                   2010 Boxer  
## 10 Bigs         M                   2004 Pit Bull (or Mi...  
## # ... with 117,361 more rows, and 3 more variables: license_issued_date <date>  
## #   license_expired_date <date>, extract_year <dbl>  
## #   borough zip  
## #   <chr>  <chr>
```

Let's make a summary table

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code)
```

```
## # A tibble: 117,371 × 9
## # Groups:   breed_rc, zip_code [18,945]
##       animal_name animal_gender animal_birth_year breed_rc      borough zip_
##       <chr>        <chr>           <dbl> <chr>        <chr>
## 1 Ali            M                2014 Basenji      Manhat...
## 2 Ian            M                2006 Unknown      Manhat...
## 3 Chewbacca     F                2012 Labrador (or Cr... Manhat...
## 4 Lola           F                2006 Miniature Pinsc... Manhat...
## 5 Lucy           F                2014 Dachshund Smoot... Brookl...
## 6 June           F                2010 Cavalier King C... Brookl...
## 7 Apple          M                2013 Havanese      Manhat...
## 8 Muneca         F                2013 Beagle        Brookl...
## 9 Benson         M                2010 Boxer        Brookl...
## 10 Bigs          M                2004 Pit Bull (or Mi... Brookl...
## # ... with 117,361 more rows, and 3 more variables: license_issued_date <date>
## #   license_expired_date <date>, extract_year <dbl>
```

Let's make a summary table

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally()                                ## # A tibble: 18,945 × 3
                                             ## # Groups:   breed_rc [311]
                                             ##   breed_rc     zip_code     n
                                             ##   <chr>          <int> <int>
                                             ## 1 Affenpinscher 10005     1
                                             ## 2 Affenpinscher 10011     1
                                             ## 3 Affenpinscher 10013     1
                                             ## 4 Affenpinscher 10014     1
                                             ## 5 Affenpinscher 10016     1
                                             ## 6 Affenpinscher 10017     1
                                             ## 7 Affenpinscher 10018     1
                                             ## 8 Affenpinscher 10019     1
                                             ## 9 Affenpinscher 10021     1
                                             ## 10 Affenpinscher 10023    1
                                             ## # ... with 18,935 more rows
```

Let's make a summary table

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  mutate(freq = n / sum(n))
```

```
## # A tibble: 18,945 × 4
## # Groups:   breed_rc [311]
##       breed_rc     zip_code     n   freq
##       <chr>          <int> <int> <dbl>
## 1 Affenpinscher  10005     1 0.0303
## 2 Affenpinscher  10011     1 0.0303
## 3 Affenpinscher  10013     1 0.0303
## 4 Affenpinscher  10014     1 0.0303
## 5 Affenpinscher  10016     1 0.0303
## 6 Affenpinscher  10017     1 0.0303
## 7 Affenpinscher  10018     1 0.0303
## 8 Affenpinscher  10019     1 0.0303
## 9 Affenpinscher  10021     1 0.0303
## 10 Affenpinscher 10023    1 0.0303
## # ... with 18,935 more rows
```

Let's make a summary table

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  mutate(freq = n / sum(n)) |>
  filter(breed_rc == "French Bulldog")
```

```
## # A tibble: 161 × 4
## # Groups: breed_rc [1]
##   breed_rc      zip_code     n    freq
##   <chr>          <int> <int>  <dbl>
## 1 French Bulldog 10001    27 0.0167
## 2 French Bulldog 10002    20 0.0123
## 3 French Bulldog 10003    36 0.0222
## 4 French Bulldog 10004     9 0.00555
## 5 French Bulldog 10005    15 0.00925
## 6 French Bulldog 10006     8 0.00494
## 7 French Bulldog 10007    17 0.0105
## 8 French Bulldog 10009    51 0.0315
## 9 French Bulldog 10010    31 0.0191
## 10 French Bulldog 10011   88 0.0543
## # ... with 151 more rows
```

Let's make a summary table

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally() |>  
mutate(freq = n / sum(n)) |>  
filter(breed_rc == "French Bulldog") ->  
nyc_fb
```

Let's make a summary table

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally() |>  
mutate(freq = n / sum(n)) |>  
filter(breed_rc == "French Bulldog") ->  
nyc_fb
```

Now we have two tables again

```
nyc_zips |> select(objectid:st_fips)

## Simple feature collection with 262 features and 6 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 7
##   objectid zip_code po_name     state borough st_fips           geometry
##       <int>    <int> <chr>      <chr> <chr>    <chr> <POLYGON [°]>
## 1        1      11372 Jackson He... NY    Queens    36 ((-73.86942 40.74916, -7...
## 2        2      11004 Glen Oaks    NY    Queens    36 ((-73.71068 40.75004, -7...
## 3        3      11040 New Hyde P... NY    Queens    36 ((-73.70098 40.7389, -73...
## 4        4      11426 Belleroose   NY    Queens    36 ((-73.7227 40.75373, -73...
## 5        5      11365 Fresh Mead... NY    Queens    36 ((-73.81089 40.72717, -7...
## 6        6      11373 Elmhurst    NY    Queens    36 ((-73.88722 40.72753, -7...
## 7        7      11001 Floral Park NY    Queens    36 ((-73.70098 40.7389, -73...
## 8        8      11375 Forest Hil... NY    Queens    36 ((-73.85625 40.73672, -7...
## 9        9      11427 Queens Vil... NY    Queens    36 ((-73.74169 40.73682, -7...
## 10       10     11374 Rego Park   NY    Queens    36 ((-73.86451 40.73407, -7...
## # ... with 252 more rows
```

Now we have two tables again

```
nyc_zips |> select(objectid:st_fips)

## Simple feature collection with 262 features and 6 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 7
##   objectid zip_code po_name     state borough st_fips           geometry
##       <int>    <int> <chr>      <chr> <chr>    <chr> <POLYGON [°]>
## 1        1     11372 Jackson He... NY     Queens 36  ((-73.86942 40.74916, -7...
## 2        2     11004 Glen Oaks    NY     Queens 36  ((-73.71068 40.75004, -7...
## 3        3     11040 New Hyde P... NY     Queens 36  ((-73.70098 40.7389, -73...
## 4        4     11426 Bellerose   NY     Queens 36  ((-73.7227 40.75373, -73...
## 5        5     11365 Fresh Mead... NY     Queens 36  ((-73.81089 40.72717, -7...
## 6        6     11373 Elmhurst    NY     Queens 36  ((-73.88722 40.72753, -7...
## 7        7     11001 Floral Park NY     Queens 36  ((-73.70098 40.7389, -73...
## 8        8     11375 Forest Hil... NY     Queens 36  ((-73.85625 40.73672, -7...
## 9        9     11427 Queens Vil... NY     Queens 36  ((-73.74169 40.73682, -7...
## 10      10     11374 Rego Park  NY     Queens 36  ((-73.86451 40.73407, -7...
## # ... with 252 more rows

nyc_fb |> select(breed_rc:n)

## # A tibble: 161 × 3
## # Groups: breed_rc [1]
##   breed_rc      zip_code     n
##   <chr>          <int> <int>
## 1 French Bulldog 10001    27
## 2 French Bulldog 10002    20
## 3 French Bulldog 10003    36
## 4 French Bulldog 10004     9
## 5 French Bulldog 10005   15
## 6 French Bulldog 10006     8
## 7 French Bulldog 10007   17
## 8 French Bulldog 10009   51
## 9 French Bulldog 10010   31
## 10 French Bulldog 10011   88
## # ... with 151 more rows
```

Now we have two tables again

```
nyc_zips |> select(objectid:st_fips)
```

```
## Simple feature collection with 262 features and 6 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 7
##   objectid zip_code po_name     state borough st_fips           geometry
##       <int>    <int> <chr>      <chr> <chr>    <chr> <POLYGON [°]>
## 1        1      11372 Jackson He... NY    Queens    36 ((-73.86942 40.74916, -7...
## 2        2      11004 Glen Oaks    NY    Queens    36 ((-73.71068 40.75004, -7...
## 3        3      11040 New Hyde P... NY    Queens    36 ((-73.70098 40.7389, -73...
## 4        4      11426 Bellerose    NY    Queens    36 ((-73.7227 40.75373, -73...
## 5        5      11365 Fresh Mead... NY    Queens    36 ((-73.81089 40.72717, -7...
## 6        6      11373 Elmhurst    NY    Queens    36 ((-73.88722 40.72753, -7...
## 7        7      11001 Floral Park NY    Queens    36 ((-73.70098 40.7389, -73...
## 8        8      11375 Forest Hil... NY    Queens    36 ((-73.85625 40.73672, -7...
## 9        9      11427 Queens Vil... NY    Queens    36 ((-73.74169 40.73682, -7...
## 10      10      11374 Rego Park   NY    Queens    36 ((-73.86451 40.73407, -7...
## # ... with 252 more rows
```

```
nyc_fb |> select(breed_rc:n)
```

```
## # A tibble: 161 × 3
## # Groups:   breed_rc [1]
##   breed_rc      zip_code     n
##   <chr>          <int> <int>
## 1 French Bulldog 10001    27
## 2 French Bulldog 10002    20
## 3 French Bulldog 10003    36
## 4 French Bulldog 10004     9
## 5 French Bulldog 10005   15
## 6 French Bulldog 10006     8
## 7 French Bulldog 10007   17
## 8 French Bulldog 10009   51
## 9 French Bulldog 10010   31
## 10 French Bulldog 10011   88
## # ... with 151 more rows
```

Join them:

```
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
```

Ready to map

```
fb_map |> select(zip_code, po_name, borough, breed_rc:freq, geometry)

## Simple feature collection with 262 features and 6 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 7
##       zip_code po_name    borough breed_rc      n     freq
##           <int> <chr>      <chr>   <chr> <int>   <dbl>
## 1       11372 Jackson H... Queens French ...  13  8.02e-3
## 2       11004 Glen Oaks  Queens French ...   1  6.17e-4
## 3       11040 New Hyde ... Queens <NA>        NA NA
## 4       11426 Bellerose  Queens French ...   1  6.17e-4
## 5       11365 Fresh Mea... Queens French ...   7  4.32e-3
## 6       11373 Elmhurst   Queens French ...  14  8.64e-3
## 7       11001 Floral Pa... Queens <NA>        NA NA
## 8       11375 Forest Hi... Queens French ...   8  4.94e-3
## 9       11427 Queens Vi... Queens French ...   2  1.23e-3
## 10      11374 Rego Park  Queens French ...   6  3.70e-3
## # ... with 252 more rows
## # ... with 252 more rows
```

A NYC map theme

Just moving the legend, really.

```
theme_nymp <- function(base_size=9, base_family="") {  
  require(grid)  
  theme_bw(base_size=base_size, base_family=base_family) %>%replace%  
    theme(axis.line=element_blank(),  
          axis.text=element_blank(),  
          axis.ticks=element_blank(),  
          axis.title=element_blank(),  
          panel.background=element_blank(),  
          panel.border=element_blank(),  
          panel.grid=element_blank(),  
          panel.spacing=unit(0, "lines"),  
          plot.background=element_blank(),  
          legend.justification = c(0,0),  
          legend.position = c(0.05, 0.58),  
          legend.direction = "horizontal"  
    )  
}
```

First cut at a map

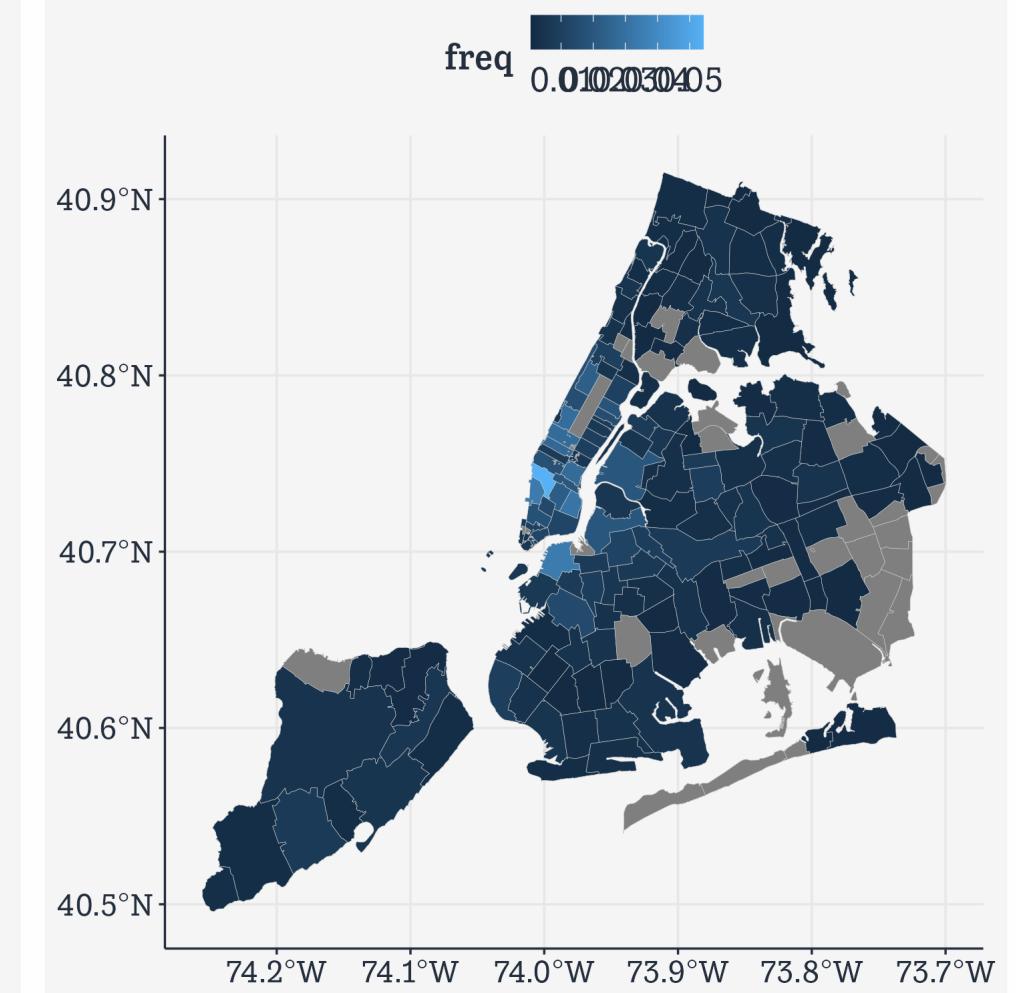
```
fb_map
## Simple feature collection with 262 features and 14 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 15
##       objectid    zip_code po_name      state borough st_fips cty_fips bld_gpostal_
##           <int>     <int> <chr>        <chr> <chr>   <chr>   <chr>   <chr>
## 1            1      11372 Jackson He... NY    Queens    36    081
## 2            2      11004 Glen Oaks    NY    Queens    36    081
## 3            3      11040 New Hyde P... NY    Queens    36    081
## 4            4      11426 Bellerose    NY    Queens    36    081
## 5            5      11365 Fresh Mead... NY    Queens    36    081
## 6            6      11373 Elmhurst    NY    Queens    36    081
## 7            7      11001 Floral Park NY    Queens    36    081
## 8            8      11375 Forest Hil... NY    Queens    36    081
## 9            9      11427 Queens Vil... NY    Queens    36    081
## 10           10     11374 Rego Park   NY    Queens    36    081
## # ... with 252 more rows, and 7 more variables: shape_leng <dbl>,
## #   shape_area <dbl>, x_id <chr>, geometry <POLYGON [°]>, breed_rc <chr>,
## #   n <int>, freq <dbl>
```

First cut at a map

