

Data Visualization - 7.

Make Maps (1)

Kieran Healy
Code Horizons

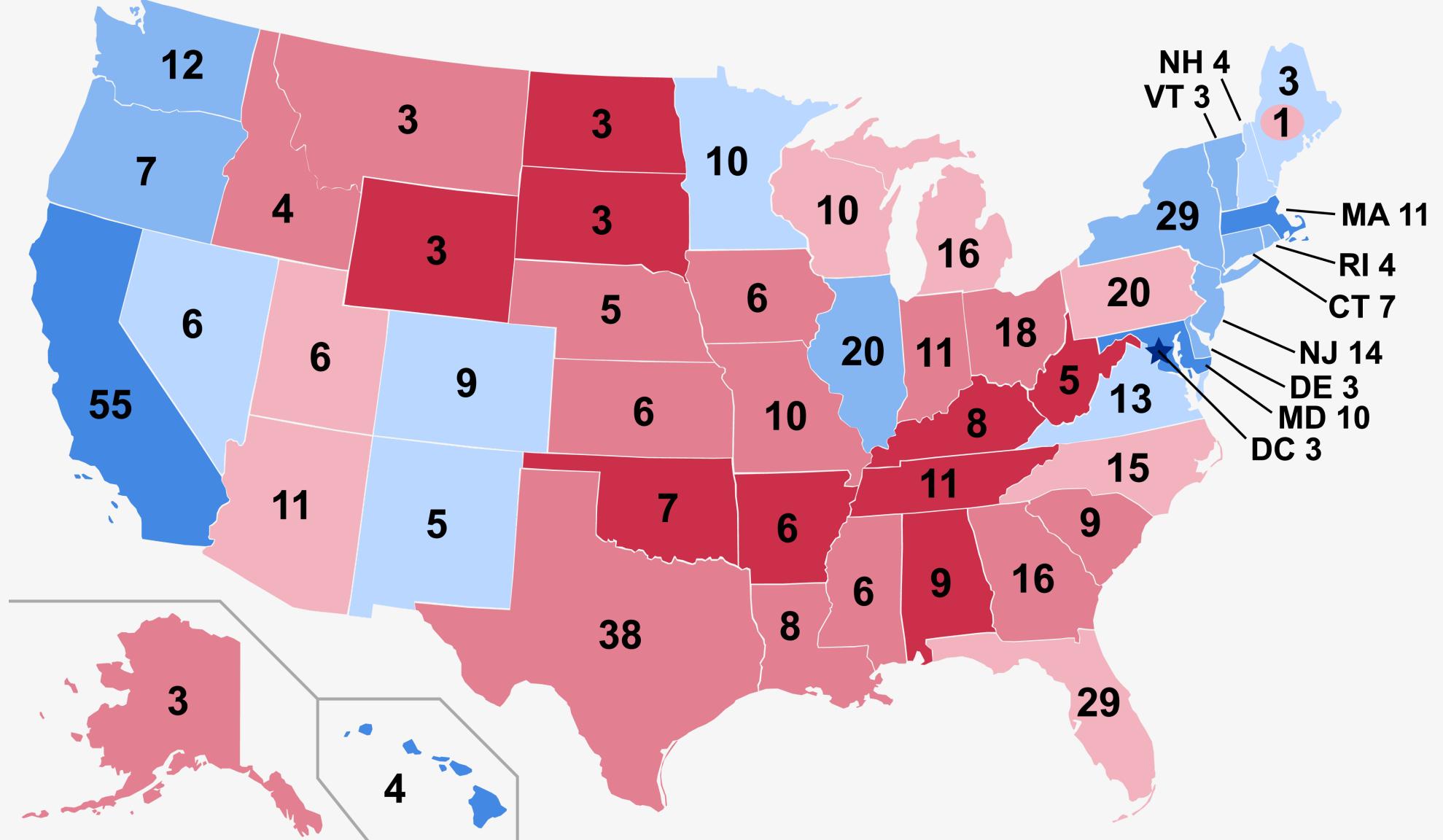
December 10, 2023

Making Maps

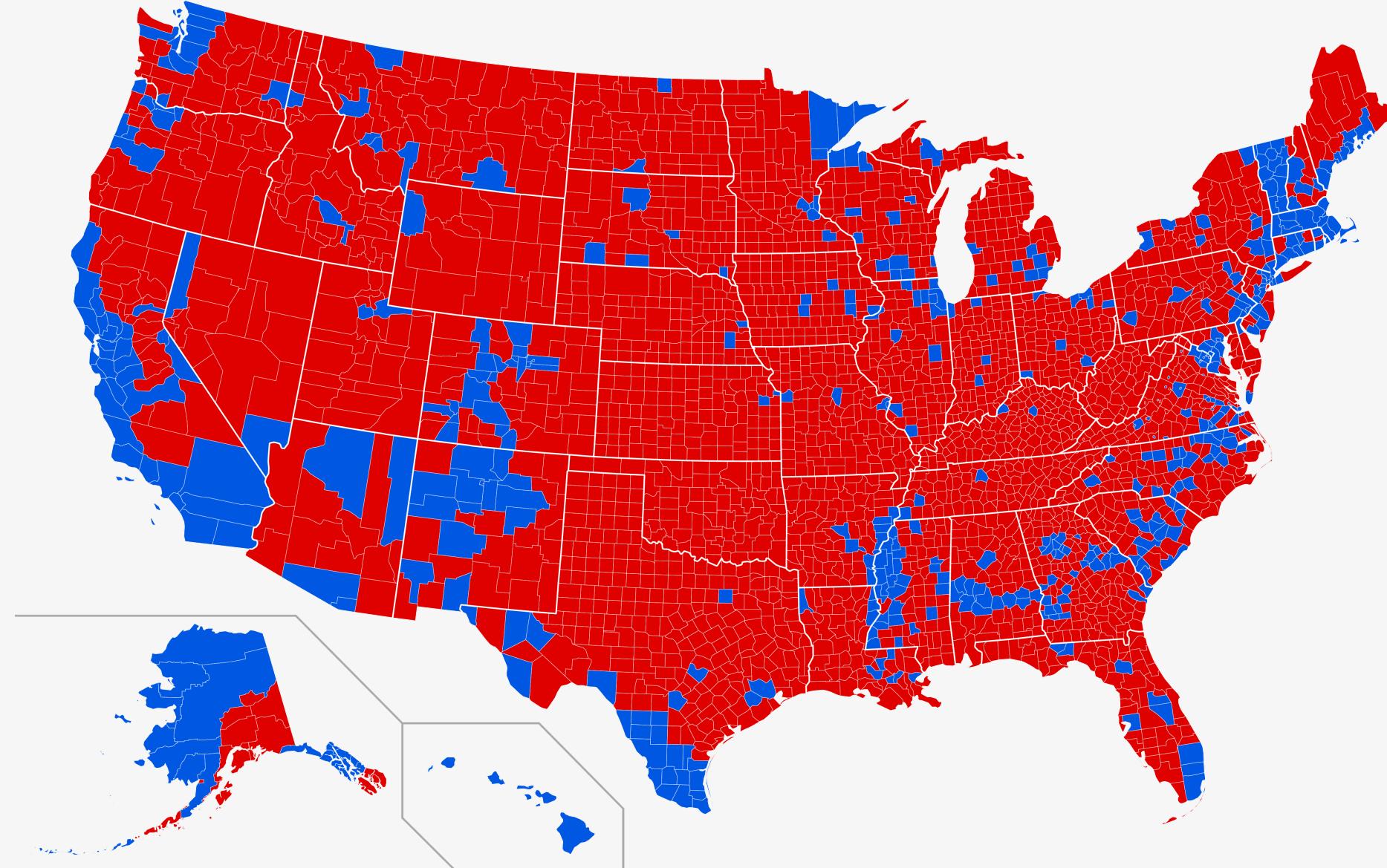
Load our packages

```
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
```

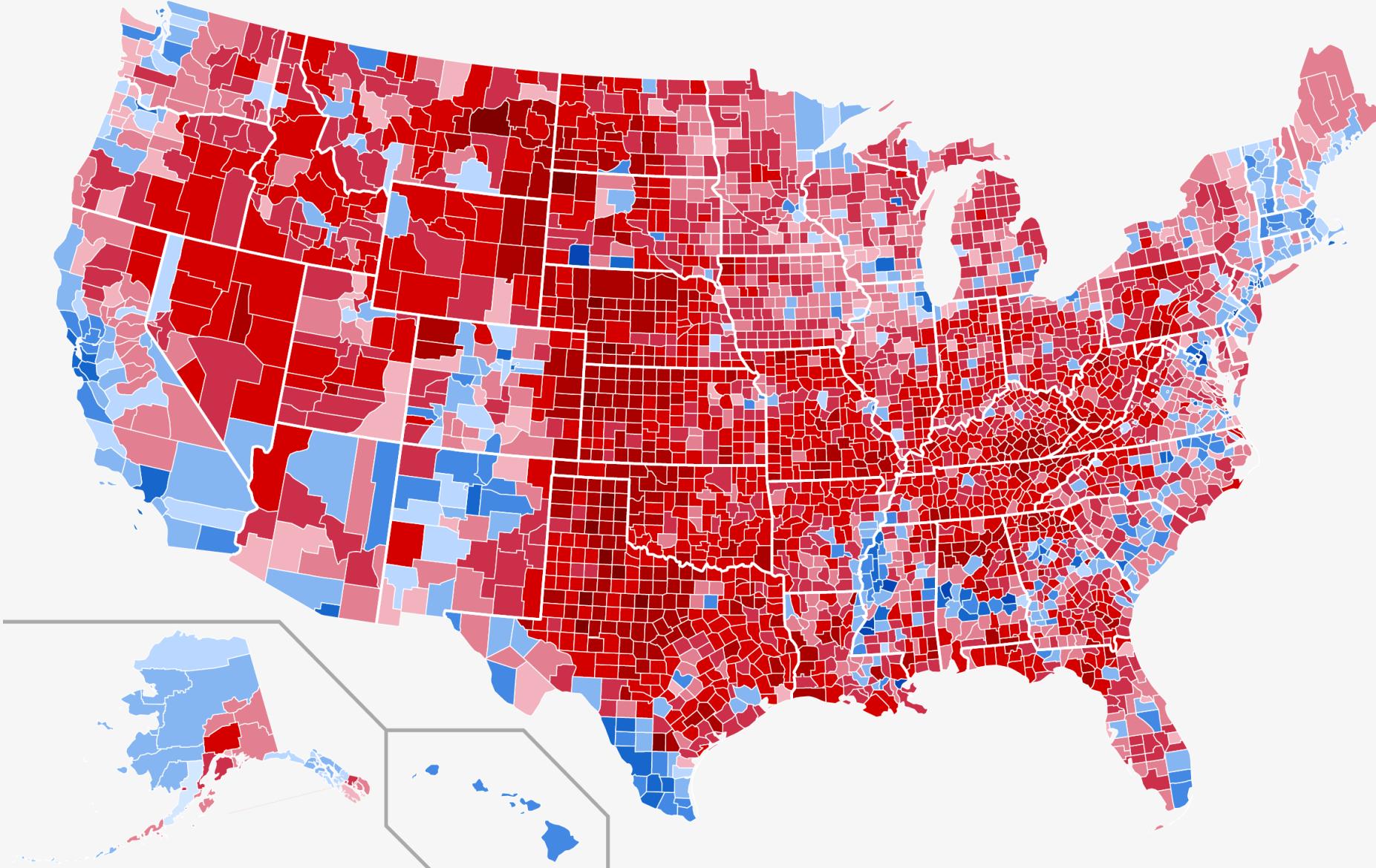

Choropleths



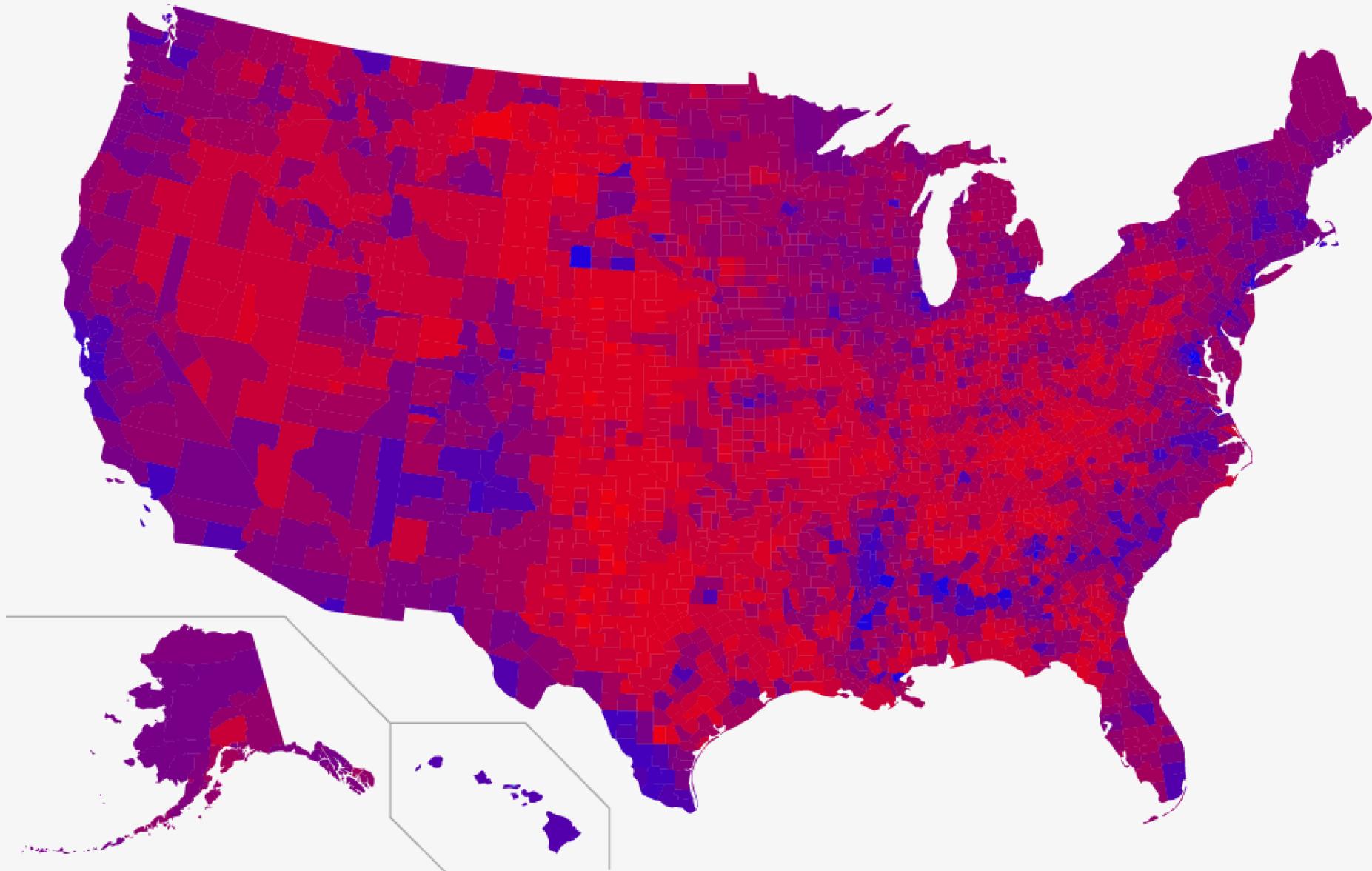
State-level; vote share; diverging; binned into four categories.



