Topic8 Mapping Electoral College Predictions

Josh Clinton

10/5/2021

Today

- Maps in `ggplot` (one way).
- Pulling it together. State level predictions and Electoral College.

Our data this time is all state-level presidential polls conducted in 2020.

```
load(file = "data/StatePresidentalVote2020.Rdata")
glimpse(dat)
```

```
## Rows: 1,545
## Columns: 19
## $ poll.id        <dbl> 1, 13, 40, 41, 30, 24, 28, 29, 49, 26, 11, 12, 14, 48, ~
## $ Geography      <chr> "UT", "CT", "WI", "CA", "MI", "FL", "GA", "MI", "WA", "~
## $ StartDate      <date> 2020-03-21, 2020-03-24, 2020-03-24, 2020-03-28, 2020-0~
## $ EndDate        <date> 2020-03-30, 2020-04-03, 2020-03-29, 2020-03-29, 2020-0~
## $ DaysinField    <dbl> 10, 11, 6, 2, 3, 5, 2, 2, 7, 3, 3, 3, 2, 2, 3, 4, 10, 1~
## $ MoE            <dbl> 2.8, 3.0, 4.2, NA, 4.0, 1.7, 3.0, 3.1, 4.1, 4.4, NA, NA~
## $ Mode           <chr> "Phone/Online", "Phone/Online", "Live phone - RDD", "Li~
## $ SampleSize     <dbl> 1331, 1000, 813, 962, 602, 3244, 1035, 1019, 583, 500, ~
## $ Biden          <dbl> 41, 47, 48, 67, 46, 46, 46, 48, 52, 42, 48, 50, 52, 38,~
## $ Trump          <dbl> 46, 34, 45, 29, 46, 40, 48, 45, 39, 49, 47, 41, 43, 49,~
## $ DemCertVote    <dbl> 38, 59, 49, 64, 51, 48, 50, 51, 58, 49, 49, 51, 49, 41,~
## $ RepCertVote    <dbl> 58, 39, 49, 34, 48, 51, 49, 48, 39, 50, 49, 48, 49, 58,~
## $ Winner         <chr> "Rep", "Dem", "Dem", "Dem", "Dem", "Rep", "Dem", "Dem",~
## $ poll.predicted <dbl> 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ Funded         <chr> "UtahPolicy.com & KUTV 2News", "Sacred Heart University~
## $ Conducted      <chr> "Y2 Analytics", "GreatBlue Research", "LHK Partners Inc~
## $ region         <fct> utah, connecticut, wisconsin, california, michigan, flo~
## $ margin         <dbl> -5, 13, 3, 38, 0, 6, -2, 3, 13, -7, 1, 9, 9, -11, 6, -1~
## $ DaysToED       <drtn> 218 days, 214 days, 219 days, 219 days, 216 days, 213 ~
```

Produce a summary by state by averaging across all polls that are done in a state.

```
State.mean <- dat %>%
  group_by(region) %>%
      summarize(BidenPct = mean(Biden, na.rm = TRUE),
                TrumpPct = mean(Trump, na.rm = TRUE),
                NumPolls = n())



State.mean
```

```
## # A tibble: 50 x 4
##    region      BidenPct TrumpPct NumPolls
##    <fct>          <dbl>    <dbl>    <int>
##  1 alabama         38.9     56.8       14
##  2 alaska          44.2     50.8       13
##  3 arizona         48.4     44.9      106
##  4 arkansas        38.1     58.6        8
##  5 california      61.8     31.8       20
##  6 colorado        53.4     40.2       28
##  7 connecticut     58.4     34.1        9
##  8 delaware        60.3     35.9        7
##  9 florida         48.6     45.5      129
## 10 georgia         47.4     46.6       73
## # ... with 40 more rows
```

Maps in R

There are several packages that let you do maps in R. Let us use one called `maps`.

```
#install.packages("maps")   # Install this if you don't already have it!
library(maps)
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

- This package contains the data needed to produce a map that we extract using the funciton `map_data`.
- The dimensions of `states48` are large because this is an object that contains the points of latitude and longitude needed to draw the map.