```
fb_map %>%  
  ggplot(mapping = aes(fill = freq))
```

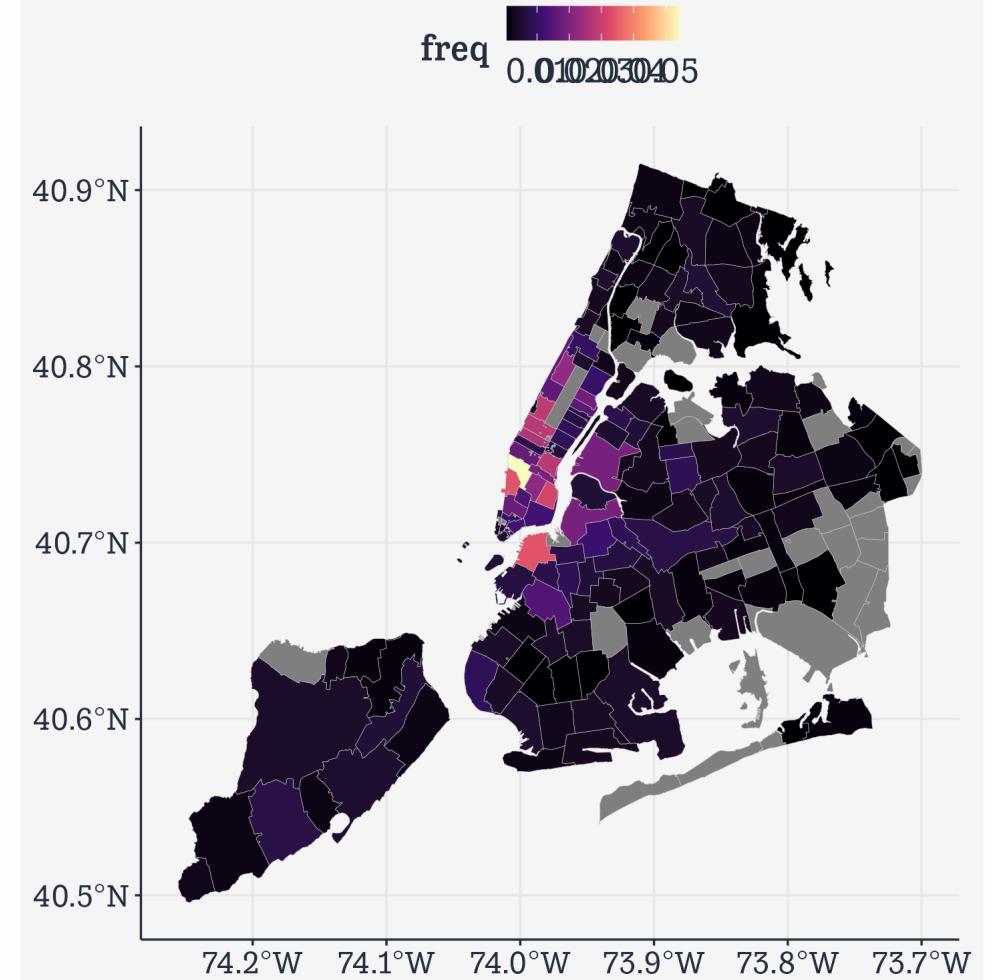
First cut at a map

```
fb_map %>%  
  ggplot(mapping = aes(fill = freq)) +  
  geom_sf(color = "gray80", size = 0.1) #<<
```



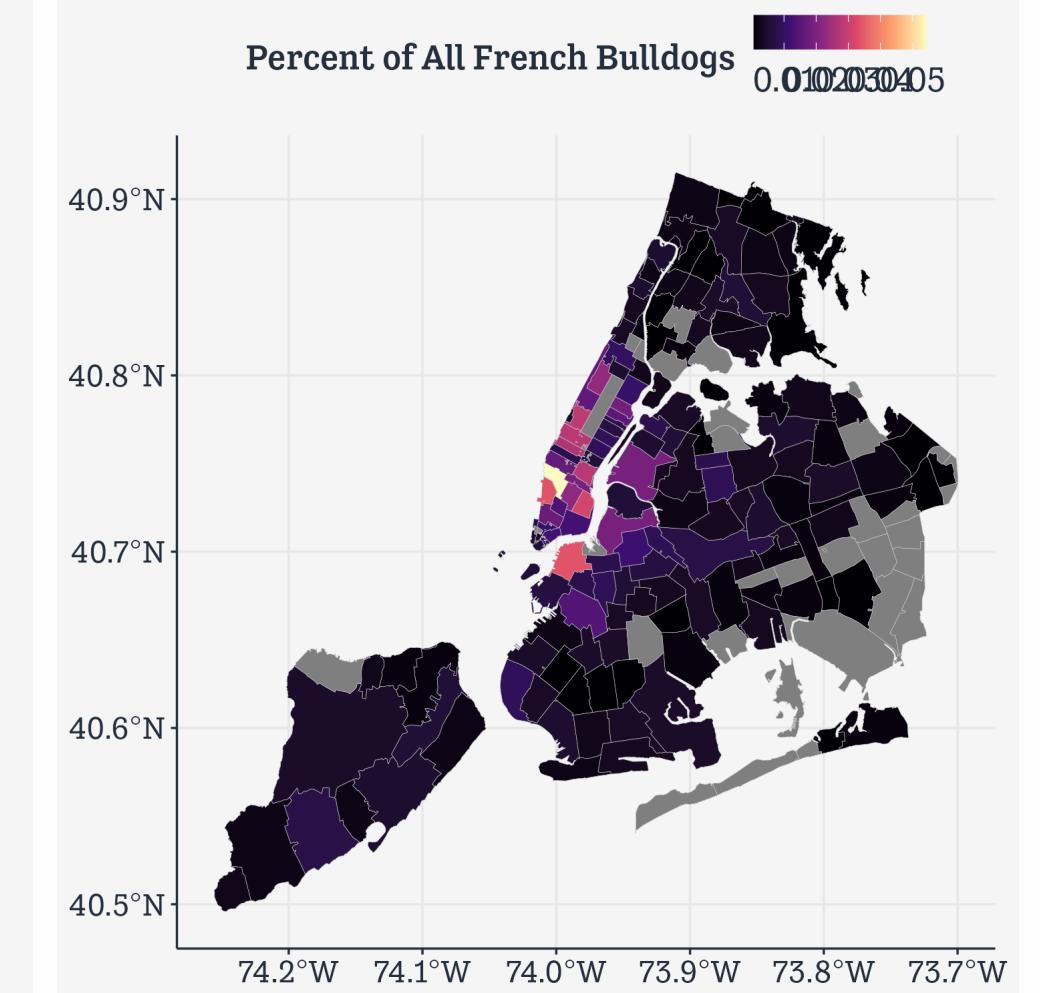
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A")
```



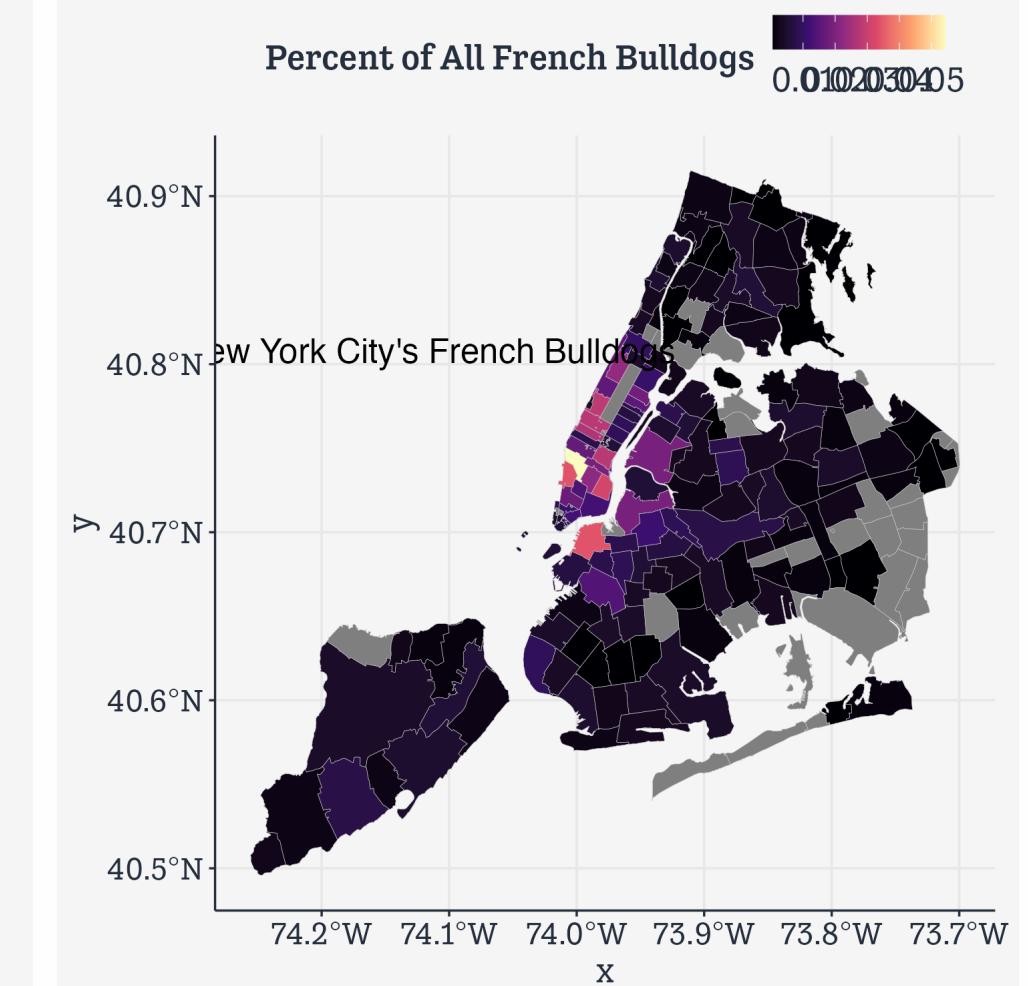
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
```



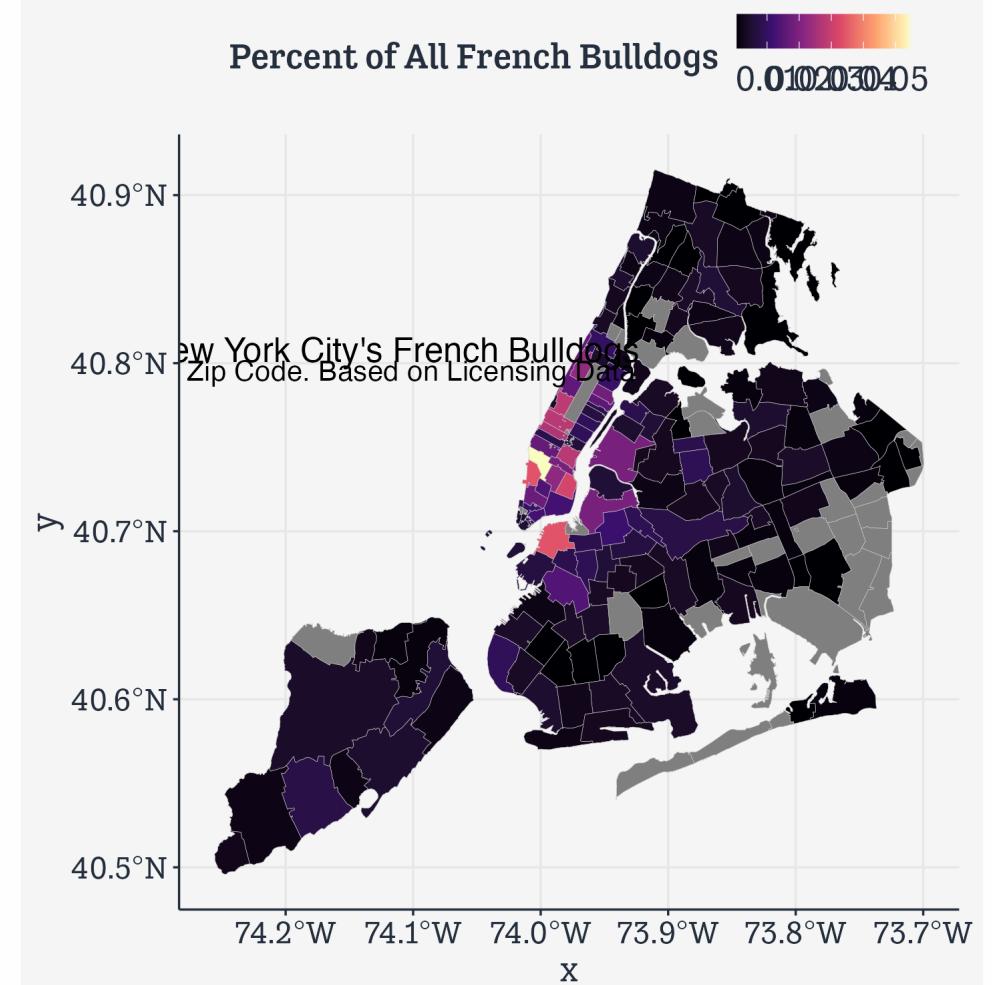
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
  annotate(geom = "text",
           x = -74.145 + 0.029,
           y = 40.82-0.012,
           label = "New York City's French Bulldog",
           size = 6)
```