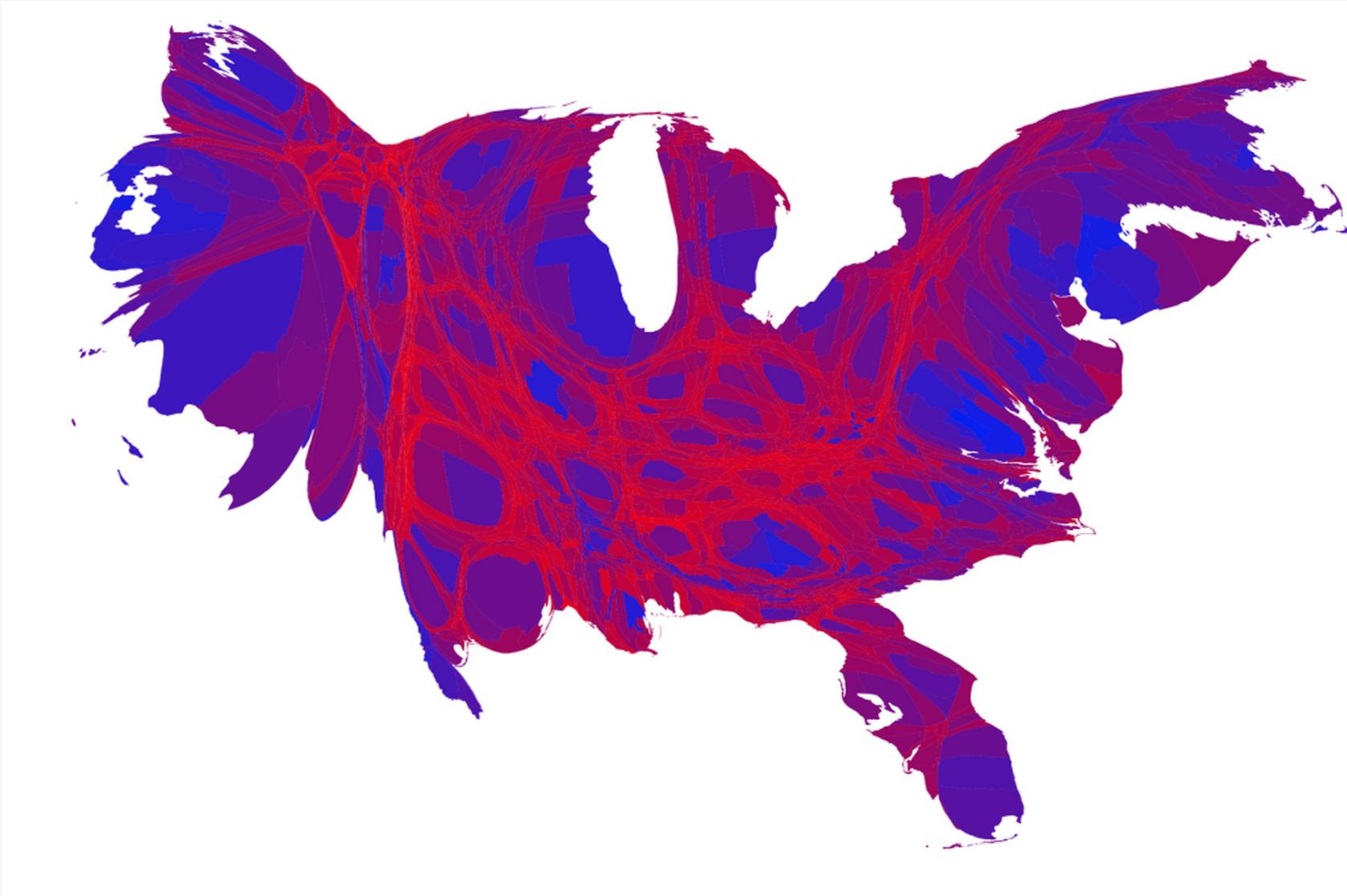
County level; winner only



County level vote share; diverging; binned into six categories

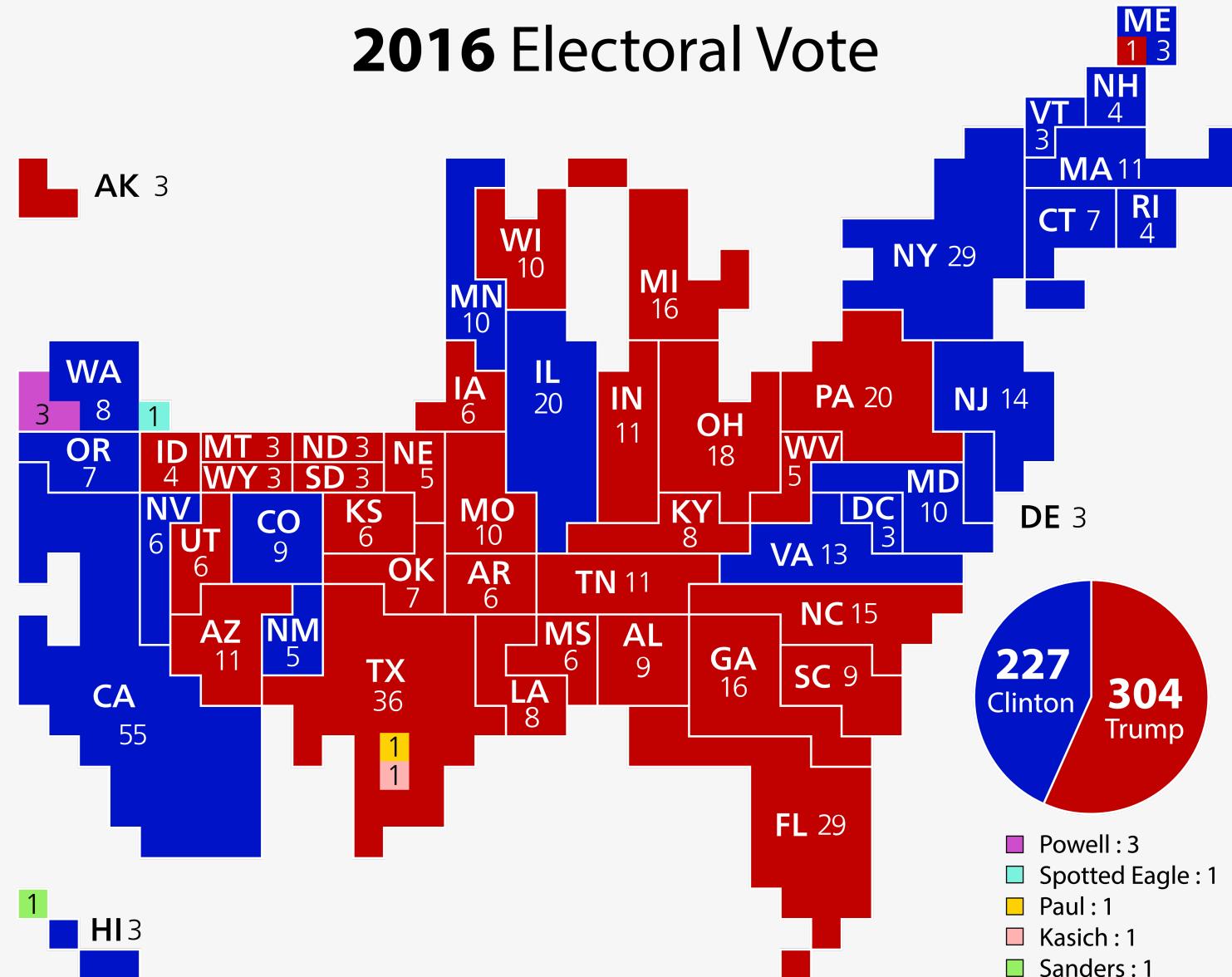


County level vote share; diverging continuous; purple midpoint

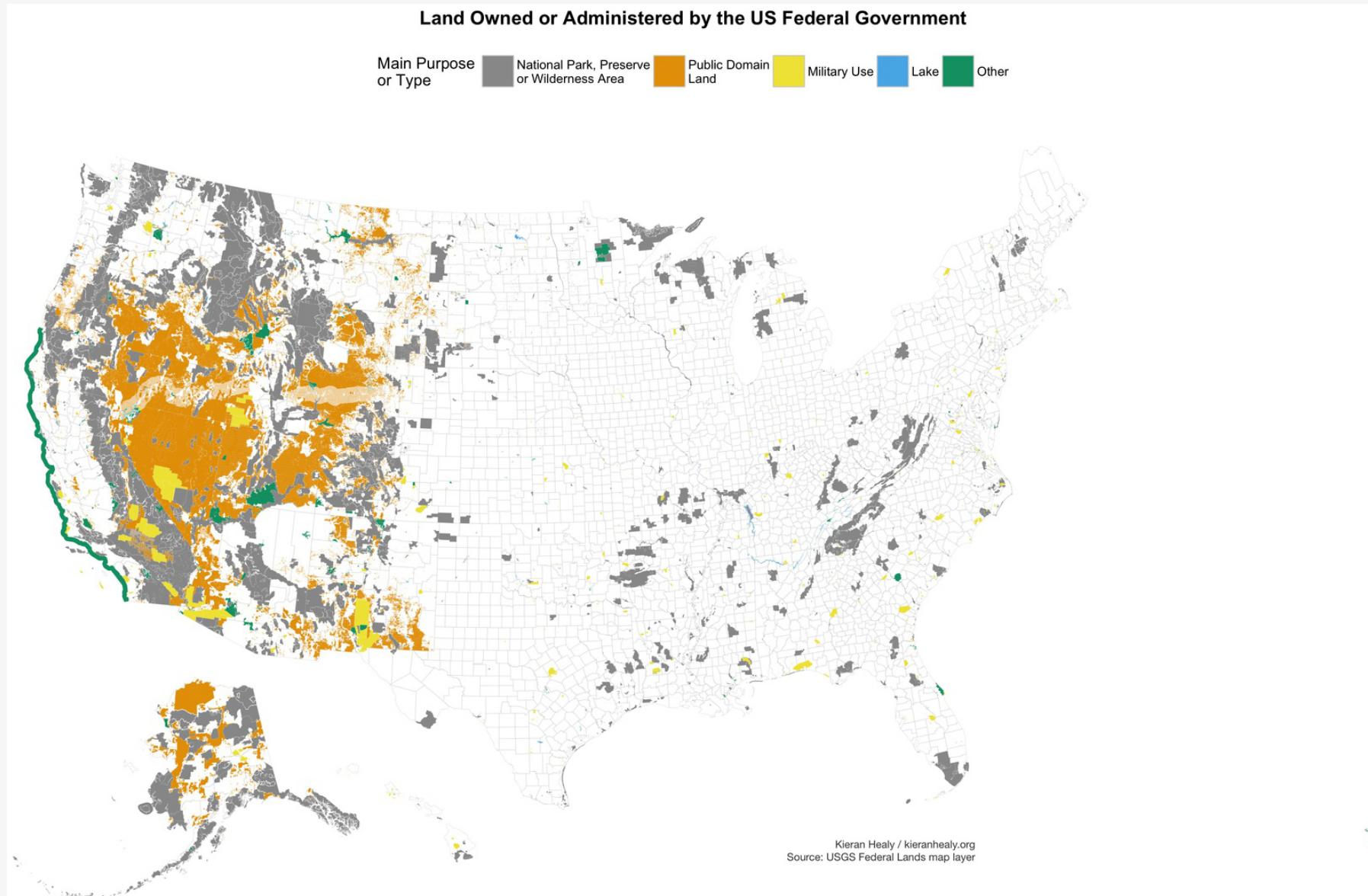


County level vote share; purple midpoint; county area deformed in proportion to population. By Mark Newman

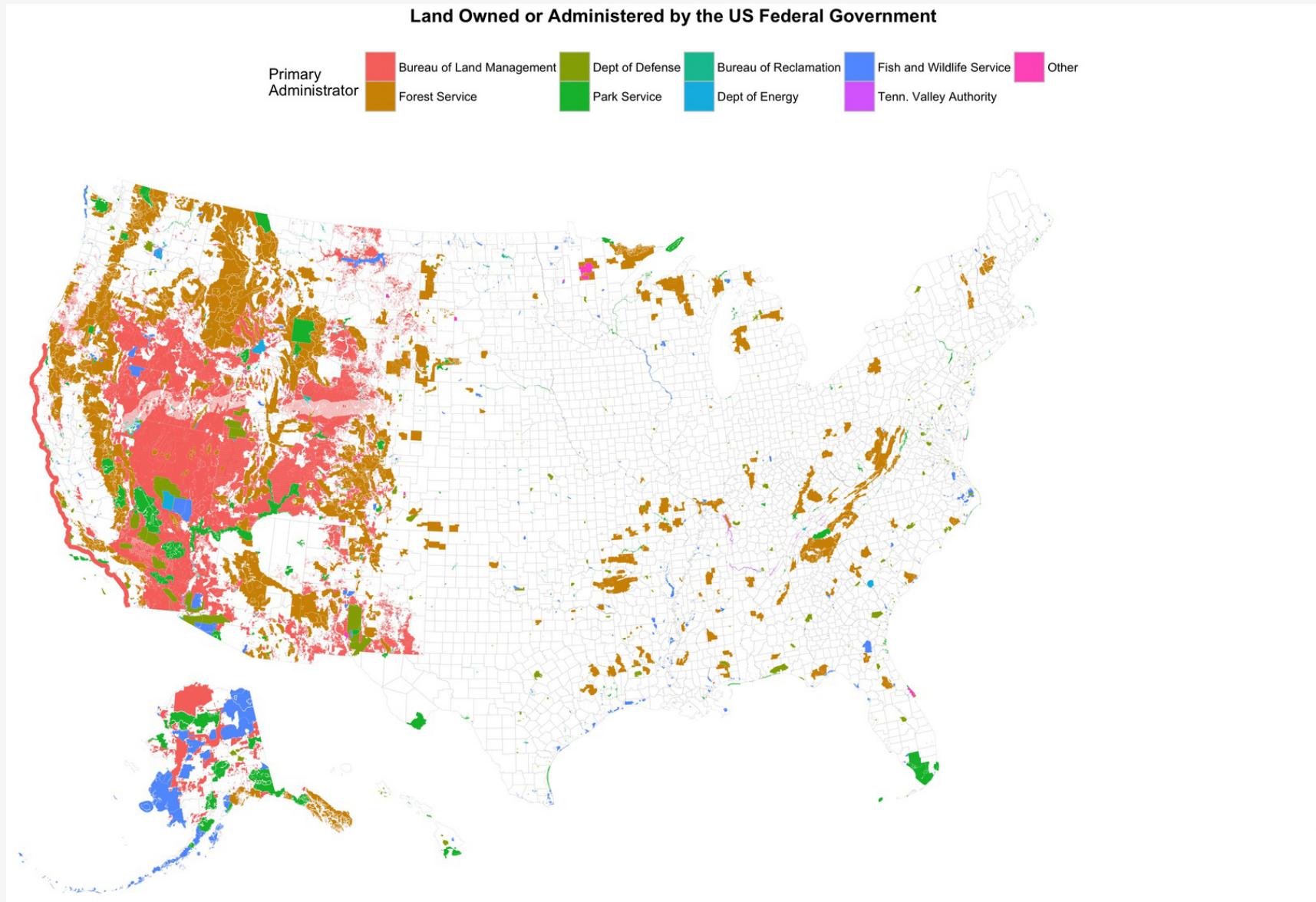
2016 Electoral Vote



Electoral college cartogram (NYT)

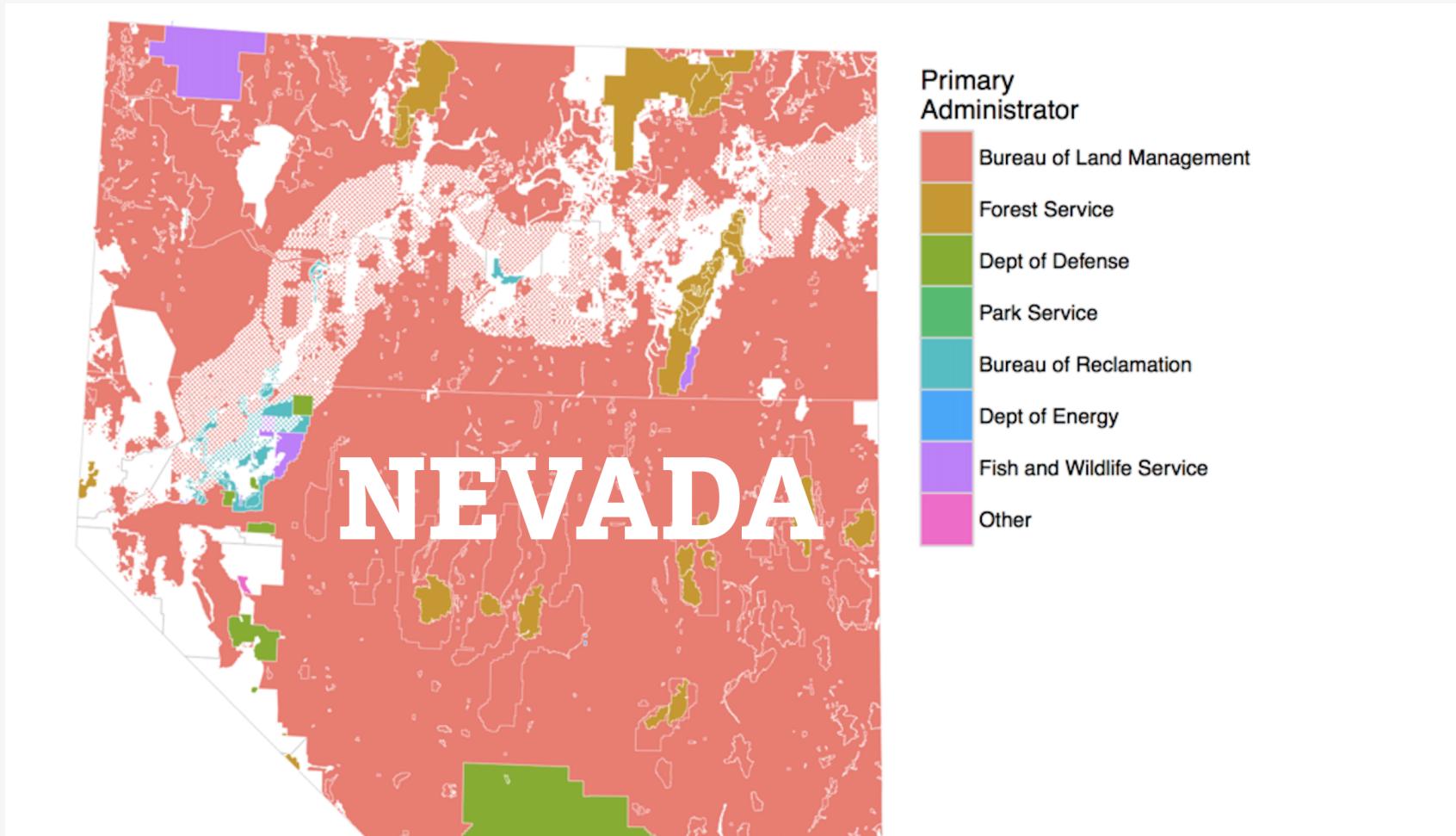


Pretty, Big, and Pretty Empty

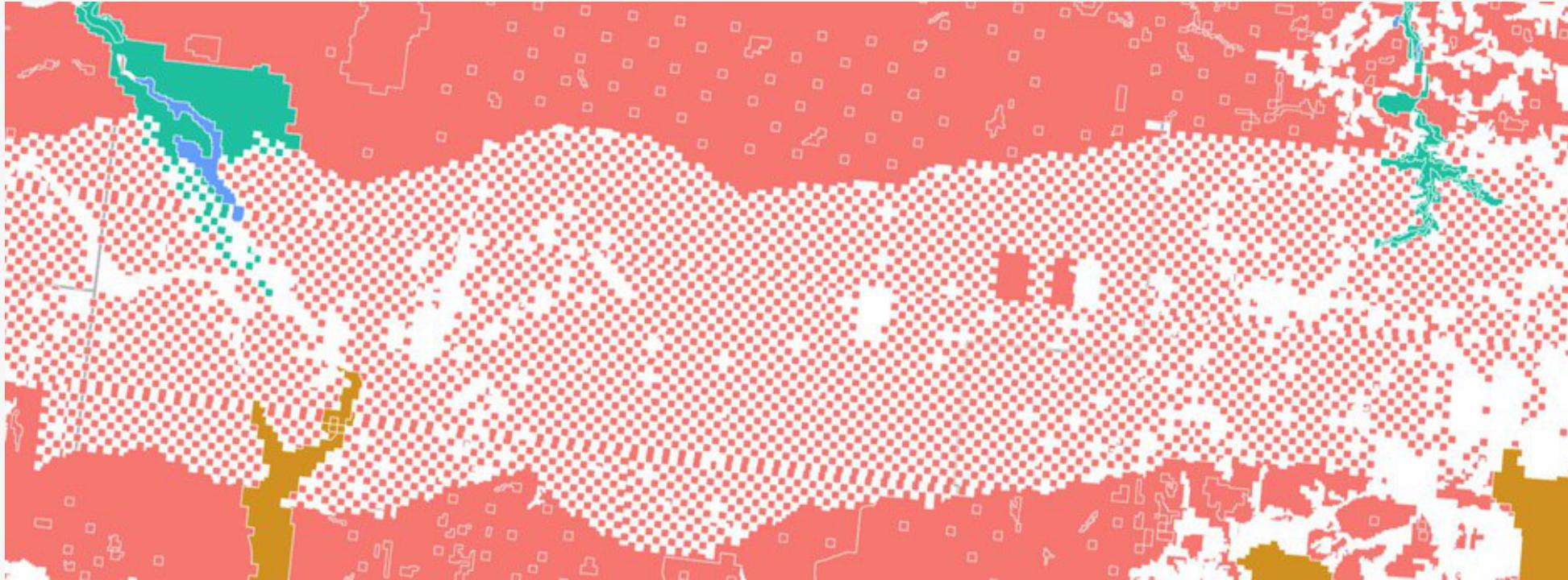


Pretty, Big, and Pretty Empty

Aside: What the hell's that?

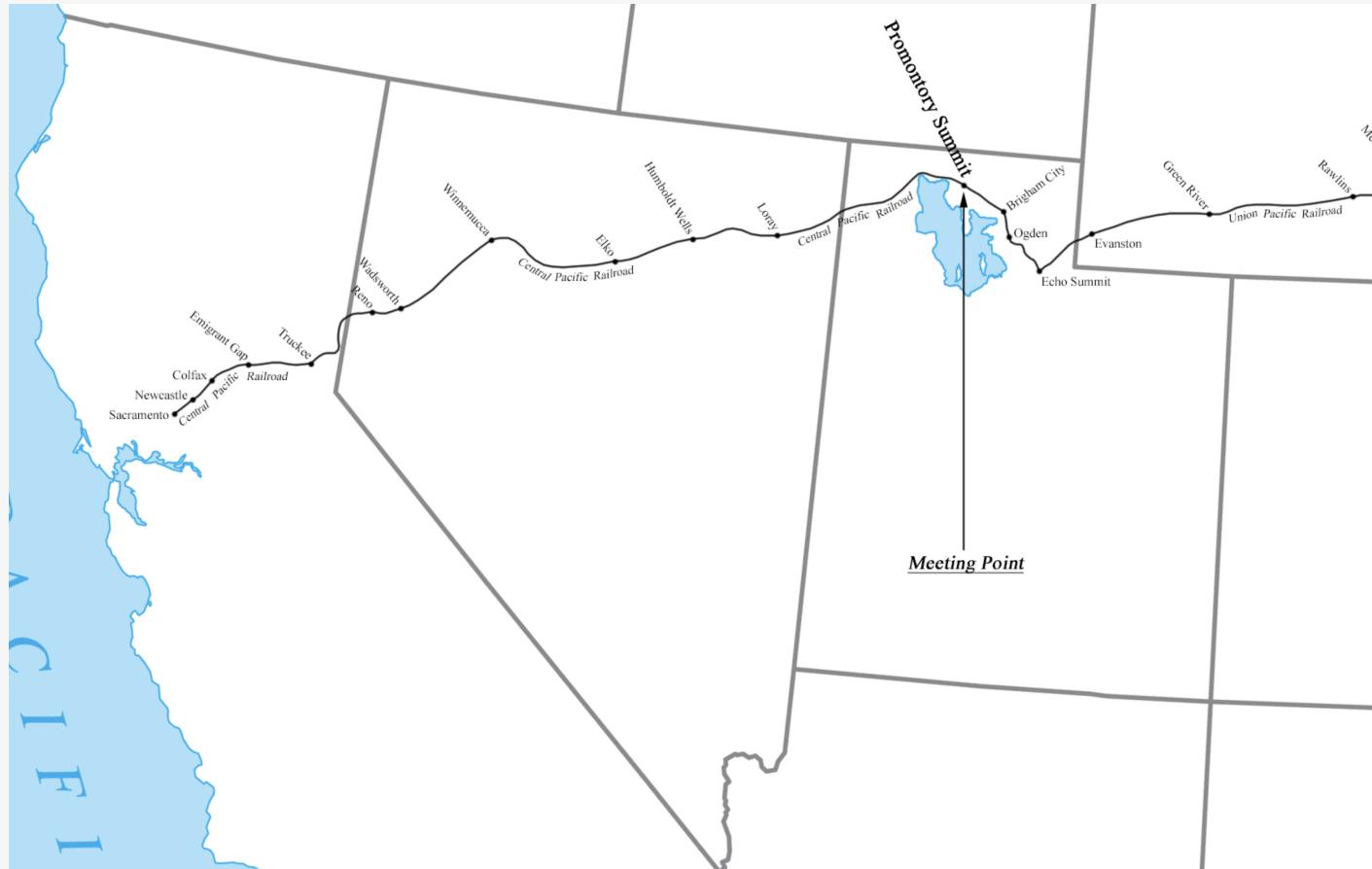


Zoom and Enhance



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.

