

Multivariate Analysis

Part 2: Visualizations

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/02/08

Slides Updated: 2023-02-07

Agenda

1. Finish up last lecture
2. Rules of visualization
3. Overtime Polls + `weighted.mean()`
4. Predicting winner with Electoral College Votes

Conditional Analysis of Polls

```
require(tidyverse)
poll <- readRDS('../data/Pres2020_PV.Rds')

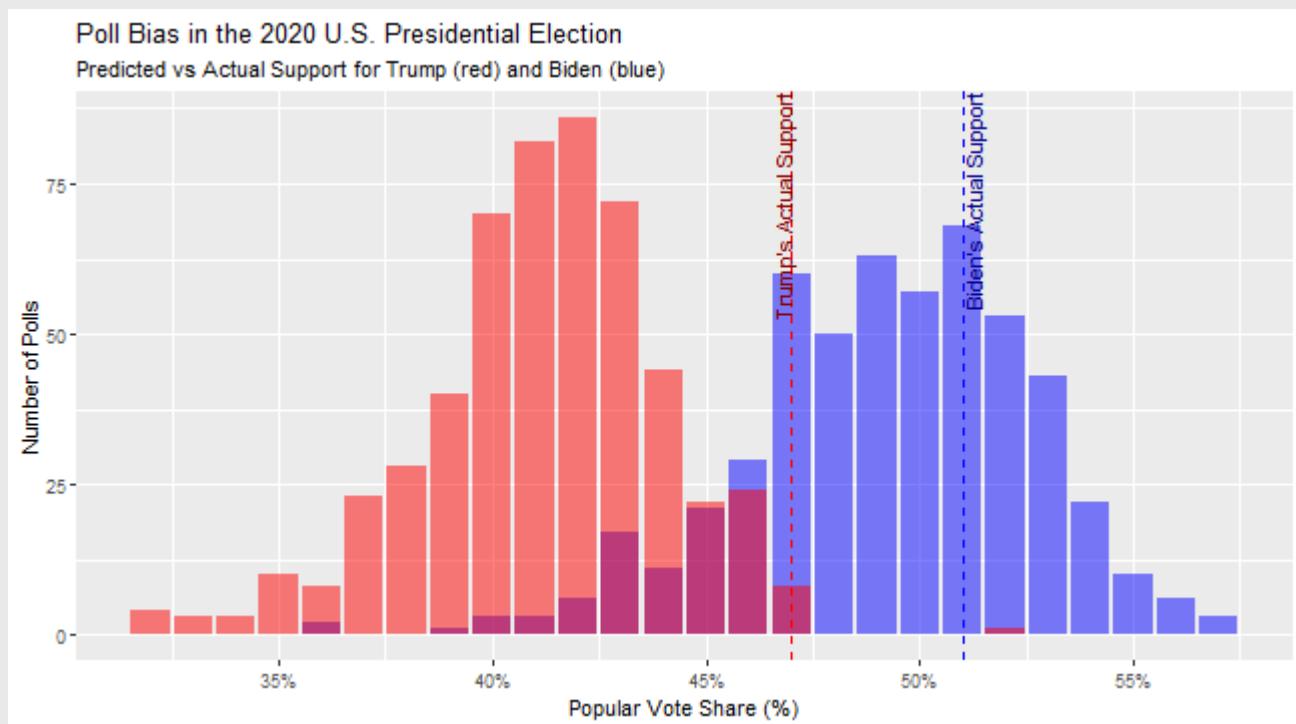
poll <- poll %>%
  mutate(Trump = Trump/100,
        Biden = Biden/100,
        margin = round(Biden - Trump, 2))
```

The Research Question

```
pRQ <- poll %>%
  ggplot() +
  geom_bar(aes(x = Biden*100),fill = 'blue',alpha = .5) +
  geom_bar(aes(x = Trump*100),fill = 'red',alpha = .5) +
  geom_vline(xintercept = 47,linetype = 'dashed',color= 'red') +
  geom_vline(xintercept = 51,linetype = 'dashed',color= 'blue')+
  annotate(geom = 'text',x = c(47),y = Inf,angle = 90,hjust = 1,vjust
= 0,label = c("Trump's Actual Support"),color = 'darkred') +
  annotate(geom = 'text',x = c(51),y = Inf,angle = 90,hjust = 1,vjust
= 1,label = c("Biden's Actual Support"),color = 'darkblue') +
  labs(title = 'Poll Bias in the 2020 U.S. Presidential Election',
       subtitle = 'Predicted vs Actual Support for Trump (red) and
Biden (blue)',
       x = 'Popular Vote Share (%)',
       y = 'Number of Polls') +
  scale_x_continuous(breaks = seq(30,60,by = 5),labels = function(x)
paste0(x,'%'))
```

The Research Question

pRQ



The Research Question

```
poll %>% # Proportion that under-predict
  summarise(propBidenUP = mean(Biden < .51),
            propTrumpUP = mean(Trump < .47))
```

```
## # A tibble: 1 × 2
##   propBidenUP propTrumpUP
##       <dbl>      <dbl>
## 1     0.612     0.983
```

```
poll %>% # Average under-prediction
  summarise(avgBidenErr = mean(.51 - Biden),
            avgTrumpErr = mean(.47 - Trump))
```

```
## # A tibble: 1 × 2
##   avgBidenErr avgTrumpErr
##       <dbl>      <dbl>
## 1     0.0175     0.0577
```

Theorizing

- **Research Question:** Why do polls under-predict Trump more than Biden?
 1. Unrepresentative samples (how were respondents contacted?)
 2. Small samples (how many respondents?)
 3. Shy Trump Voters / trolls (lying respondents)
 4. Timing (closer to the election → less biased)

Theorizing

- A fifth explanation?
- Anti-Trump media!

Donald J. Trump

@realDonaldTrump

Following

Any negative polls are fake news, just like the CNN, ABC, NBC polls in the election. Sorry, people want border security and extreme vetting.

RETWEETS LIKES

19,266 80,481

7:01 AM - 6 Feb 2017

Theorizing

- Theory #1: Does the "mode" of the survey matter?
 - I.e., if you only call people on landlines, who do you reach?
 - And how might they differ from the general population?
- Assumption 1: Younger people do not use landlines, meaning that surveys which rely on random digit dialing (RDD) will get disproportionately older respondents.
- Assumption 2: Younger voters are more progressive, making them less likely to support Trump.
- Theory: Surveys that use RDD will find more support for Trump than Biden.

Analyzing

- Plot the Biden-Trump vote margin by mode type

```
poll %>%  
  count(Mode)
```

```
## # A tibble: 9 × 2  
##   Mode          n  
##   <chr>     <int>  
## 1 IVR           1  
## 2 IVR/Online    47  
## 3 Live phone - RBS    13  
## 4 Live phone - RDD    51  
## 5 Online        366  
## 6 Online/Text    1  
## 7 Phone - unknown  1  
## 8 Phone/Online    19  
## 9 <NA>          29
```

- So many modes of interviewing people!

(Soft) Rules of Visualization

- Variable `type` informs visualization

1. Univariate

- Categorical data: `geom_bar()`
- Continuous data: `geom_histogram()` or `geom_density()`

2. Bivariate

- Categorical X Categorical: `geom_bar()`
- Binary X Continuous: `geom_histogram()` or `geom_density()`
- Categorical X Continuous: `geom_boxplot()` or `geom_violin()`
- Continuous X Continuous: `geom_point()`

Beyond Bivariate

1. Trivariate

- Categorical X Categorical X Continuous: `geom_tile()`
- Continuous X Continuous X Categorical: `geom_point() + color`
- Continuous X Continuous X Continuous: `geom_point() + color/size`
- Latitude X Longitude X Categorical / Continuous: Maps!
- Var X Var X Time: Animated!
- (Beyond the scope of this course, but get creative!)

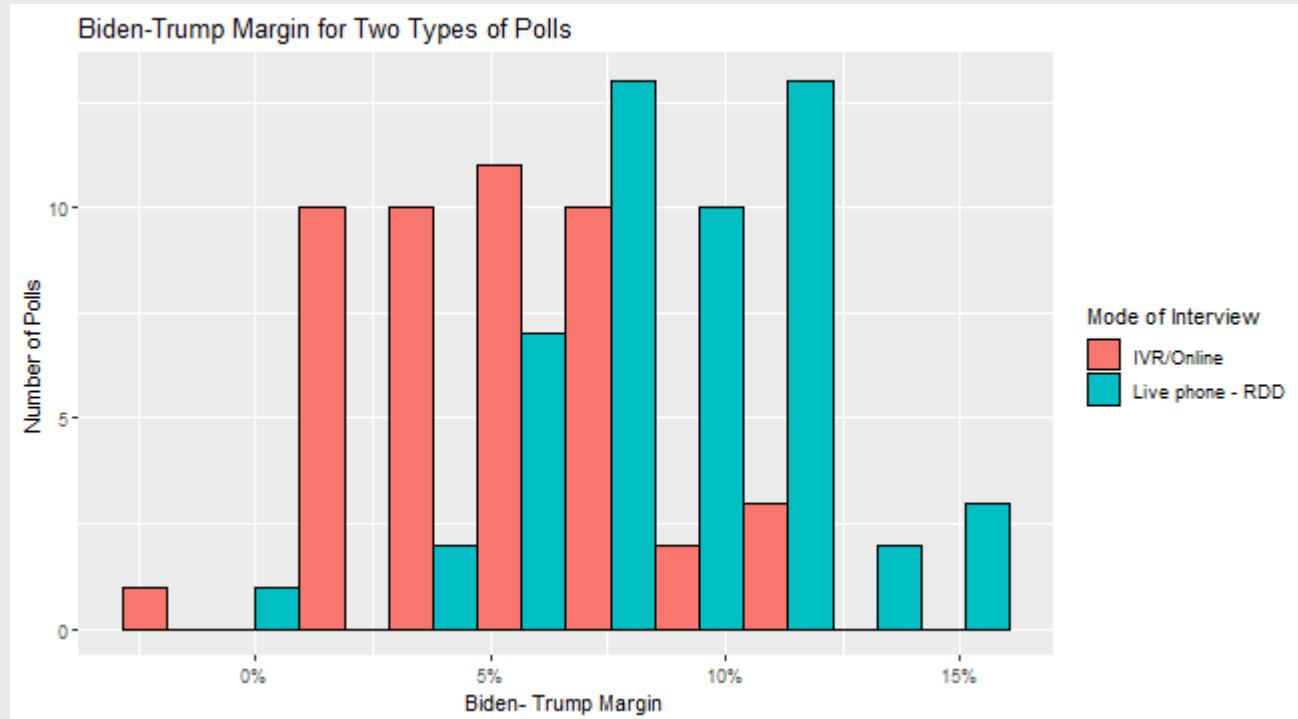
Analyzing

- For now, just focus on `IVR/Online` versus `Live phone - RDD`
- Since `margin` is a continuous variable, use `geom_histogram`

```
pMode <- poll %>%
  filter(Mode == "IVR/Online" | Mode == "Live phone - RDD") %>%
  ggplot(aes(x= margin, fill = Mode)) +
  labs(y = "Number of Polls",
       x = "Biden- Trump Margin",
       title = "Biden-Trump Margin for Two Types of Polls",
       fill = "Mode of Interview") +
  geom_histogram(bins=10, color="black", position="dodge") +
  scale_x_continuous(breaks=seq(-.1,.2,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

Mode Matters!

pMode



- But results are **inconsistent** with our [theory](#)!

Visualization

- How can we improve this? Perhaps `geom_density()` and `geom_vline()`?

```
toplot <- poll %>%
  filter(Mode == "IVR/Online" | Mode == "Live phone - RDD")

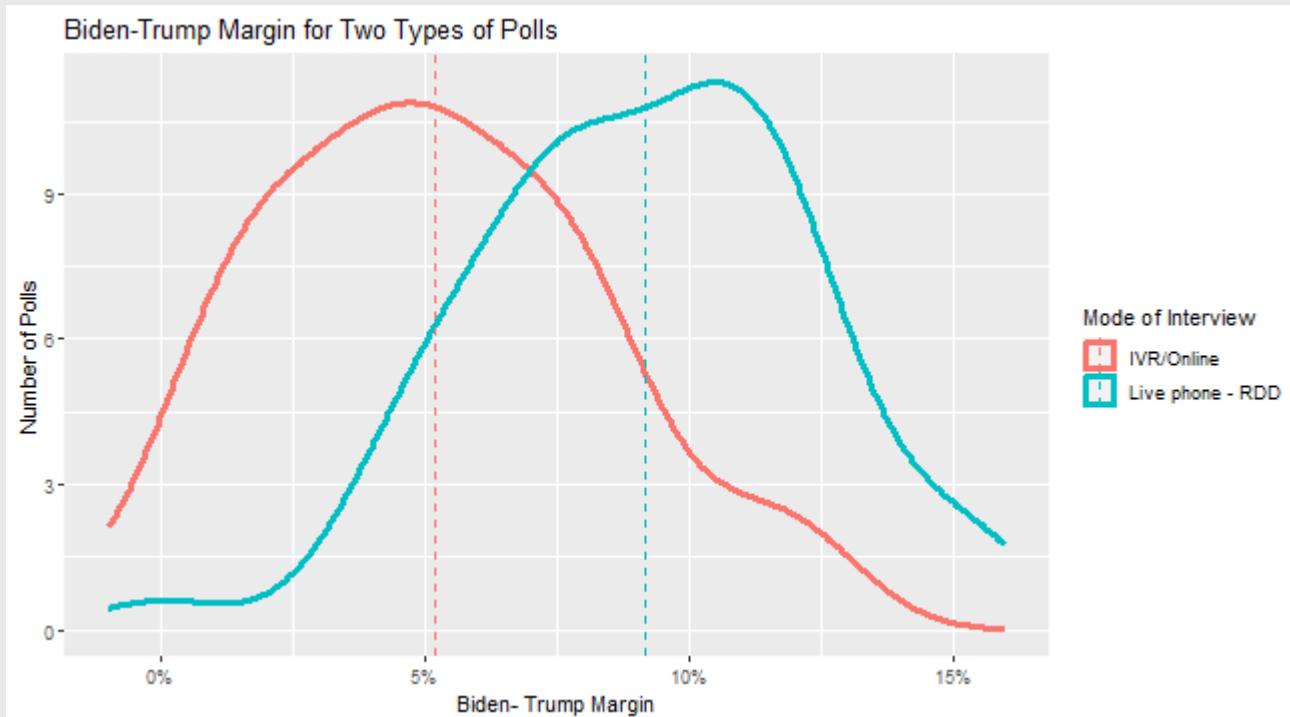
pModeDens <- toplot %>%
  ggplot(aes(x= margin, color = Mode)) +
  labs(y = "Number of Polls",
       x = "Biden- Trump Margin",
       title = "Biden-Trump Margin for Two Types of Polls",
       color = "Mode of Interview") +
  geom_density(lwd = 1.2) +
  scale_x_continuous(breaks=seq(-.1,.2,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  geom_vline(data = toplot %>%
              group_by(Mode) %>%
              summarise(margin = mean(margin)),aes(xintercept =
margin,color = Mode),linetype = 'dashed')
```

```
## Warning: Using `size` aesthetic for lines was deprecated in
ggplot2
## 3.4.0.
```

Visualization

- How can we improve this? Perhaps `geom_density()` and `geom_vline()`?

pModeDens



More Modes

- `geom_histogram()` and `geom_density()` less useful for more comparisons
- First, let's drop modes that were hardly used

```
(toKeep <- poll %>%  
  count(Mode) %>%  
  filter(n > 5,  
         !is.na(Mode)))
```

```
## # A tibble: 5 × 2  
##   Mode           n  
##   <chr>        <int>  
## 1 IVR/Online     47  
## 2 Live phone - RBS    13  
## 3 Live phone - RDD    51  
## 4 Online        366  
## 5 Phone/Online    19
```