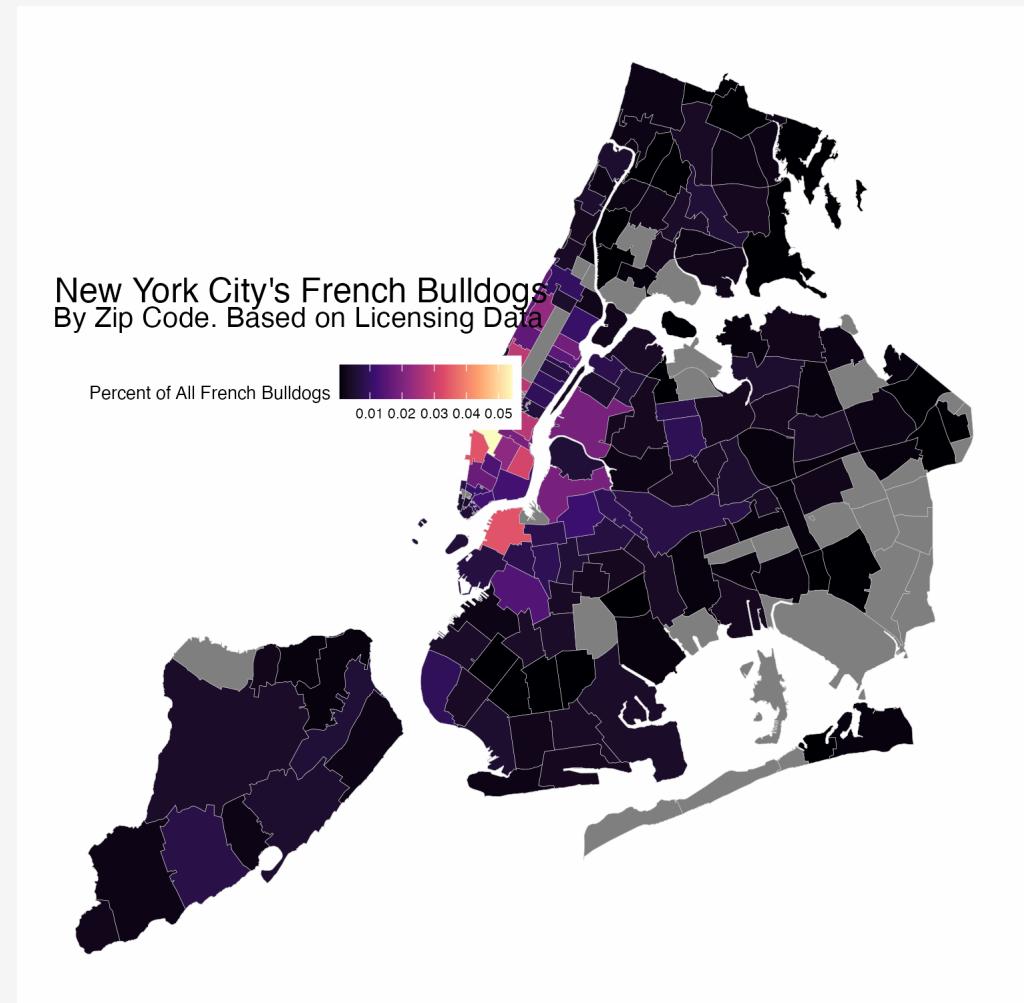
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
  annotate(geom = "text",
           x = -74.145 + 0.029,
           y = 40.82-0.012,
           label = "New York City's French Bulldog",
           size = 6) +
  annotate(geom = "text",
           x = -74.1468 + 0.029,
           y = 40.8075-0.012,
           label = "By Zip Code. Based on Licensing Data",
           size = 5)
```



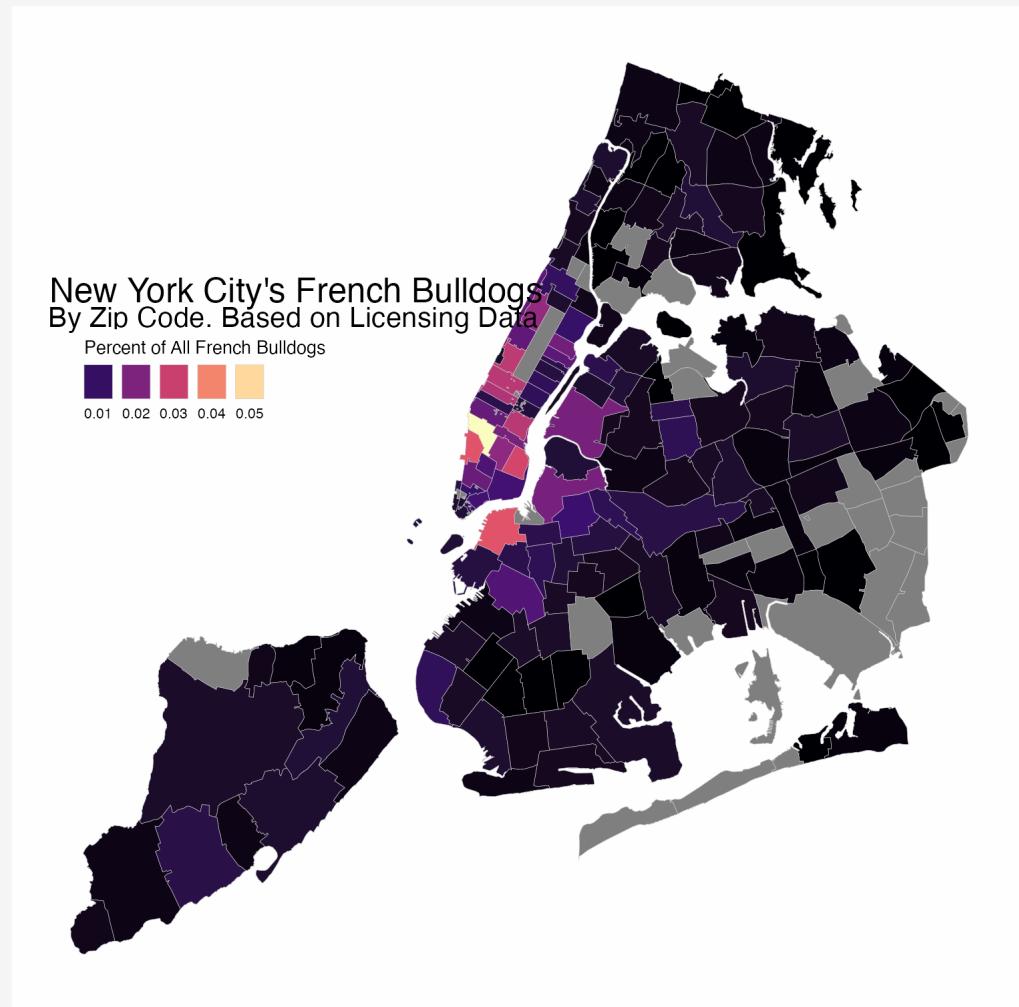
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
  annotate(geom = "text",
           x = -74.145 + 0.029,
           y = 40.82-0.012,
           label = "New York City's French Bulldog",
           size = 6) +
  annotate(geom = "text",
           x = -74.1468 + 0.029,
           y = 40.8075-0.012,
           label = "By Zip Code. Based on Licensing",
           size = 5) +
kjhsldes::kjh_them_nymap()
```



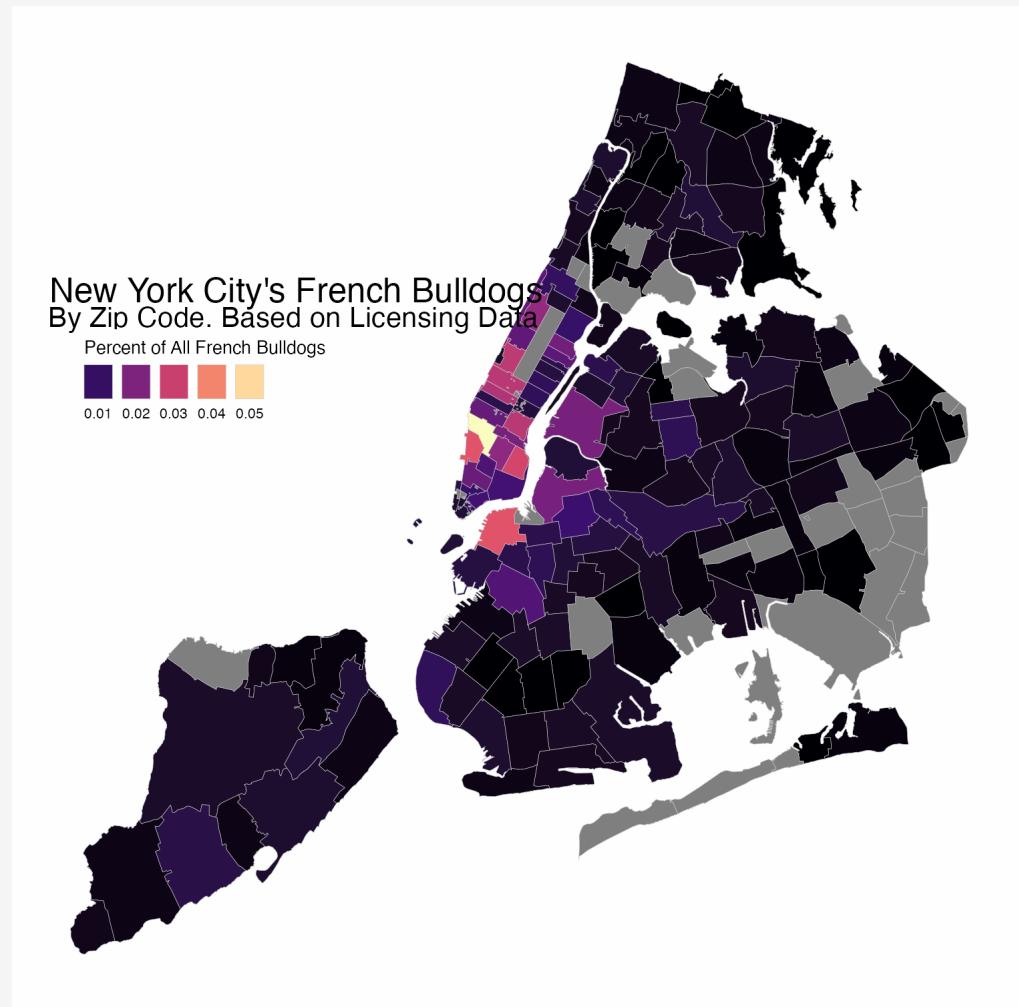
First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
  annotate(geom = "text",
           x = -74.145 + 0.029,
           y = 40.82-0.012,
           label = "New York City's French Bulldog",
           size = 6) +
  annotate(geom = "text",
           x = -74.1468 + 0.029,
           y = 40.8075-0.012,
           label = "By Zip Code. Based on Licensing Data",
           size = 5) +
  kjhslides::kjh_theme_nymap() +
  guides(fill =
    guide_legend(title.position = "top",
                label.position = "bottom",
                keywidth = 1,
                nrow = 1))
```



First cut at a map

```
fb_map %>%
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_viridis_c(option = "A") +
  labs(fill = "Percent of All French Bulldogs")
  annotate(geom = "text",
           x = -74.145 + 0.029,
           y = 40.82-0.012,
           label = "New York City's French Bulldog",
           size = 6) +
  annotate(geom = "text",
           x = -74.1468 + 0.029,
           y = 40.8075-0.012,
           label = "By Zip Code. Based on Licensing Data",
           size = 5) +
  kjhslides::kjh_theme_nymap() +
  guides(fill =
    guide_legend(title.position = "top",
                 label.position = "bottom",
                 keywidth = 1,
                 nrow = 1))
```



Use a different palette

```
library(colorspace)
```

Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_c
```

Use a different palette

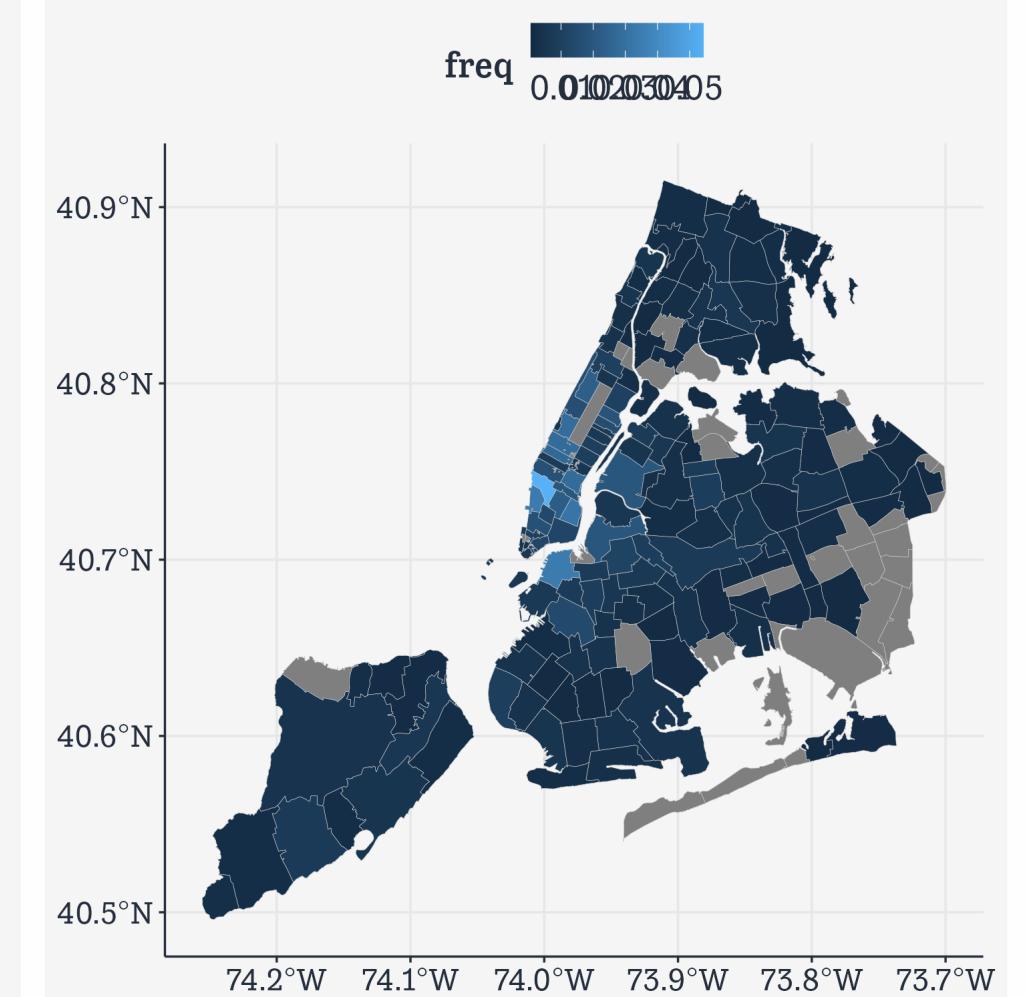
```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map
## Simple feature collection with 262 features and 14 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 15
##       objectid zip_code po_name      state borough st_fips cty_fips bld_gpostal
##          <int>    <int> <chr>      <chr> <chr>   <chr>   <chr>   <chr>
## 1           1     11372 Jackson He... NY    Queens    36    081
## 2           2     11004 Glen Oaks    NY    Queens    36    081
## 3           3     11040 New Hyde P... NY    Queens    36    081
## 4           4     11426 Bellerose    NY    Queens    36    081
## 5           5     11365 Fresh Mead... NY    Queens    36    081
## 6           6     11373 Elmhurst    NY    Queens    36    081
## 7           7     11001 Floral Park NY    Queens    36    081
## 8           8     11375 Forest Hil... NY    Queens    36    081
## 9           9     11427 Queens Vil... NY    Queens    36    081
## 10          10    11374 Rego Park   NY    Queens    36    081
## # ... with 252 more rows, and 7 more variables: shape_leng <dbl>,
## #   shape_area <dbl>, x_id <chr>, geometry <POLYGON [°]>, breed_rc <chr>,
## #   n <int>, freq <dbl>
```

Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq))
```

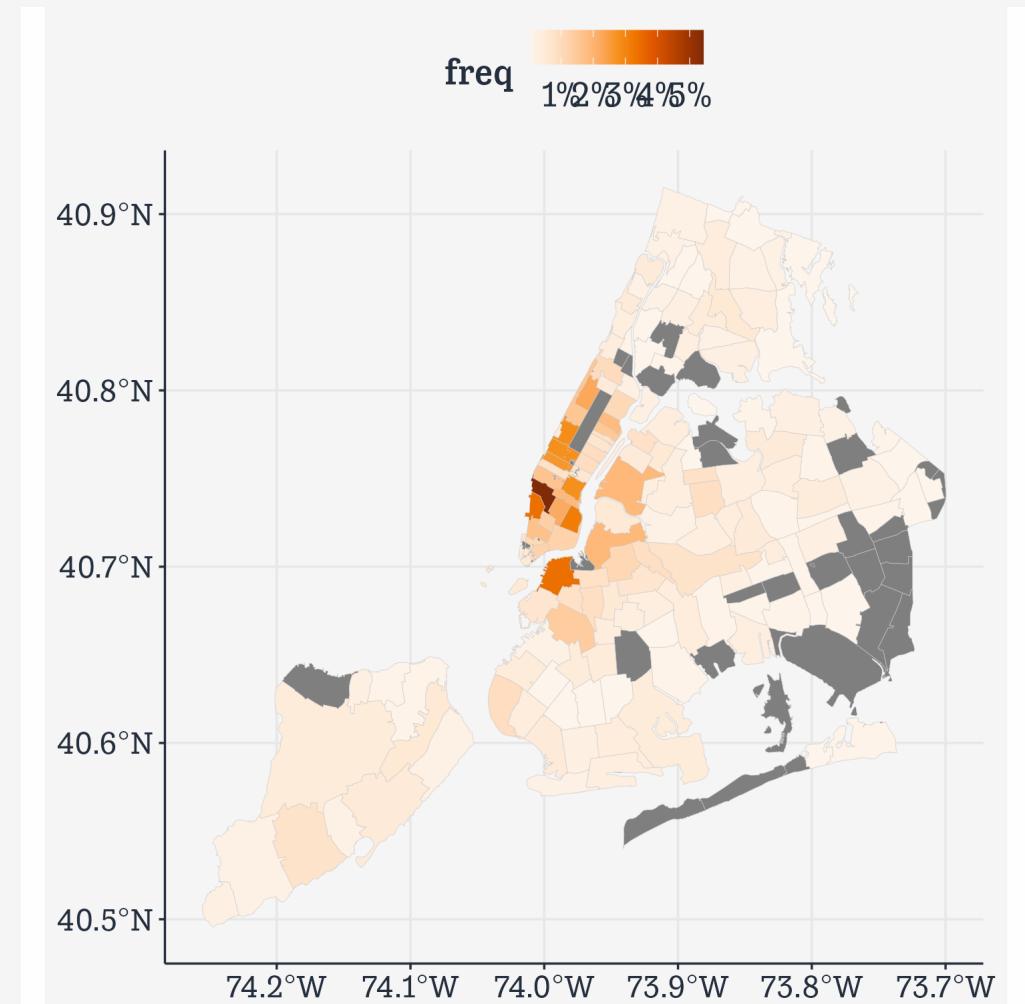
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1)
```



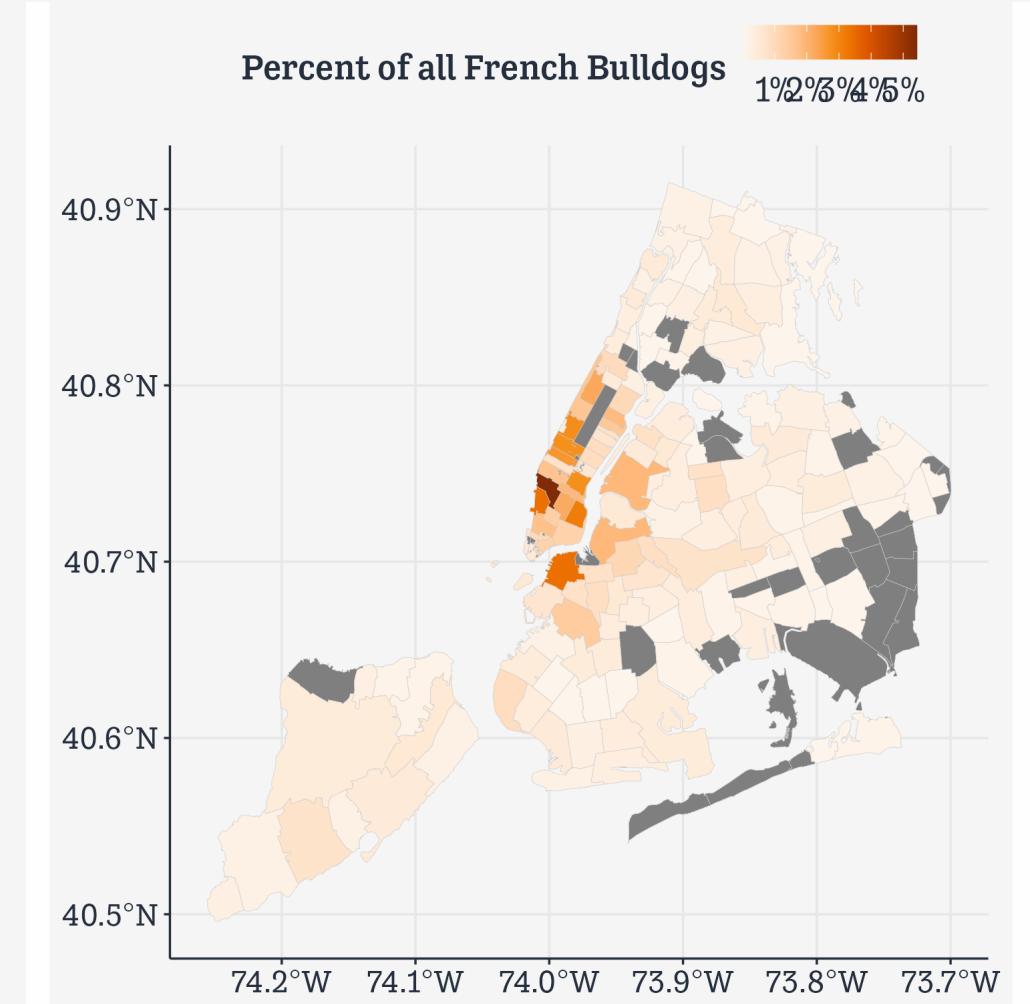
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent())
```



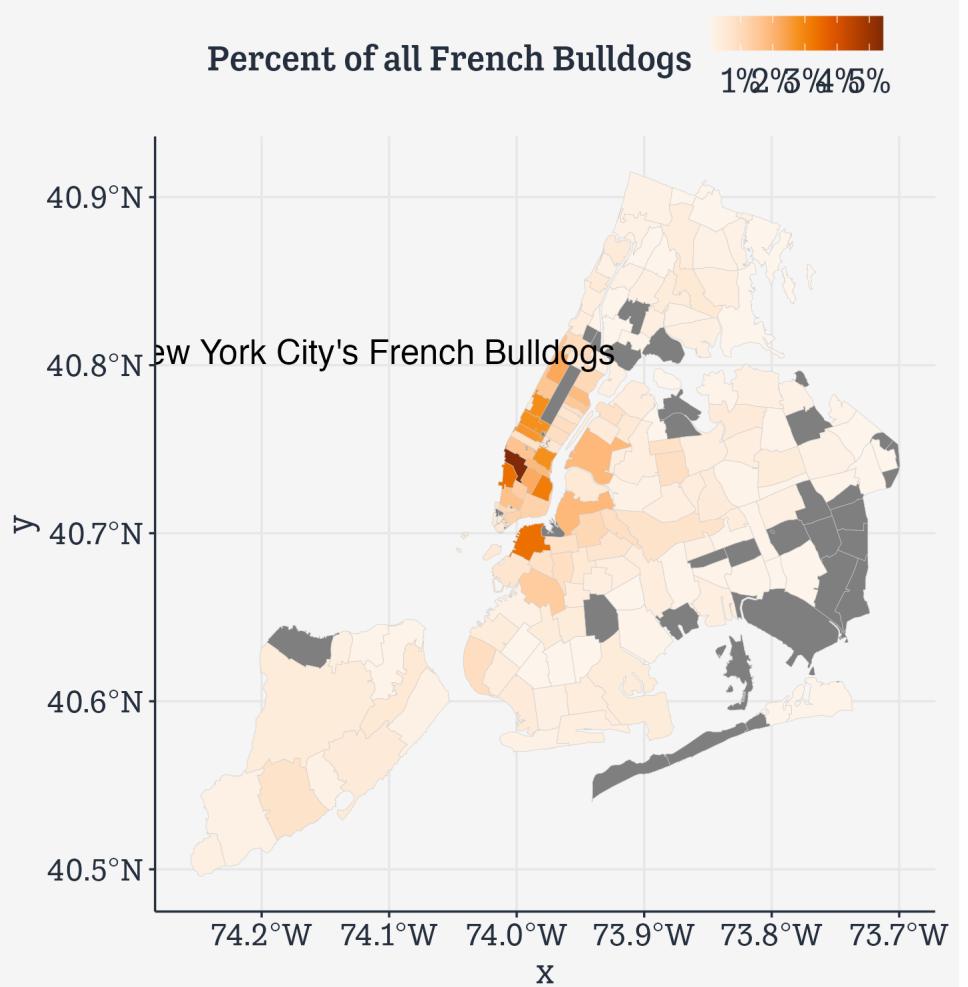
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs")
```