```
states48 <- map_data("state")
dim(states48)
```

```
## [1] 15537     6
```

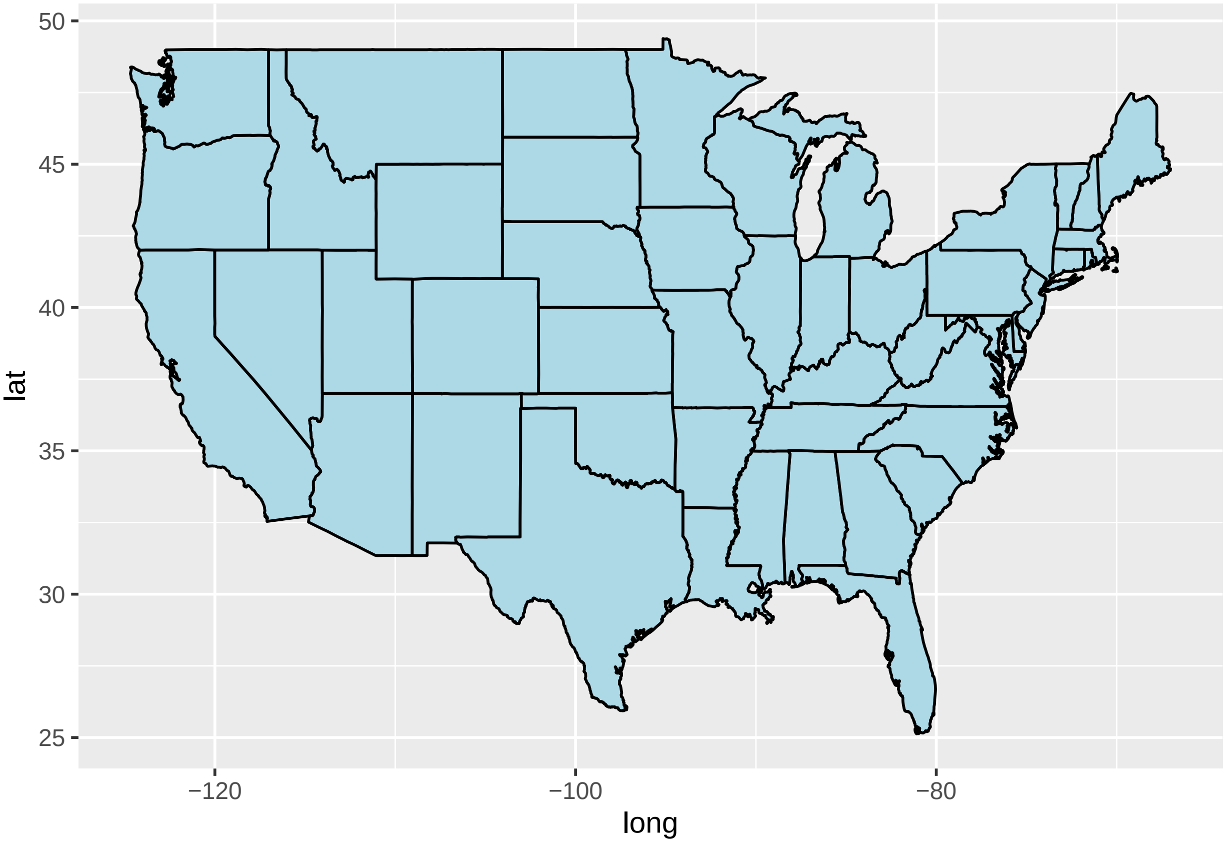
```
states48[1,]
```