```
toplot <- poll %>% filter(Mode %in% toKeep$Mode)
```

More Modes

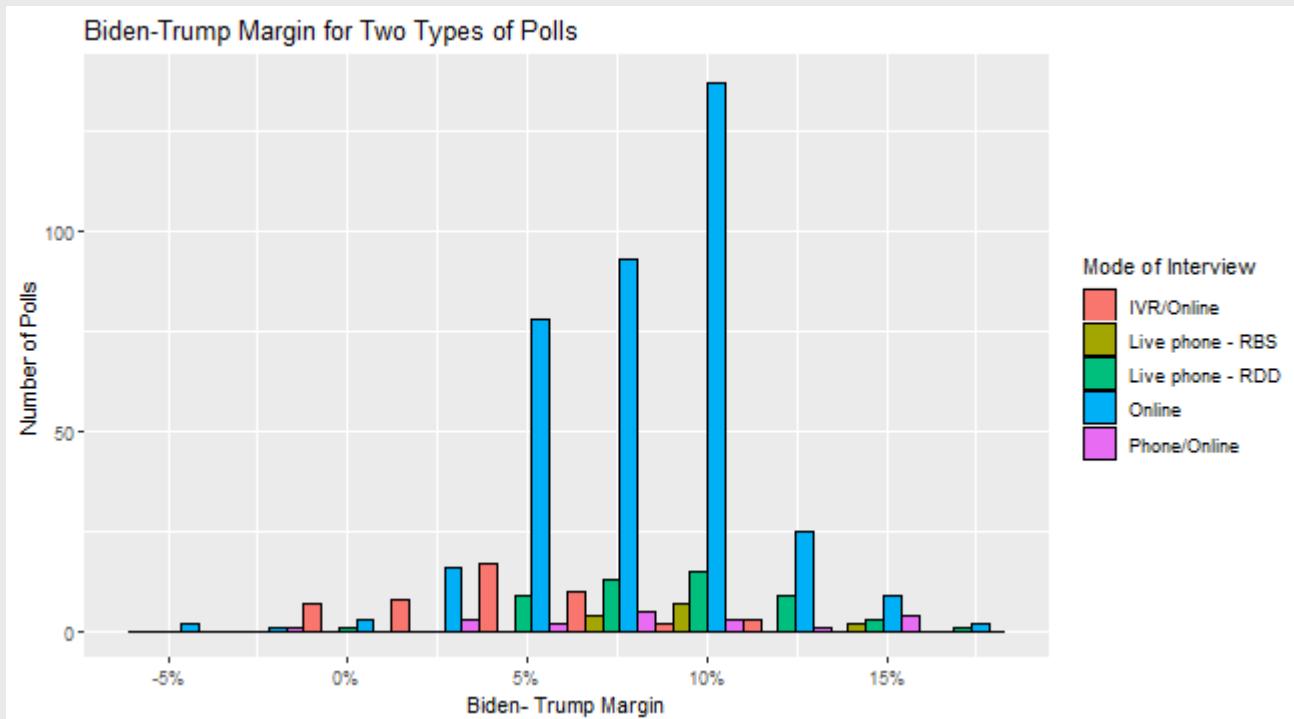
- How hard is `geom_histogram()` with more categories?

```
pModeHist <- toplot %>%
  ggplot(aes(x= margin, fill = Mode)) +
  labs(y = "Number of Polls",
       x = "Biden- Trump Margin",
       title = "Biden-Trump Margin for Two Types of Polls",
       fill = "Mode of Interview") +
  geom_histogram(color = 'black',position = 'dodge',bins = 10) +
  scale_x_continuous(breaks=seq(-.1,.2,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

More Modes

- How hard is `geom_histogram()` with more categories?

```
pModeHist
```



More Modes

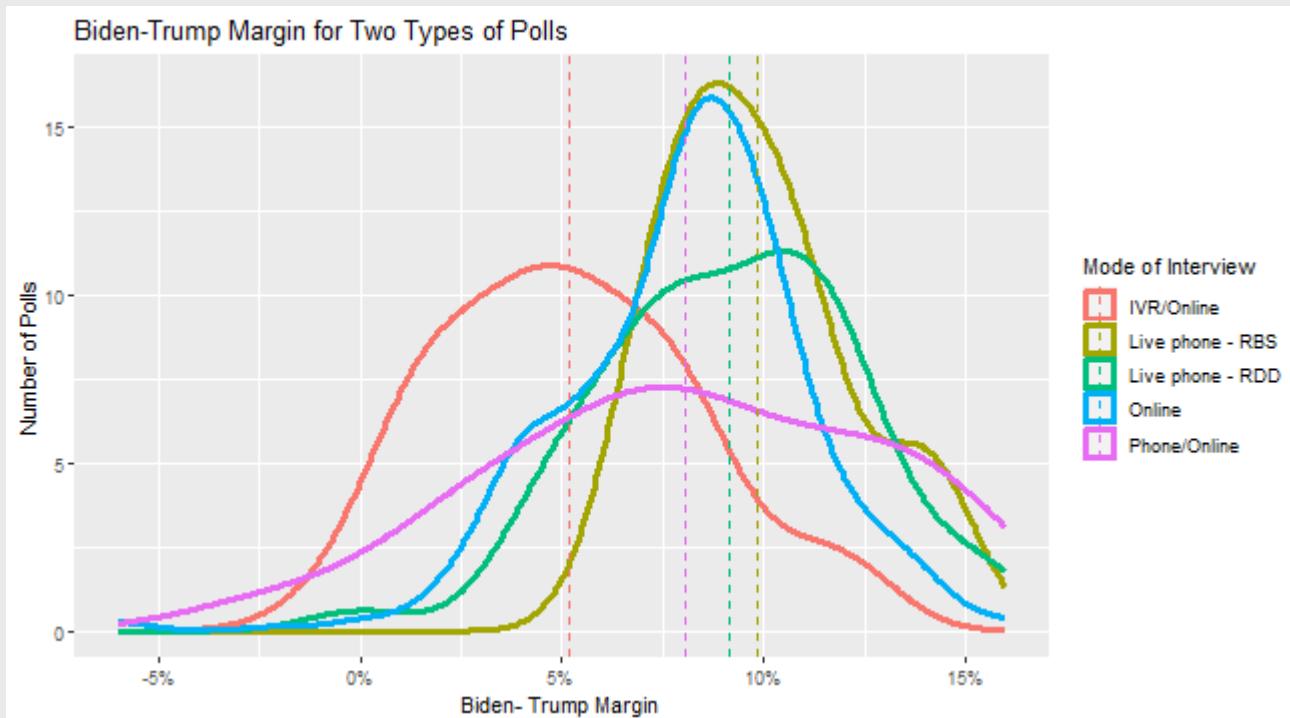
- How hard is `geom_density()` with more categories?

```
pModeDens <- toplot %>%
  ggplot(aes(x= margin, color = Mode)) +
  labs(y = "Number of Polls",
       x = "Biden- Trump Margin",
       title = "Biden-Trump Margin for Two Types of Polls",
       color = "Mode of Interview") +
  geom_density(lwd = 1.2) +
  scale_x_continuous(breaks=seq(-.1,.2,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  geom_vline(data = toplot %>%
              group_by(Mode) %>%
              summarise(margin = mean(margin)),aes(xintercept =
margin,color = Mode),linetype = 'dashed')
```

More Modes

- How hard is `geom_density()` with more categories?

pModeDens



geom_boxplot()

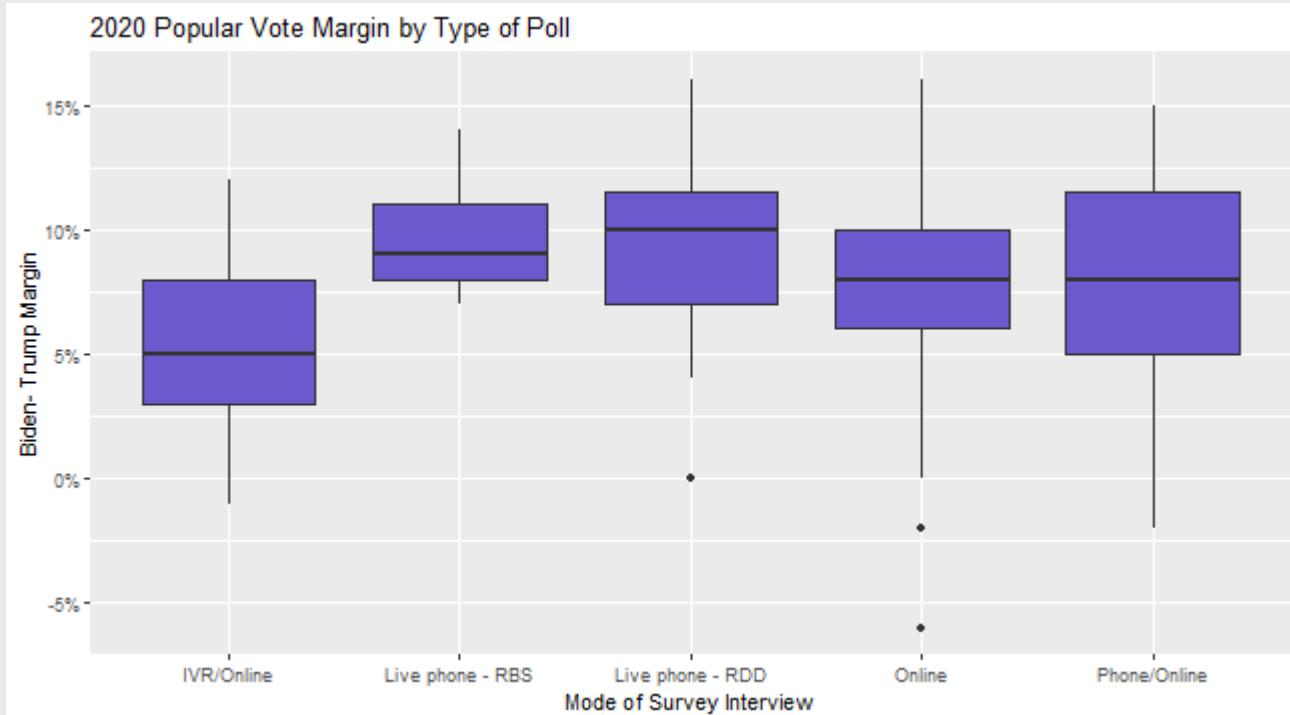
- More categories requires more compact ways of visualizing distributions

```
pModeBox <- toplot %>%  
  ggplot(aes(x = Mode, y = margin)) +  
    labs(x = "Mode of Survey Interview",  
         y = "Biden- Trump Margin",  
         title = "2020 Popular Vote Margin by Type of Poll") +  
    geom_boxplot(fill = "slateblue") +  
    scale_y_continuous(breaks=seq(-.1,.2,by=.05),  
                       labels= scales::percent_format(accuracy = 1))
```

geom_boxplot()

- More categories requires more compact ways of visualizing distributions

pModeBox



Ordering Unordered Categories

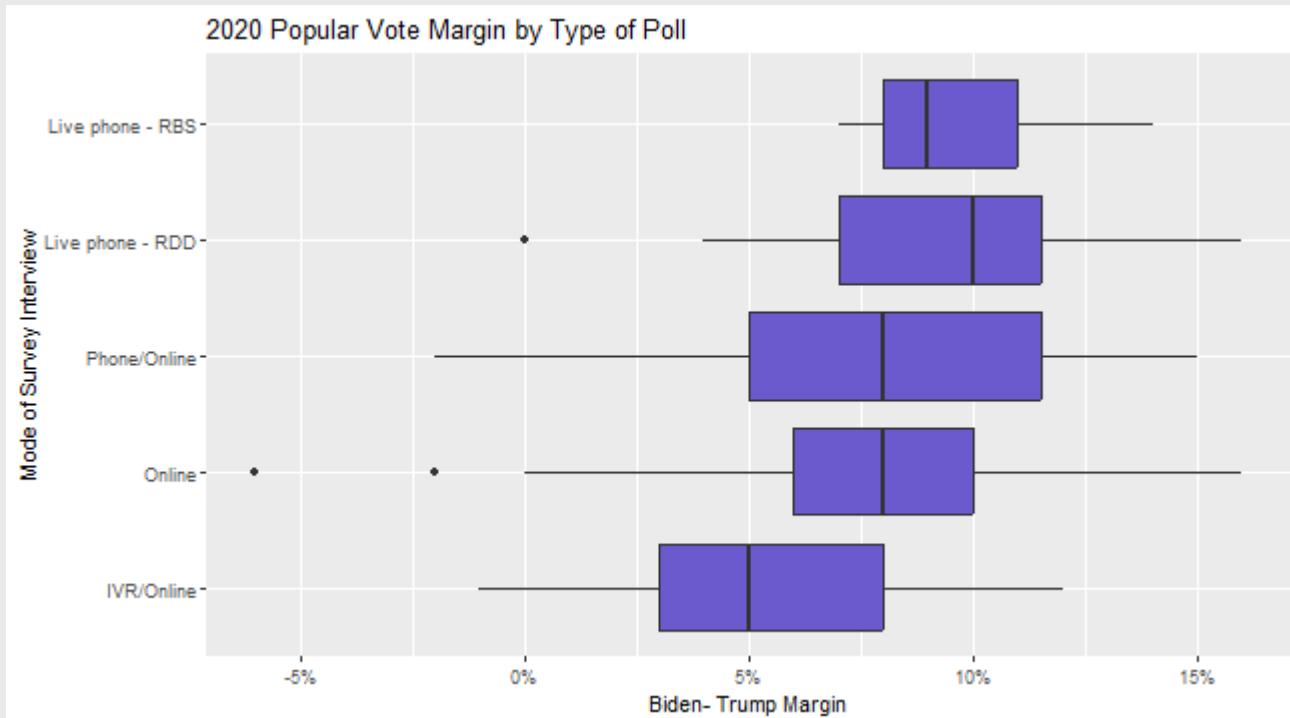
- We can use `reorder()` to arrange categories by the data

```
pModeBox <- toplot %>%  
  ggplot(aes(x = reorder(Mode,margin), y = margin)) +  
    labs(x = "Mode of Survey Interview",  
         y = "Biden- Trump Margin",  
         title = "2020 Popular Vote Margin by Type of Poll") +  
    geom_boxplot(fill = "slateblue") +  
    scale_y_continuous(breaks=seq(-.1,.2,by=.05),  
                       labels= scales::percent_format(accuracy = 1))
```

Ordering Unordered Categories

- We can use `reorder()` to arrange categories by the data

```
pModeBox + coord_flip()
```



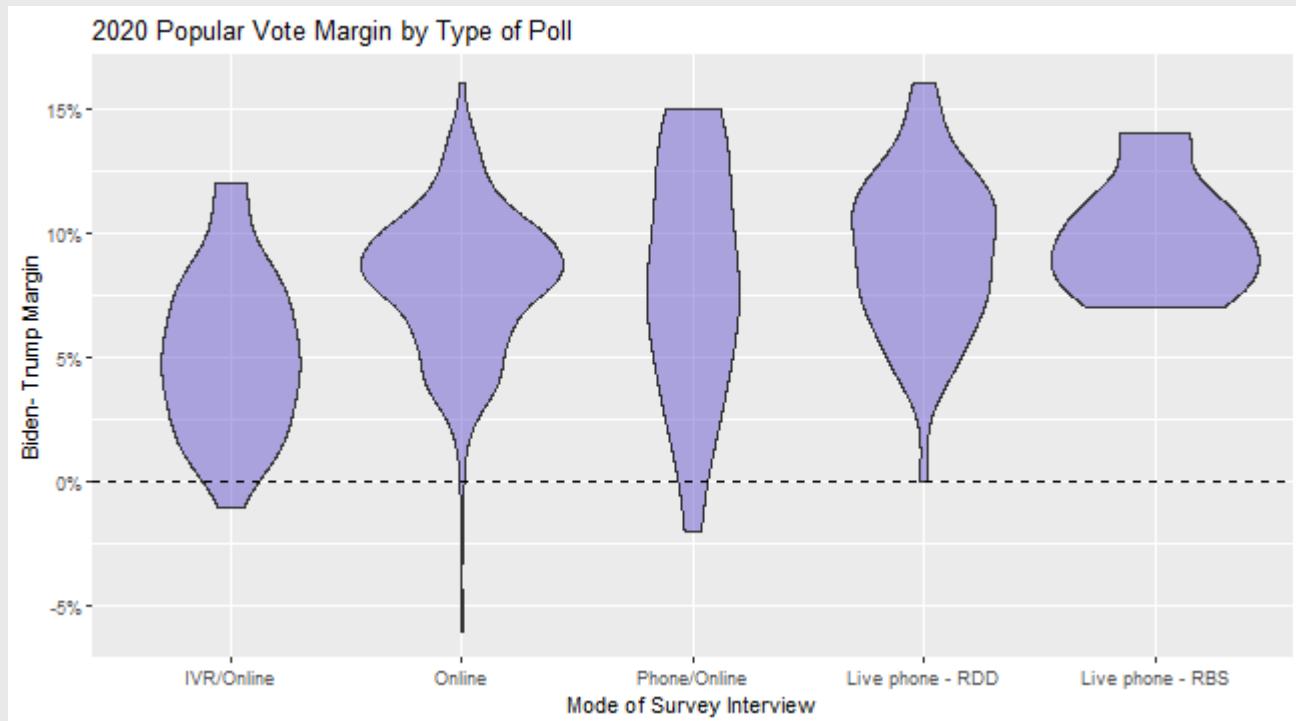
geom_violin()