Maps as polygons

Packages

```
library(here)      # manage file paths
library(socviz)    # data and some useful functions
library(tidyverse) # your friend and mine
library(maps)      # Some basic maps
library(ggforce)   # ggplot extensions
```

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>

What is this, at root?

```
us_states
```

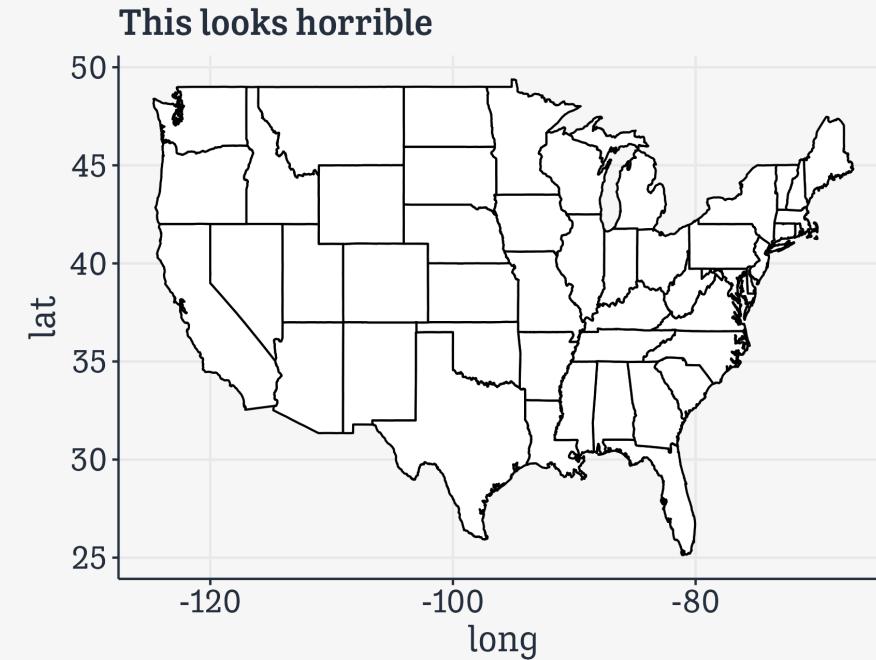
```
# A tibble: 15,537 × 6
  long     lat group order region subregion
  <dbl>   <dbl> <dbl> <int> <chr>   <chr>
1 -87.5  30.4     1     1 alabama <NA>
2 -87.5  30.4     1     2 alabama <NA>
3 -87.5  30.4     1     3 alabama <NA>
4 -87.5  30.3     1     4 alabama <NA>
5 -87.6  30.3     1     5 alabama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

It's a series of rows defining `x` and `y` coordinatates 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.

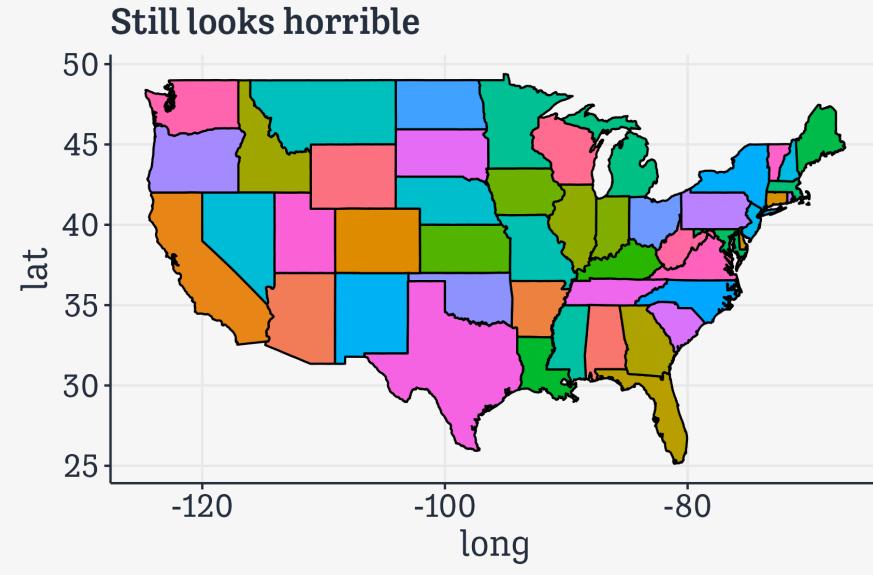
Like this, with `geom_polygon()`

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



We can show 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 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

```
1 us_states ← as_tibble(map_data("state"))
```

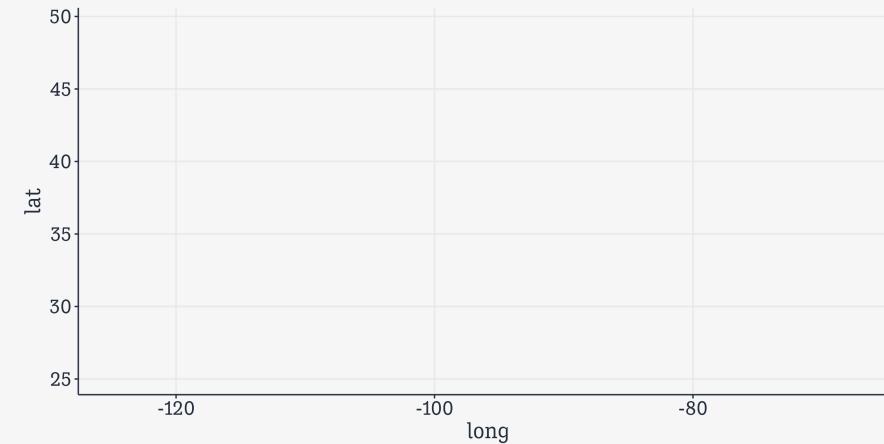
Fix the projection

```
1 us_states ← as_tibble(map_data("state"))
2
3 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>
# i 15,527 more rows
```

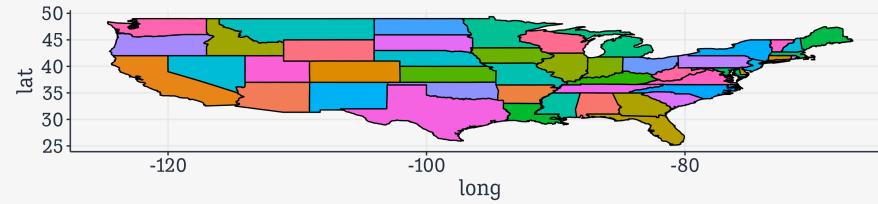
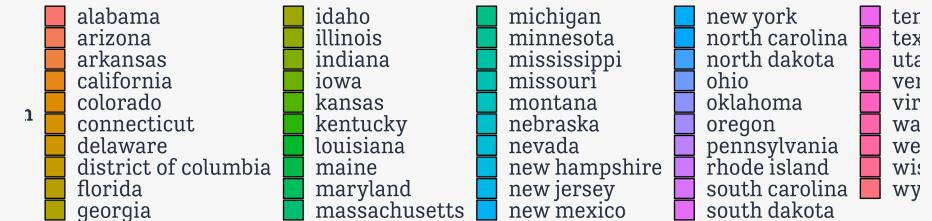
Fix the projection

```
1 us_states ← as_tibble(map_data("state"))
2
3 us_states ▷
4   ggplot(mapping = aes(x = long,
5                         y = lat,
6                         fill = region,
7                         group = group))
```



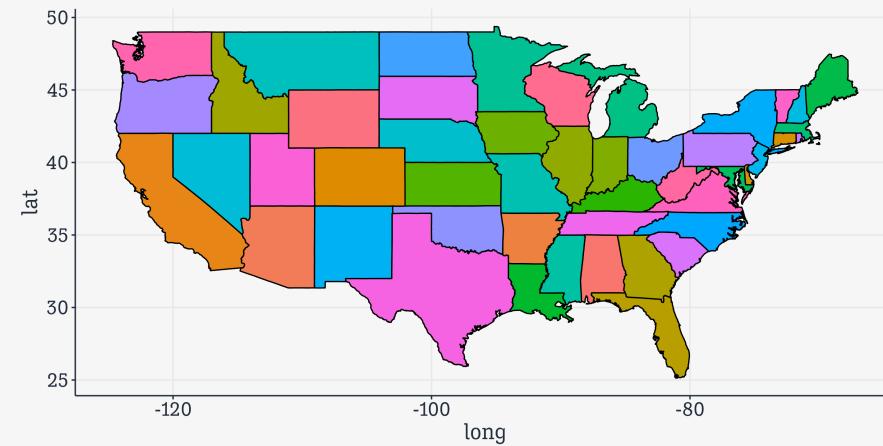
Fix the projection

```
1 us_states ← as_tibble(map_data("state"))
2
3 us_states >
4   ggplot(mapping = aes(x = long,
5                         y = lat,
6                         fill = region,
7                         group = group)) +
8   geom_polygon(color = "black")
```



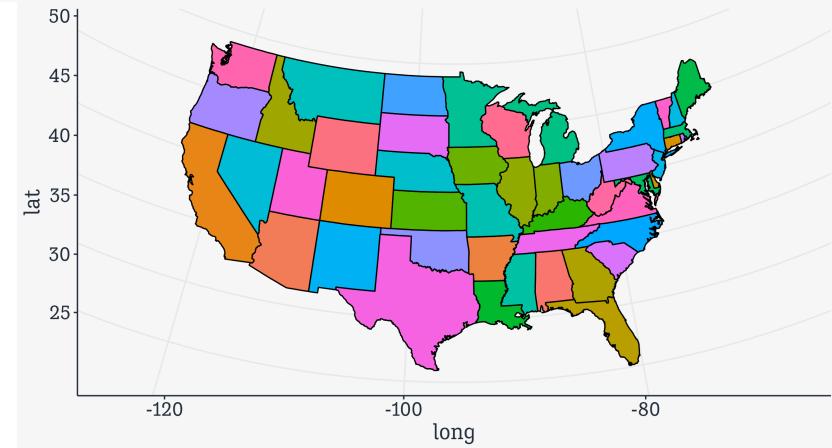
Fix the projection

```
1 us_states <- as_tibble(map_data("state"))
2
3 us_states >>
4   ggplot(mapping = aes(x = long,
5                         y = lat,
6                         fill = region,
7                         group = group)) +
8   geom_polygon(color = "black") +
9   guides(fill = "none")
```

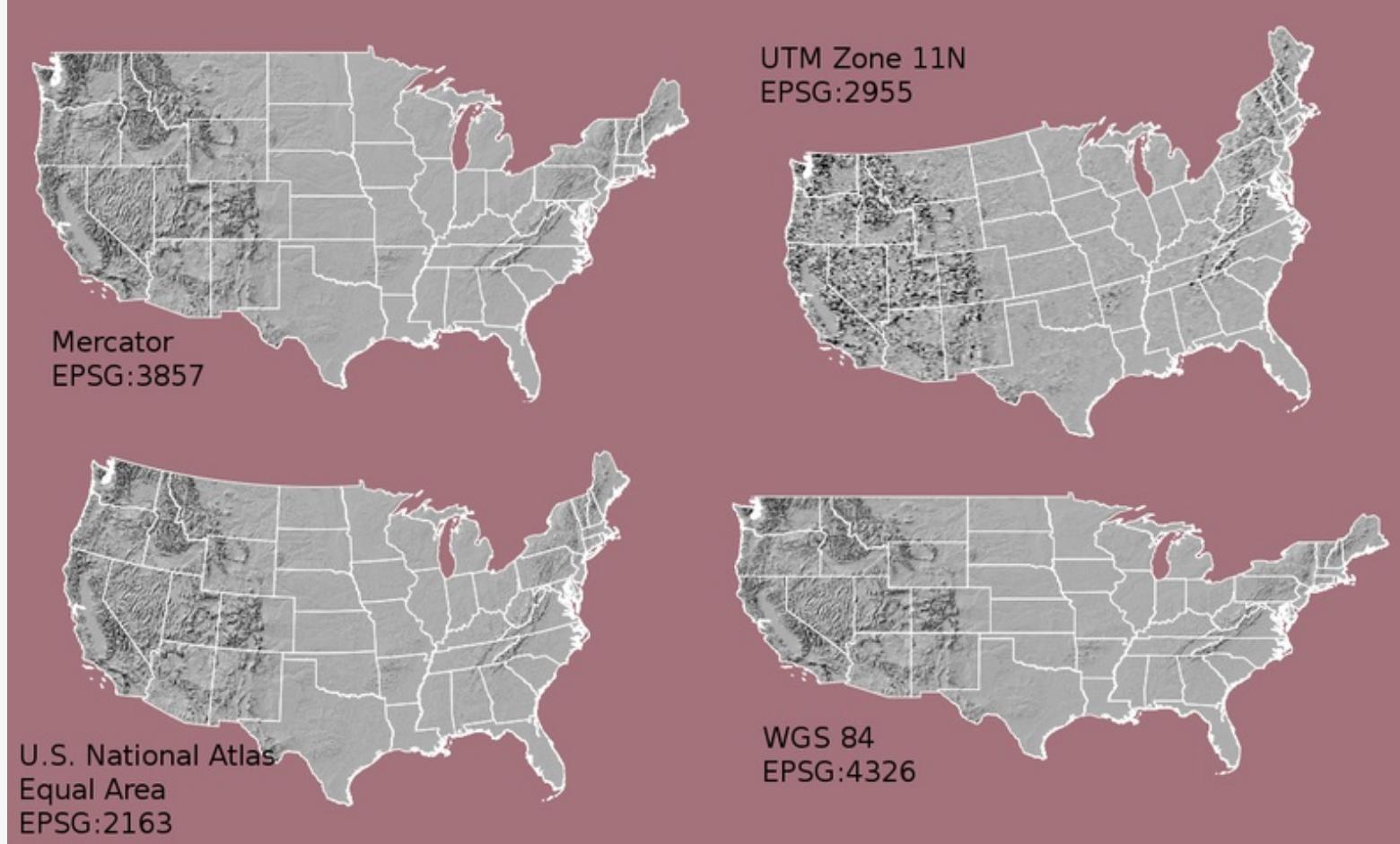


Fix the projection

```
1 us_states <- as_tibble(map_data("state"))
2
3 us_states >>
4   ggplot(mapping = aes(x = long,
5                         y = lat,
6                         fill = region,
7                         group = group)) +
8   geom_polygon(color = "black") +
9   guides(fill = "none") +
10  coord_map(projection = "albers",
11             lat0 = 39,
12             lat1 = 45)
```



U.S. Map Projections

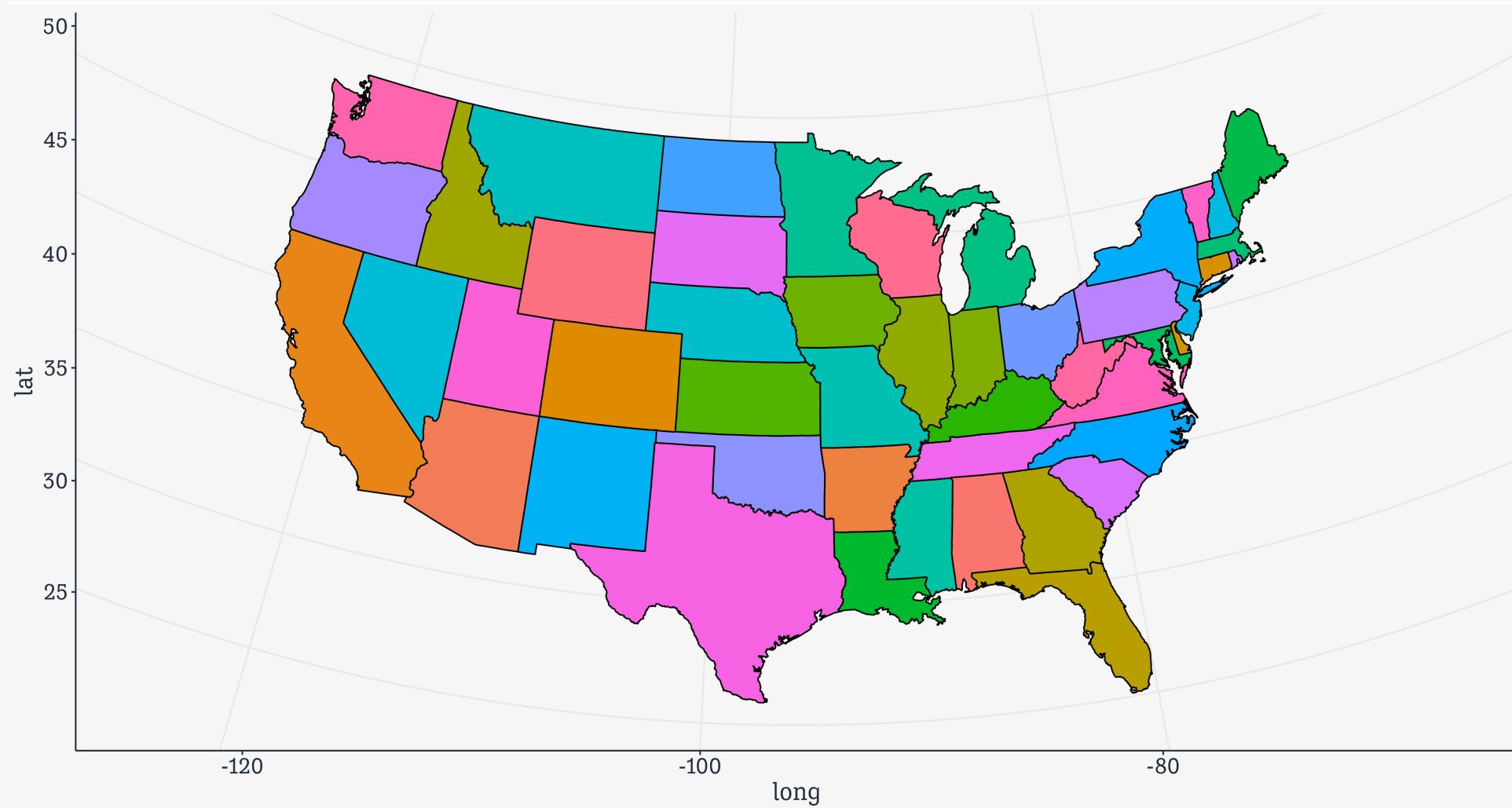


A selection of projections

U.S. Map Projections



Albers is the standard



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>
# i 15,527 more rows
```

election

```
# A tibble: 51 × 22
  state      st     fips total_vote vote_margin
  <chr>      <chr> <dbl>      <dbl>      <dbl>
  winner     party  pct_margin r_points
  <chr>      <chr> <dbl>      <dbl>      <dbl>
  Trump      Repu... 0.277      27.7      588708
  1 Alabama   AL        1  2123372
  Trump      Repu... 0.147      14.7      46933
  2 Alaska    AK        2  318608
  Trump      Repu... 0.035       3.5      91234
  3 Arizona   AZ        4  2604657
  Trump      Repu... 0.269      26.9      304378
  4 Arkansas  AR        5  1130635
  Trump      Repu... 0.300      -30.0      4269978
  5 Californ... CA        6  14237893
  Clint... Demo... 0.0491     -4.91      136386
  6 Colorado   CO        8  2780247
  Clint... Demo... 0.136      -13.6      224357
  7 Connecti... CT        9  1644920
```