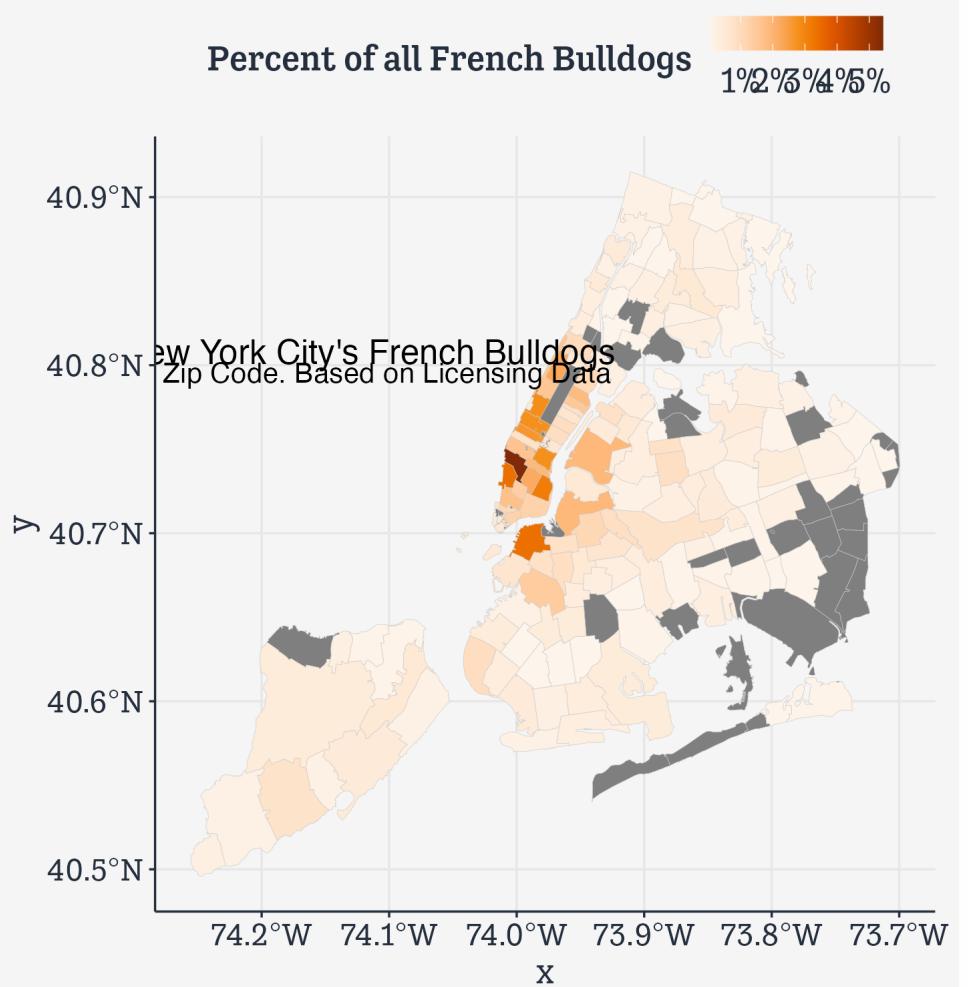
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82 - 0.012,
          label = "New York City's French Bulldogs",
          size = 6)
```



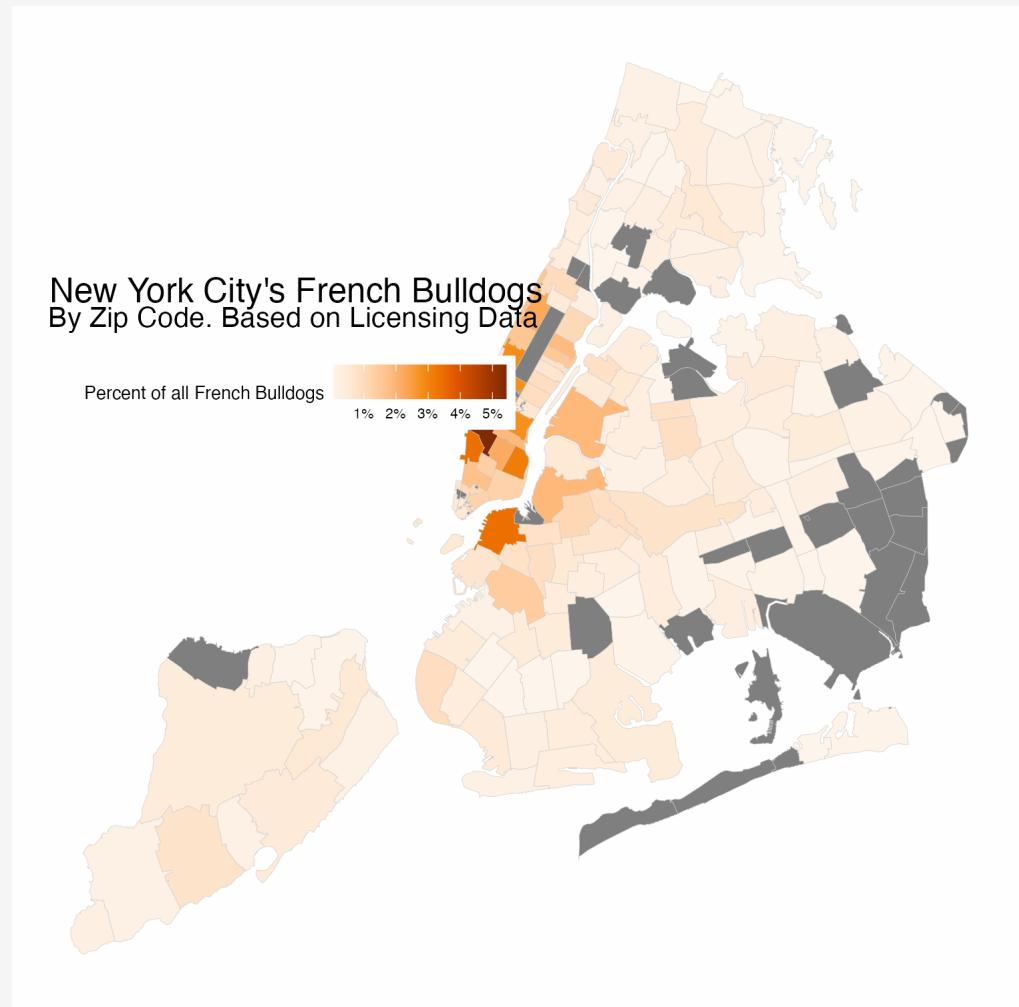
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldogs",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5)
```



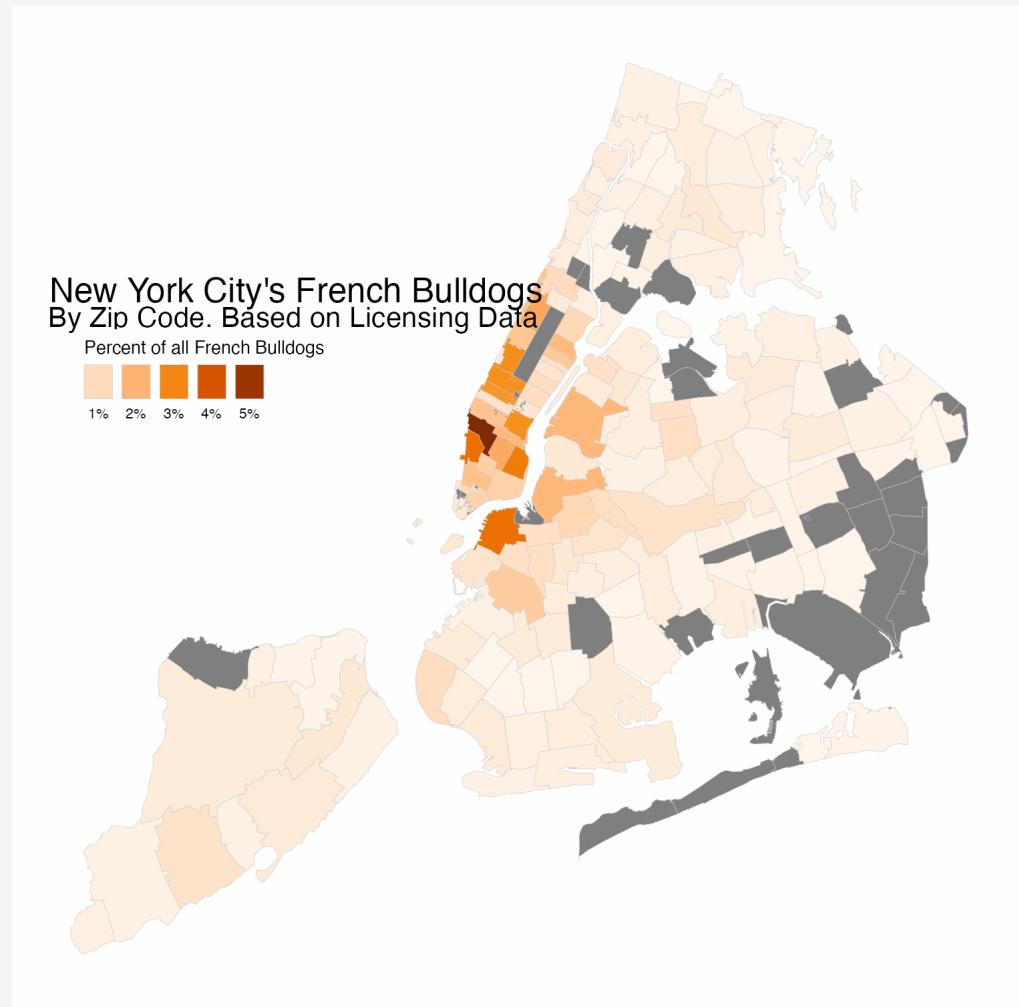
Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5) +
  kjhslides::kjh_theme_nymap()
```



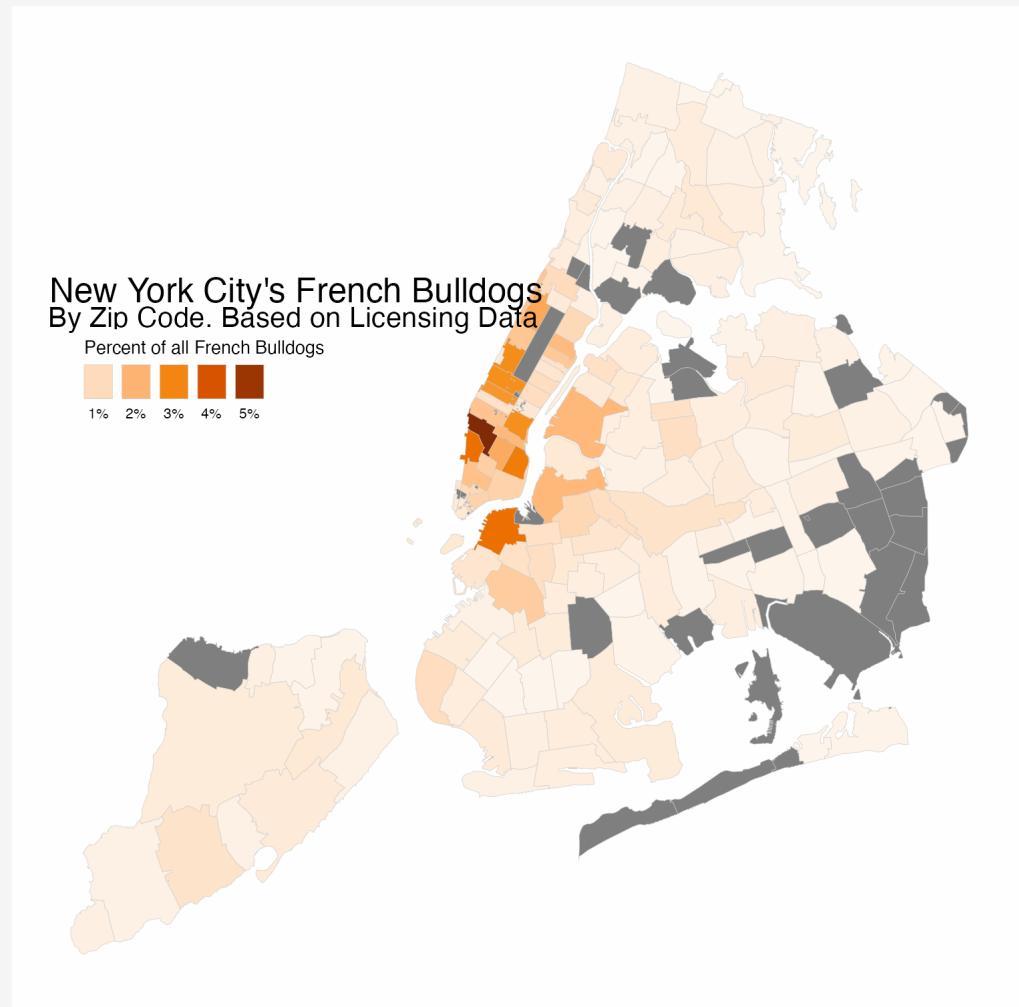
Use a different palette

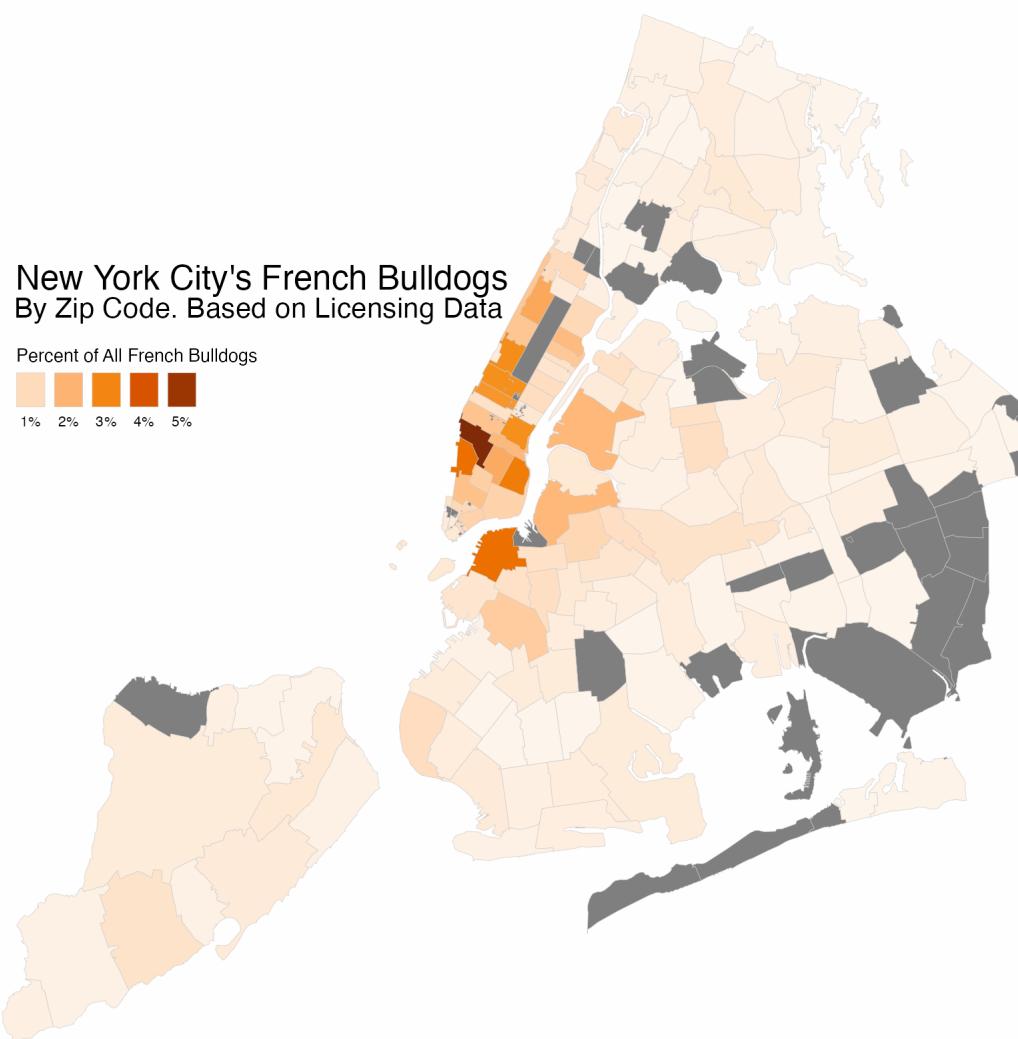
```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent() +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82 - 0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5) +
  kjhslides::kjh_theme_nymap() +
  guides(fill =
        guide_legend(title.position = "top",
                     label.position = "bottom",
                     keywidth = 1,
                     nrow = 1))
```



Use a different palette

```
library(colorspace)
fb_map <- left_join(nyc_zips, nyc_fb, by = "zip_code")
fb_map |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent() +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82 - 0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5) +
  kjhslides::kjh_theme_nymap() +
  guides(fill =
        guide_legend(title.position = "top",
                     label.position = "bottom",
                     keywidth = 1,
                     nrow = 1))
```





NYC Dogs Map mark 2

Keep the Zero count zips