- Boxplots are cleaner than densities and histograms for multiple categories
- But we lose ability to see distributions within the 80% box

```
pModeViol <- toplot %>%
  ggplot(aes(x = reorder(Mode,margin), y = margin)) +
  labs(x = "Mode of Survey Interview",
       y = "Biden- Trump Margin",
       title = "2020 Popular Vote Margin by Type of Poll") +
  geom_violin(fill = "slateblue",alpha = .5) +
  scale_y_continuous(breaks=seq(-.1,.2,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

geom_violin()

```
pModeViol + geom_hline(yintercept = 0,linetype = 'dashed')
```



Continuous by Continuous

- For **conditional relationships** between two **continuous variables**, use `geom_point()`
- **Theory:** Are polls politically biased?
 - I.e., a Biden-friendly poll might **under**predict Trump support and **over**predict Biden support
- **Data:** Trump support conditional on Biden support

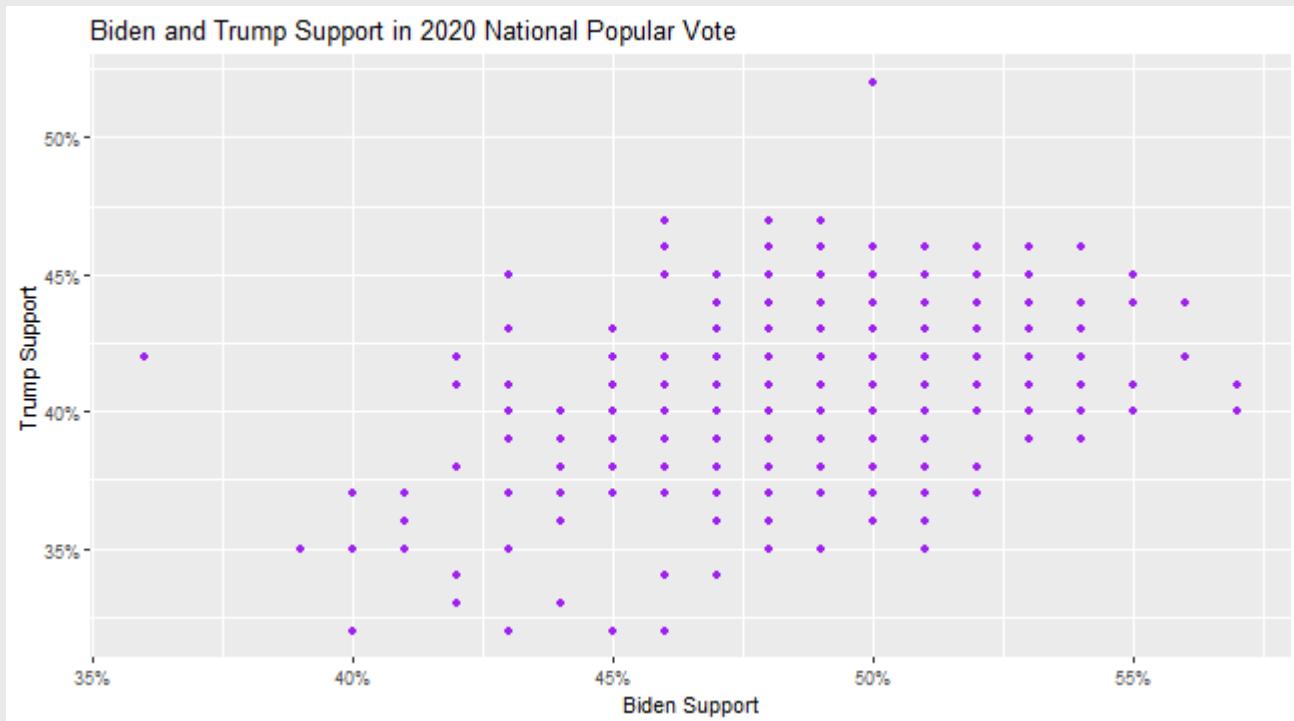
Analysis

- Plot Trump support versus Biden support

```
pSupp <- poll %>%
  ggplot(aes(x = Biden, y = Trump)) +
  labs(title="Biden and Trump Support in 2020 National Popular Vote",
       y = "Trump Support",
       x = "Biden Support") +
  geom_point(color="purple") +
  scale_y_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  scale_x_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

geom_scatter()

pSupp



- How many observations are at each point?

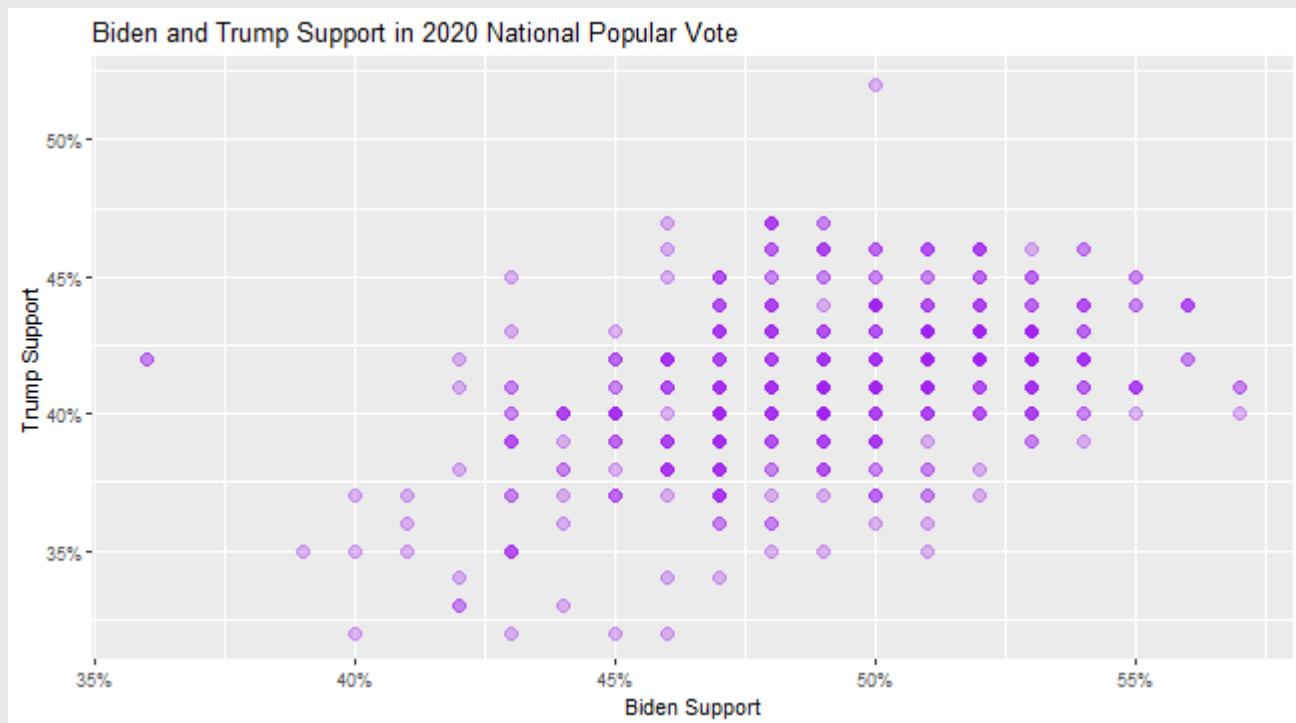
Tweaking alpha

- We can set the transparency of each point such that multiple points will show up darker
 - I.e., `alpha=.3` means that a single point will be 70% transparent, but 3 points on top of each other will be 10% transparent

```
pSupp <- poll %>%
  ggplot(aes(x = Biden, y = Trump)) +
  labs(title="Biden and Trump Support in 2020 National Popular Vote",
       y = "Trump Support",
       x = "Biden Support") +
  geom_point(color="purple",alpha = .3,size = 3) +
  scale_y_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  scale_x_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

Tweaking alpha

pSupp



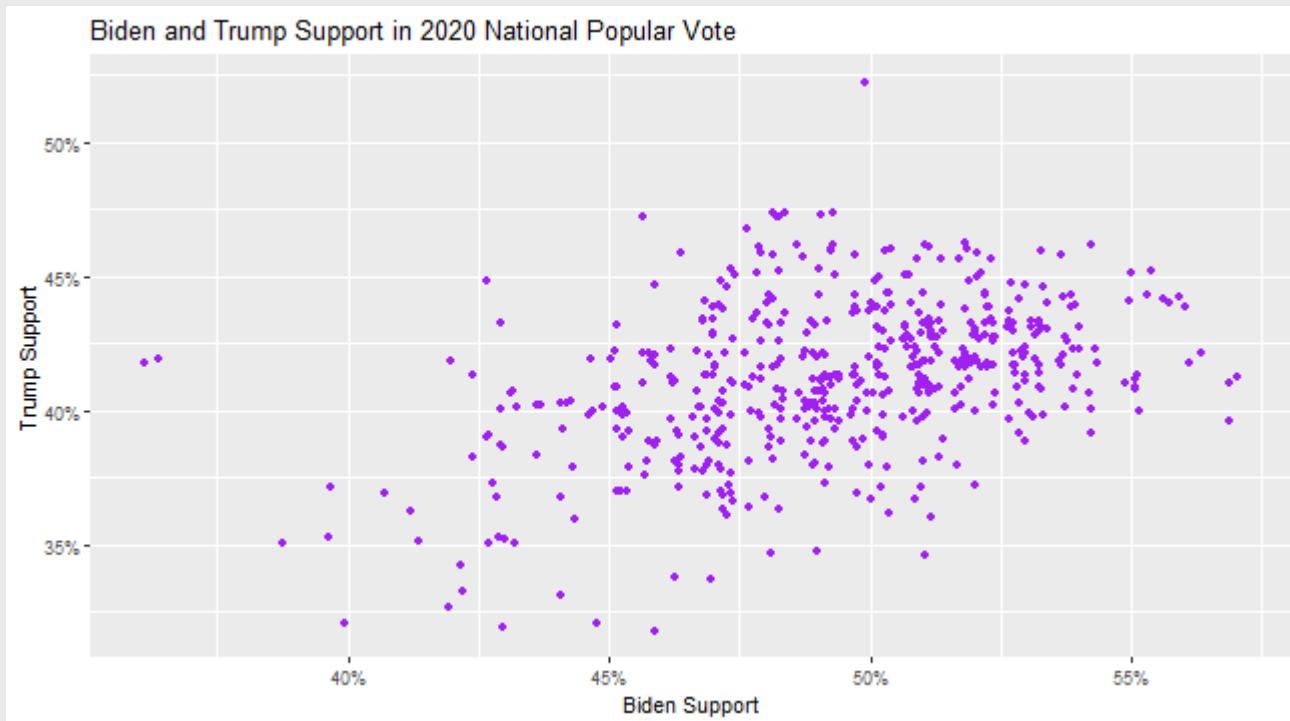
geom_jitter()

- Instead, we could "jitter" the points
 - This adds some random noise to each point to shake them off each other

```
pSupp <- poll %>%
  ggplot(aes(x = Biden, y = Trump)) +
  labs(title="Biden and Trump Support in 2020 National Popular Vote",
       y = "Trump Support",
       x = "Biden Support") +
  geom_jitter(color="purple") +
  scale_y_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  scale_x_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

geom_jitter()

pSupp



size

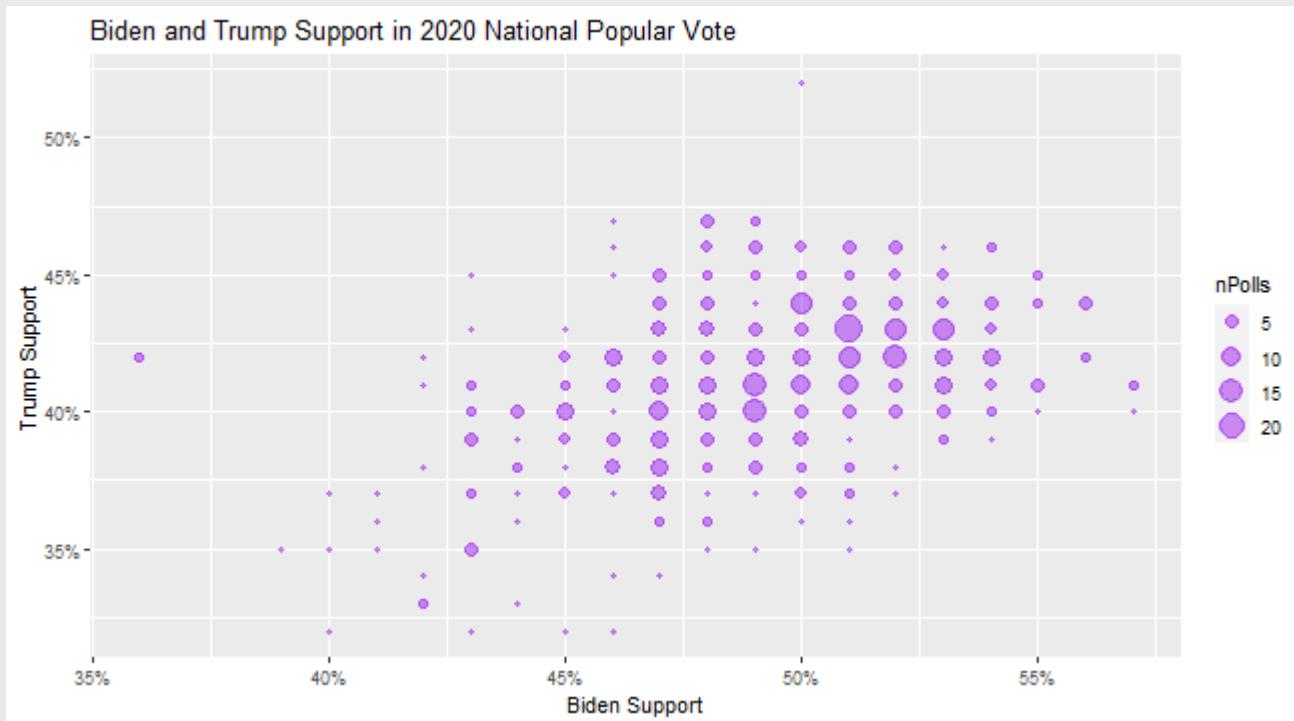
- Finally, we could simply count the number of polls at each x,y coordinate
 - Then size the points by the number of polls

```
pSupp <- poll %>%
  group_by(Biden,Trump) %>%
  summarise(nPolls = n()) %>%
  ggplot(aes(x = Biden, y = Trump,size = nPolls)) +
  labs(title="Biden and Trump Support in 2020 National Popular Vote",
       y = "Trump Support",
       x = "Biden Support") +
  geom_point(color="purple",alpha = .5) +
  scale_y_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1)) +
  scale_x_continuous(breaks=seq(0,1,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

```
## `summarise()` has grouped output by 'Biden'. You can
## override using the `groups` argument.
```

size

pSupp



Theory

- These results indicate that polls which predict greater support for Biden **also** predict greater support for Trump
 - Is this consistent with the theory?
 - Recall that **Biden-biased** polls should underpredict Trump support and overpredict Biden support
 - In the **data**, this would suggest a **negative** relationship
 - But we find a **positive** relationship
- **Inconsistent** with the theory, but raises another puzzle
- Why do polls that underpredict support for Biden also underpredict support for Trump?
 - **Third party bias?** Polls bias against 3rd party candidates
 - **Timing of poll?** Fewer uncertain responses closer to election

Dates and More Dimensions

- To test the [timing of polls theory](#), let's look at how predicted support varies over time