```
##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
```

Creating the canvas

- The map is drawn using `geom_polygon` which uses the `long` and `lat` to draw the points in `ggplot`.
- Note that the default is pretty ugly (and distracting)

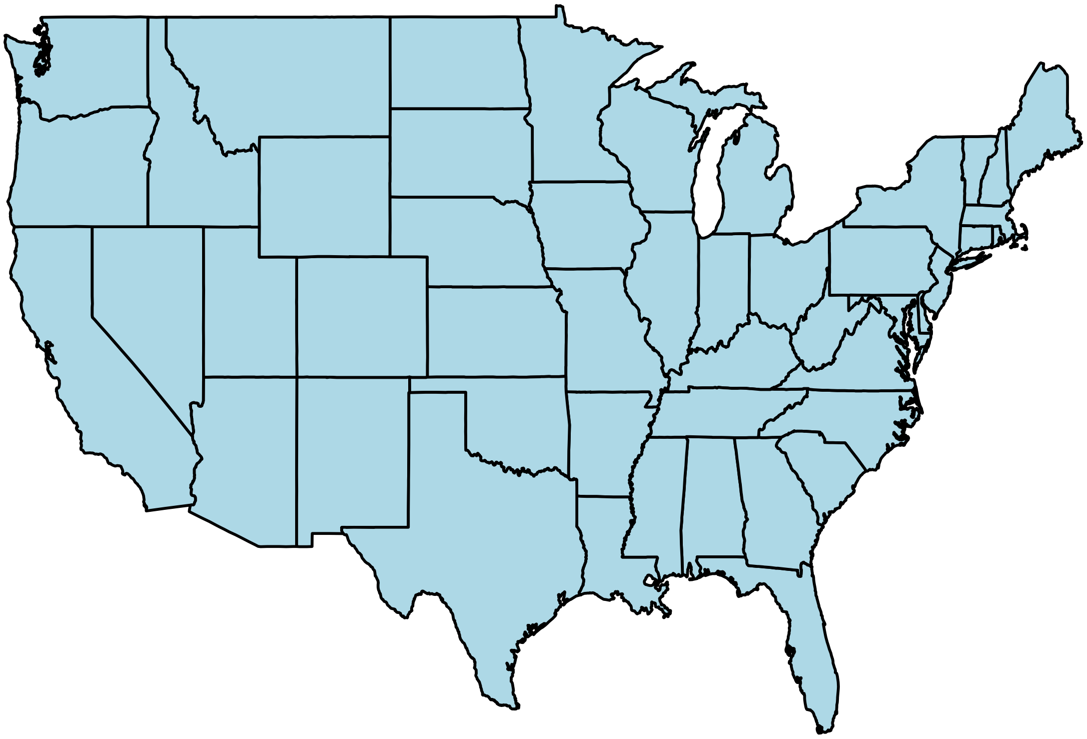
```
ggplot() +
  geom_polygon(data=states48, aes(x=long, y=lat, group=group),
               color="black",
               fill="lightblue" )
```



Let’s clean in up and save it as an object to be used later. (Note that we can create a custom theme to apply to a ggplot)

```
plotmap <- ggplot() +
  geom_polygon(data=states48, aes(x=long, y=lat, group=group),
               color="black",
               fill="lightblue" ) +
  labs(y = "") +
  labs(x = "") + scale_x_continuous(breaks=NULL, labels=NULL) +
  scale_y_continuous(breaks=NULL, labels=NULL) +
    theme(
    panel.border = element_blank(),
    panel.background = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
    )

plotmap
```



Get Some Data!