```
nyc_license
## # A tibble: 493,072 × 9
##   animal_name animal_gender animal_birth_year breed_rc      borough zip
##   <chr>        <chr>           <dbl> <chr>        <chr>
## 1 Paige         F              2014 Pit Bull (or Mi... Manhat...
## 2 Yogi          M              2010 Boxer          Bronx
## 3 Ali            M              2014 Basenji       Manhat...
## 4 Queen         F              2013 Akita Crossbreed Manhat...
## 5 Lola           F              2009 Maltese       Manhat...
## 6 Ian            M              2006 Unknown       Manhat...
## 7 Buddy          M              2008 Unknown       Manhat...
## 8 Chewbacca     F              2012 Labrador (or Cr... Manhat...
## 9 Heidi-Bo      F              2007 Dachshund Smoot... Brookl...
## 10 Massimo       M             2009 Bull Dog, French Brookl...
## # ... with 493,062 more rows, and 3 more variables: license_issued_date <date>
## #   license_expired_date <date>, extract_year <dbl>
```

Keep the Zero count zips

```
nyc_license |>  
filter(extract_year == 2018)  
  
## # A tibble: 117,371 × 9  
##   animal_name animal_gender animal_birth_year breed_rc  
##   <chr>        <chr>                  <dbl> <chr>  
## 1 Ali          M                   2014 Basenji  
## 2 Ian          M                   2006 Unknown  
## 3 Chewbacca    F                   2012 Labrador (or Cr...  
## 4 Lola          F                   2006 Miniature Pinsc...  
## 5 Lucy          F                   2014 Dachshund Smoot...  
## 6 June          F                   2010 Cavalier King C...  
## 7 Apple         M                   2013 Havanese  
## 8 Muneca        F                   2013 Beagle  
## 9 Benson        M                   2010 Boxer  
## 10 Bigs         M                   2004 Pit Bull (or Mi...  
## # ... with 117,361 more rows, and 3 more variables: license_issued_date <date>  
## #   license_expired_date <date>, extract_year <dbl>  
## #   borough zip  
## #   <chr>  <chr>
```

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code)

## # A tibble: 117,371 × 9
## # Groups: breed_rc, zip_code [18,945]
##   animal_name animal_gender animal_birth_year breed_rc      borough zip_
##   <chr>        <chr>           <dbl> <chr>       <chr>
## 1 Ali          M              2014 Basenji    Manhat...
## 2 Ian          M              2006 Unknown    Manhat...
## 3 Chewbacca    F              2012 Labrador (or Cr... Manhat...
## 4 Lola         F              2006 Miniature Pinsc... Manhat...
## 5 Lucy         F              2014 Dachshund Smoot... Brookl...
## 6 June         F              2010 Cavalier King C... Brookl...
## 7 Apple        M              2013 Havanese    Manhat...
## 8 Muneca       F              2013 Beagle     Brookl...
## 9 Benson       M              2010 Boxer     Brookl...
## 10 Bigs        M             2004 Pit Bull (or Mi... Brookl...
## # ... with 117,361 more rows, and 3 more variables: license_issued_date <date>
## #   license_expired_date <date>, extract_year <dbl>
```

Keep the Zero count zips

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally()  
  
## # A tibble: 18,945 × 3  
## # Groups: breed_rc [311]  
##   breed_rc     zip_code     n  
##   <chr>        <int> <int>  
## 1 Affenpinscher 10005     1  
## 2 Affenpinscher 10011     1  
## 3 Affenpinscher 10013     1  
## 4 Affenpinscher 10014     1  
## 5 Affenpinscher 10016     1  
## 6 Affenpinscher 10017     1  
## 7 Affenpinscher 10018     1  
## 8 Affenpinscher 10019     1  
## 9 Affenpinscher 10021     1  
## 10 Affenpinscher 10023    1  
## # ... with 18,935 more rows
```

Keep the Zero count zips

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally() |>  
ungroup()  
  
## # A tibble: 18,945 × 3  
##   breed_rc     zip_code     n  
##   <chr>          <int> <int>  
## 1 Affenpinscher 10005     1  
## 2 Affenpinscher 10011     1  
## 3 Affenpinscher 10013     1  
## 4 Affenpinscher 10014     1  
## 5 Affenpinscher 10016     1  
## 6 Affenpinscher 10017     1  
## 7 Affenpinscher 10018     1  
## 8 Affenpinscher 10019     1  
## 9 Affenpinscher 10021     1  
## 10 Affenpinscher 10023    1  
## # ... with 18,935 more rows
```

Keep the Zero count zips

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally() |>  
ungroup() |>  
complete(zip_code, breed_rc,  
        fill = list(n = 0))  
  
## # A tibble: 137,151 × 3  
##   zip_code breed_rc      n  
##       <int> <chr>     <int>  
## 1     1135 Affenpinscher     0  
## 2     1135 Afghan Hound     0  
## 3     1135 Afghan Hound Crossbreed 0  
## 4     1135 Airedale Terrier    0  
## 5     1135 Akita           0  
## 6     1135 Akita Crossbreed    0  
## 7     1135 Alaskan Malamute    0  
## 8     1135 American Bully     0  
## 9     1135 American English Coonhound 0  
## 10    1135 American Eskimo Dog    0  
## # ... with 137,141 more rows
```

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  ungroup() |>
  complete(zip_code, breed_rc,
           fill = list(n = 0)) |>
# Regroup to get the right denominator
  group_by(breed_rc)
```

```
## # A tibble: 137,151 × 3
## # Groups:   breed_rc [311]
##       zip_code breed_rc      n
##           <int> <chr>     <int>
## 1        1135 Affenpinscher     0
## 2        1135 Afghan Hound     0
## 3        1135 Afghan Hound Crossbreed     0
## 4        1135 Airedale Terrier     0
## 5        1135 Akita             0
## 6        1135 Akita Crossbreed     0
## 7        1135 Alaskan Malamute     0
## 8        1135 American Bully     0
## 9        1135 American English Coonhound     0
## 10       1135 American Eskimo Dog     0
## # ... with 137,141 more rows
```

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  ungroup() |>
  complete(zip_code, breed_rc,
           fill = list(n = 0)) |>
# Regroup to get the right denominator
  group_by(breed_rc) |>
  mutate(freq = n / sum(n))
```



```
## # A tibble: 137,151 × 4
## # Groups:   breed_rc [311]
##       zip_code breed_rc
##             <int> <chr>
## 1          1135 Affenpinscher
## 2          1135 Afghan Hound
## 3          1135 Afghan Hound Crossbreed
## 4          1135 Airedale Terrier
## 5          1135 Akita
## 6          1135 Akita Crossbreed
## 7          1135 Alaskan Malamute
## 8          1135 American Bully
## 9          1135 American English Coonhound
## 10         1135 American Eskimo Dog
## # ... with 137,141 more rows
```

zip_code	breed_rc	n	freq
1135	Affenpinscher	0	0
1135	Afghan Hound	0	0
1135	Afghan Hound Crossbreed	0	0
1135	Airedale Terrier	0	0
1135	Akita	0	0
1135	Akita Crossbreed	0	0
1135	Alaskan Malamute	0	0
1135	American Bully	0	0
1135	American English Coonhound	0	0
1135	American Eskimo Dog	0	0

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  ungroup() |>
  complete(zip_code, breed_rc,
           fill = list(n = 0)) |>
# Regroup to get the right denominator
  group_by(breed_rc) |>
  mutate(freq = n / sum(n)) |>
  filter(breed_rc == "French Bulldog")
```

```
## # A tibble: 441 × 4
## # Groups:   breed_rc [1]
##       zip_code breed_rc      n   freq
##       <int> <chr>     <int> <dbl>
## 1     1135 French Bulldog     0     0
## 2     1175 French Bulldog     0     0
## 3     1305 French Bulldog     0     0
## 4     6403 French Bulldog     0     0
## 5     6473 French Bulldog     0     0
## 6     6518 French Bulldog     0     0
## 7     6615 French Bulldog     0     0
## 8     6901 French Bulldog     0     0
## 9     7002 French Bulldog     0     0
## 10    7010 French Bulldog     0     0
## # ... with 431 more rows
```

Keep the Zero count zips

```
nyc_license |>  
filter(extract_year == 2018) |>  
group_by(breed_rc, zip_code) |>  
tally() |>  
ungroup() |>  
complete(zip_code, breed_rc,  
         fill = list(n = 0)) |>  
# Regroup to get the right denominator  
group_by(breed_rc) |>  
mutate(freq = n / sum(n)) |>  
filter(breed_rc == "French Bulldog") ->  
nyc_fb2
```

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  ungroup() |>
  complete(zip_code, breed_rc,
           fill = list(n = 0)) |>
# Regroup to get the right denominator
  group_by(breed_rc) |>
  mutate(freq = n / sum(n)) |>
  filter(breed_rc == "French Bulldog") ->
  nyc_fb2

fb_map2 <- left_join(nyc_zips,
                      nyc_fb2,
                      by = "zip_code")
```

Keep the Zero count zips

```
nyc_license |>
  filter(extract_year == 2018) |>
  group_by(breed_rc, zip_code) |>
  tally() |>
  ungroup() |>
  complete(zip_code, breed_rc,
           fill = list(n = 0)) |>
# Regroup to get the right denominator
  group_by(breed_rc) |>
  mutate(freq = n / sum(n)) |>
  filter(breed_rc == "French Bulldog") ->
  nyc_fb2

fb_map2 <- left_join(nyc_zips,
                      nyc_fb2,
                      by = "zip_code")
```

Ready to map, again

```
fb_map2 |> select(zip_code, po_name, borough, breed_rc:freq, geometry)

## Simple feature collection with 262 features and 6 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 7
##       zip_code po_name     borough breed_rc      n    freq               geometry
##           <int> <chr>       <chr>   <chr> <int>  <dbl>             <POLYGON [°]>
## 1       11372 Jackson He... Queens French ...  13 8.02e-3 ((-73.86942 40.74916, -7...
## 2       11004 Glen Oaks    Queens French ...    1 6.17e-4 ((-73.71068 40.75004, -7...
## 3       11040 New Hyde P... Queens French ...    0 0          ((-73.70098 40.7389, -73...
## 4       11426 Bellerose    Queens French ...    1 6.17e-4 ((-73.7227 40.75373, -73...
## 5       11365 Fresh Mead... Queens French ...    7 4.32e-3 ((-73.81089 40.72717, -7...
## 6       11373 Elmhurst     Queens French ...   14 8.64e-3 ((-73.88722 40.72753, -7...
## 7       11001 Floral Park Queens French ...    0 0          ((-73.70098 40.7389, -73...
## 8       11375 Forest Hil... Queens French ...    8 4.94e-3 ((-73.85625 40.73672, -7...
## 9       11427 Queens Vil... Queens French ...    2 1.23e-3 ((-73.74169 40.73682, -7...
## 10      11374 Rego Park    Queens French ...   6 3.70e-3 ((-73.86451 40.73407, -7...
## # ... with 252 more rows
```

This time, a number of previous **NA** rows are now zeroes instead.

Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_
```

Now redraw the map

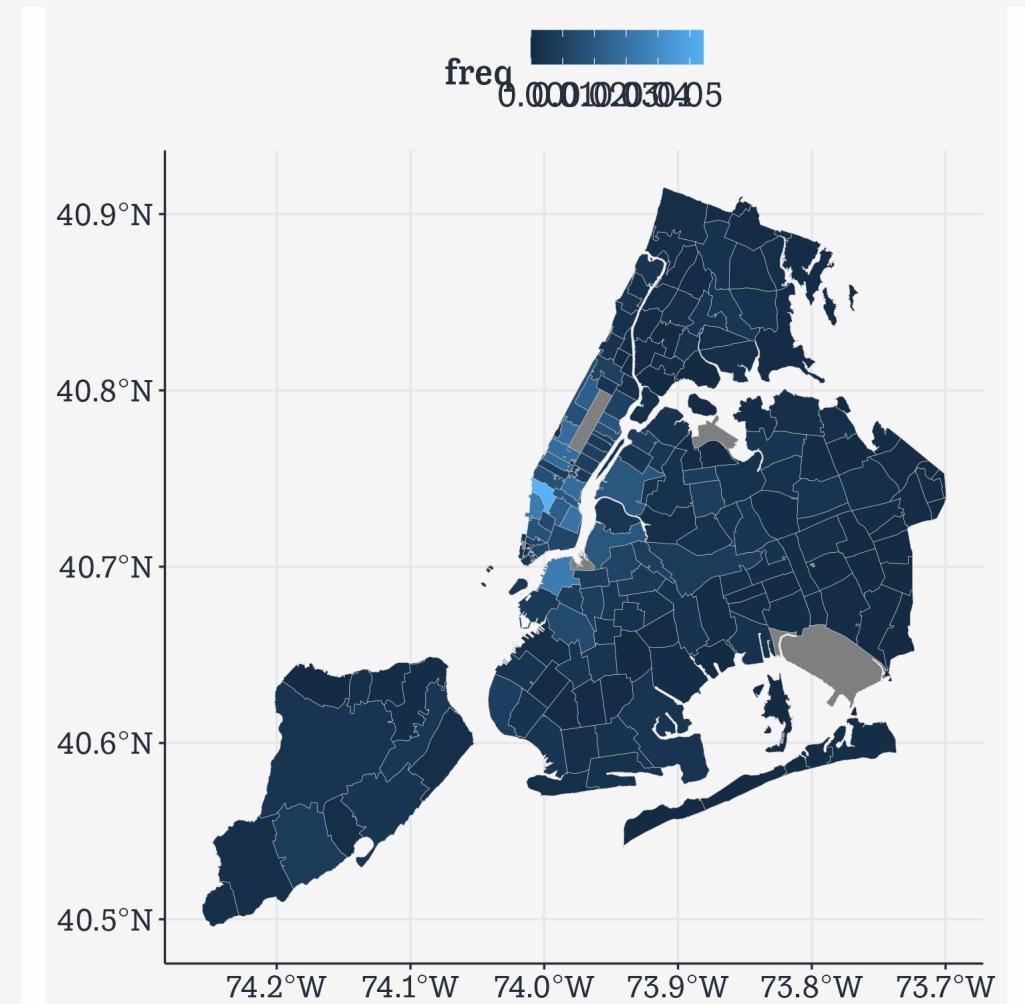
```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip")
fb_map2
## Simple feature collection with 262 features and 14 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -74.25576 ymin: 40.49584 xmax: -73.6996 ymax: 40.91517
## Geodetic CRS: WGS 84
## # A tibble: 262 × 15
##       objectid    zip_code po_name      state borough st_fips cty_fips bld_gpostal_
##           <int>     <int> <chr>      <chr> <chr>   <chr>   <chr>   <chr>
## 1           1      11372 Jackson He... NY    Queens    36    081
## 2           2      11004 Glen Oaks    NY    Queens    36    081
## 3           3      11040 New Hyde P... NY    Queens    36    081
## 4           4      11426 Bellerose    NY    Queens    36    081
## 5           5      11365 Fresh Mead... NY    Queens    36    081
## 6           6      11373 Elmhurst    NY    Queens    36    081
## 7           7      11001 Floral Park NY    Queens    36    081
## 8           8      11375 Forest Hil... NY    Queens    36    081
## 9           9      11427 Queens Vil... NY    Queens    36    081
## 10          10     11374 Rego Park   NY    Queens    36    081
## # ... with 252 more rows, and 7 more variables: shape_leng <dbl>,
## #   shape_area <dbl>, x_id <chr>, geometry <POLYGON [°]>, breed_rc <chr>,
## #   n <int>, freq <dbl>
```

Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq))
```

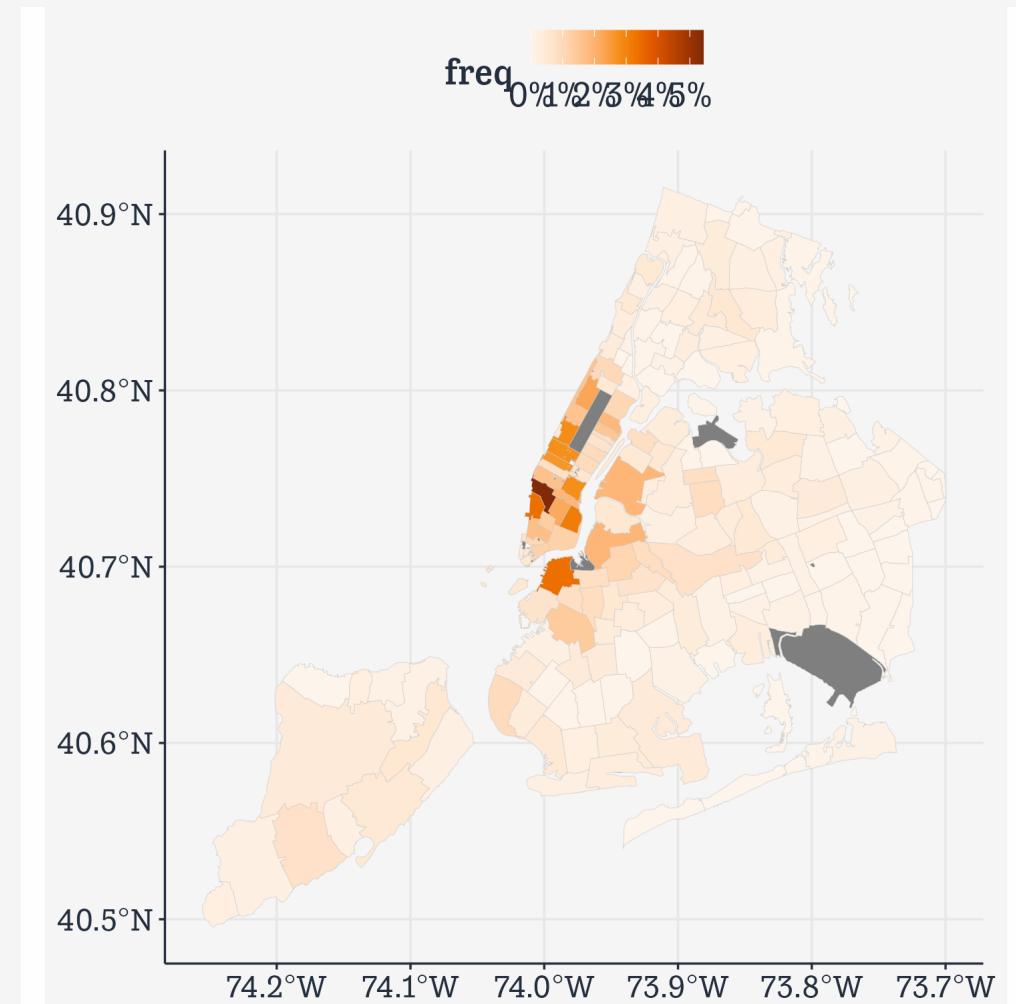
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1)
```



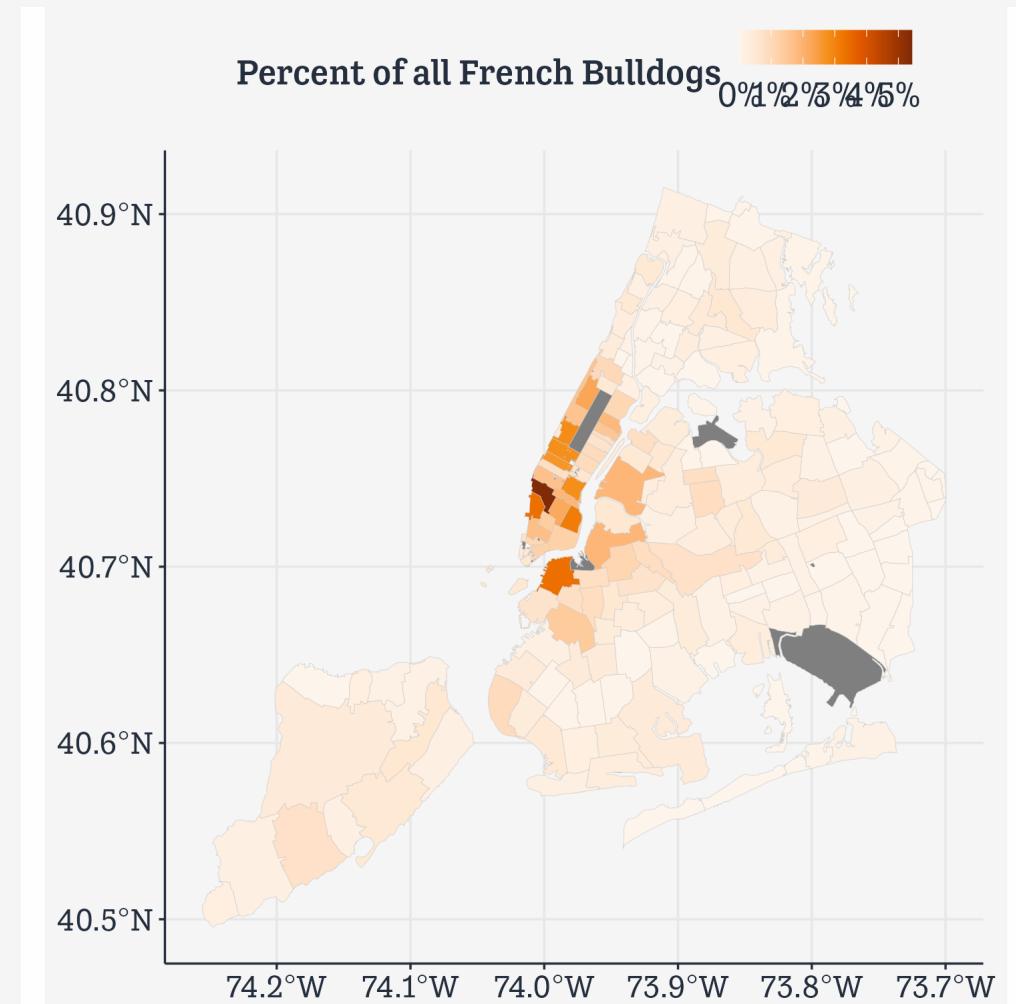
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent())
```



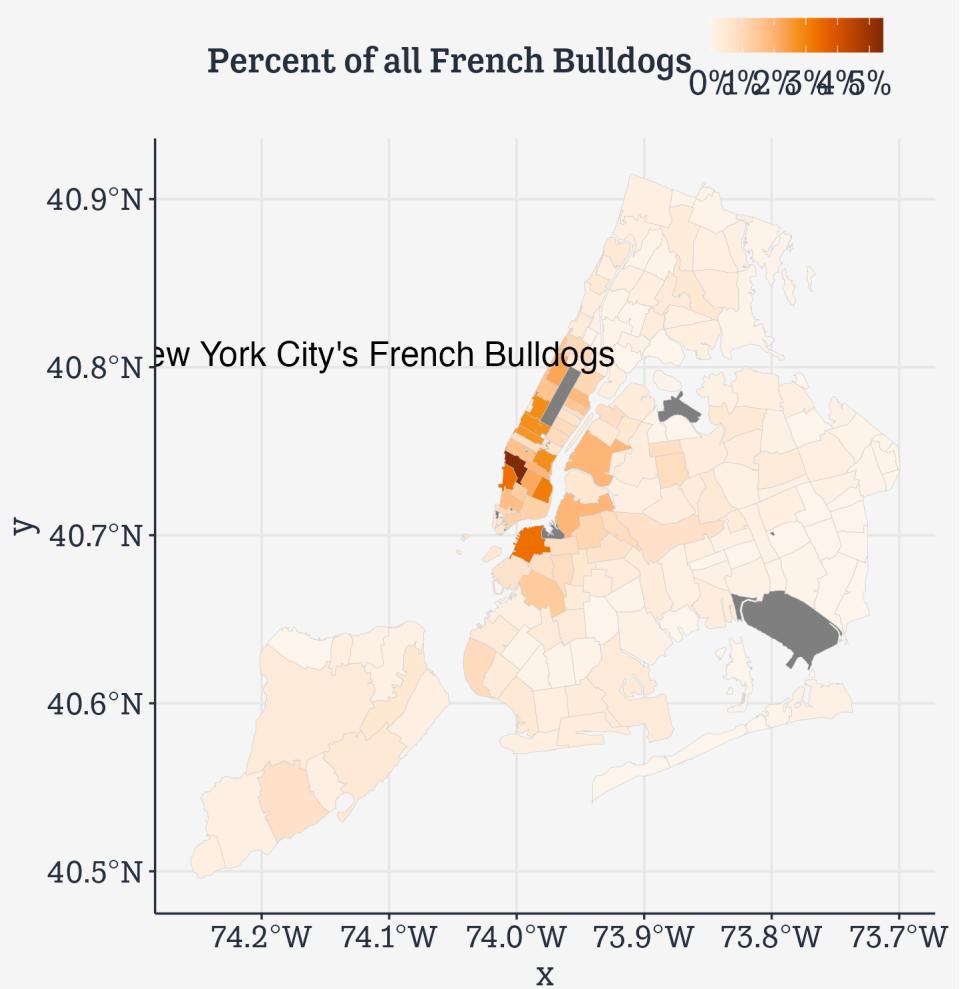
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs")
```



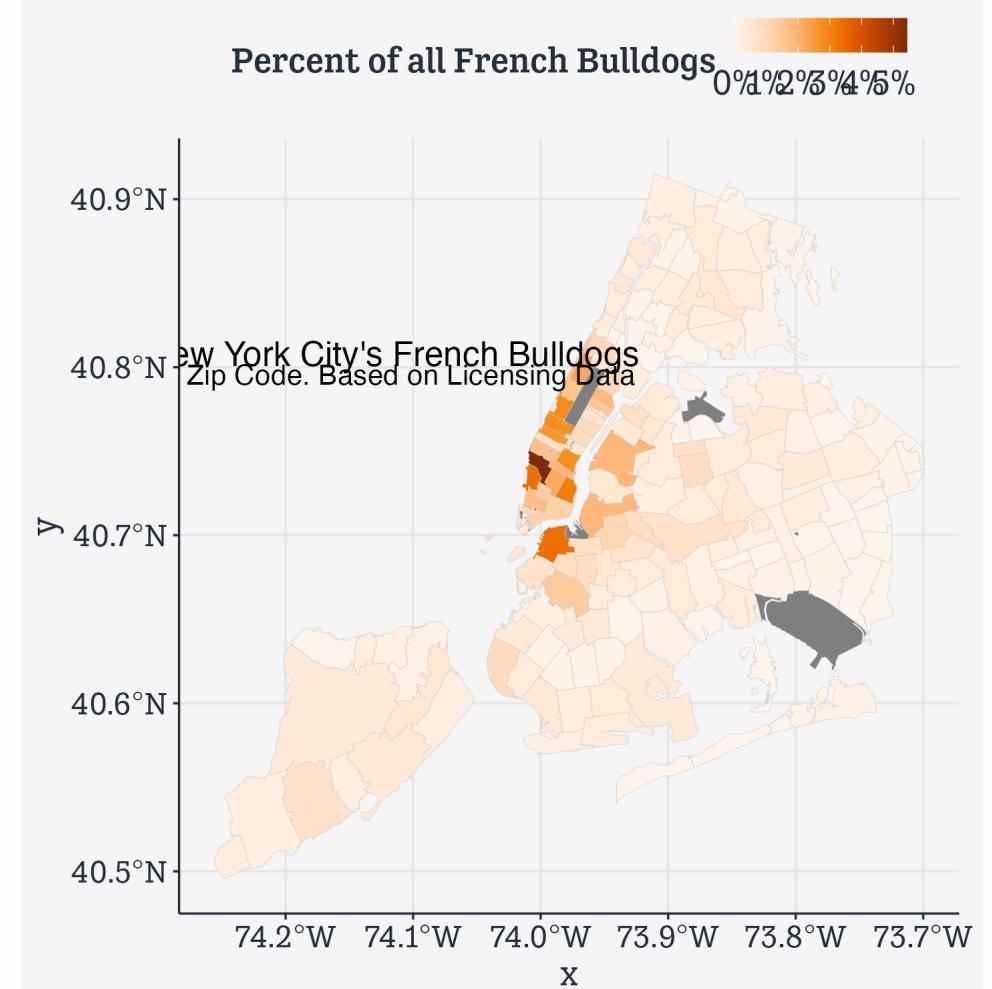
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82 - 0.012,
          label = "New York City's French Bulldog",
          size = 6)
```



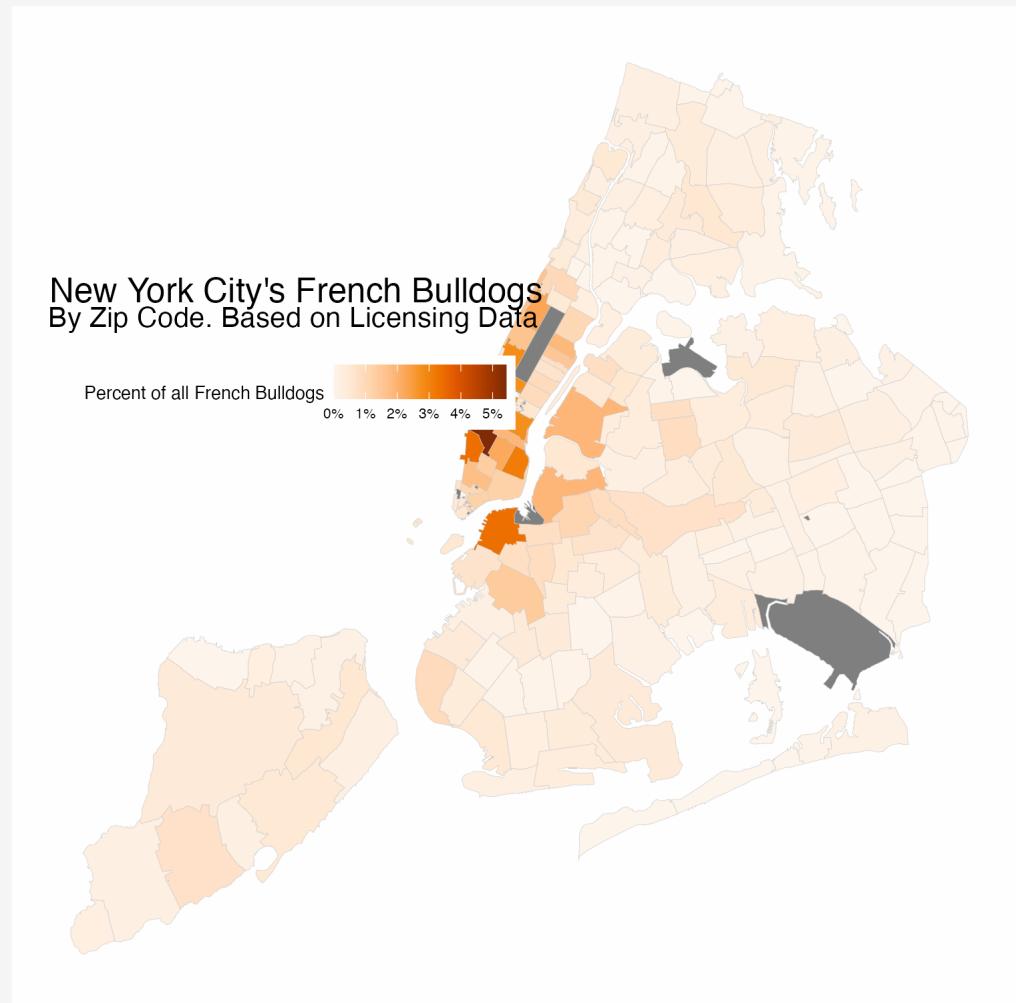
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5)
```