```
poll %>%  
  select(StartDate,EndDate,DaysinField)
```

```
## # A tibble: 528 × 3  
##   StartDate   EndDate   DaysinField  
##   <chr>       <chr>        <dbl>  
## 1 10/31/2020 11/2/2020         3  
## 2 10/31/2020 11/2/2020         3  
## 3 10/29/2020 11/2/2020         5  
## 4 11/1/2020   11/1/2020         1  
## 5 11/1/2020   11/1/2020         1  
## 6 10/30/2020 11/1/2020         3  
## 7 10/31/2020 11/2/2020         3  
## 8 10/30/2020 11/1/2020         3  
## 9 10/29/2020 11/1/2020         4  
## 10 10/29/2020 11/1/2020        4  
## # ... with 518 more rows
```

Dates

- Need to convert `StartDate` and `EndDate` into `date` class

```
election.day <- as.Date("11/3/2020", "%m/%d/%Y")  
election.day
```

```
## [1] "2020-11-03"
```

```
election.day16 <- as.Date("11/8/2016", "%m/%d/%Y")  
election.day16
```

```
## [1] "2016-11-08"
```

```
election.day - election.day16
```

```
## Time difference of 1456 days
```

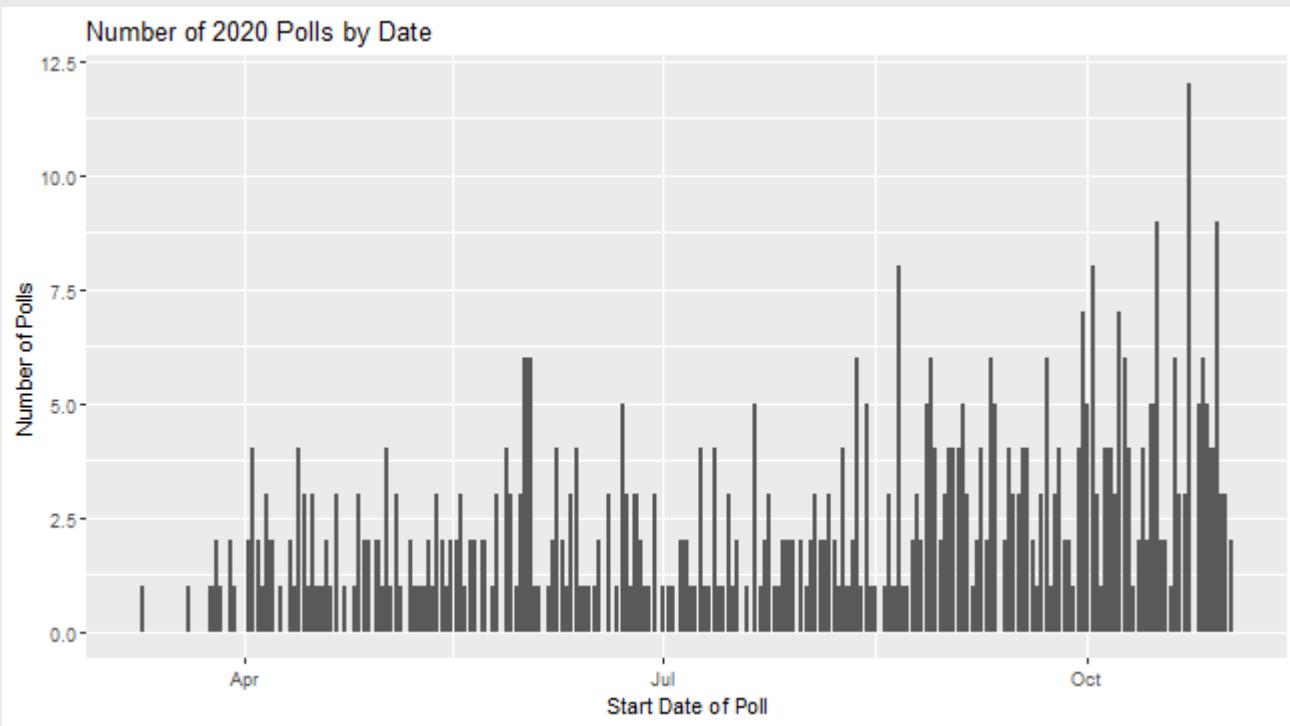
Wrangling

```
toplot <- poll %>%
  mutate(EndDate = as.Date(EndDate, "%m/%d/%Y"),
        StartDate = as.Date(StartDate, "%m/%d/%Y"),
        DaysToED = as.numeric(election.day - EndDate),
        margin = Biden - Trump)
```

- How many polls were conducted over time?

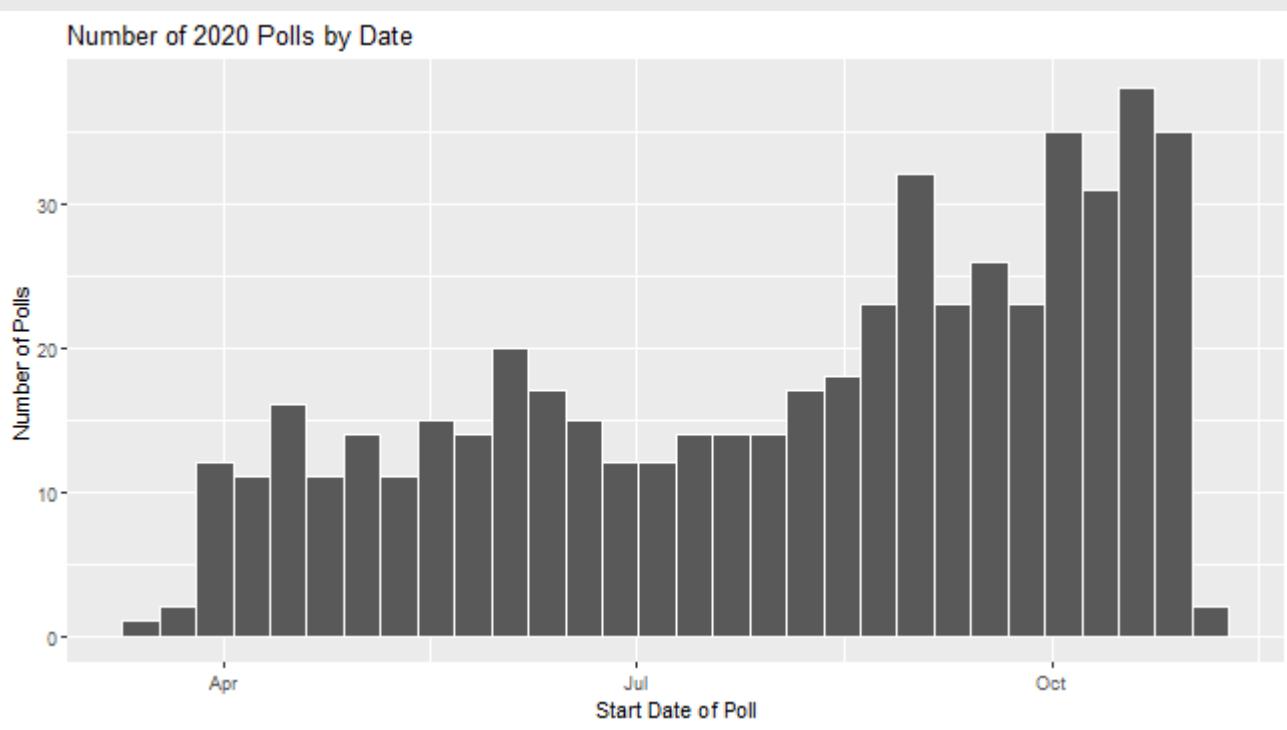
Polls by Date (`geom_bar`)

```
toplot %>%
  ggplot(aes(x = StartDate)) +
  geom_bar() + labs(title = 'Number of 2020 Polls by Date',x = 'Start
Date of Poll',y = 'Number of Polls')
```



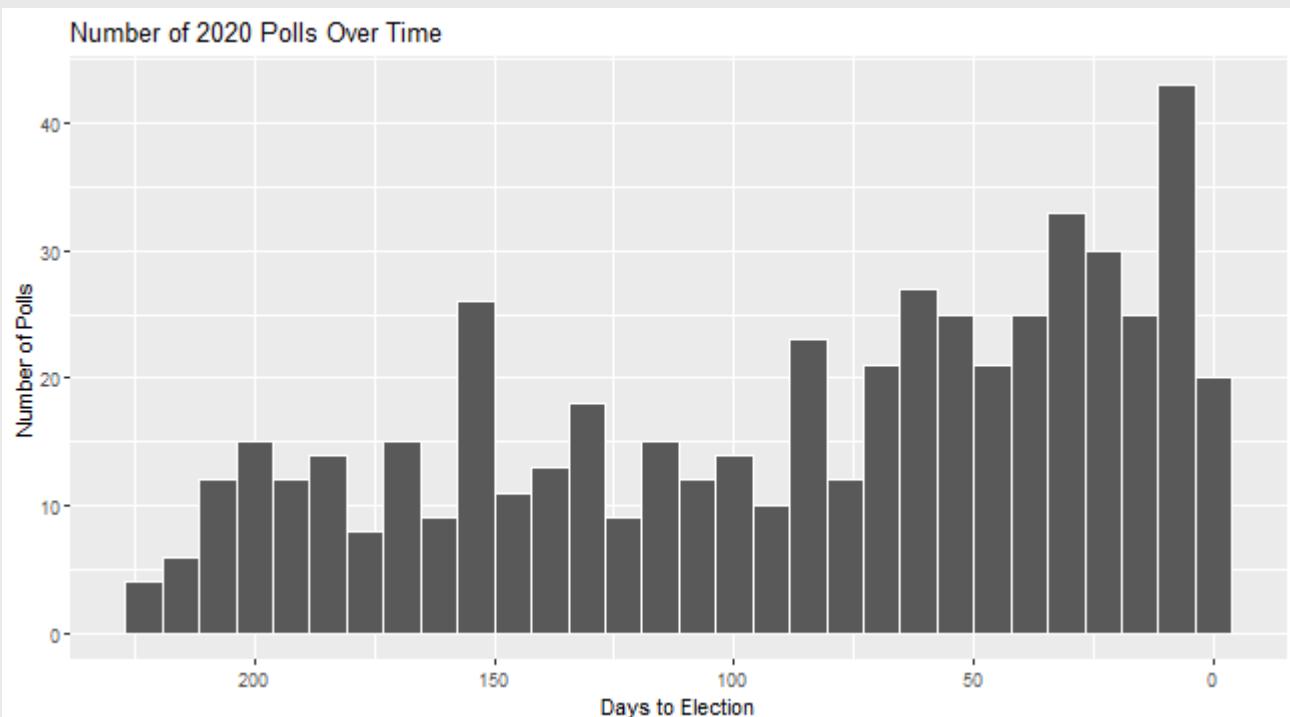
Polls by Date (geom_histogram)

```
toplot %>%
  ggplot(aes(x = StartDate)) +
  geom_histogram(bins = 30,color = 'white') + labs(title = 'Number of
2020 Polls by Date',x = 'Start Date of Poll',y = 'Number of Polls')
```



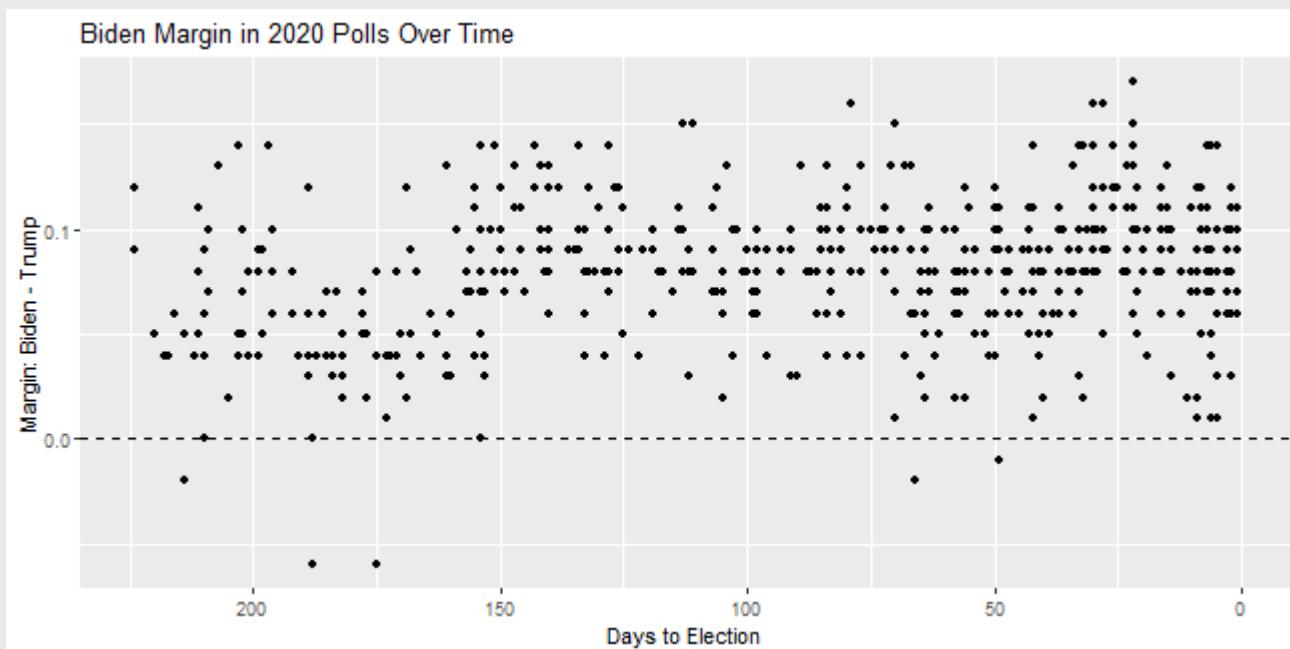
Polls by Time to Election