To merge on `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 Clinton Demo...     0.300  
6 colorado     Colo... CO      8     2780247     136386 Clinton Demo...     0.0491  
7 connecticut   Conn... CT      9    1644920     224357 Clinton Demo...     0.136  
8 delaware     Dela... DE     10     443814      50476  Clinton Demo...     0.114  
9 district of... Dist... DC     11     311268     270107 Clinton Demo...     0.868  
10 florida      Flor... FL     12    9502747     112911 Trump   Repu...     0.0119  
# i 41 more rows  
# i 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>
# i 15,527 more rows
```

election

```
# A tibble: 51 × 23
  region      state st fips total_vote
  <chr>        <chr> <chr> <dbl>      <dbl>
1 alabama     Alab... AL 1 2123372
2 alaska       Alas... AK 2 318608
3 arizona     Ariz... AZ 4 2604657
4 arkansas    Arka... AR 5 1130635
5 california  Cali... CA 6 14237893
6 colorado    Colo... CO 8 2780247
7 connecticut Conn... CT 9 1644920
8 delaware    Dela... DE 10 1365000
9 florida     Flori... FL 11 2963528
10 georgia    Georg... GA 12 2779593
11 hawaii     Hawa... HI 13 900000
12 idaho       Idaho... ID 14 1147745
13 illinois   Illino... IL 15 2926685
14 indiana    Indiana... IN 16 2059885
15 kansas     Kansas... KS 17 1073555
16 kentucky   Kentuc... KY 18 1274745
17 louisiana Louisiana... LA 19 1970000
18 maine       Maine... ME 20 740000
19 maryland   Maryland... MD 21 1030000
20 massachusetts Massachusetts... MA 22 900000
21 mississippi Mississippi... MS 23 600000
22 missouri   Missouri... MO 24 1000000
23 montana     Montana... MT 25 500000
24 nebraska   Nebraska... NE 26 500000
25 new hampshire New Hampshire... NH 27 400000
26 new jersey New Jersey... NJ 28 900000
27 new mexico New Mexico... NM 29 500000
28 new york   New York... NY 30 2000000
29 north dakota North Dakota... ND 31 200000
30 ohio       Ohio... OH 32 1300000
31 oklahoma   Oklahoma... OK 33 700000
32 rhode island Rhode Island... RI 34 400000
33 south carolina South Carolina... SC 35 700000
34 south dakota South Dakota... SD 36 300000
35 tennessee Tennessee... TN 37 1000000
36 vermont     Vermont... VT 38 200000
37 virginia   Virginia... VA 39 1300000
38 washington Washington... WA 40 1000000
39 west virginia West Virginia... WV 41 500000
40 wisconsin  Wisconsin... WI 42 700000
41 wyoming     Wyoming... WY 43 200000
42 alaska      Alaska... AK 44 300000
43 arizona    Arizona... AZ 45 500000
44 colorado   Colorado... CO 46 1000000
45 connecticut Connecticut... CT 47 500000
46 delaware   Delaware... DE 48 300000
47 hawaii     Hawaii... HI 49 200000
48 idaho      Idaho... ID 50 500000
49 illinois  Illinois... IL 51 1500000
50 kentucky   Kentucky... KY 52 500000
51 mississippi Mississippi... MS 53 300000
```

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>    <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
# i 15,527 more rows
# i 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

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) %+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(0,0)  
  )  
}
```

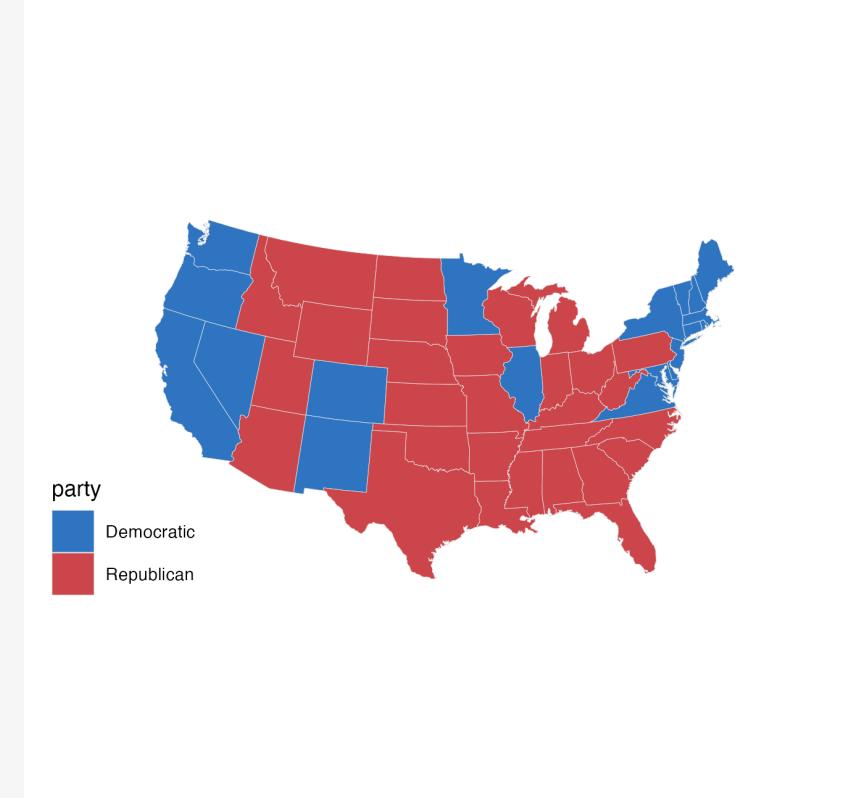
Add the theme function at the end

```
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()
```



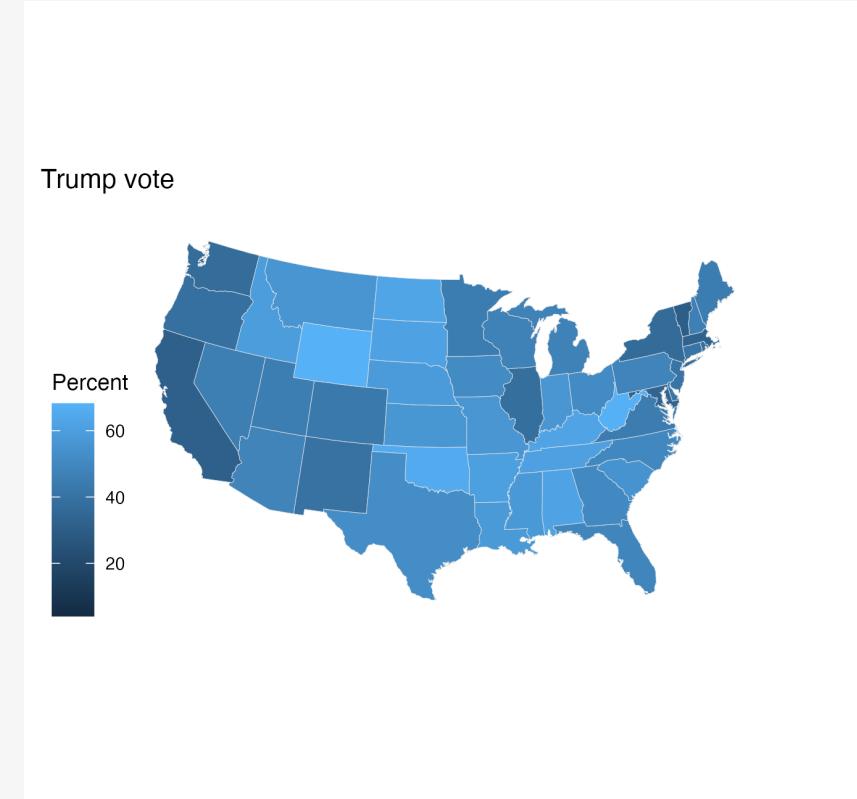
Fix the Party Colors

```
## Hex color codes for Democratic Blue and Repu  
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()
```



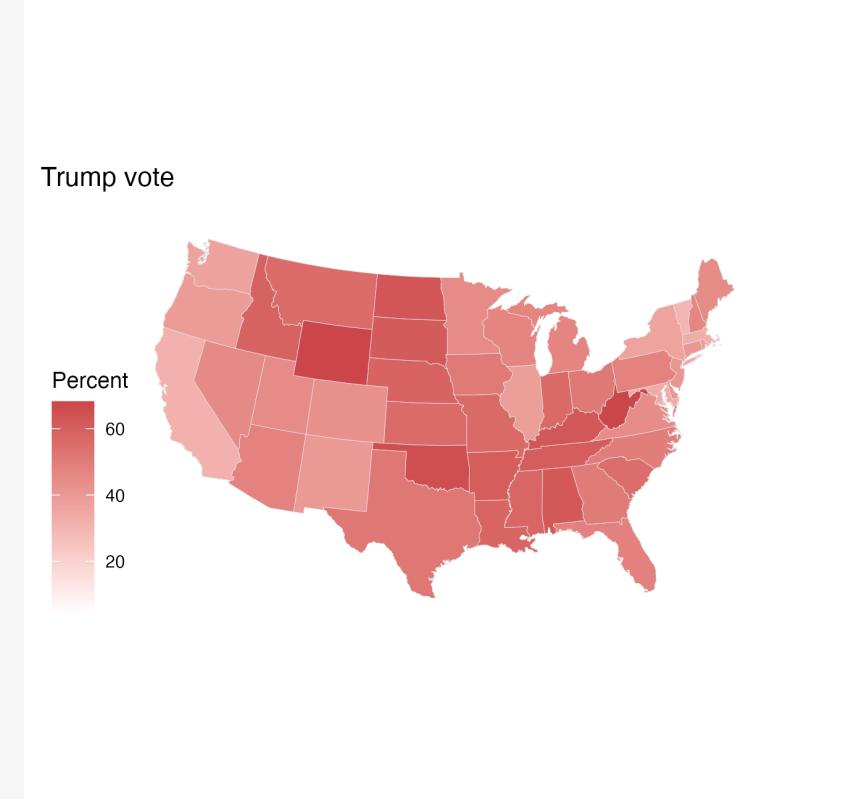
Continuous measures are *gradients*

```
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()
```



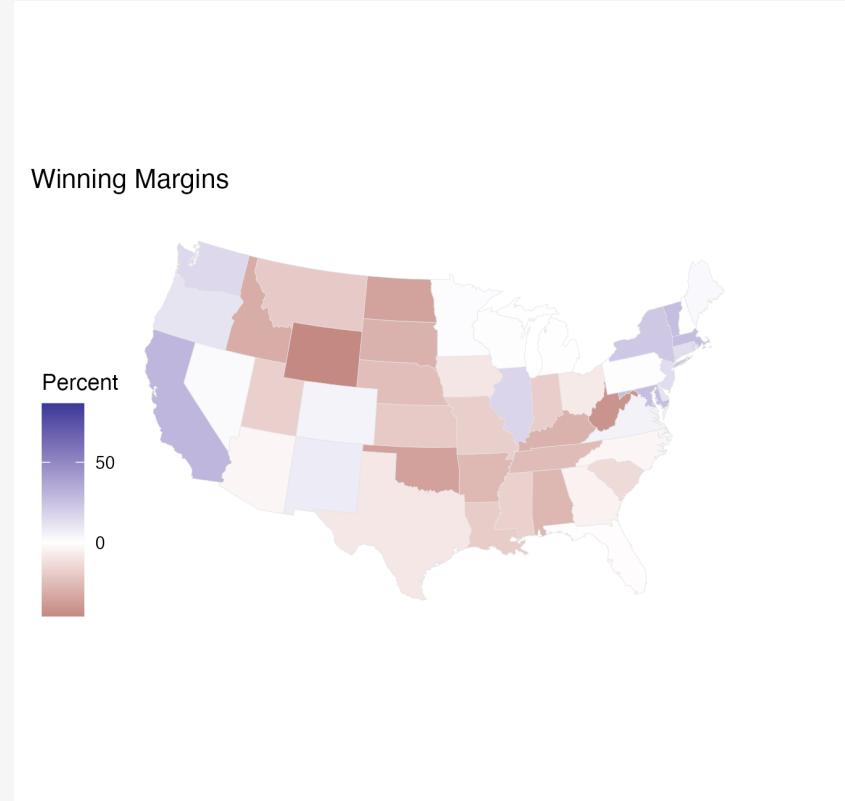
Fix the gradient with a scale function

```
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()
```



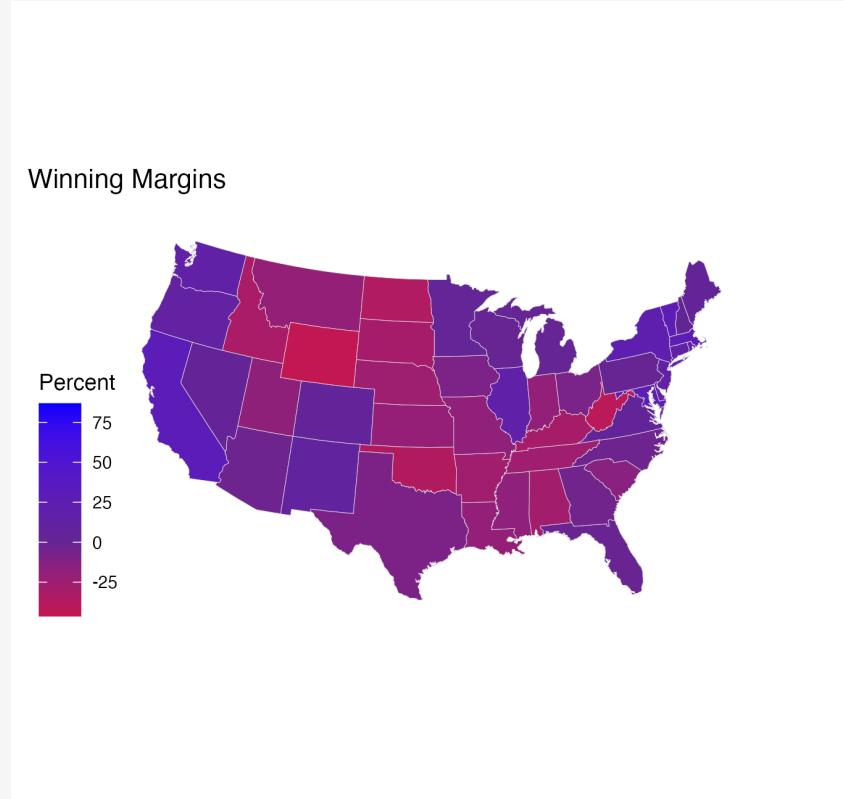
Some gradients are *diverging*

```
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()
```