```
states48.join <- inner_join(states48,State.mean)
```

```
## Joining, by = "region"
```

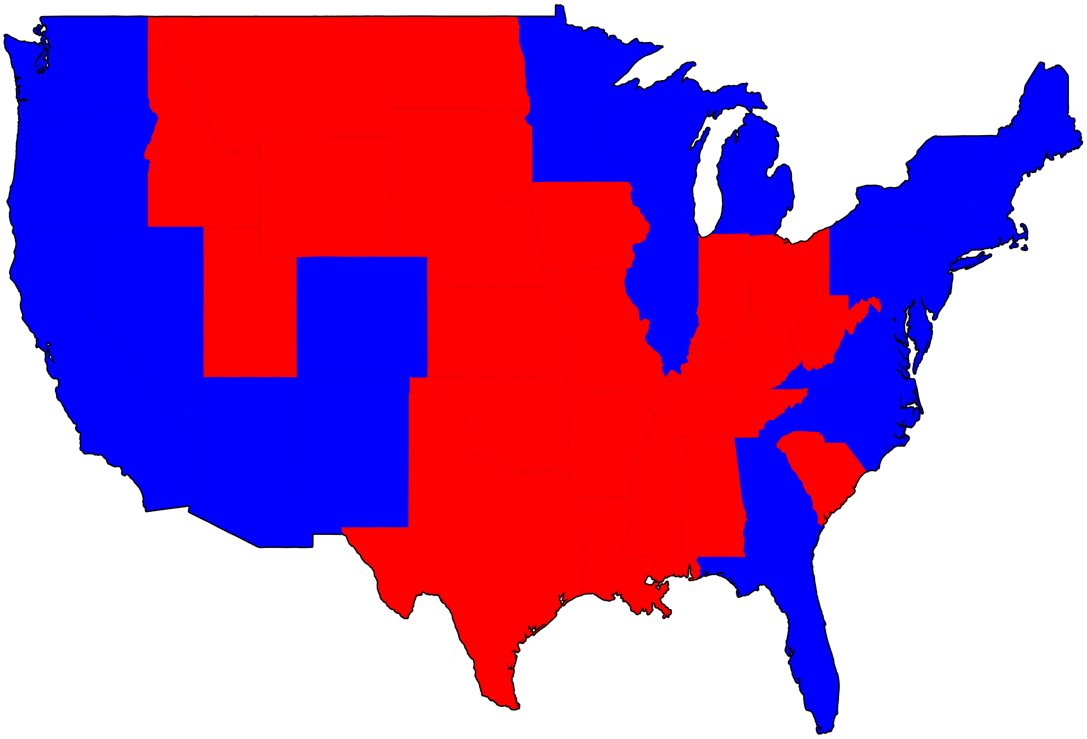
```
states48.join[1,]
```