```
toplot %>%
  ggplot(aes(x = DaysToED)) +
  geom_histogram(bins = 30,color = 'white') + labs(title = 'Number of
2020 Polls Over Time',x = 'Days to Election',y = 'Number of Polls') +
  scale_x_reverse()
```



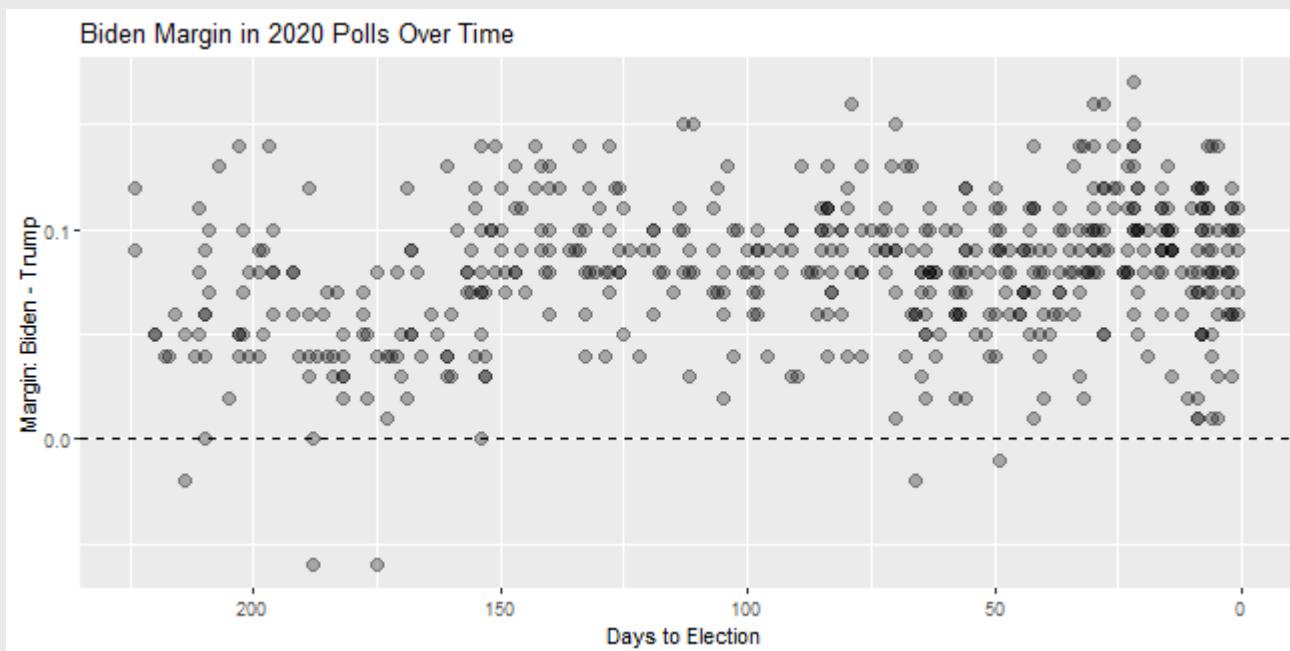
Margin by Time to Election

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_point() +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



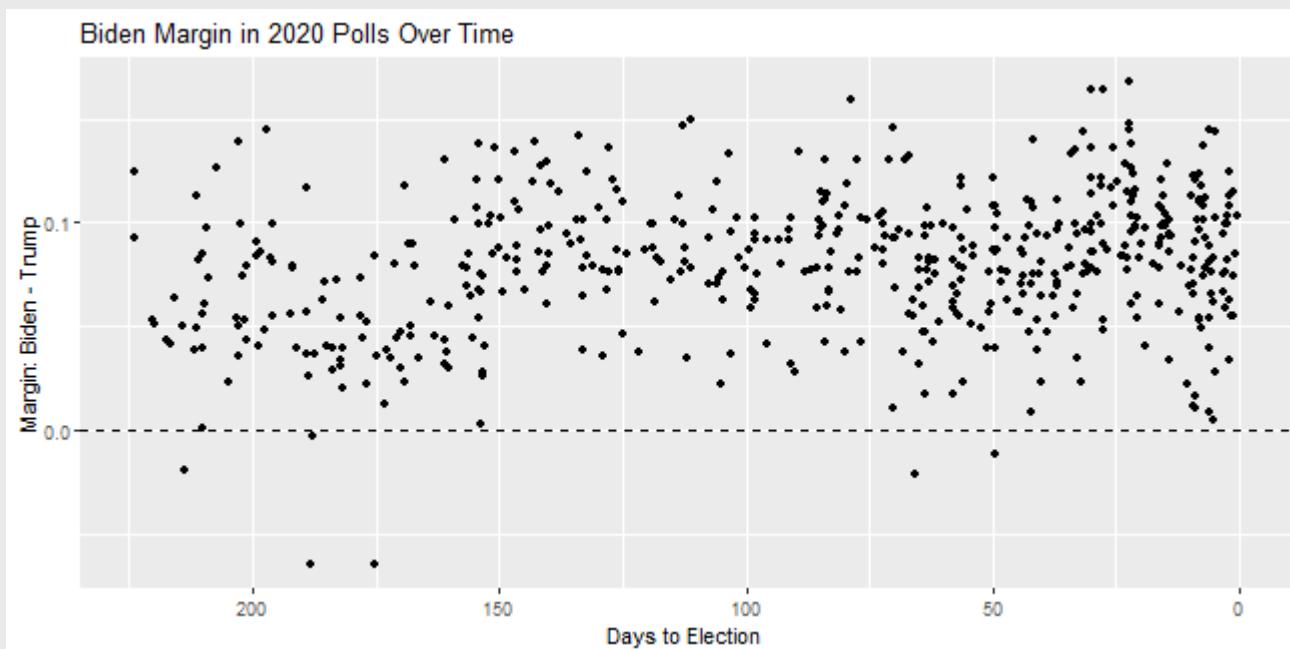
Margin by Time to Election

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_point(alpha = .3,size = 3) +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



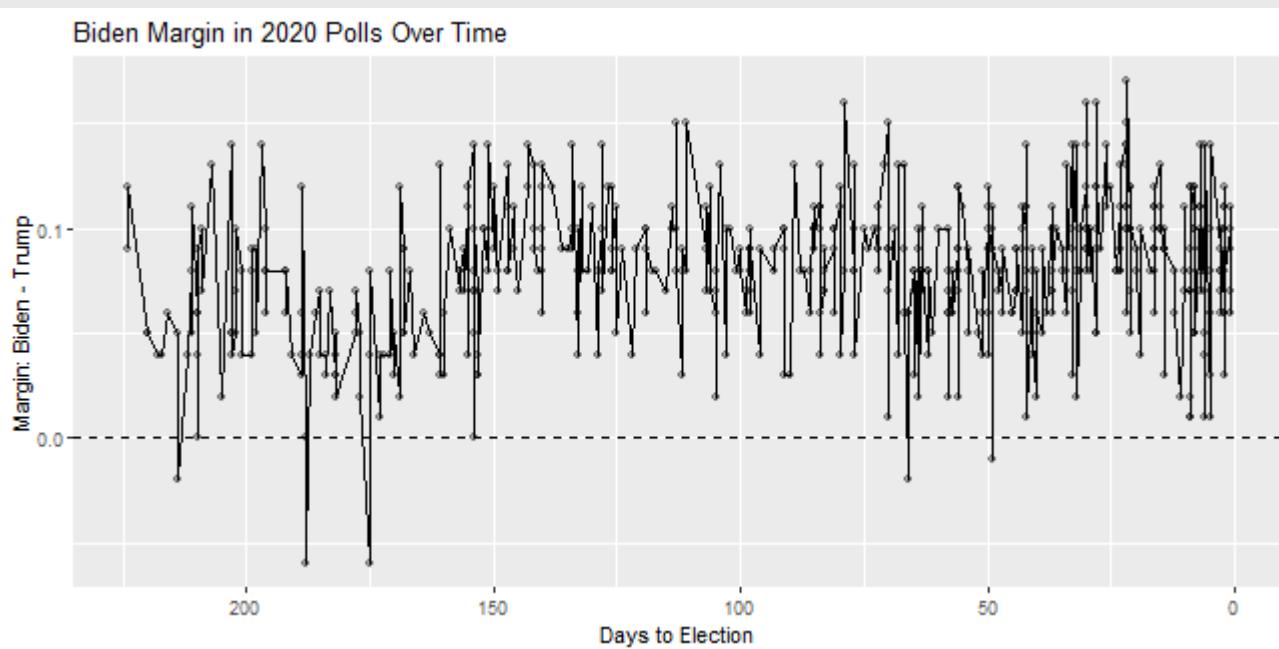
Margin by Time to Election

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_jitter(height = .005) +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



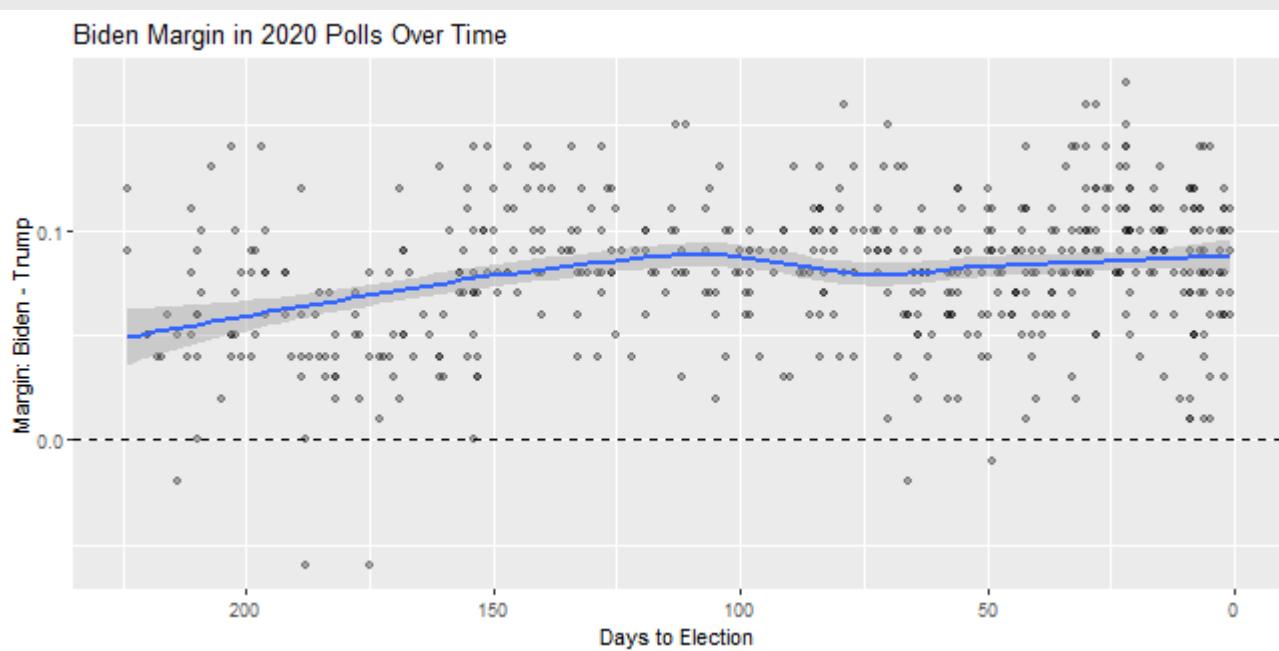
Highlight Trends

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_point(alpha = .3) +
  geom_line() +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



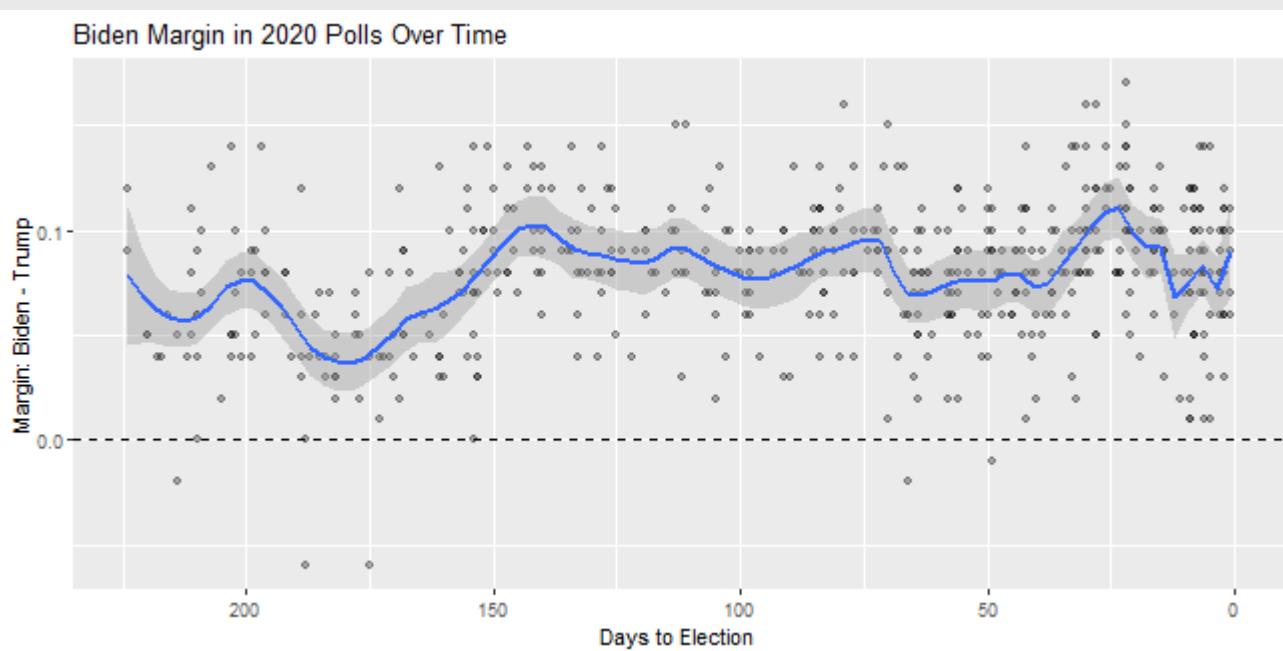
Highlight Trends

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_point(alpha = .3) +
  geom_smooth() +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



Highlight Trends

```
toplot %>%
  ggplot(aes(x = DaysToED,y = margin)) +
  geom_point(alpha = .3) +
  geom_smooth(span = .1) +
  labs(title = 'Biden Margin in 2020 Polls Over Time',
       x = 'Days to Election',y = 'Margin: Biden - Trump') +
  scale_x_reverse() + geom_hline(yintercept = 0,linetype = 'dashed')
```



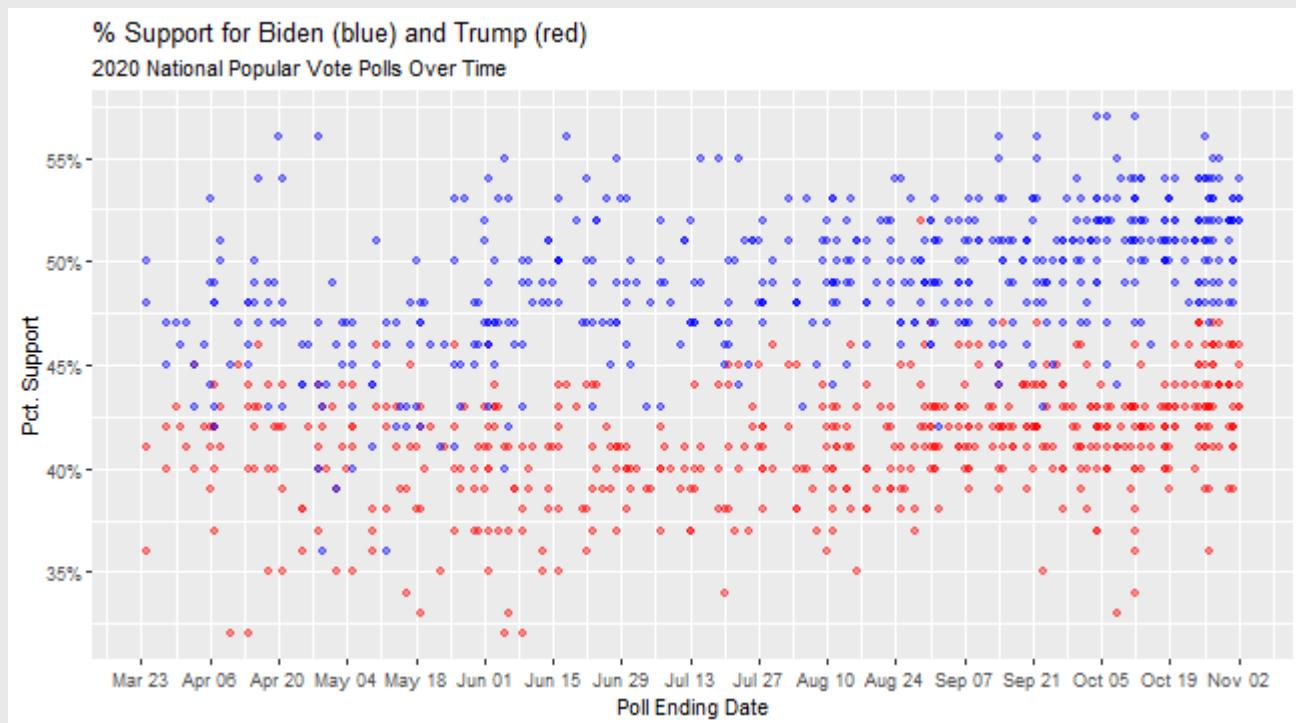
Multiple Variables over time

- We can look at multiple continuous variables at the same time
 - Use `color` or `fill` to distinguish