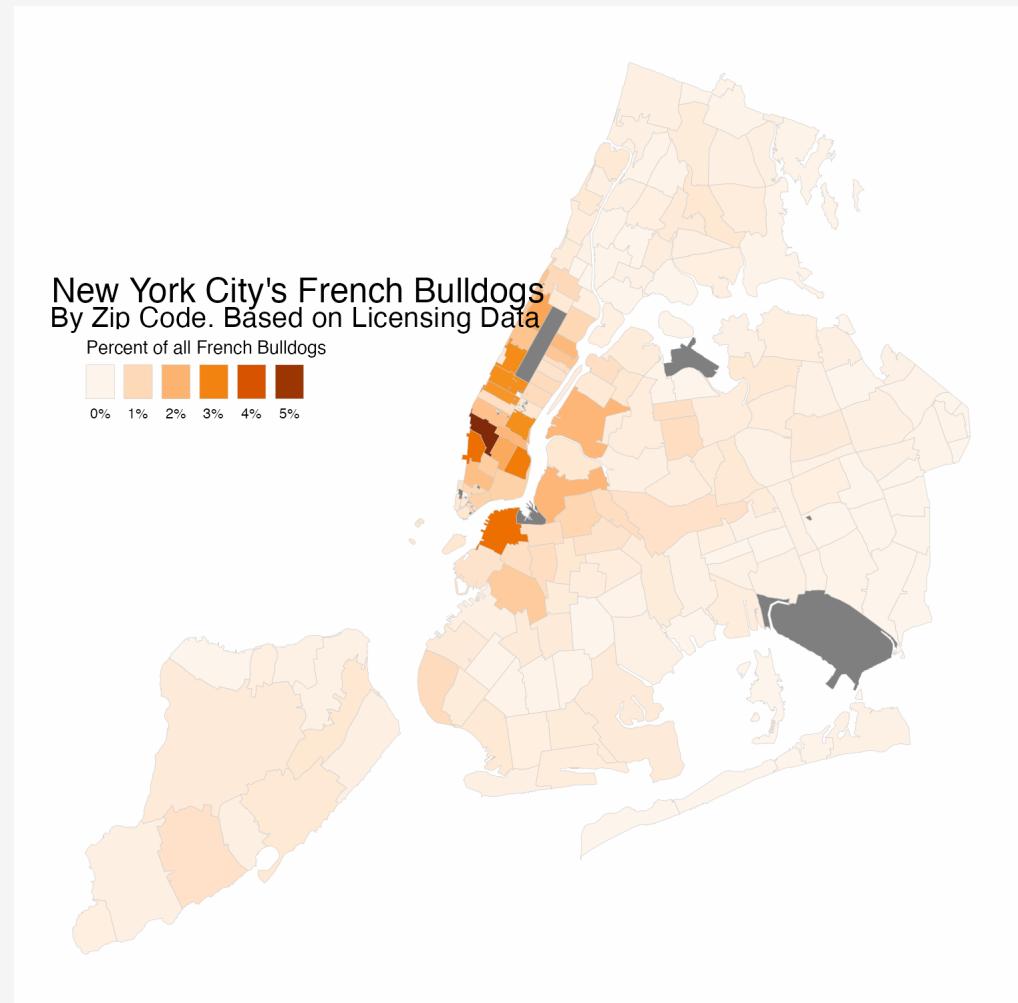
Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing",
          size = 5) +
  kjhslides::kjh_theme_nymap()
```



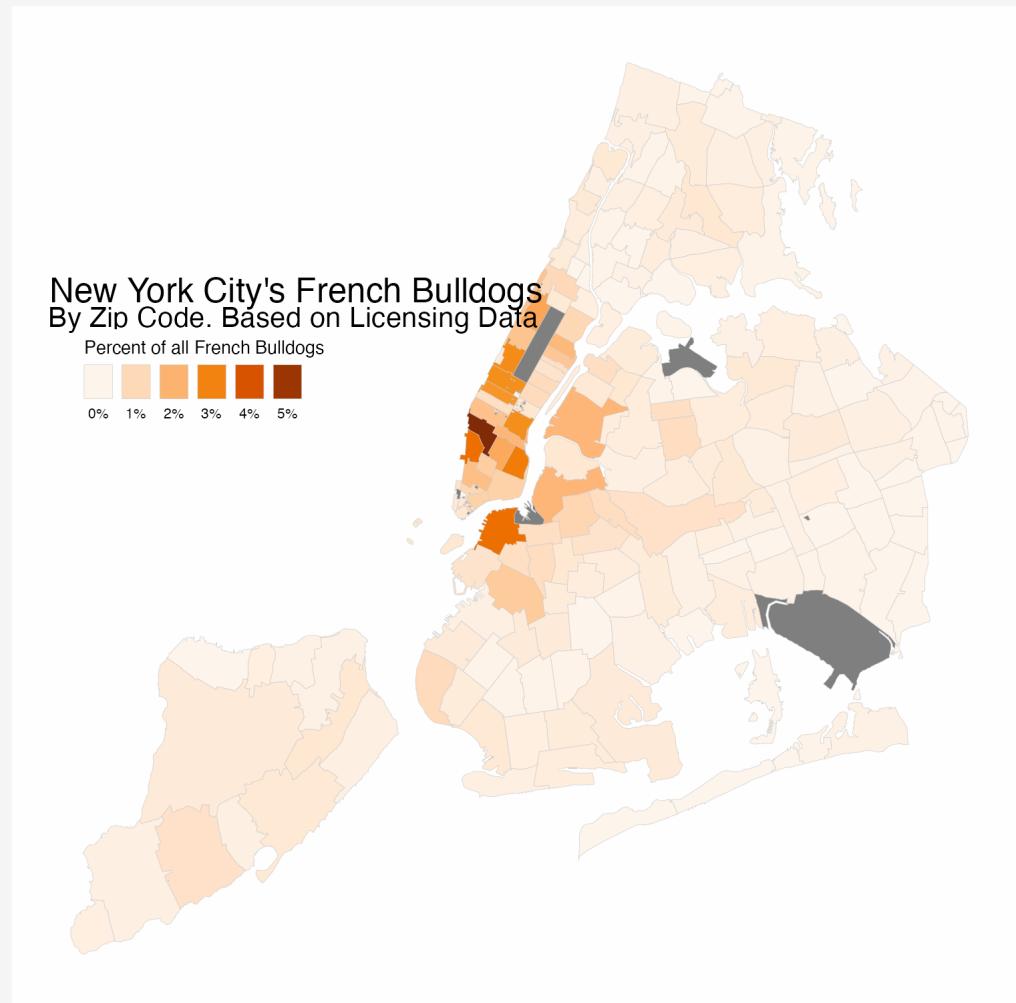
Now redraw the map

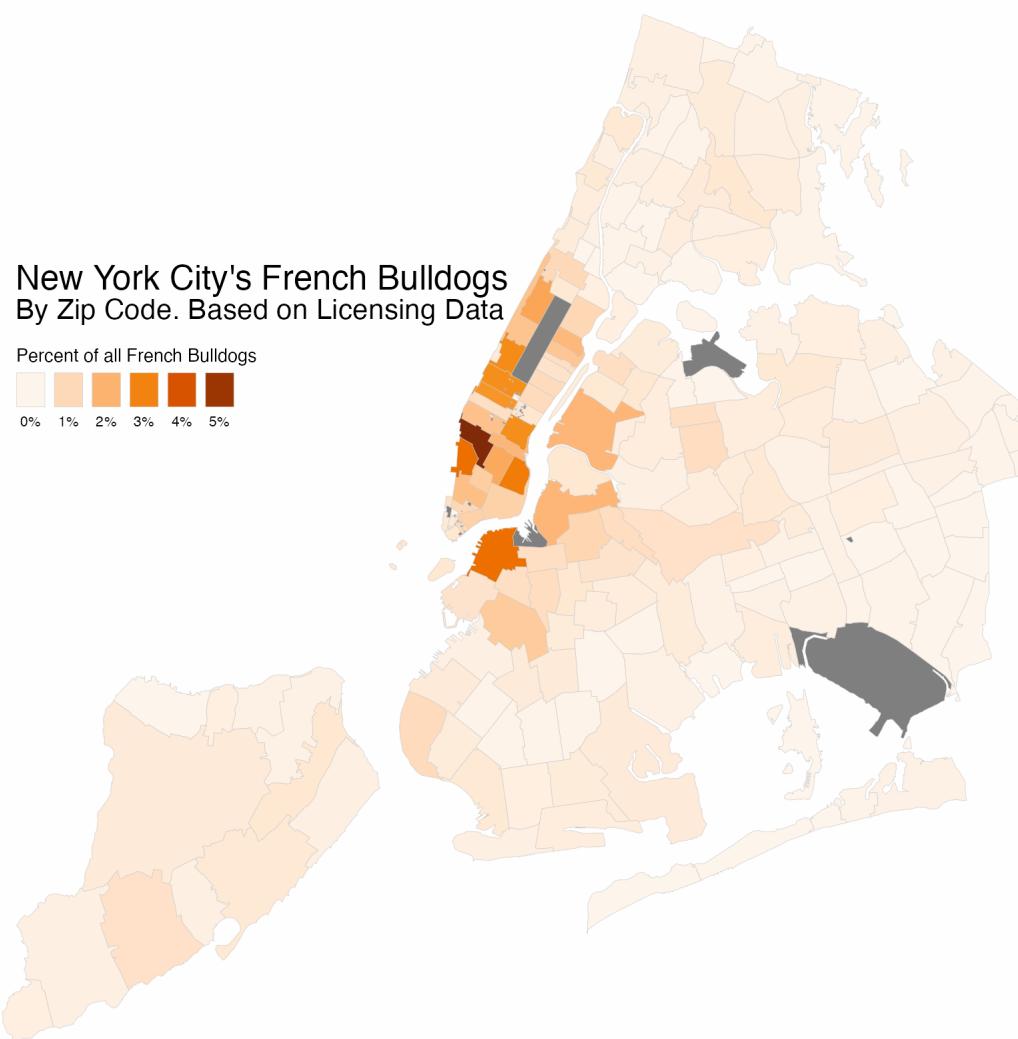
```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5) +
  kjhslides::kjh_nymap() +
  guides(fill =
    guide_legend(title.position = "top",
                label.position = "bottom",
                keywidth = 1,
                nrow = 1))
```



Now redraw the map

```
fb_map2 <- left_join(nyc_zips, nyc_fb2, by = "zip_"
fb_map2 |>
  ggplot(mapping = aes(fill = freq)) +
  geom_sf(color = "gray80", size = 0.1) +
  scale_fill_continuous_sequential(
    palette = "Oranges",
    labels = scales::label_percent()) +
  labs(fill = "Percent of all French Bulldogs") +
  annotate(geom = "text",
          x = -74.145 + 0.029,
          y = 40.82-0.012,
          label = "New York City's French Bulldog",
          size = 6) +
  annotate(geom = "text",
          x = -74.1468 + 0.029,
          y = 40.7955,
          label = "By Zip Code. Based on Licensing Data",
          size = 5) +
  kjhslides::kjh_theme_nymap() +
  guides(fill =
            guide_legend(title.position = "top",
                         label.position = "bottom",
                         keywidth = 1,
                         nrow = 1))
```





Zero areas properly zero, missing areas properly missing.