```
##        long      lat group order  region subregion BidenPct TrumpPct NumPolls
## 1 -87.46201 30.38968     1     1 alabama      <NA> 38.92857 56.78571       14
```

- `inner_join()` merge dataframes if common in both (must have same name)
- `outer_join()` merge dataframes if in either

Produce a map that indicates predicted winners depending on the average polling done in each state using different polygons.

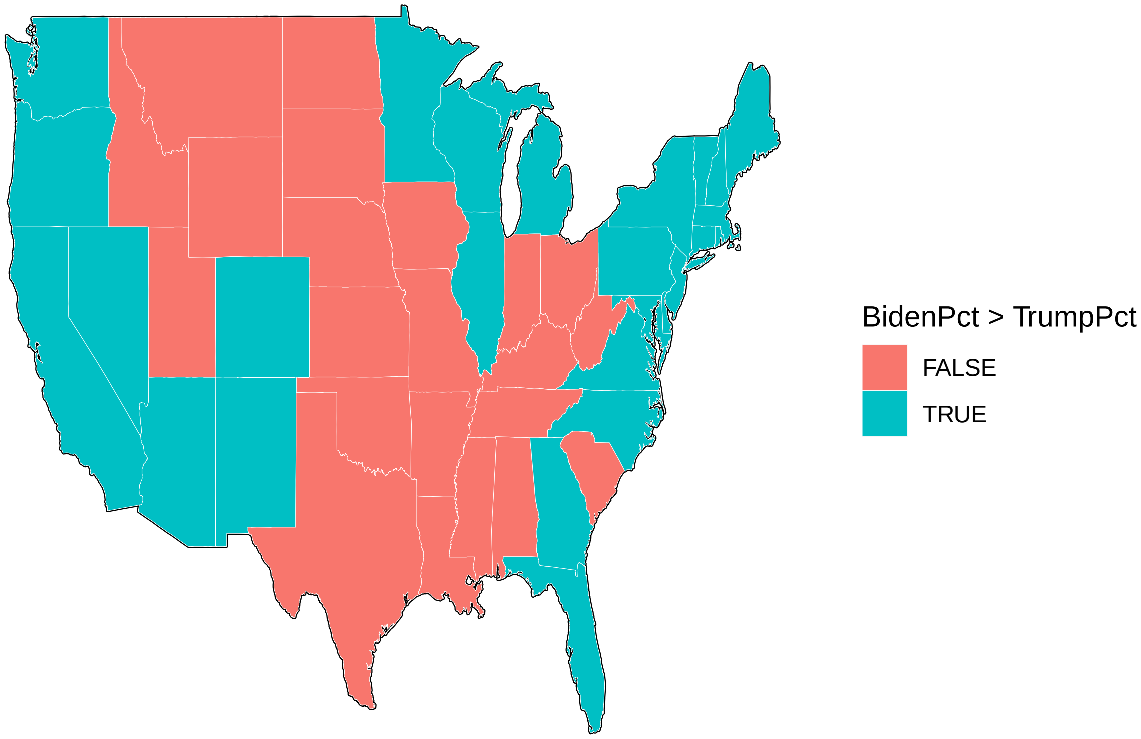
```
plotmap +
  geom_polygon(data=states48.join,
               aes(x=long, y=lat, group=group)) +