```
BidenTrumpplot <- toplot %>%
  ggplot() +
  geom_point(aes(x = EndDate, y = Trump),
             color = "red", alpha=.4) +
  geom_point(aes(x = EndDate, y = Biden),
             color = "blue", alpha=.4) +
  labs(title="% Support for Biden (blue) and Trump (red)",
       subtitle = "2020 National Popular Vote Polls Over Time",
       y = "Pct. Support",
       x = "Poll Ending Date") +
  scale_x_date(date_breaks = "2 week", date_labels = "%b %d") +
  scale_y_continuous(breaks=seq(.3,.7,by=.05),
                     labels= scales::percent_format(accuracy = 1))
```

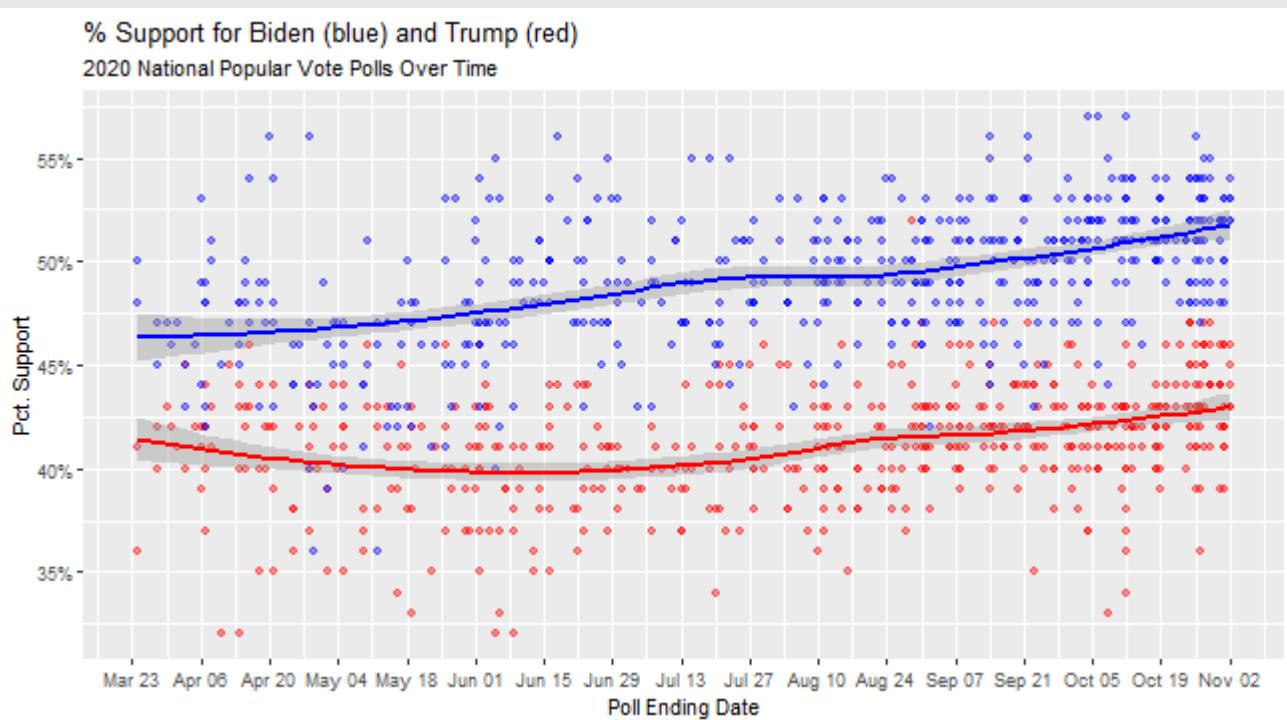
Multiple Variables over time

BidenTrumpplot



Highlighting Trends

```
BidenTrumpplot +  
  geom_smooth(aes(x = EndDate, y = Trump),  
              color = "red", se=T) +  
  geom_smooth(aes(x = EndDate, y = Biden),  
              color = "blue", se=T)
```



Typing back to theory

- Evidence of **both** Biden and Trump support increasing toward election
- Does this mean third party support waned?
- Does this mean that undecided voters made a decision?

2. Overtime Errors

- Wrangling to convert character dates to `date` class
- Plotting prediction errors over time

```
toplot <- poll %>%
  mutate(EndDate = as.Date(EndDate, "%m/%d/%Y"),
        StartDate = as.Date(StartDate, "%m/%d/%Y"),
        DaysToED = as.numeric(as.Date('11/3/2020',format =
'%m/%d/%Y') - EndDate),
        margin = Biden - Trump,
        errDem = Biden - DemCertVote / 100,
        errRep = Trump - RepCertVote / 100)
```

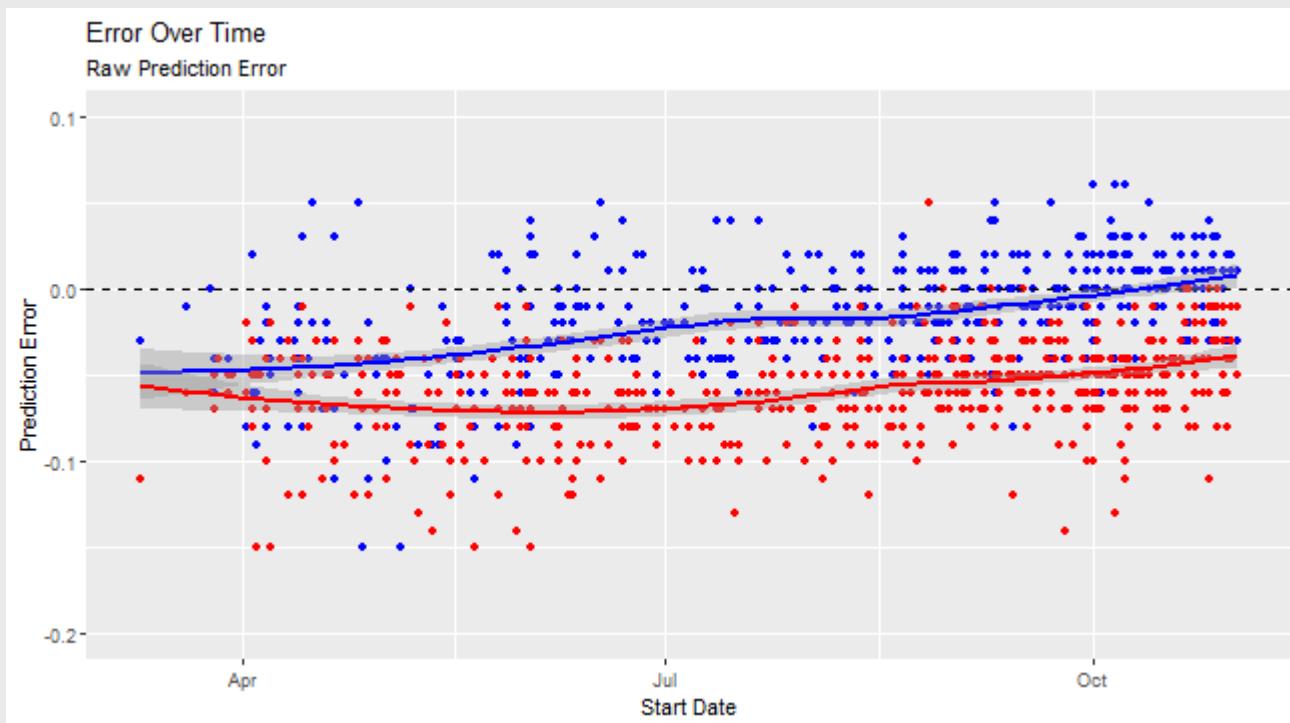
Overtime Errors

- Wrangling to convert character dates to date class
- Plotting prediction errors over time

```
pErrOverTime_Raw <- toplot %>%  
  ggplot(aes(x = StartDate)) +  
  geom_point(aes(y = errDem),color = 'blue') +  
  geom_point(aes(y = errRep),color = 'red') +  
  geom_hline(yintercept = 0,linetype = 'dashed') +  
  geom_smooth(aes(y = errDem),color = 'blue') +  
  geom_smooth(aes(y = errRep),color = 'red') +  
  labs(title = 'Error Over Time',subtitle = 'Raw Prediction Error',x  
= 'Start Date',y = 'Prediction Error') +  
  ylim(c(-.2,.1))
```

Overtime Errors

pErrOverTime_Raw



Poll Quality

- Polls got better
 - ...but more for Biden than for Trump
- Poll quality:
 - Sample size
 - "Margin of Error" ([MoE](#))

Weighting by Sample Size

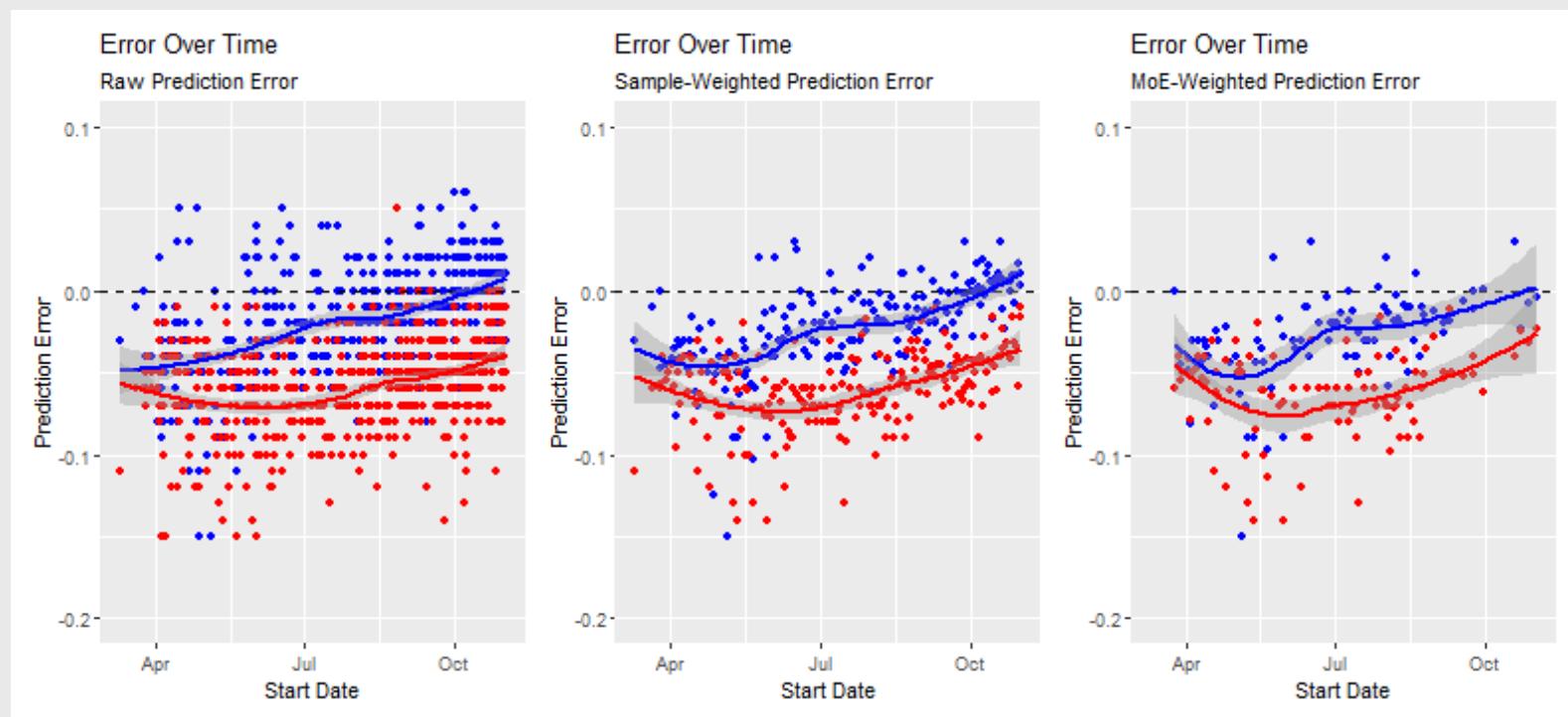
```
pErrOverTime_SampleSize <- toplot %>%
  group_by(StartDate) %>%
  summarise(errDem = weighted.mean(errDem,w = SampleSize),
            errRep = weighted.mean(errRep,w = SampleSize)) %>%
  ggplot(aes(x = StartDate)) +
  geom_point(aes(y = errDem),color = 'blue') +
  geom_point(aes(y = errRep),color = 'red') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  geom_smooth(aes(y = errDem),color = 'blue') +
  geom_smooth(aes(y = errRep),color = 'red') +
  labs(title = 'Error Over Time',subtitle = 'Sample-Weighted
Prediction Error',x = 'Start Date',y = 'Prediction Error') +
  ylim(c(-.2,.1))
```

Weighting by Inverse of MoE

```
pErrOverTime_MoE <- topplot %>%
  group_by(StartDate) %>%
  summarise(errDem = weighted.mean(errDem,w = 1/MoE),
            errRep = weighted.mean(errRep,w = 1/MoE)) %>%
  ggplot(aes(x = StartDate)) +
  geom_point(aes(y = errDem),color = 'blue') +
  geom_point(aes(y = errRep),color = 'red') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  geom_smooth(aes(y = errDem),color = 'blue') +
  geom_smooth(aes(y = errRep),color = 'red') +
  labs(title = 'Error Over Time',subtitle = 'MoE-Weighted Prediction
Error',x = 'Start Date',y = 'Prediction Error') +
  ylim(c(-.2,.1))
```

Comparing