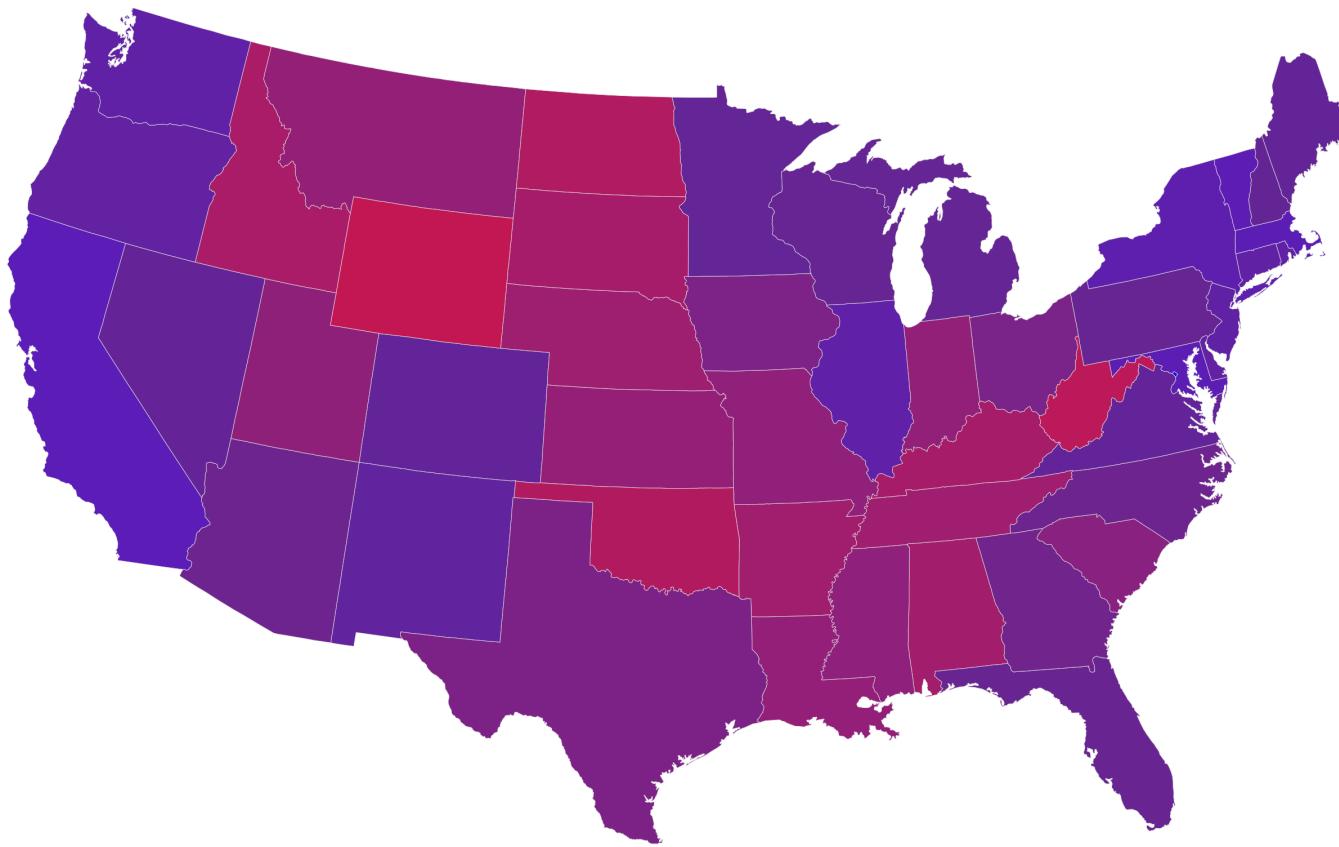


Purple America Map

```
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



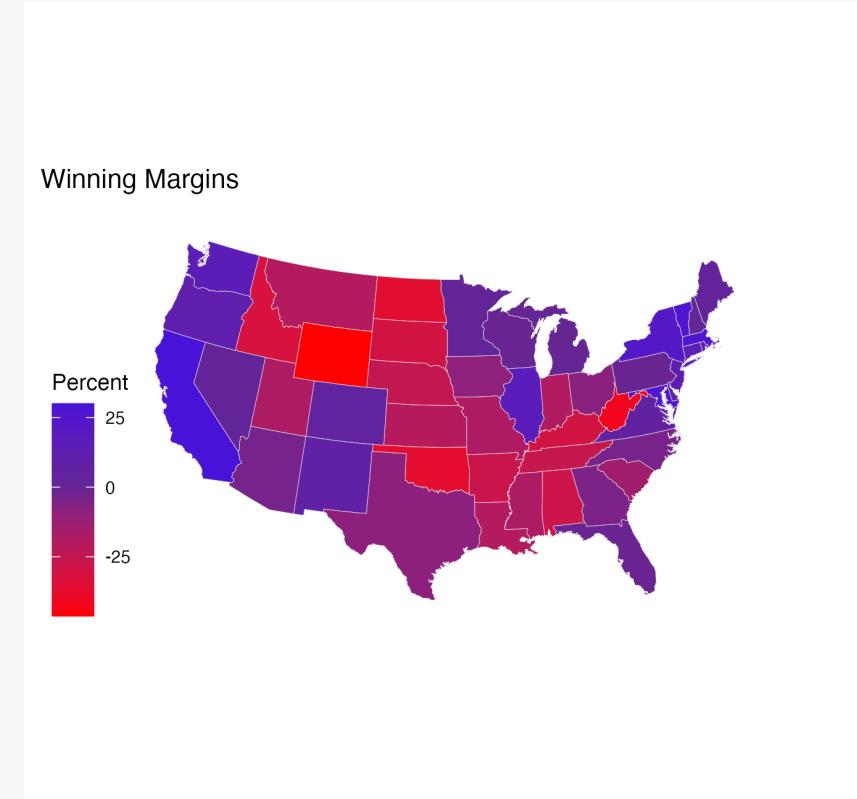
Take a closer look at this, though.



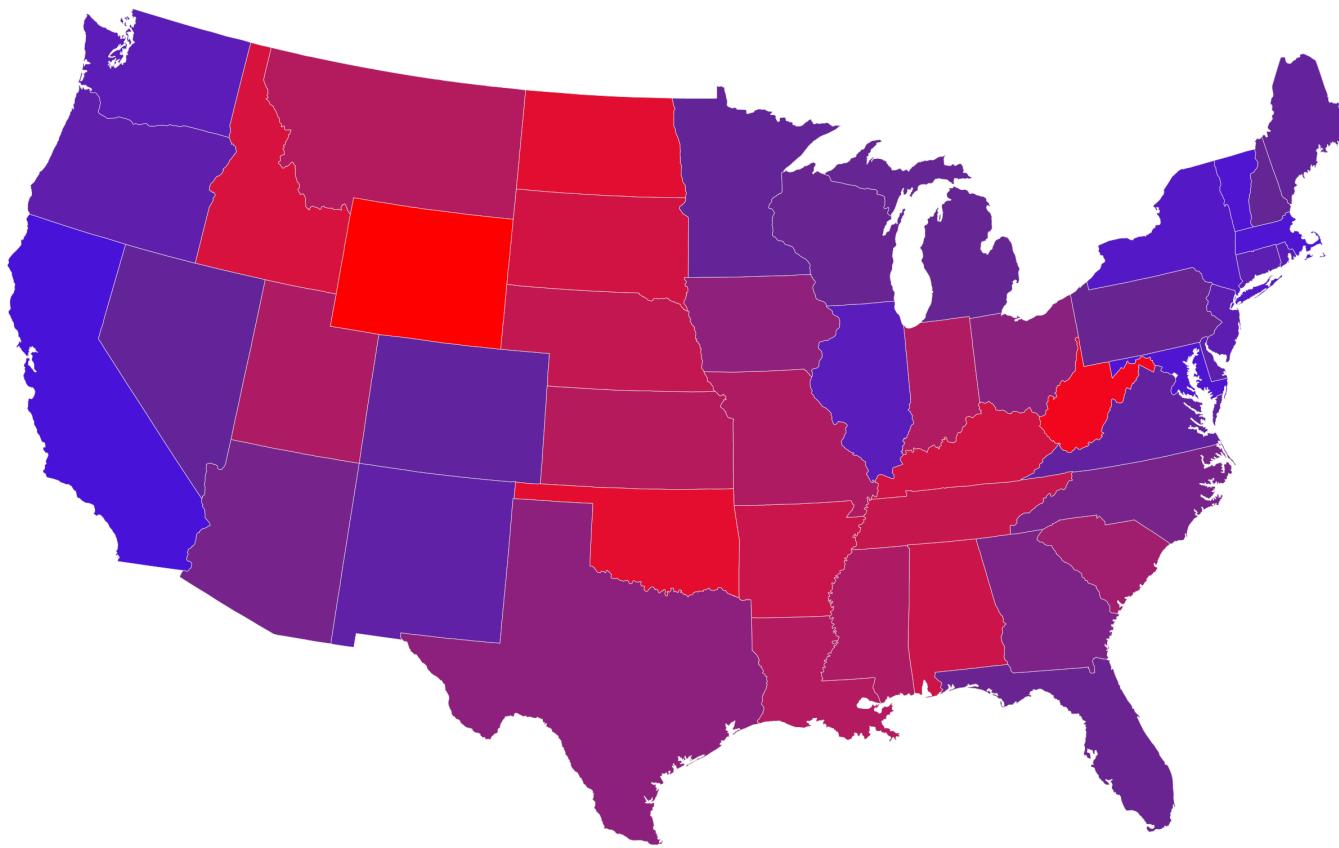
Washington, DC

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()
```



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  
# i 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>    <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
# i 3,185 more rows
# i 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>,
```

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 13307 Webster County  GA    [ 10,    50) [25.0,50.0)  
2 38049 McHenry County  ND    [  0,    10) [ 0.0, 2.0)  
3 12021 Collier County FL    [ 100,   500) [ 5.0,10.0)  
4 32033 White Pine County NV    [  0,    10) [ 2.0, 5.0)  
5 31175 Valley County   NE    [  0,    10) [ 0.0, 2.0)  
6 48221 Hood County     TX    [ 100,   500) [ 0.0, 2.0)  
7 20103 Leavenworth County KS    [ 100,   500) [ 5.0,10.0)  
8 38071 Ramsey County   ND    [  0,    10) [ 0.0, 2.0)  
9 29007 Audrain County   MO    [ 10,    50) [ 5.0,10.0)  
10 13265 Taliaferro County  GA   [  0,    10) [50.0,85.3]
```

Joined table

```
county_full ← as_tibble(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
# i 191,372 more rows
# i 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>
```

County Population Density

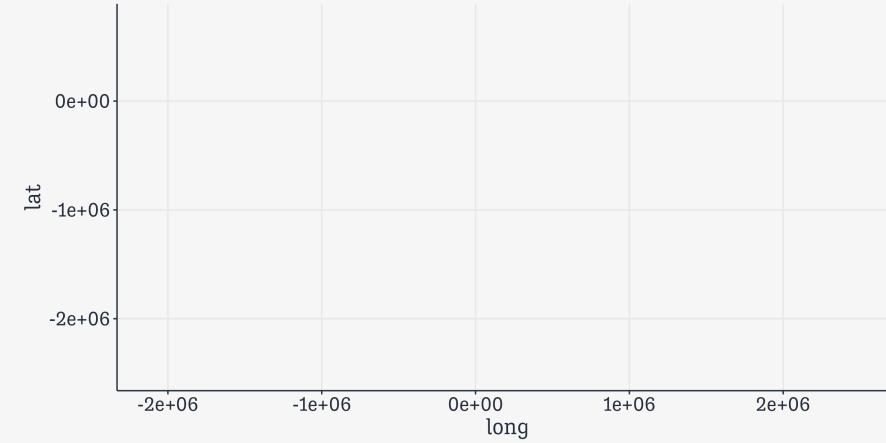
```
1 county_full <- as_tibble(left_join(county_map, county_data, by
```

County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full
# A tibble: 191,382 × 38
#>   long      lat order hole piece group    id   name state
#>   <dbl>     <dbl> <int> <lgl> <fct> <fct> <chr> <chr> <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
#> # i 191,372 more rows
#> # i 28 more variables: pop_dens <fct>, pop_dens4 <fct>, pop_dens6
```

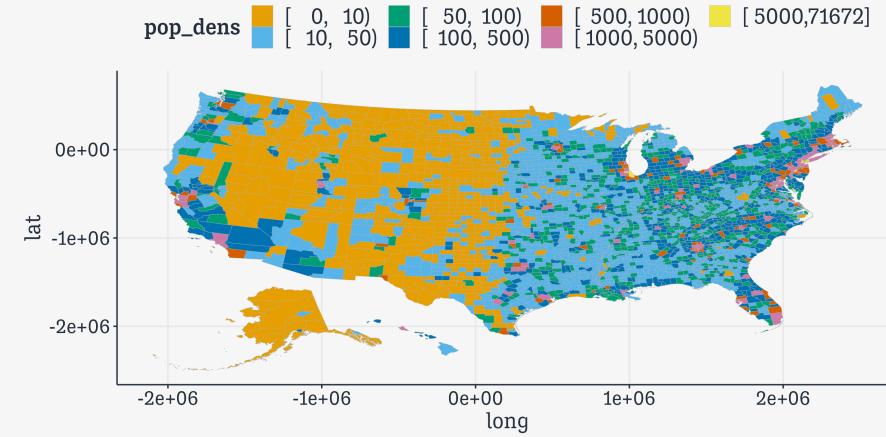
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by =  
2  
3  
4 county_full %>%  
5   ggplot(mapping = aes(x = long, y = lat,  
6                       fill = pop_dens,  
7                       group = group))
```



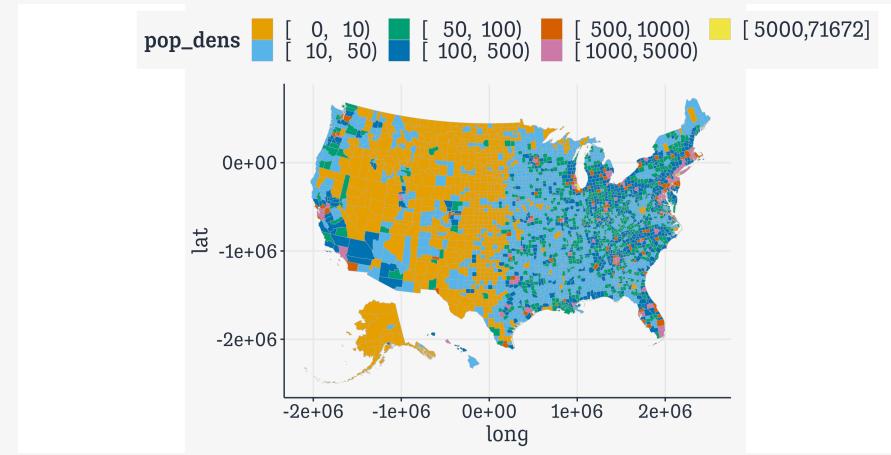
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1)
```



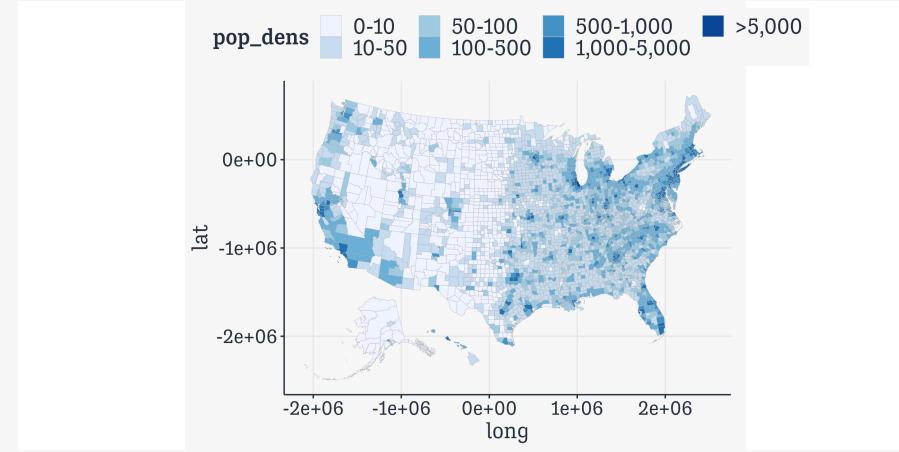
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed()
```



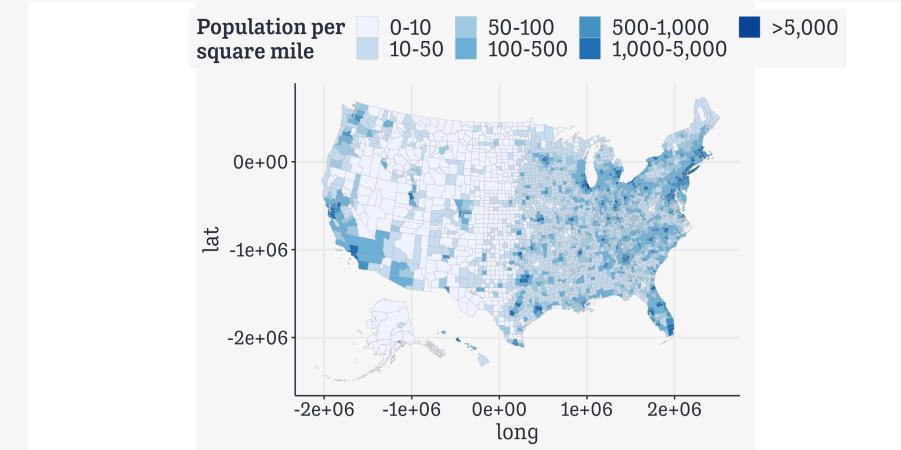
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed() +
11  scale_fill_brewer(palette="Blues",
12                    labels = c("0-10", "10-50", "50-100",
13                    "100-500", "500-1,000",
14                    "1,000-5,000", ">5,000"))
```



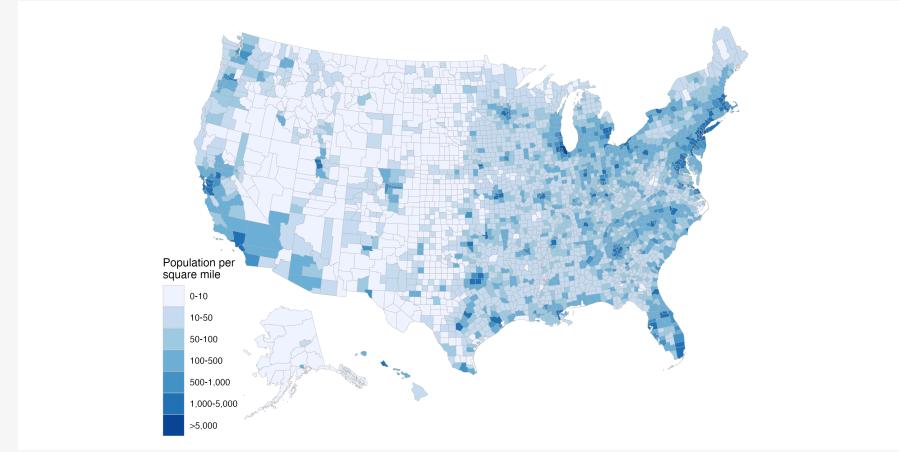
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed() +
11  scale_fill_brewer(palette="Blues",
12                    labels = c("0-10", "10-50", "50-100",
13                    "100-500", "500-1,000",
14                    "1,000-5,000", ">5,000")) +
15  labs(fill = "Population per\nsquare mile")
```



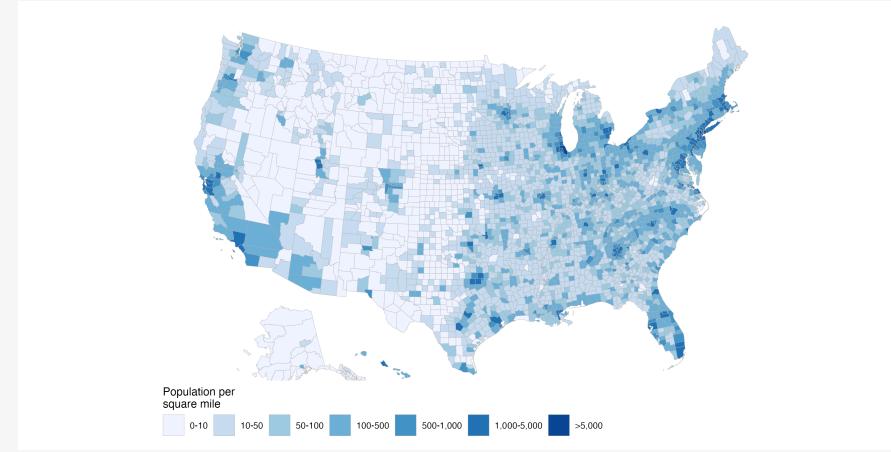
County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed() +
11  scale_fill_brewer(palette="Blues",
12                    labels = c("0-10", "10-50", "50-100",
13                    "100-500", "500-1,000",
14                    "1,000-5,000", ">5,000")) +
15  labs(fill = "Population per\nsquare mile") +
16  kjhslides::kjh_theme_map()
```



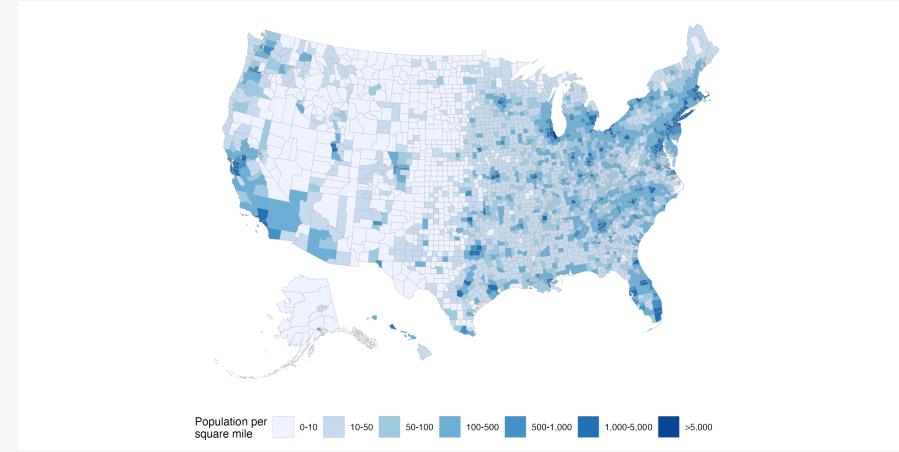
County Population Density

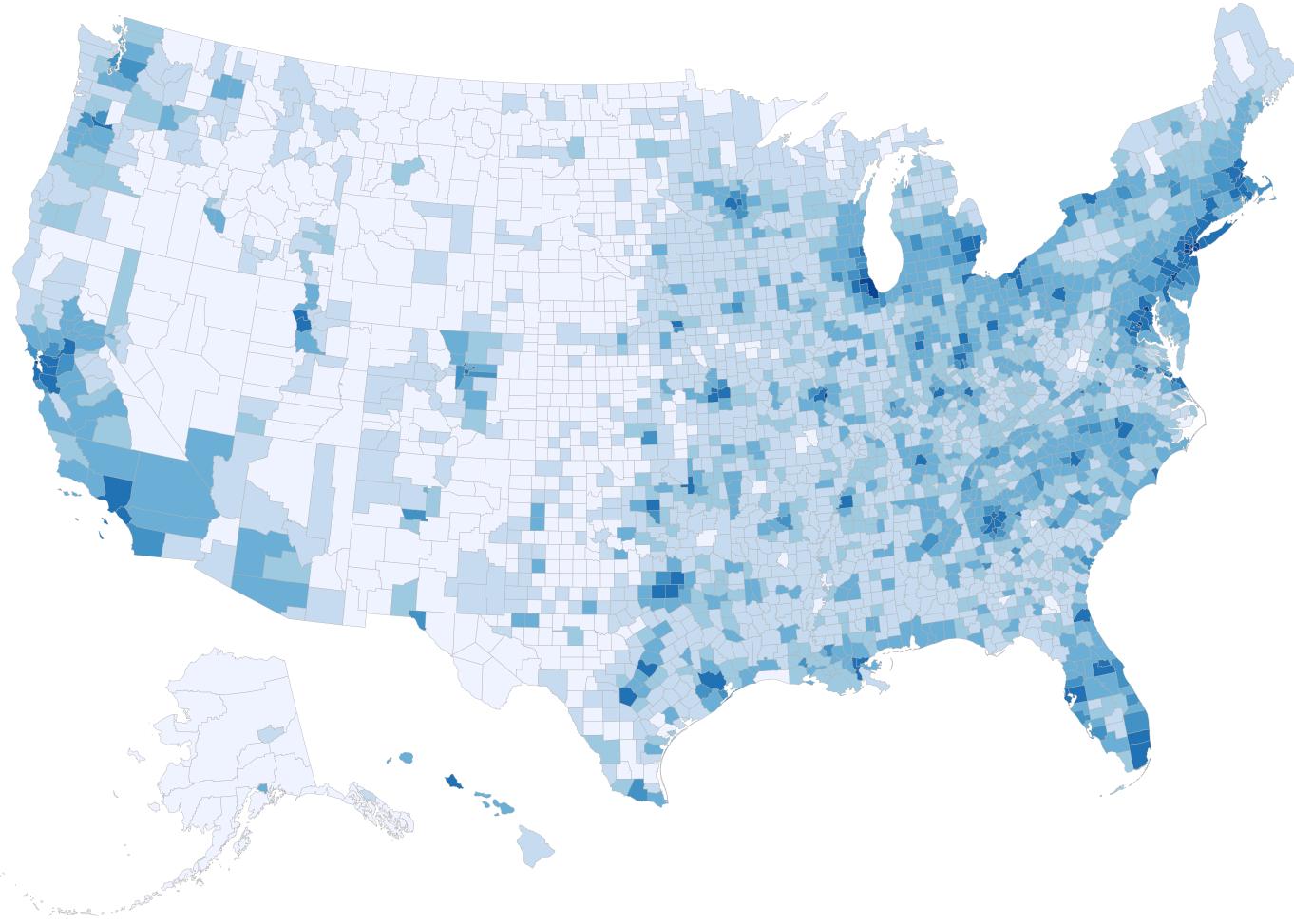
```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed() +
11  scale_fill_brewer(palette="Blues",
12                    labels = c("0-10", "10-50", "50-100",
13                    "100-500", "500-1,000",
14                    "1,000-5,000", ">5,000")) +
15  labs(fill = "Population per\nsquare mile") +
16  kjhslides::kjh_theme_map() +
17  guides(fill = guide_legend(nrow = 1))
```



County Population Density

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3
4 county_full %>
5   ggplot(mapping = aes(x = long, y = lat,
6                     fill = pop_dens,
7                     group = group)) +
8   geom_polygon(color = "gray70",
9                 size = 0.1) +
10  coord_fixed() +
11  scale_fill_brewer(palette="Blues",
12                    labels = c("0-10", "10-50", "50-100",
13                      "100-500", "500-1,000",
14                      "1,000-5,000", ">5,000")) +
15  labs(fill = "Population per\nsquare mile") +
16  kjhslides::kjh_theme_map() +
17  guides(fill = guide_legend(nrow = 1)) +
18  theme(legend.position = "bottom")
```