  geom_polygon(data=filter(states48.join,BidenPct > TrumpPct),
               aes(x=long, y=lat, group=group),fill="Blue") +
  geom_polygon(data=filter(states48.join,BidenPct < TrumpPct),
               aes(x=long, y=lat, group=group),fill="Red")
```



But we can do the same thing using `fill` – here defined to be which candidate is larger.

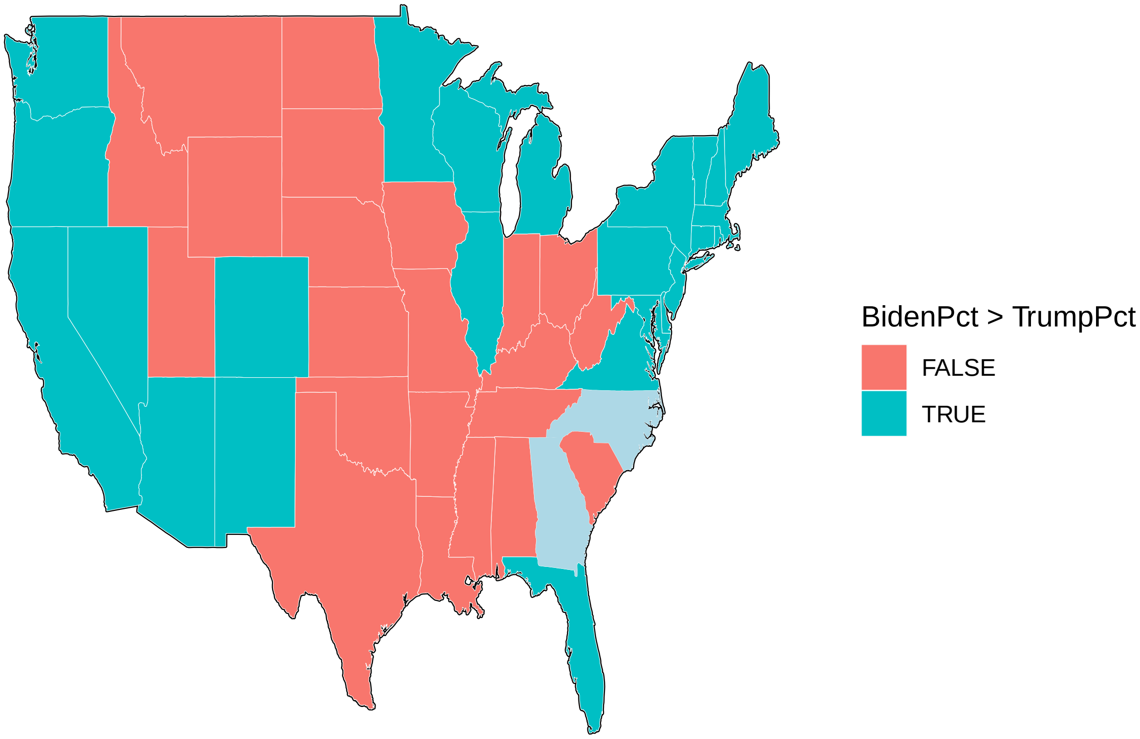
```
statewinner <- plotmap +
  geom_polygon(data=states48.join,
               aes(x=long, y=lat, group=group,
                   fill=BidenPct > TrumpPct),
               color="white", size = .1)