```
require(patchwork)  
pErrOverTime_Raw + pErrOverTime_SampleSize + pErrOverTime_MoE
```



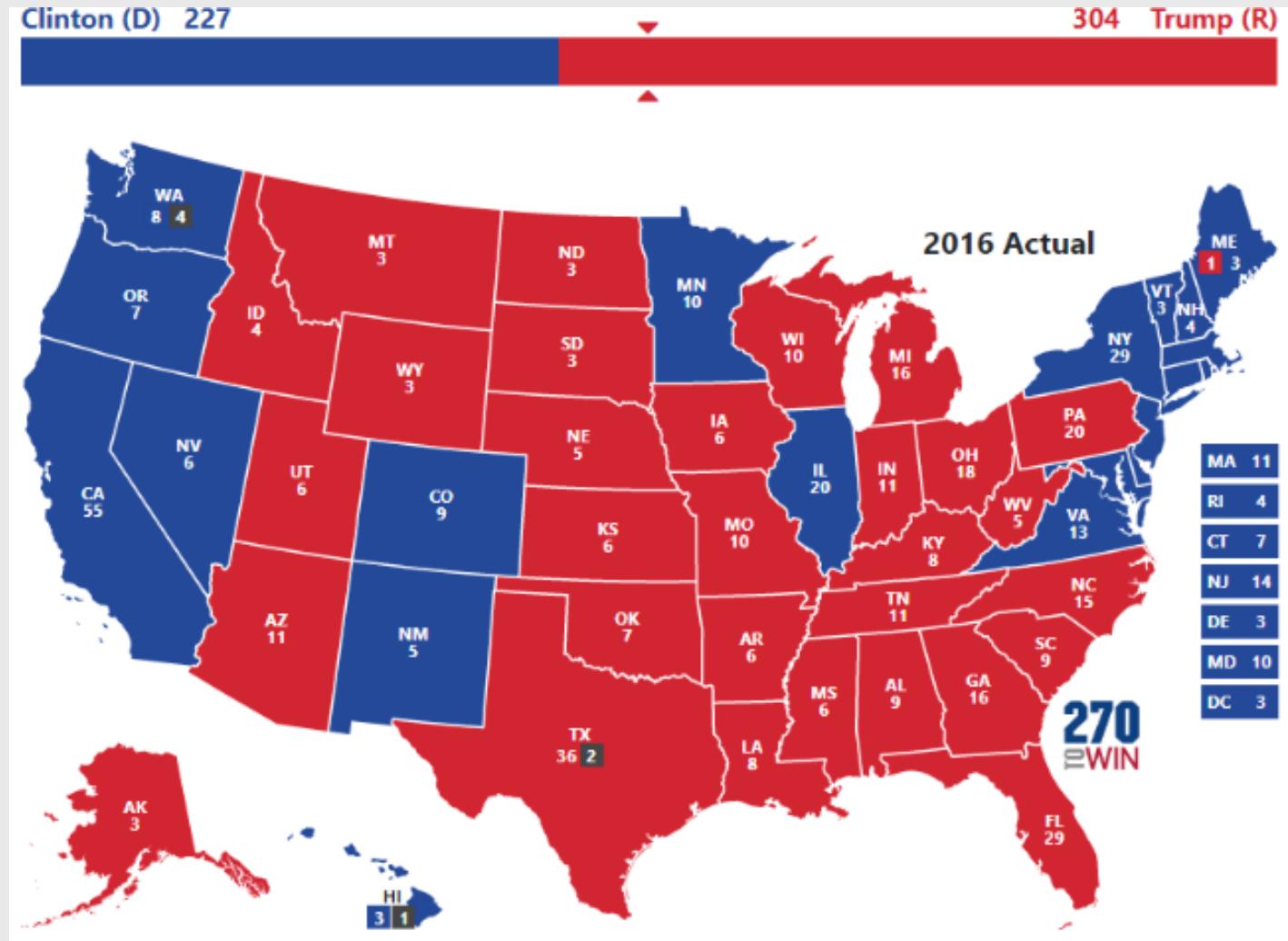
3. The Electoral College

- U.S. presidents determined by the **Electoral College** (EC)
 - Each state allocated votes equal to their "congressional delegation"
 - One vote per Senator + one vote per **Member of the House** (MotH)
 - 100 Senators + 435 MotHs + 3 votes for DC = 538 total votes
 - To win the presidency, you need at least 270 EC votes
- What does this mean for representation?
 - Population divided equally* across 435 MotHs
 - In 2020, each MotH represented 761,179 people
 - Geographic distribution of this population is **non-uniform**

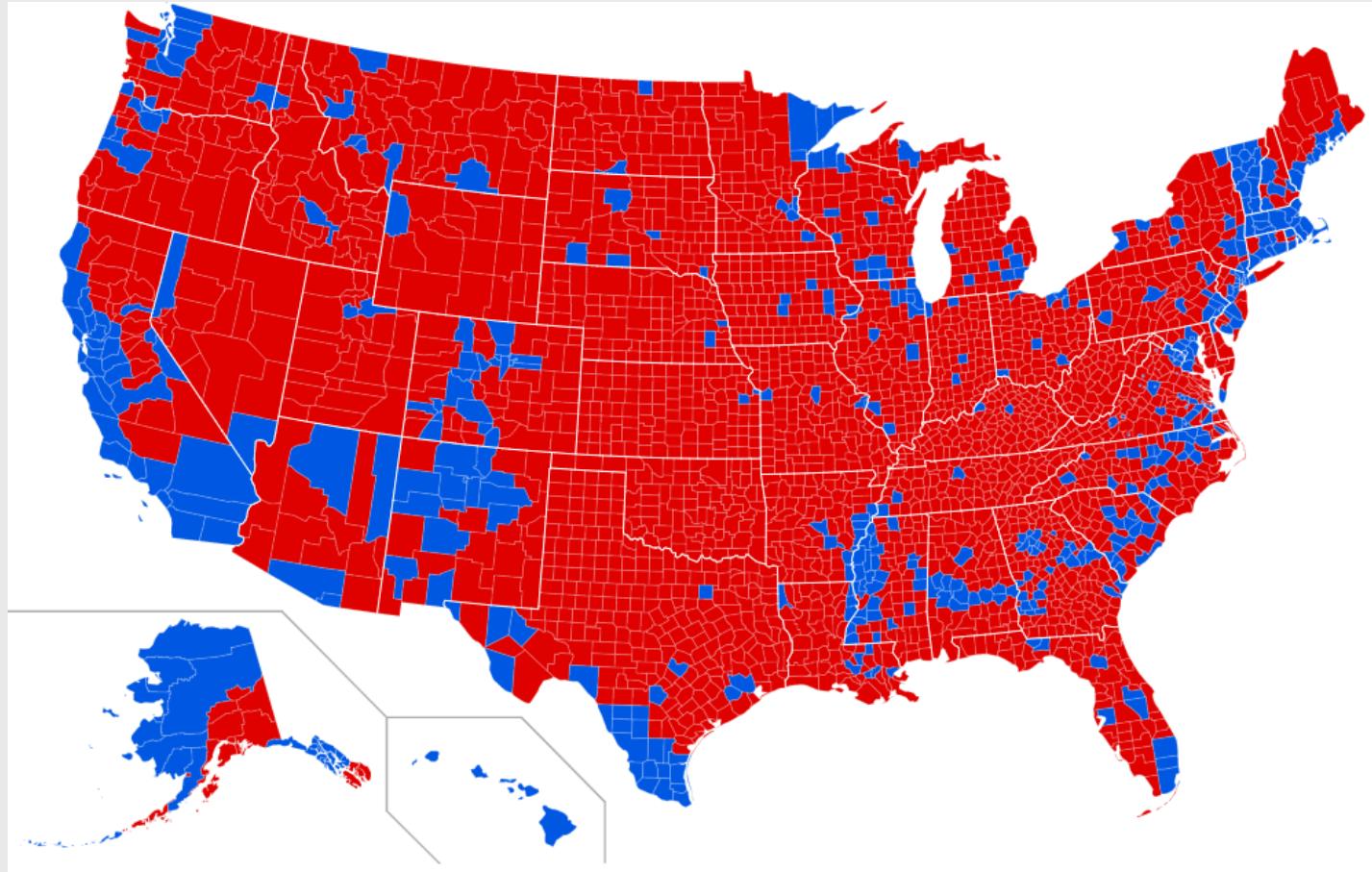
Assigning Districts

- Every 10 years, U.S. takes a census of its population
- Based on these results, congressional districts are reapportioned
- In 2010, Florida's pop. was ~19m & average pop. per MoTH was ~710k
 - How many MoTHs? $\text{ceiling}(19000000 / 710000) = 27$
 - Every state has 2 senators, so FL had $27 + 2 = 29$ EC votes
 - In 2020, FL pop. was 21m → each resident had 0.00000137 of a vote
- What about Vermont?
 - $3 \text{ EC votes} / 624,340 \text{ residents} = 0.00000481$
- Vermonters have roughly 3.5 times the influence of Floridians!

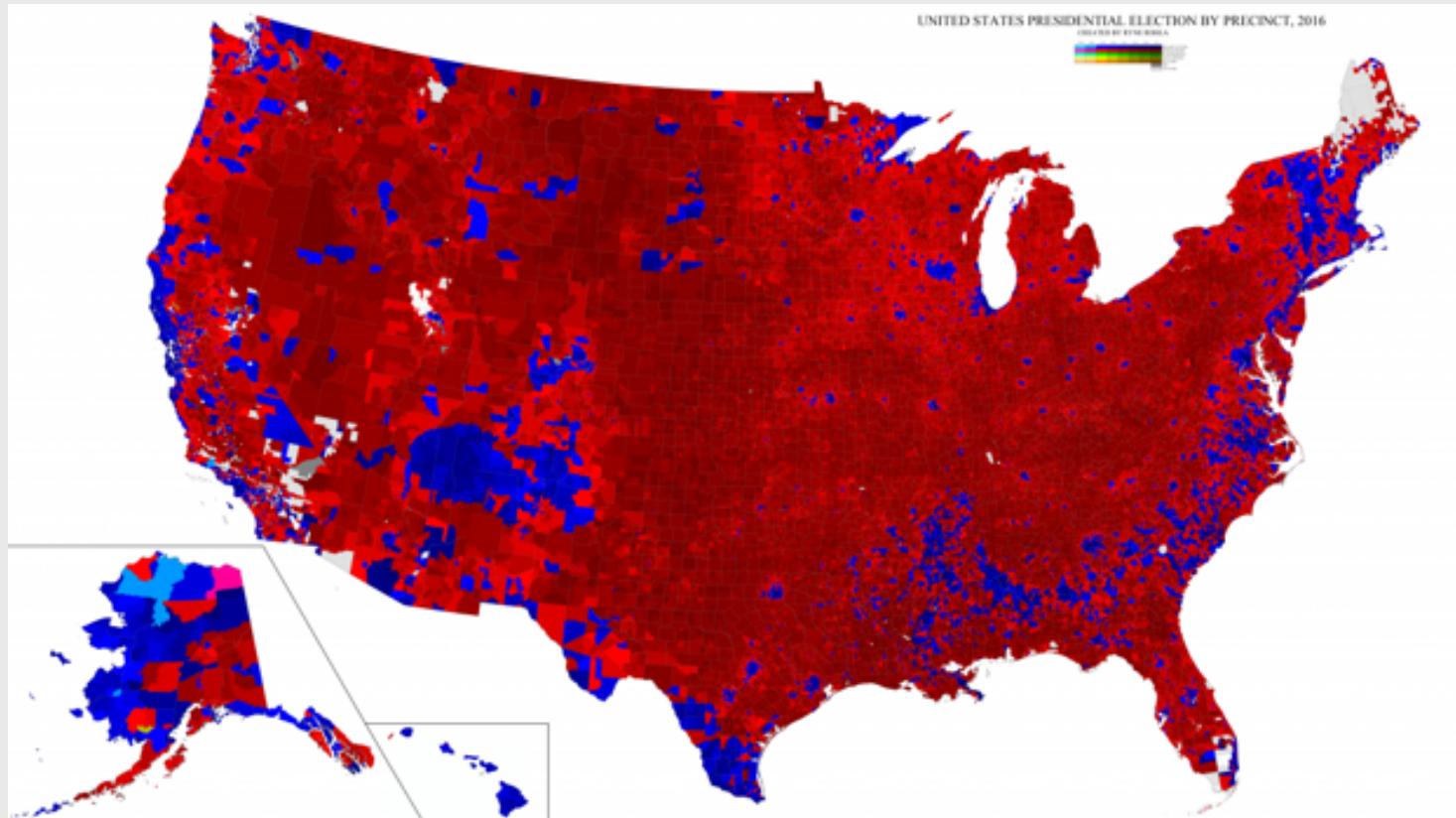
Data Viz Misleads!



Data Viz Misleads!



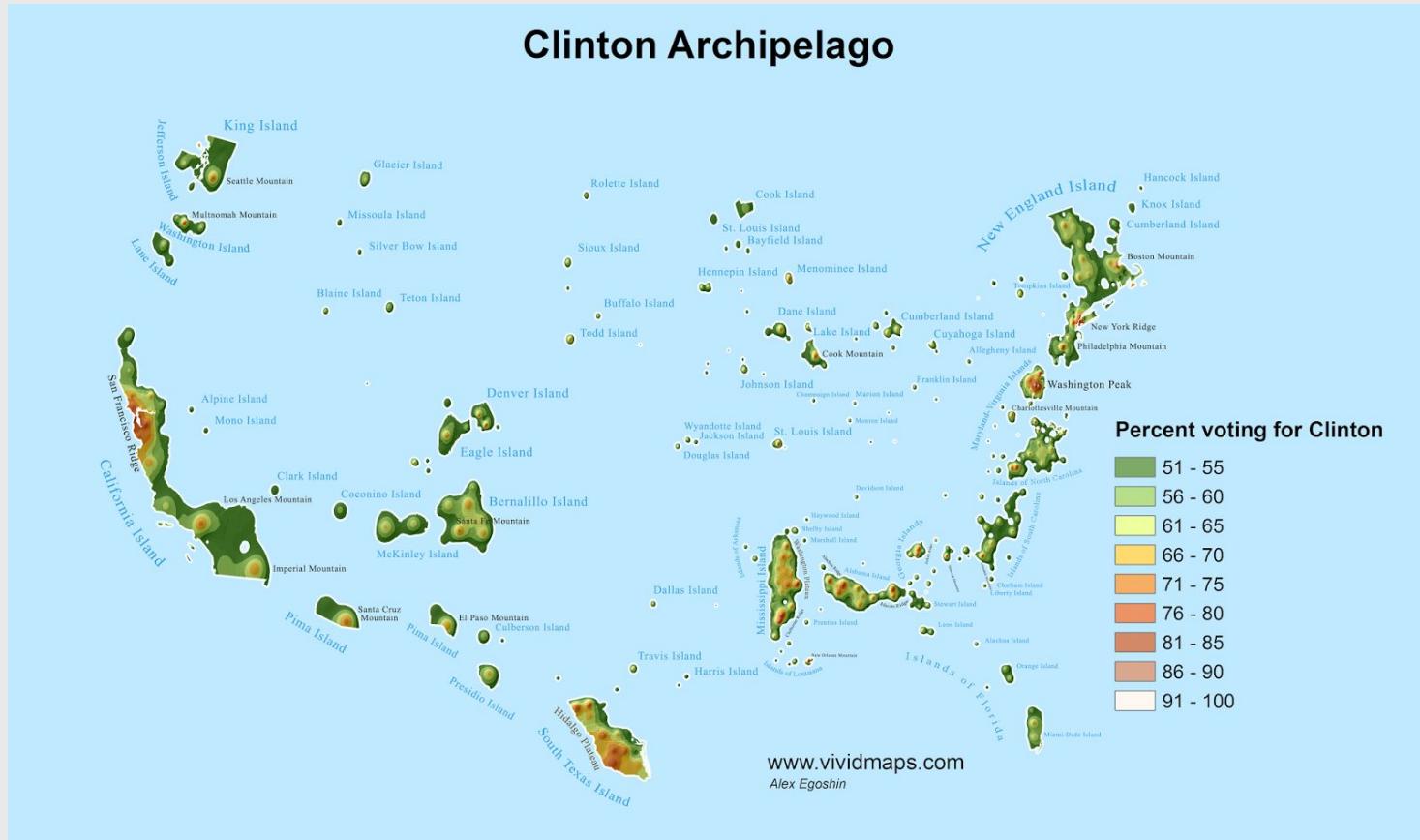
Data Viz Misleads!



Data Viz Misleads!

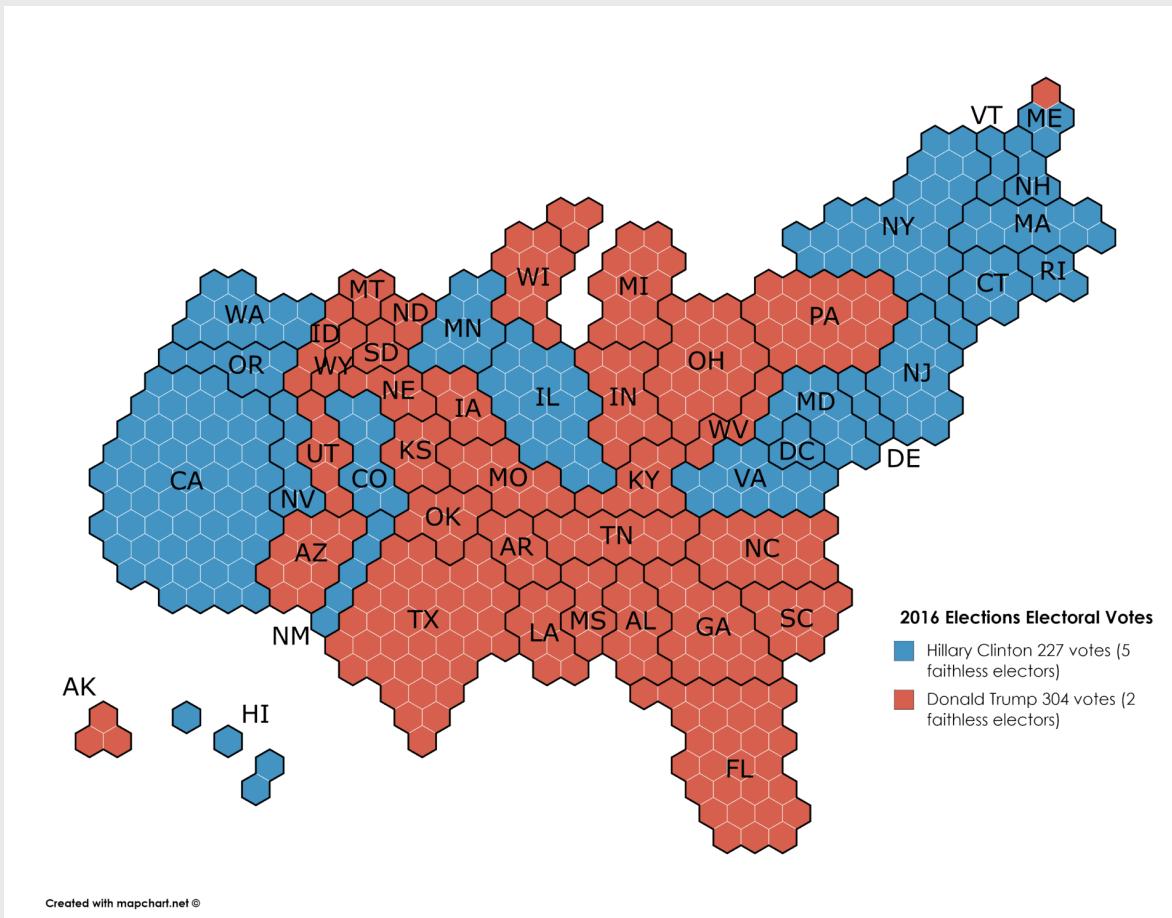


Data Viz Misleads!



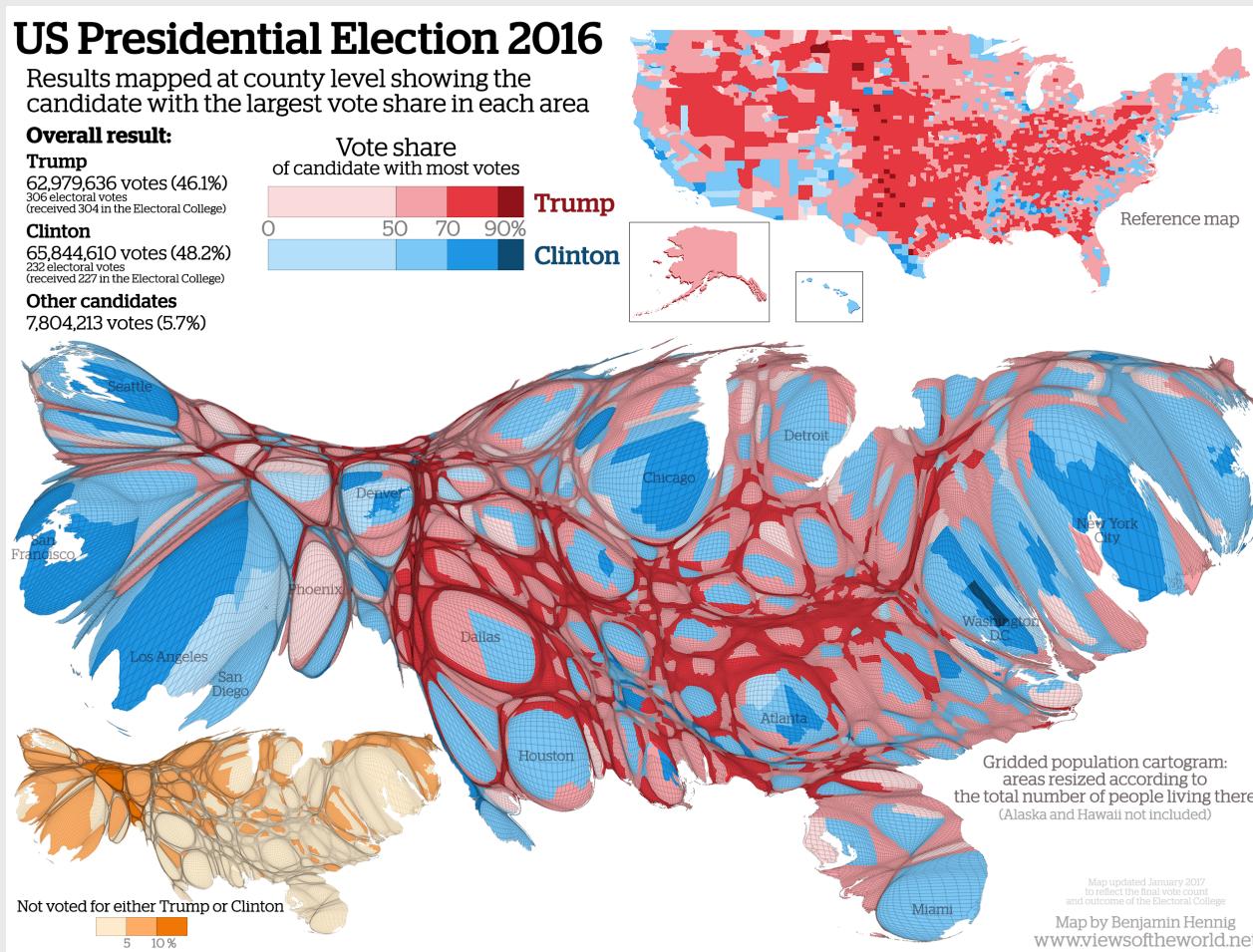
Correcting Maps

- Fundamental challenge due to distribution of population across land



Correcting Maps

- Fundamental challenge due to distribution of population across land



Introducing Data

- We will attempt to use state polls to predict 2020 winner
- National polls are great at predicting popular vote share...
- ...but presidents aren't elected based on popular vote
- With a few exceptions, states have a "winner take all" system
 - Winner of state's popular vote gets **all** of the state's EC votes
- Thus we need good state-level polls
- Load `Pres2020_StatePolls.Rds` to `statePoll` object