Population Density by County, binned

Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
```

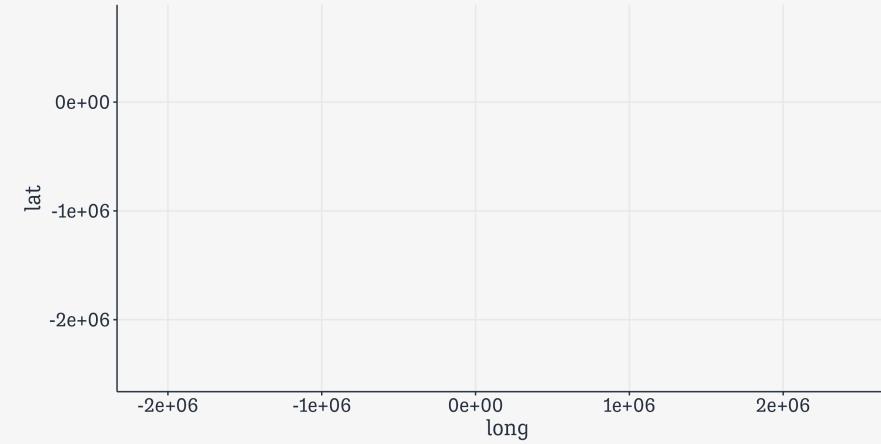
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full
```

```
# A tibble: 191,382 × 38
      long      lat order hole piece group     id   name state
  <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
# i 191,372 more rows
# i 28 more variables: pop_dens <fct>, pop_dens4 <fct>, pop_dens6
```

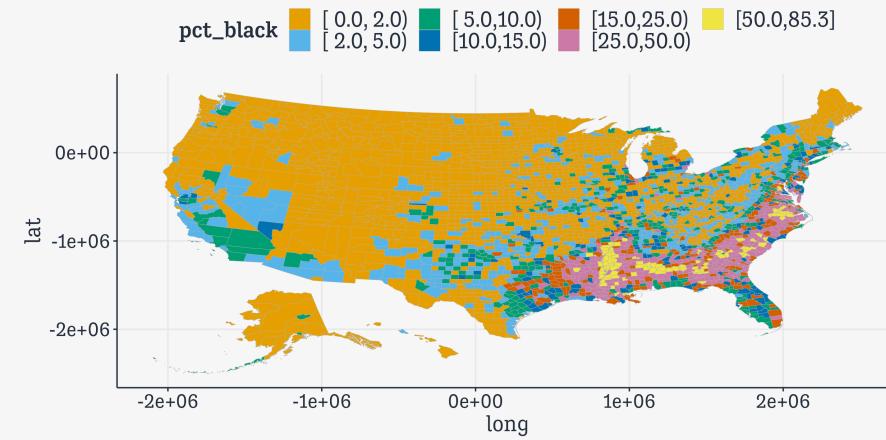
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full ▷
4   ggplot(mapping = aes(x = long, y = lat,
5                     fill = pct_black,
6                     group = group))
```



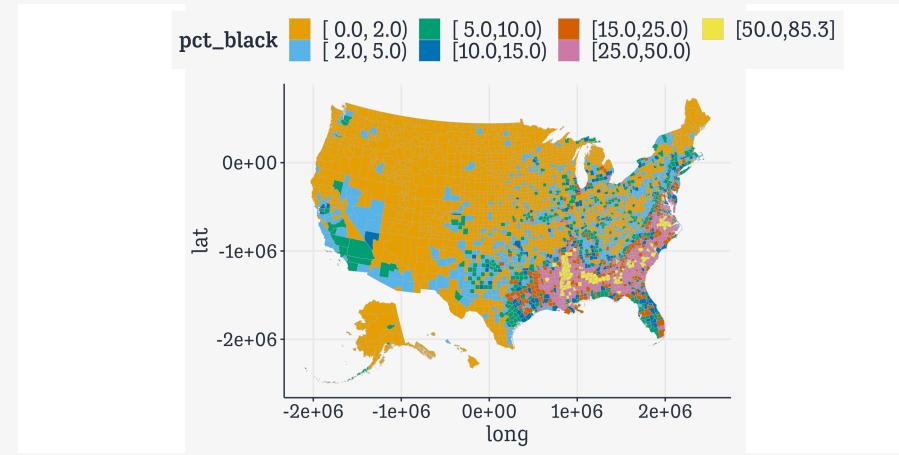
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full ▷
4   ggplot(mapping = aes(x = long, y = lat,
5                     fill = pct_black,
6                     group = group)) +
7   geom_polygon(color = "gray70",
8               size = 0.1)
```



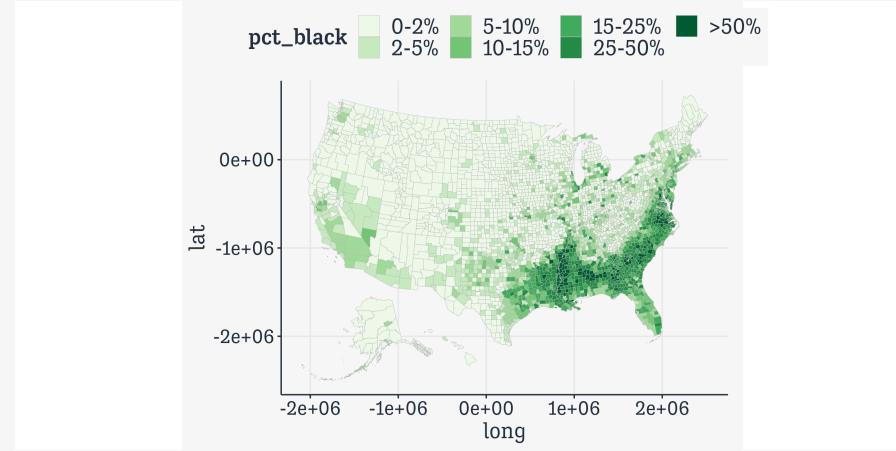
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full ▷
4     ggplot(mapping = aes(x = long, y = lat,
5                         fill = pct_black,
6                         group = group)) +
7     geom_polygon(color = "gray70",
8                  size = 0.1) +
9     coord_fixed()
```



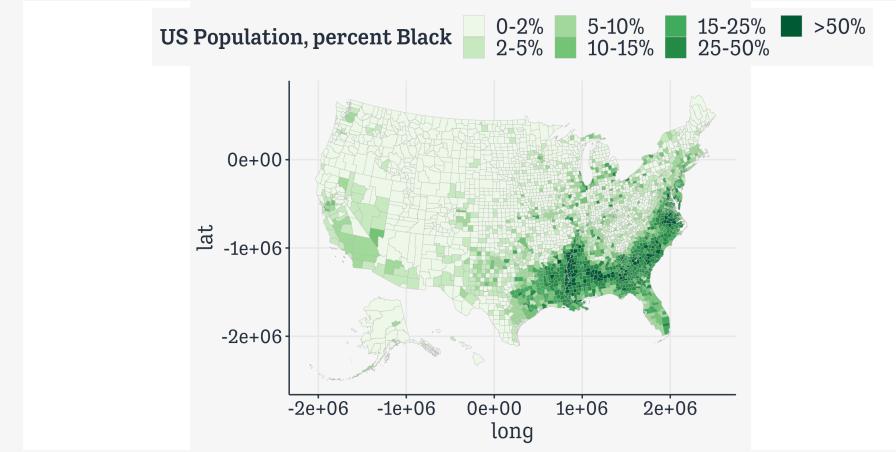
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full ▷
4   ggplot(mapping = aes(x = long, y = lat,
5                     fill = pct_black,
6                     group = group)) +
7   geom_polygon(color = "gray70",
8                 size = 0.1) +
9   coord_fixed() +
10  scale_fill_brewer(palette="Greens",
11                    labels = c("0-2%", "2-5%", "5-10%",
12                      "10-15%", "15-25%",
13                      "25-50%", ">50%"))
```



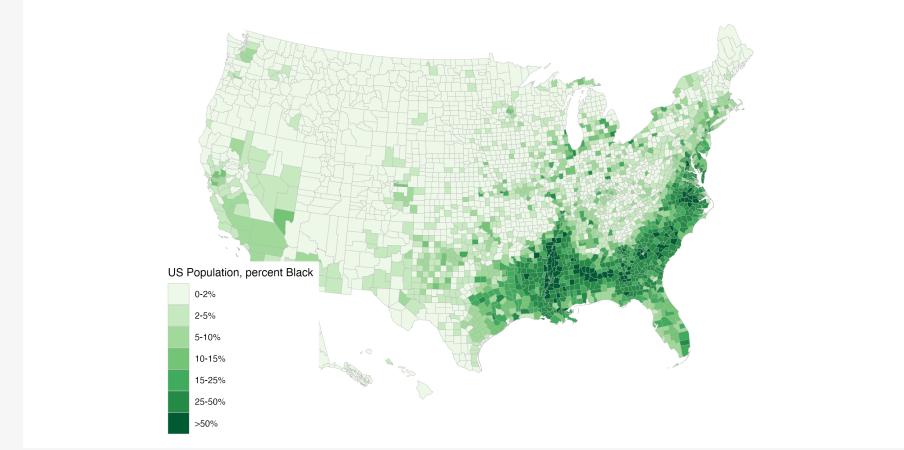
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full %>
4   ggplot(mapping = aes(x = long, y = lat,
5                       fill = pct_black,
6                       group = group)) +
7   geom_polygon(color = "gray70",
8               size = 0.1) +
9   coord_fixed() +
10  scale_fill_brewer(palette="Greens",
11                     labels = c("0-2%", "2-5%", "5-10%",
12                           "10-15%", "15-25%",
13                           "25-50%", ">50%")) +
14  labs(fill = "US Population, percent Black")
```



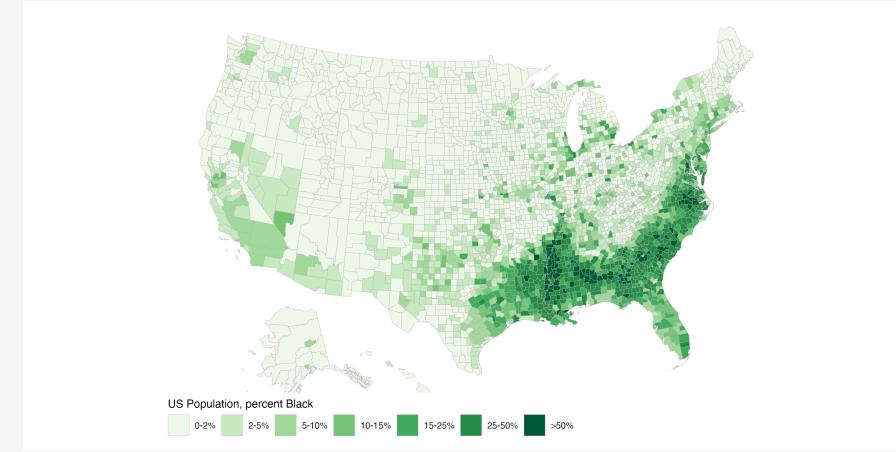
Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full >
4   ggplot(mapping = aes(x = long, y = lat,
5                     fill = pct_black,
6                     group = group)) +
7   geom_polygon(color = "gray70",
8                 size = 0.1) +
9   coord_fixed() +
10  scale_fill_brewer(palette="Greens",
11                    labels = c("0-2%", "2-5%", "5-10%",
12                           "10-15%", "15-25%",
13                           "25-50%", ">50%")) +
14  labs(fill = "US Population, percent Black") +
15  kjhslides::kjh_theme_map()
```



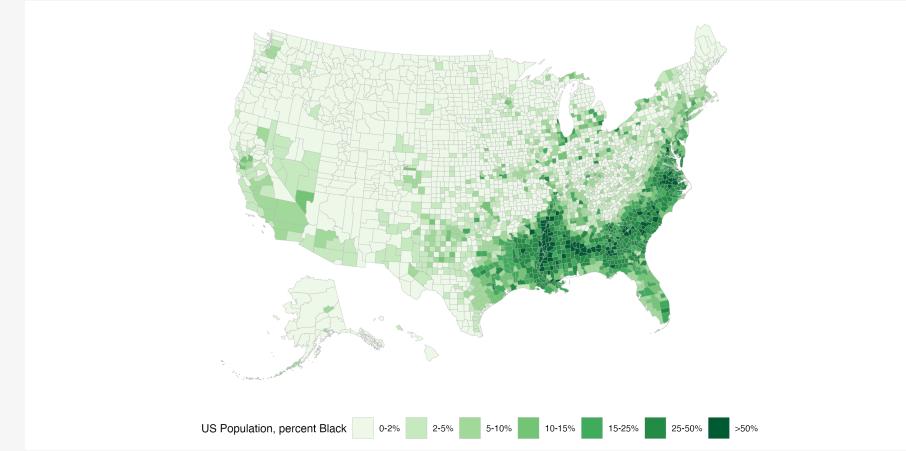
Same again for Percent Black

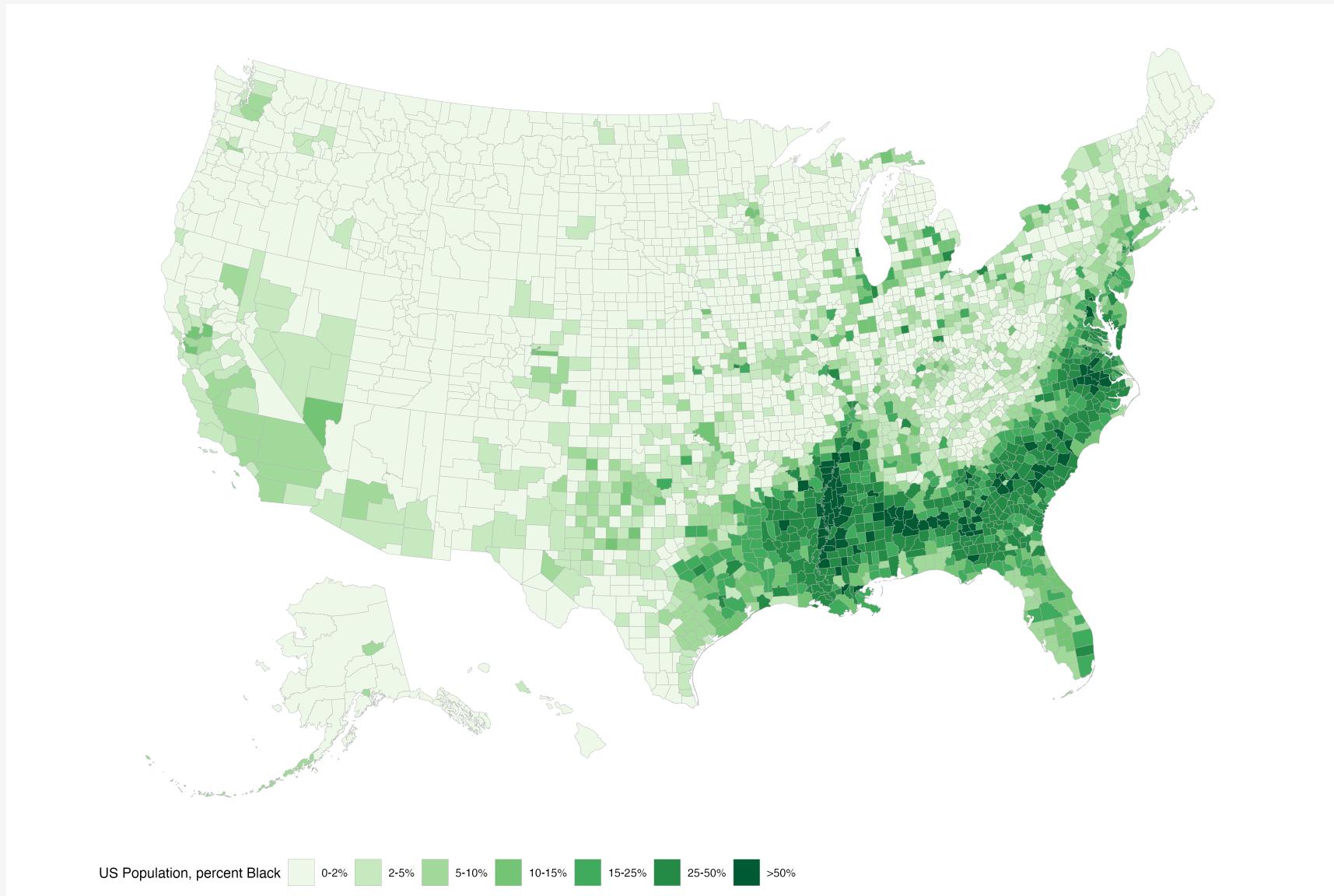
```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full >
4   ggplot(mapping = aes(x = long, y = lat,
5                     fill = pct_black,
6                     group = group)) +
7   geom_polygon(color = "gray70",
8                 size = 0.1) +
9   coord_fixed() +
10  scale_fill_brewer(palette="Greens",
11                    labels = c("0-2%", "2-5%", "5-10%",
12                           "10-15%", "15-25%",
13                           "25-50%", ">50%")) +
14  labs(fill = "US Population, percent Black") +
15  kjhslides::kjh_theme_map() +
16  guides(fill = guide_legend(nrow = 1))
```