statewinner
```



- We can also add more nuanced information by denoting the location of states where the margin is less than 2 – i.e., very close.
- Because we are adding on top of the canvass we are just “repainting” the impacted states.

```
statewinner +
  geom_polygon(data=filter(states48.join,
               BidenPct > TrumpPct & BidenPct - TrumpPct < 2),
               aes(x=long, y=lat, group=group),
               fill="light blue") +
  geom_polygon(data=filter(states48.join,
               BidenPct < TrumpPct & BidenPct - TrumpPct > 2),
               aes(x=long, y=lat, group=group),
               fill="light red")
```



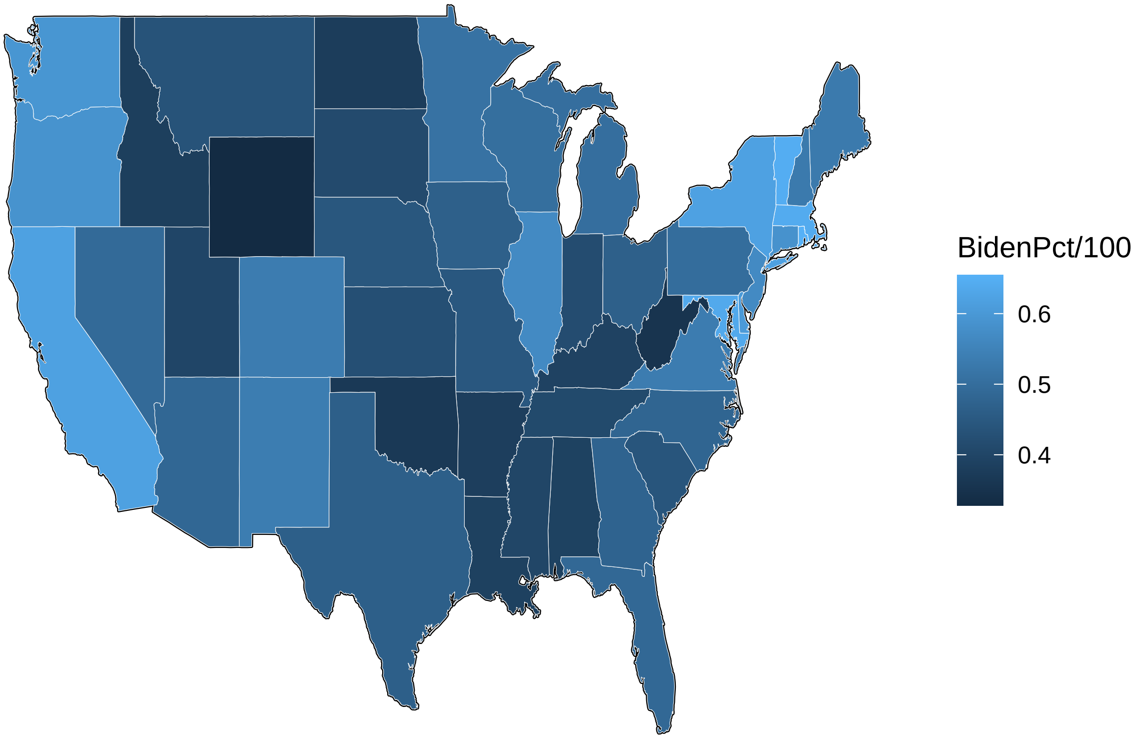
IN-CLASS: Can you plot the location of "Blow out" states? How do you define that?

Using Fill as a variable

- Instead of using different colors to define categories of difference we can also use the `fill` to try to convey information about the variation of continuous differences – e.g., the level of support.

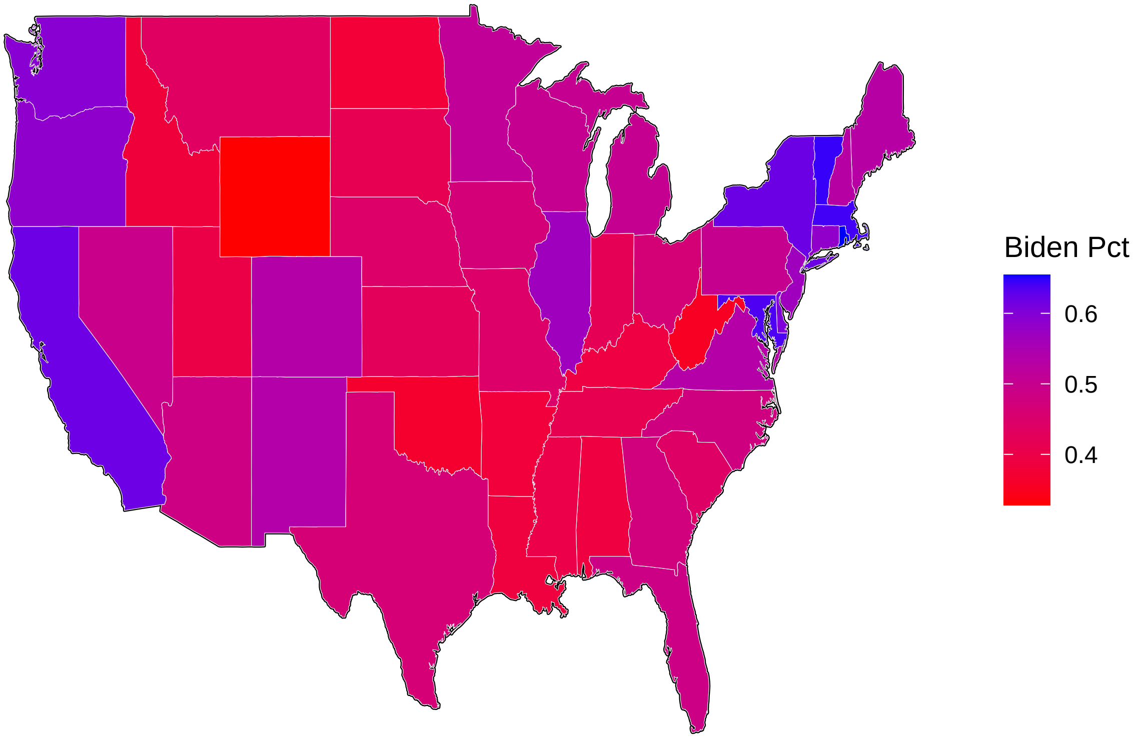
```
plot1 <- plotmap +
  geom_polygon(data=states48.join,
               aes(x=long, y=lat, group=group,
                   fill=BidenPct/100),
               color="WHITE", size = .1)

plot1
```

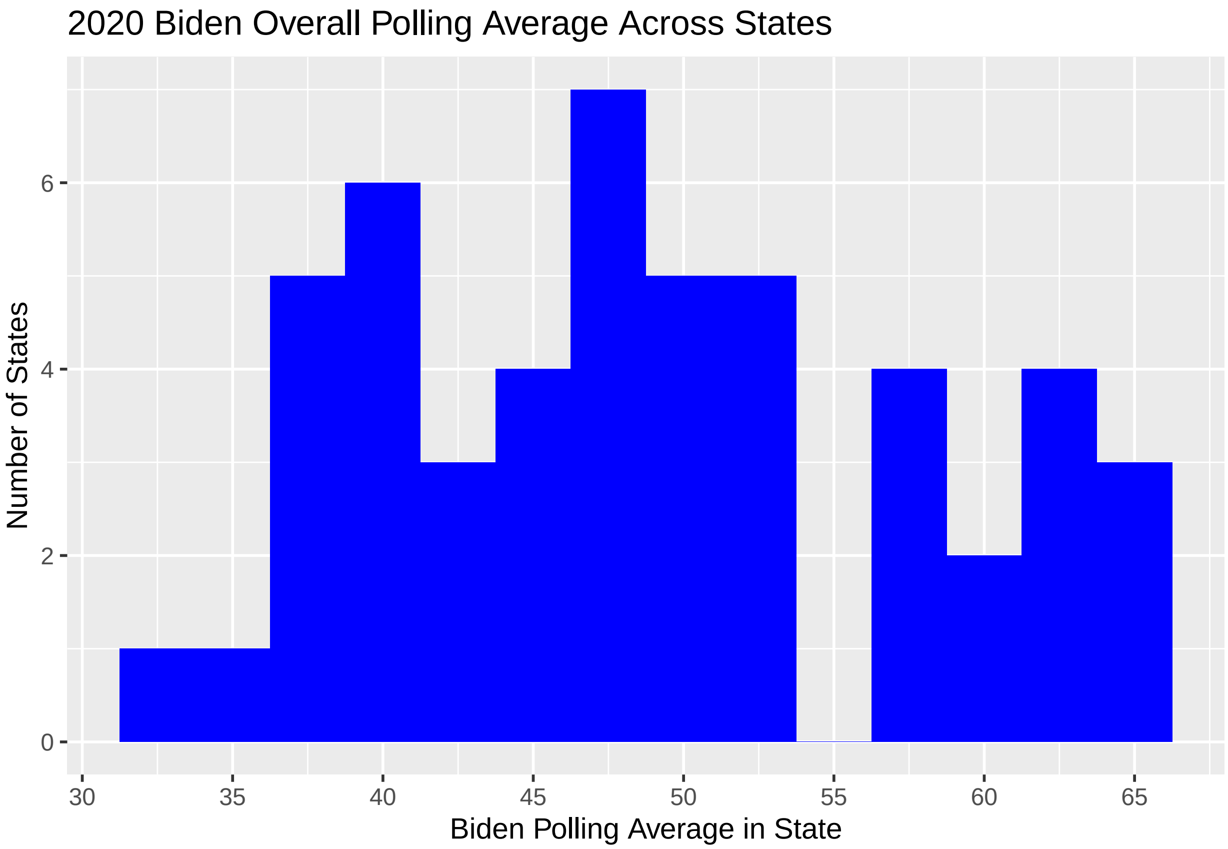


And we can also define the scale in terms of the color associated with the min and max values.

```
plot1 + scale_fill_continuous(name="Biden Pct",
                              low = "Red", high = "Blue")
```



- But how much better is this than a histogram? Does the geography really matter here?
- Why a histogram and not a barplot? What would a barplot look like?


2020 Biden Overall Polling Average Across States
Number of States
0
2
4
6
30
35
40
45
50
55
60
65
Biden Polling Average in State