Same again for Percent Black

```
1 county_full ← as_tibble(left_join(county_map, county_data, by
2
3 county_full ▷
4   ggplot(mapping = aes(x = long, y = lat,
5                       fill = pct_black,
6                       group = group)) +
7   geom_polygon(color = "gray70",
8               size = 0.1) +
9   coord_fixed() +
10  scale_fill_brewer(palette="Greens",
11                     labels = c("0-2%", "2-5%", "5-10%",
12                           "10-15%", "15-25%",
13                           "25-50%", ">50%")) +
14  labs(fill = "US Population, percent Black") +
15  kjhslides::kjh_theme_map() +
16  guides(fill = guide_legend(nrow = 1)) +
17  theme(legend.position = "bottom")
```





Percent Black, by County, binned

Big counties, few people, rare events

Example: Reverse coding

Code Reverse

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

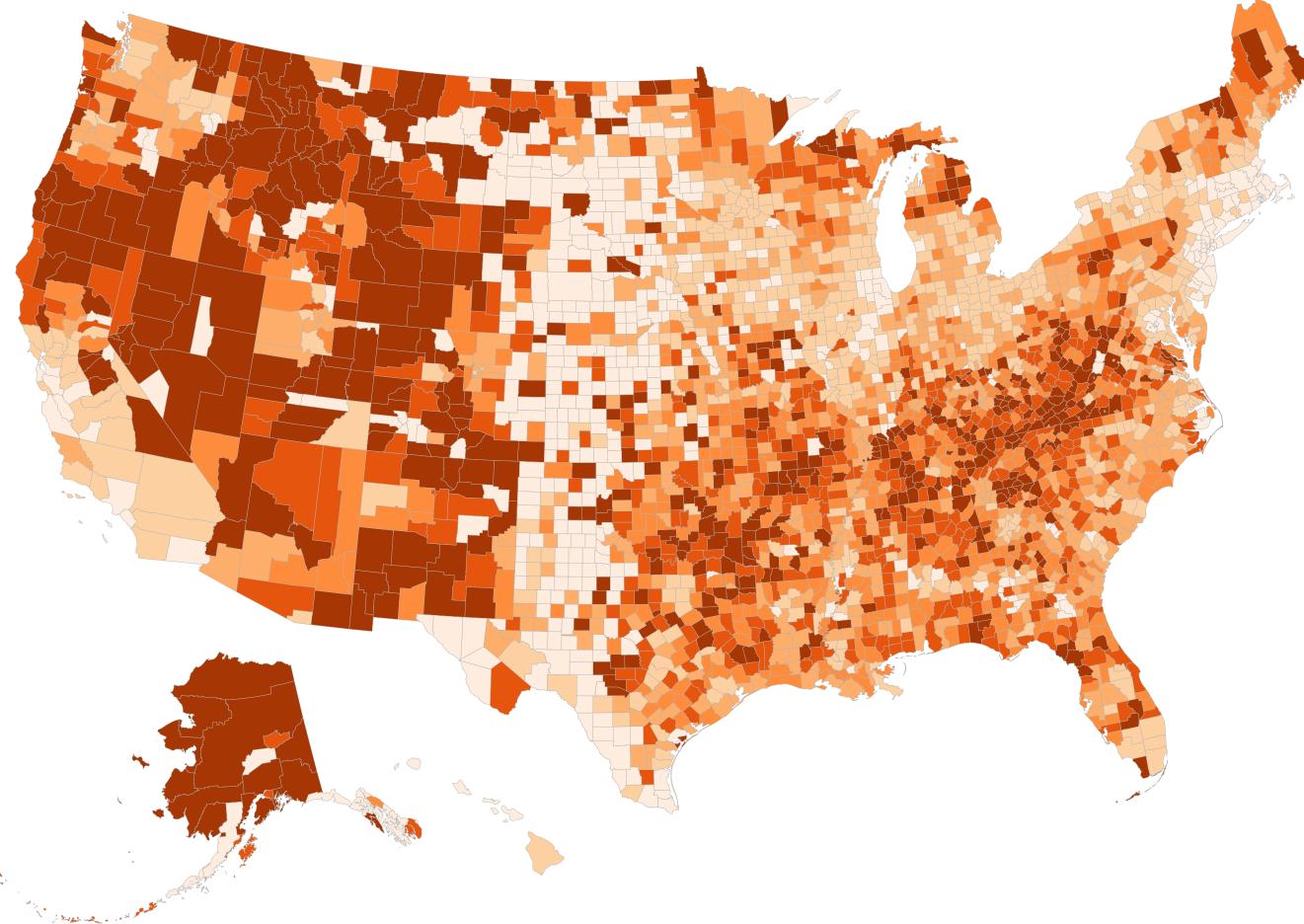
```
[1] "#FEEDDE" "#FDD0A2" "#FDAE6B" "#FD8D3C" "#E6550D" "#A63603"
```



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

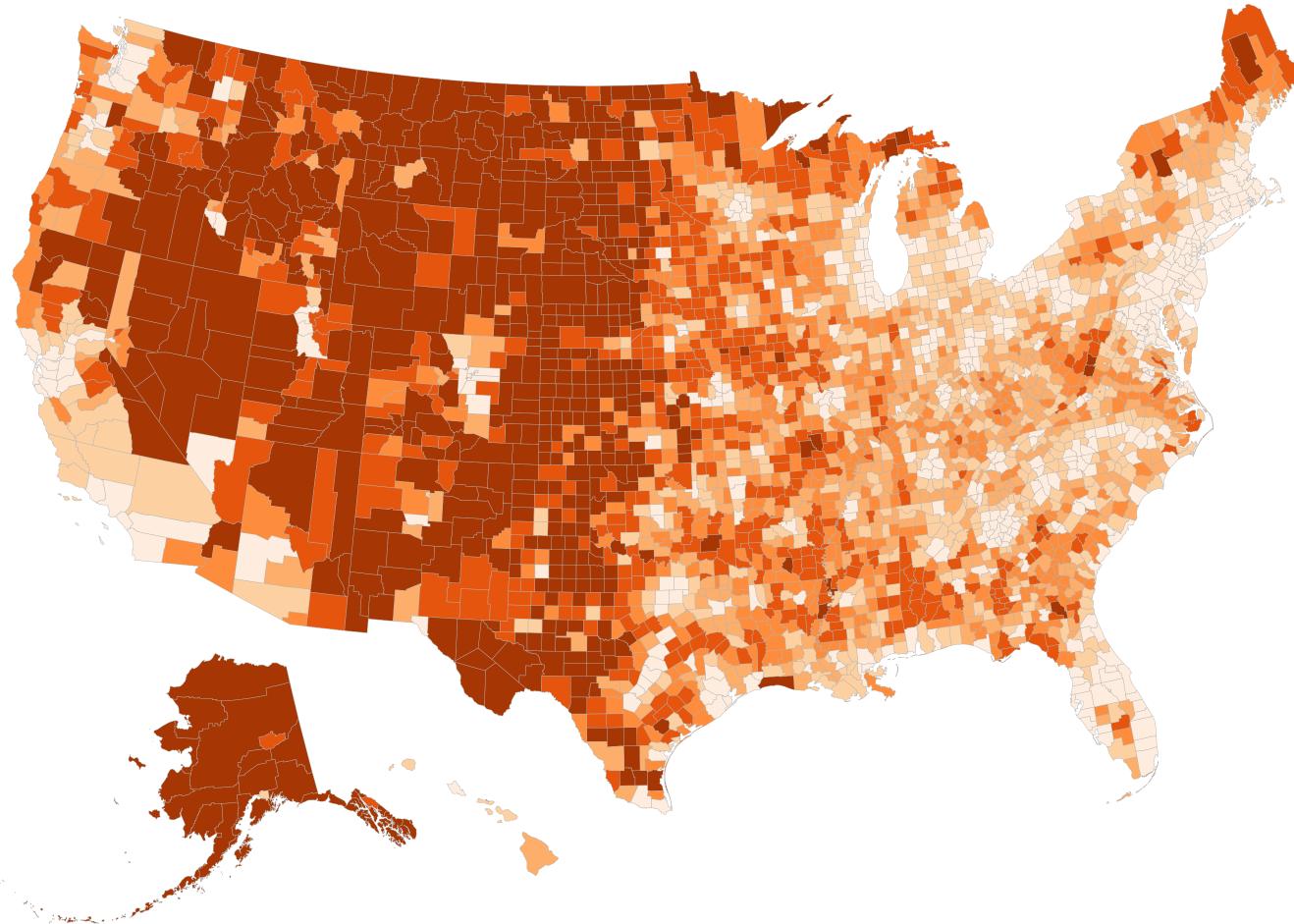


Regular palette

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

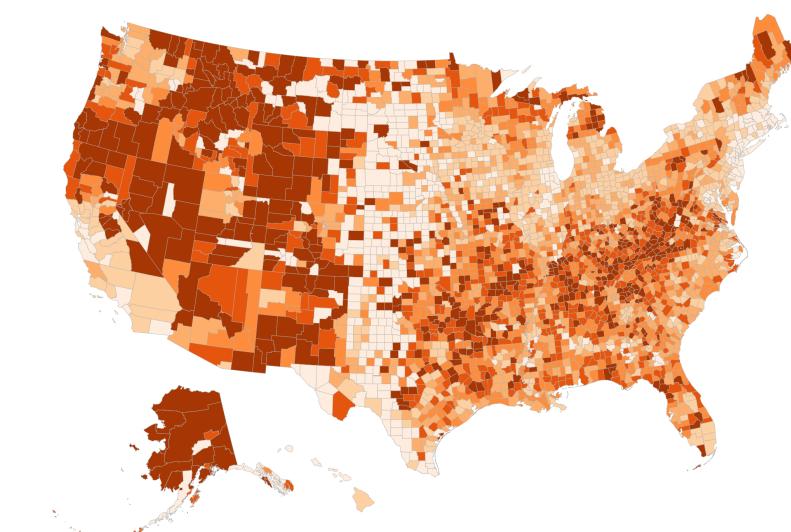


Persons per square mile [0, 9] [9, 25] [25, 45] [45, 82] [82, 215] [215,71672]

Reverse-coded density

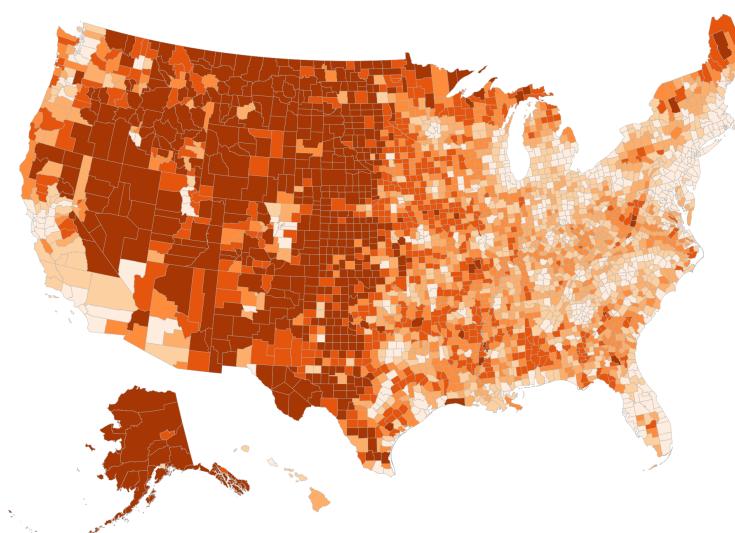
Comparison

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]

Small multiples for maps

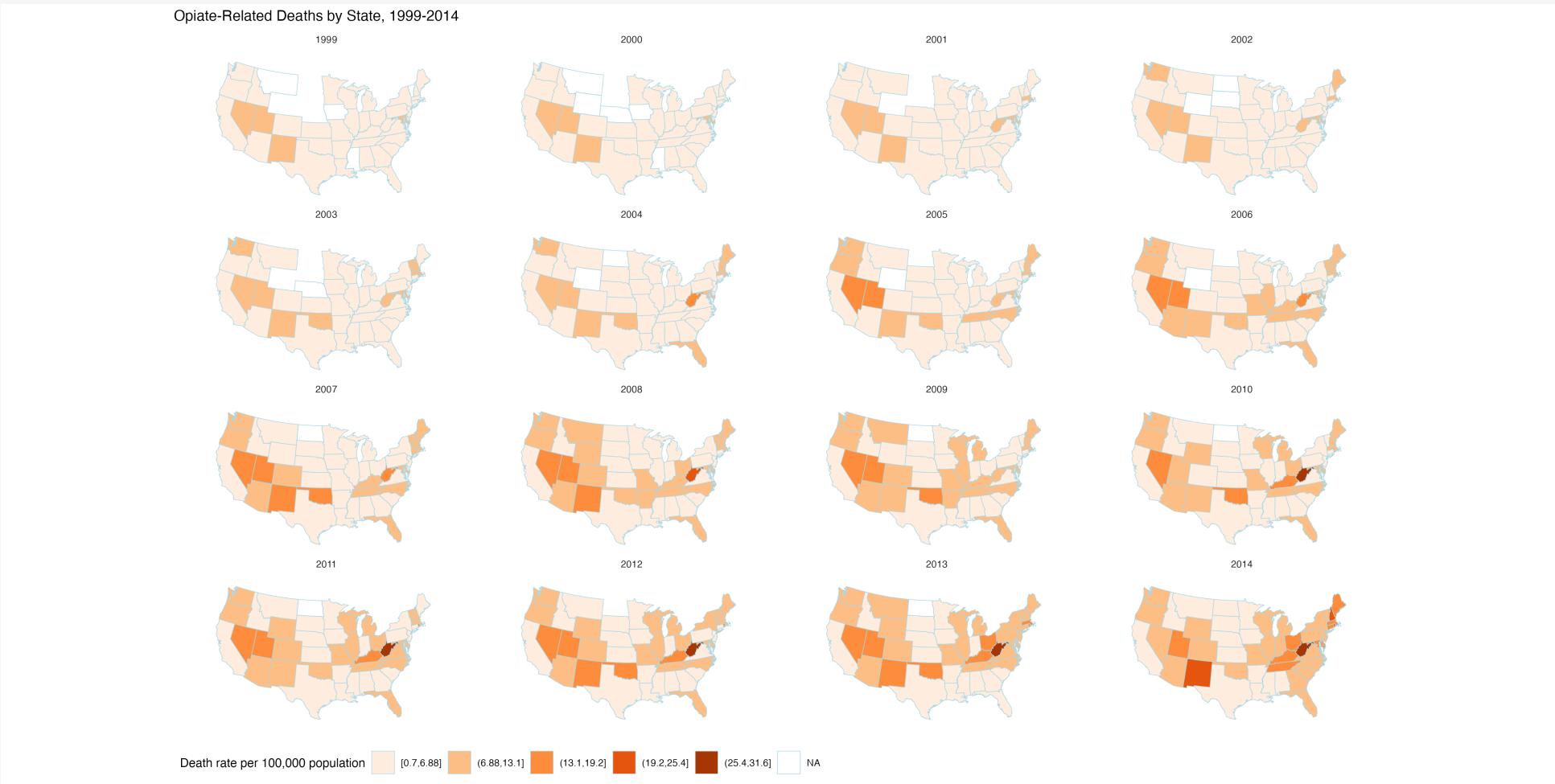
Opiate-related Mortality, 1999-2014

```
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
# i 790 more rows
# i 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")
```



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

```
1 ## Put this in an object called `st_top`  
2 opiates
```

```
# A tibble: 800 × 11  
#>   year state    fips deaths population crude adjusted  
#>   <int> <chr>    <int>  <int>      <dbl> <dbl> <dbl>  
#>   adjusted_se region abbr  
#>   <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  
#> # i 790 more rows  
#> # i 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
1 ## Put this in an object called `st_top`  
2 opiates %>  
3   filter(year == max(year),  
4         abbr != "DC")
```

```
# A tibble: 50 × 11  
#>   year state    fips deaths population crude adjusted  
#>   <int> <chr> <int> <int> <dbl> <dbl> <dbl>  
#>   <dbl> <ord> <chr>  
#> 1 2014 alabama 1 270 4849377 5.6 5.6  
#> 0.3 South AL  
#> 2 2014 alaska 2 76 736732 10.3 10.6  
#> 1.2 West AK  
#> 3 2014 arizona 4 589 6731484 8.7 8.8  
#> 0.4 West AZ  
#> 4 2014 arkans... 5 173 2966369 5.8 6.3  
#> 0.5 South AR  
#> 5 2014 califo... 6 2024 38802500 5.2 5  
#> 0.1 West CA  
#> 6 2014 colora... 8 517 5355866 9.7 9.4  
#> 0.4 West CO  
#> 7 2014 connec... 9 525 3596677 14.6 15.2  
#> 0.7 North... CT  
#> 8 2014 delawa... 10 124 935614 13.3 13.9  
#> 1.3 South DE  
#> 9 2014 florida 12 1399 19893297 7 7.2  
#> 0.2 South FL  
#> 10 2014 georgia 13 710 10097343 7 7  
#> 0.3 South GA  
#> # i 40 more rows  
#> # i 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
1 ## Put this in an object called `st_top`  
2 opiates %>  
3   filter(year == max(year),  
4         abbr != "DC") %>  
5   group_by(region)
```

```
# A tibble: 50 × 11  
# Groups:   region [4]  
    year state    fips deaths population crude adjusted  
    <int> <chr>    <int>  <int>     <dbl>    <dbl>  
 1 2014 alabama    1    270  4849377    5.6    5.6  
 0.3 South AL  
 2 2014 alaska     2     76  736732    10.3   10.6  
 1.2 West AK  
 3 2014 arizona    4    589  6731484    8.7    8.8  
 0.4 West AZ  
 4 2014 arkans...   5    173  2966369    5.8    6.3  
 0.5 South AR  
 5 2014 califo...   6    2024 38802500    5.2    5  
 0.1 West CA  
 6 2014 colora...   8    517  5355866    9.7    9.4  
 0.4 West CO  
 7 2014 connec...   9    525  3596677   14.6   15.2  
 0.7 North... CT  
 8 2014 delawa...  10    124  935614    13.3   13.9  
 1.3 South DE  
 9 2014 florida   12   1399 19893297     7    7.2  
 0.2 South FL  
10 2014 georgia   13   710  10097343     7    7  
 0.3 South GA  
# i 40 more rows
```

The two leading states in each region in 2014

```
1 ## Put this in an object called `st_top`  
2 opiates %>  
3   filter(year == max(year),  
4         abbr != "DC") %>  
5   group_by(region) %>  
6   slice_max(order_by = adjusted,  
7             n = 2)
```

```
# A tibble: 8 × 11  
# Groups:   region [4]  
#> #>   year state    fips deaths population crude adjusted  
#> #>   adjusted_se region abbr  
#> #>   <int> <chr>    <int>  <int>     <dbl>    <dbl>  
#> #>   <dbl> <ord>    <chr>  
#> 1 2014 new ham...    33    297  1326813  22.4   23.4  
#> 1.4 North... NH  
#> 2 2014 rhode i...    44    205  1055173  19.4   19.8  
#> 1.4 North... RI  
#> 3 2014 ohio        39    2106 11594163  18.2   19.1  
#> 0.4 Midwe... OH  
#> 4 2014 missouri     29    696  6063589  11.5    12  
#> 0.5 Midwe... MO  
#> 5 2014 new mex...    35    402  2085572  19.3   20.2  
#> West NM  
#> 6 2014 utah        49    455  2942902  15.5   16.8  
#> 0.8 West UT  
#> 7 2014 west vi...    54    554  1850326  29.9   31.6  
#> 1.4 South WV  
#> 8 2014 kentucky     21    729  4413457  16.5   16.8  
#> 0.6 South KY  
#> # i 1 more variable: division_name <chr>
```

The two leading states in each region in 2014

```
1 ## Put this in an object called `st_top`  
2 opiates %>  
3   filter(year == max(year),  
4         abbr != "DC") %>  
5   group_by(region) %>  
6   slice_max(order_by = adjusted,  
7             n = 2)
```

```
# A tibble: 8 × 11  
# Groups:   region [4]  
#> #>   year state    fips deaths population crude adjusted  
#> #>   adjusted_se region abbr  
#> #>   <int> <chr>    <int>  <int>     <dbl>    <dbl>  
#> #>   <dbl> <ord>    <chr>  
#> 1 2014 new ham...    33    297  1326813  22.4   23.4  
#> 1.4 North... NH  
#> 2 2014 rhode i...    44    205  1055173  19.4   19.8  
#> 1.4 North... RI  
#> 3 2014 ohio        39    2106 11594163  18.2   19.1  
#> 0.4 Midwe... OH  
#> 4 2014 missouri     29    696  6063589  11.5    12  
#> 0.5 Midwe... MO  
#> 5 2014 new mex...    35    402  2085572  19.3   20.2  
#> West NM  
#> 6 2014 utah        49    455  2942902  15.5   16.8  
#> 0.8 West UT  
#> 7 2014 west vi...    54    554  1850326  29.9   31.6  
#> 1.4 South WV  
#> 8 2014 kentucky     21    729  4413457  16.5   16.8  
#> 0.6 South KY  
#> # i 1 more variable: division_name <chr>
```

Opiates Time Series plot

```
1 st_top ← opiates ▷ filter(year = max(year), abbr ≠ "DC")
```

Opiates Time Series plot

```
1 st_top ← opiates ▷ filter(year = max(year), abbr ≠ "DC")
2   group_by(region)
```

Opiates Time Series plot

```
1 st_top ← opiates ▷ filter(year = max(year), abbr ≠ "DC")
2   group_by(region) ▷
3     slice_max(order_by = adjusted, n = 2)
```

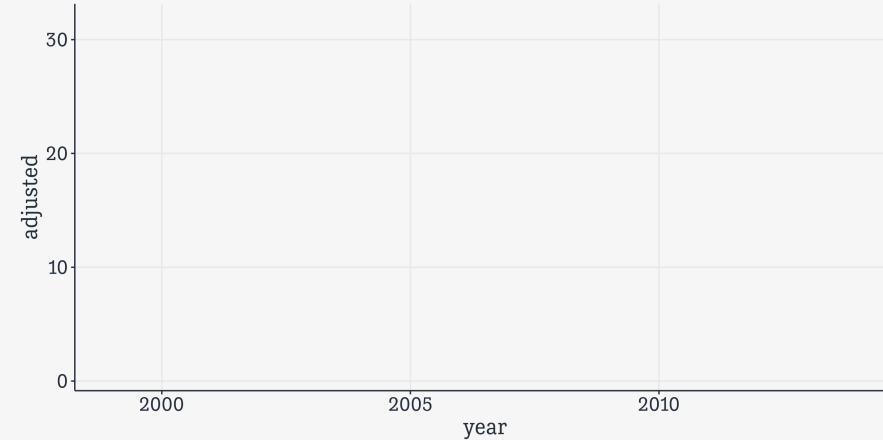
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2 group_by(region) %>%
3 slice_max(order_by = adjusted, n = 2)
4
5 opiates
```

```
# A tibble: 800 × 11
   year state    fips deaths population crude adjusted
   <int> <chr>    <int>   <int>      <dbl>    <dbl>
adjusted_se region abbr
<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
# i 790 more rows
# i 1 more variable: division_name <chr>
```

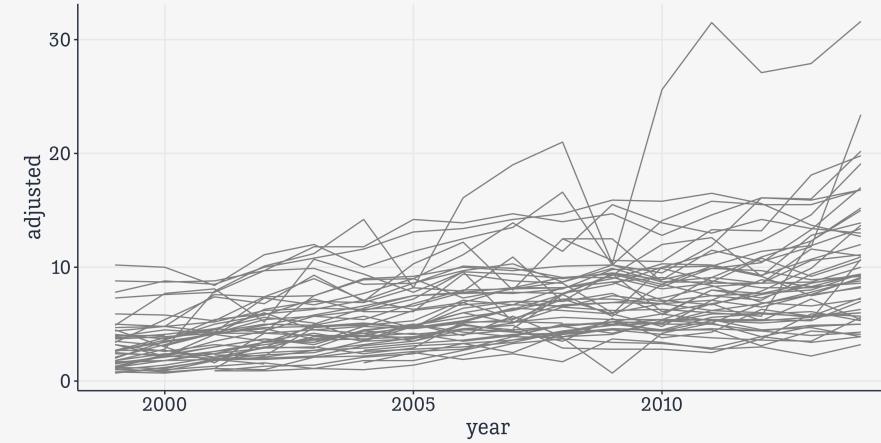
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2   group_by(region) %>%
3     slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6   ggplot(aes(x = year,
7               y = adjusted))
```



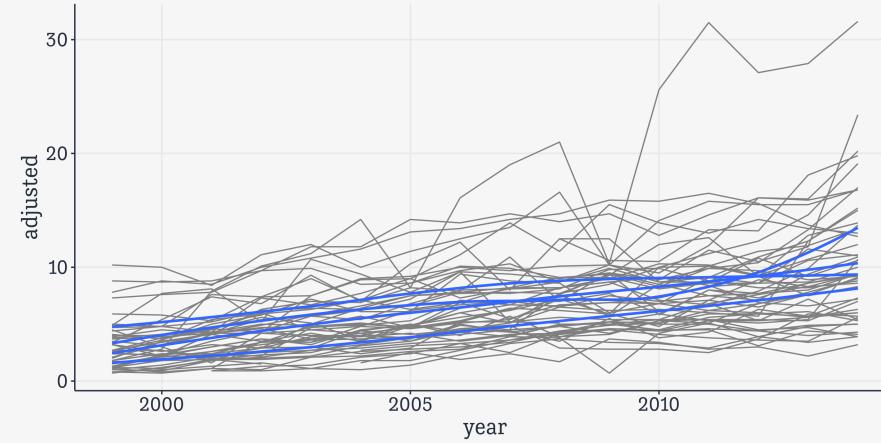
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2   group_by(region) %>%
3     slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6   ggplot(aes(x = year,
7             y = adjusted)) +
8   geom_line(aes(group = state),
9             color = "gray50")
```



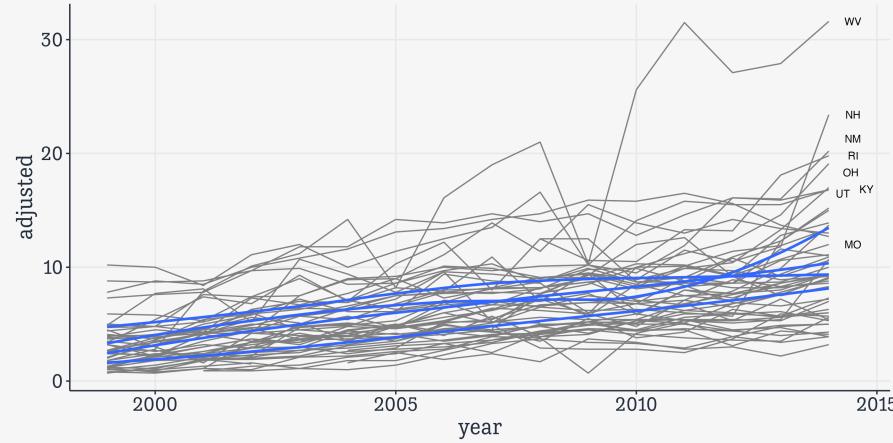
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2   group_by(region) %>%
3     slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6   ggplot(aes(x = year,
7             y = adjusted)) +
8   geom_line(aes(group = state),
9             color = "gray50") +
10  geom_smooth(aes(group = region),
11             se = FALSE)
```



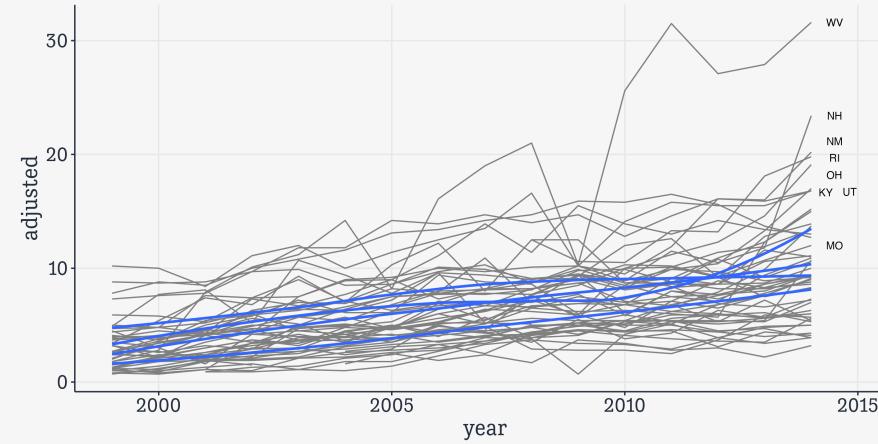
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2 group_by(region) %>%
3 slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6 ggplot(aes(x = year,
7             y = adjusted)) +
8 geom_line(aes(group = state),
9             color = "gray50") +
10 geom_smooth(aes(group = region),
11              se = FALSE) +
12 ggrepel::geom_text_repel(
13   data = st_top,
14   mapping = aes(x = year,
15                 y = adjusted,
16                 label = abbr),
17   size = 3,
18   segment.color = NA,
19   nudge_x = 0.5)
```



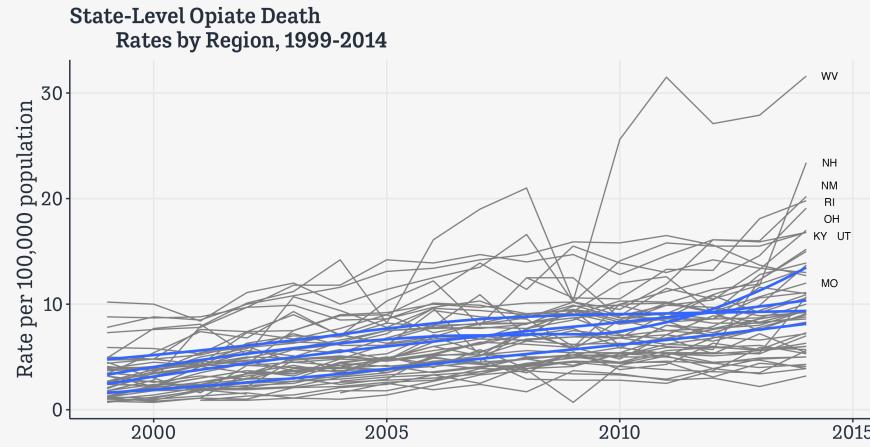
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2   group_by(region) %>%
3     slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6   ggplot(aes(x = year,
7             y = adjusted)) +
8   geom_line(aes(group = state),
9             color = "gray50") +
10  geom_smooth(aes(group = region),
11             se = FALSE) +
12  ggrepel::geom_text_repel(
13    data = st_top,
14    mapping = aes(x = year,
15                  y = adjusted,
16                  label = abbr),
17    size = 3,
18    segment.color = NA,
19    nudge_x = 0.5) +
20  coord_cartesian(c(min(opiates$year),
21                    max(opiates$year) + 1))
```



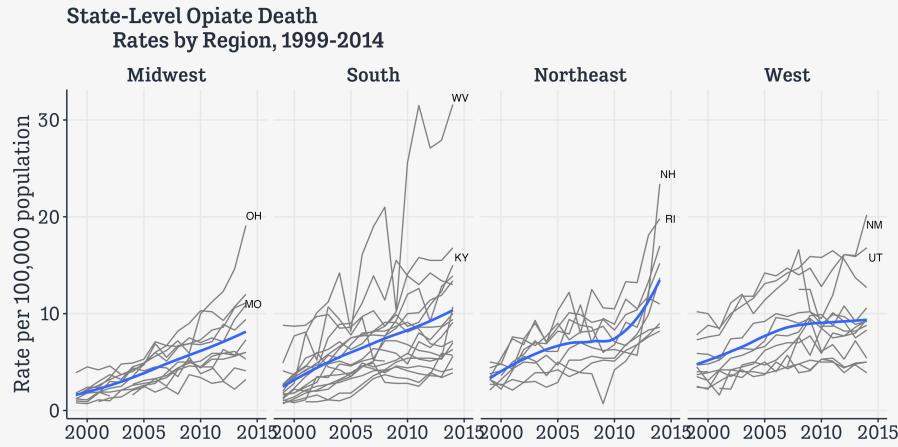
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2 group_by(region) %>%
3 slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6 ggplot(aes(x = year,
7             y = adjusted)) +
8 geom_line(aes(group = state),
9             color = "gray50") +
10 geom_smooth(aes(group = region),
11              se = FALSE) +
12 ggrepel::geom_text_repel(
13   data = st_top,
14   mapping = aes(x = year,
15                 y = adjusted,
16                 label = abbr),
17   size = 3,
18   segment.color = NA,
19   nudge_x = 0.5) +
20 coord_cartesian(c(min(opiates$year),
21                   max(opiates$year) + 1)) +
22 labs(x = NULL,
23       y = "Rate per 100,000 population",
24       title = "State-Level Opiate Death
25         Rates by Region, 1999-2014")
```



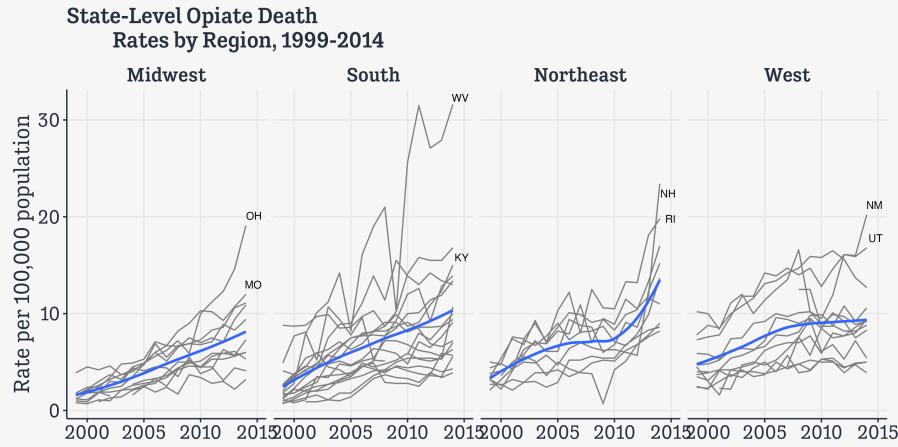
Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2   group_by(region) %>%
3     slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6   ggplot(aes(x = year,
7             y = adjusted)) +
8   geom_line(aes(group = state),
9             color = "gray50") +
10  geom_smooth(aes(group = region),
11             se = FALSE) +
12  ggrepel::geom_text_repel(
13    data = st_top,
14    mapping = aes(x = year,
15                  y = adjusted,
16                  label = abbr),
17    size = 3,
18    segment.color = NA,
19    nudge_x = 0.5) +
20  coord_cartesian(c(min(opiates$year),
21                    max(opiates$year) + 1)) +
22  labs(x = NULL,
23        y = "Rate per 100,000 population",
24        title = "State-Level Opiate Death
25          Rates by Region, 1999-2014") +
26  facet_wrap(~ reorder(region, adjusted,
27                      na.rm = TRUE),
28            nrow = 1)
```

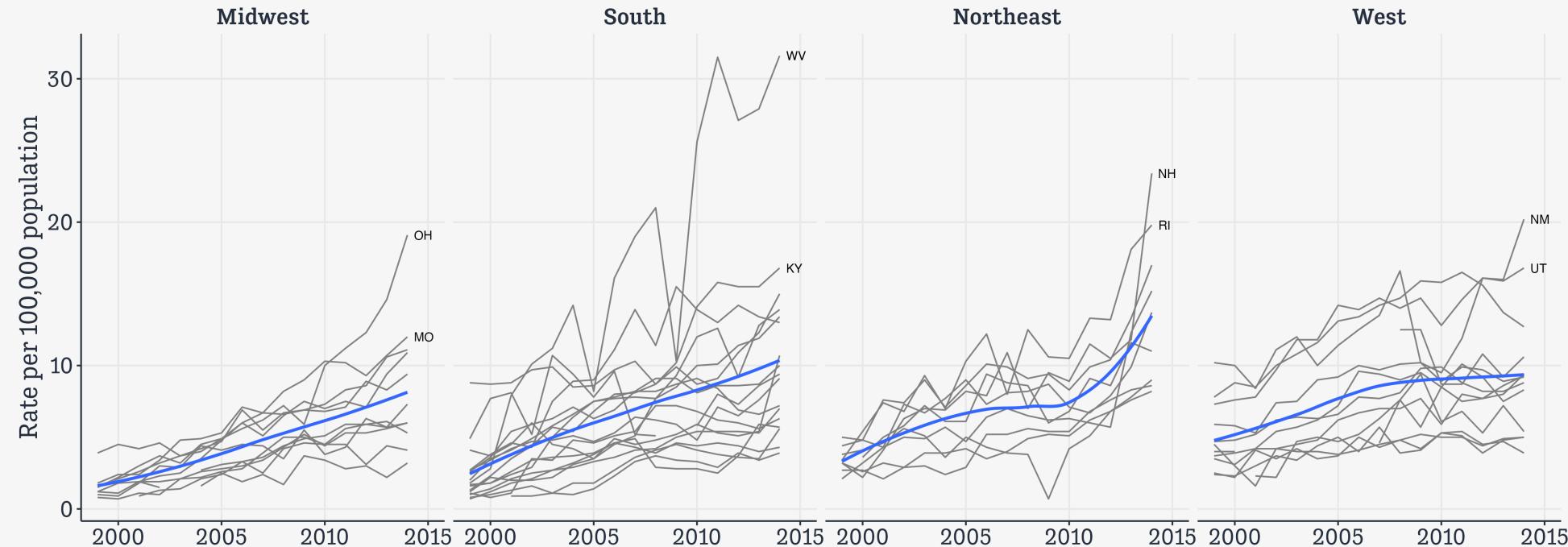


Opiates Time Series plot

```
1 st_top ← opiates %>% filter(year == max(year), abbr ≠ "DC")
2 group_by(region) %>%
3 slice_max(order_by = adjusted, n = 2)
4
5 opiates %>%
6 ggplot(aes(x = year,
7             y = adjusted)) +
8 geom_line(aes(group = state),
9             color = "gray50") +
10 geom_smooth(aes(group = region),
11              se = FALSE) +
12 ggrepel::geom_text_repel(
13   data = st_top,
14   mapping = aes(x = year,
15                 y = adjusted,
16                 label = abbr),
17   size = 3,
18   segment.color = NA,
19   nudge_x = 0.5) +
20 coord_cartesian(c(min(opiates$year),
21                   max(opiates$year) + 1)) +
22 labs(x = NULL,
23       y = "Rate per 100,000 population",
24       title = "State-Level Opiate Death
25           Rates by Region, 1999-2014") +
26 facet_wrap(~ reorder(region, adjusted,
27                      na.rm = TRUE),
28                      nrow = 1)
```



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



Regional trends in opiate-related mortality.