```
require(tidyverse)
statePoll <- readRDS('..../data/Pres2020_StatePolls.Rds')
```

Look at the data

```
glimpse(statePoll)
```

```
## Rows: 1,545
## Columns: 19
## $ StartDate      <date> 2020-03-21, 2020-03-24, 2020-03-24...
## $ EndDate        <date> 2020-03-30, 2020-04-03, 2020-03-29...
## $ DaysinField    <dbl> 10, 11, 6, 2, 3, 5, 2, 2, 7, 3, 3, ...
## $ MoE            <dbl> 2.8, 3.0, 4.2, NA, 4.0, 1.7, 3.0, 3...
## $ Mode           <chr> "Phone/Online", "Phone/Online", "Li...
## $ SampleSize     <dbl> 1331, 1000, 813, 962, 602, 3244, 10...
## $ Biden          <dbl> 41, 47, 48, 67, 46, 46, 46, 48, 52, ...
## $ Trump          <dbl> 46, 34, 45, 29, 46, 40, 48, 45, 39, ...
## $ Winner         <chr> "Rep", "Dem", "Dem", "Dem", "Dem", ...
## $ poll.predicted <dbl> 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, ...
## $ Funded         <chr> "UtahPolicy.com & KUTV 2News", "Sac...
## $ Conducted      <chr> "Y2 Analytics", "GreatBlue Research...
## $ margin          <dbl> -5, 13, 3, 38, 0, 6, -2, 3, 13, -7, ...
## $ DaysToED        <drtn> 218 days, 214 days, 219 days, 219 ...
## $ StateName       <chr> "Utah", "Connecticut", "Wisconsin", ...
## $ EV              <int> 6, 7, 10, 55, 16, 29, 16, 16, 12, 1...
## $ State           <chr> "UT", "CT", "WI", "CA", "MI", "FL", ...
## $ BidenCertVote   <dbl> 38, 59, 49, 64, 51, 48, 50, 51, 58, ...
```

Scientific Goal: Predict Winner

- Need to calculate predicted probability of Biden victory
 1. Fraction of polls with Biden in the lead
 2. Fraction of respondents supporting Biden
 3. Fraction of two way respondents supporting Biden
- Wrangling time!

```
statePoll <- statePoll %>%
  mutate(Biden2W = Biden / (Biden+Trump),
        Biden = Biden / 100,
        Trump = Trump/100)

stateProbs <- statePoll %>%
  group_by(StateName, State) %>%
  summarise(BidenProb1 = mean(Biden > Trump),
            BidenProb2 = mean(Biden),
            BidenProb3 = mean(Biden2W),
            EV = mean(EV), .groups = 'drop') # This is just the value
```

of EV

Looking at data

```
stateProbs
```

```
## # A tibble: 50 × 6
##   StateName    State BidenProb1 BidenProb2 BidenProb3     EV
##   <chr>        <chr>      <dbl>      <dbl>      <dbl> <dbl>
## 1 Alabama      AL         0          0.389     0.407     9
## 2 Alaska       AK         0          0.442     0.466     3
## 3 Arizona      AZ        0.840     0.484     0.519    11
## 4 Arkansas     AR         0          0.381     0.395     6
## 5 California   CA         1          0.618     0.661    55
## 6 Colorado     CO         1          0.534     0.571     9
## 7 Connecticut  CT         1          0.584     0.631     7
## 8 Delaware     DE         1          0.603     0.627     3
## 9 Florida      FL        0.798     0.486     0.517    29
## 10 Georgia     GA        0.548     0.474     0.504    16
## # ... with 40 more rows
```

Visualizing Data

- R has fancy packages that allow for interactive plots ([plotly](#))

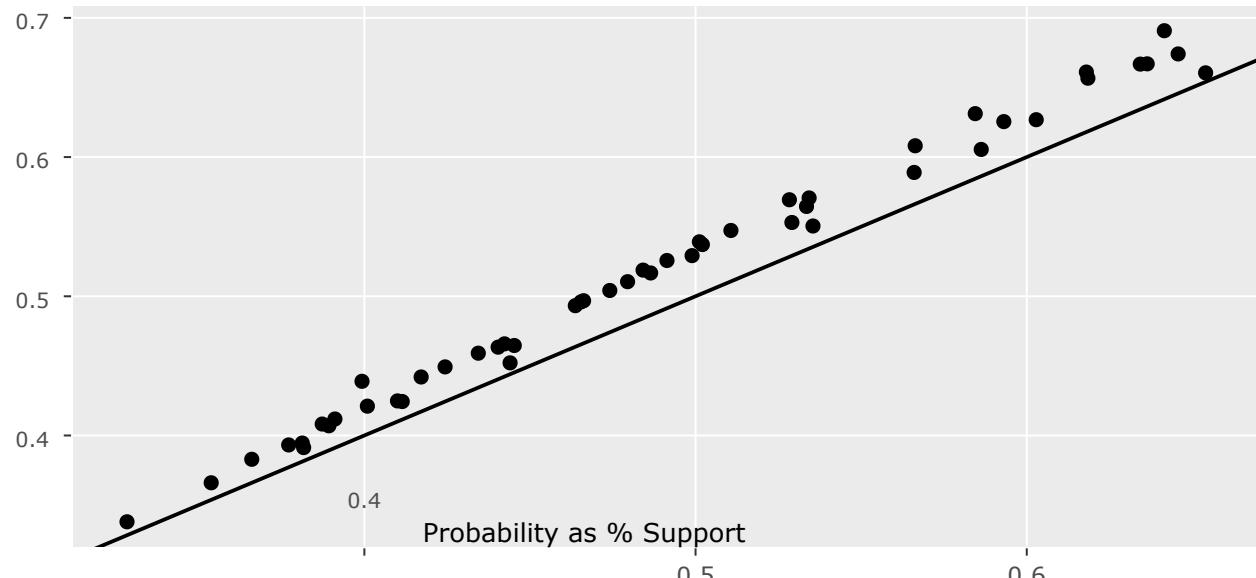
```
# install.packages('plotly')
require(plotly)

gg <- stateProbs %>%
  ggplot(aes(x=BidenProb2, y=BidenProb3, text=paste(StateName))) +
  geom_point() +
  geom_abline(intercept=0, slope=1) +
  labs(x= "Probability as % Support",
       y = "Probability as Two-Party % Support",
       title = "Comparing Probability of Winning Measures")
```

Visualizing Data

```
ggplotly(gg, tooltip = 'text')
```

Comparing Probability of Winning Measures



Visualizing Data

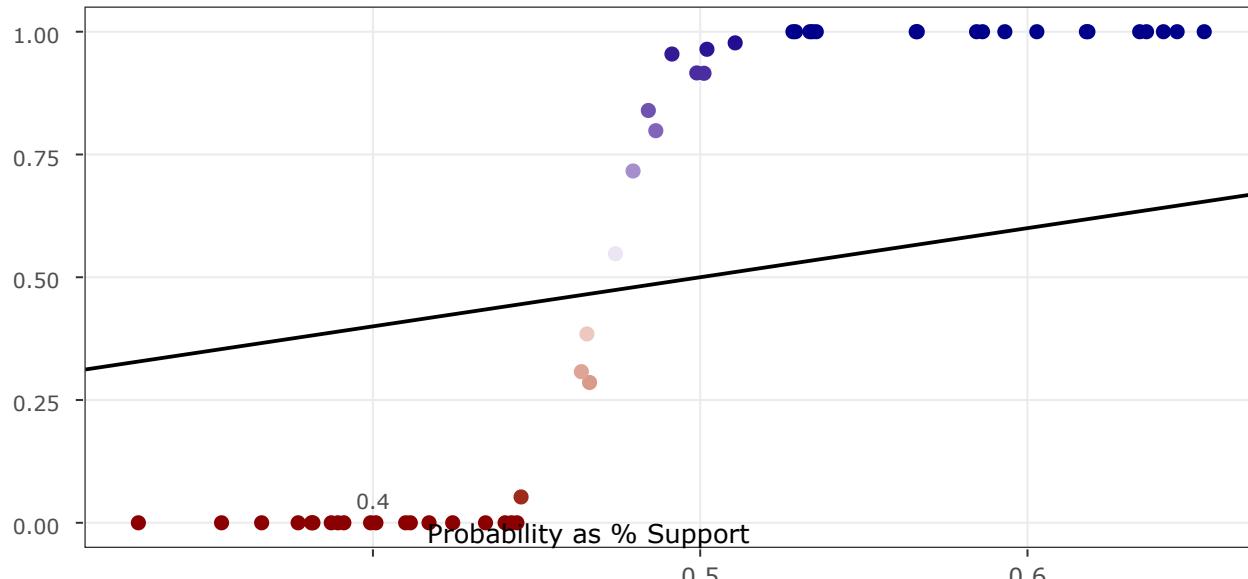
- What about the % of polls that predict a Biden victory?
 - I.e., we know Trump won't win California or Biden won't win Tennessee

```
gg <- stateProbs %>%
  ggplot(aes(x=BidenProb2, y=BidenProb1, text=paste(StateName), color =
BidenProb1)) +
  geom_point() +
  geom_abline(intercept=0, slope=1) +
  labs(x= "Probability as % Support",
       y = "Probability as % Polls Winning",
       title = "Comparing Probability of Winning Measures") +
  scale_color_gradient2(guide = 'none', low = "darkred", mid =
"white", high = "darkblue", midpoint = .5) +
  theme_bw()
```

Visualizing Data

```
ggplotly(gg, tooltip = 'text')
```

Comparing Probability of Winning Measures



Visualizing Data

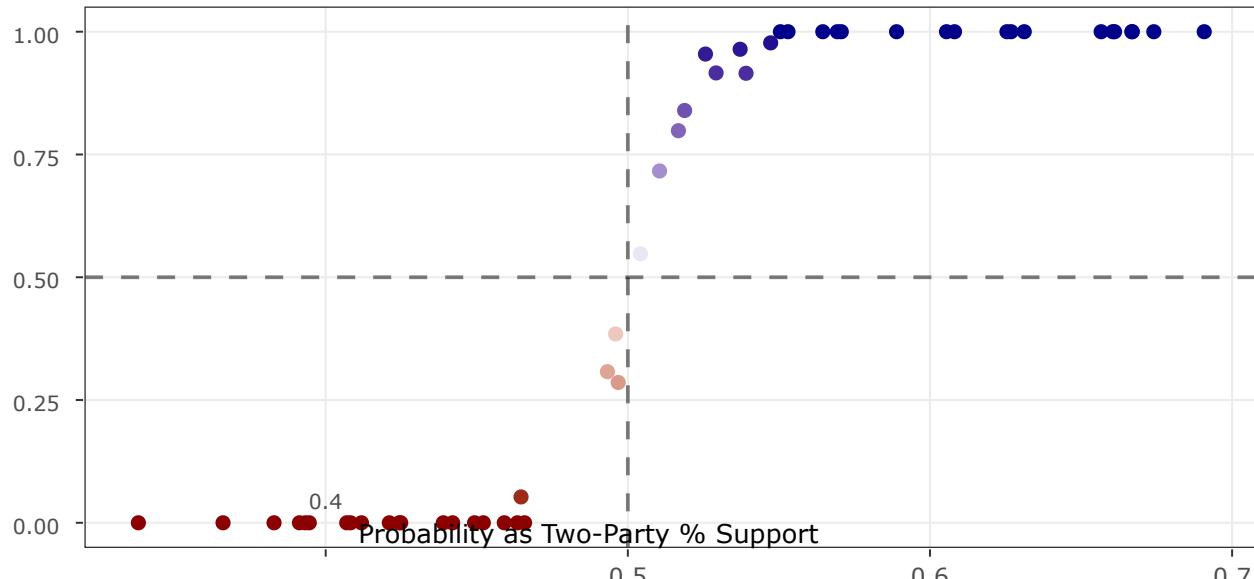
- Comparing to two-way

```
gg <- stateProbs %>%
  ggplot(aes(x=BidenProb3, y=BidenProb1, text=paste(StateName), color =
BidenProb1)) +
  geom_point() +
  labs(x= "Probability as Two-Party % Support",
       y = "Probability as % Polls Winning",
       title = "Comparing Probability of Winning Measures") +
  scale_color_gradient2(guide = 'none', low = "darkred", mid =
"white", high = "darkblue", midpoint = .5) +
  theme_bw() +
  geom_hline(yintercept = .5, linetype = 'dashed', alpha = .5) +
  geom_vline(xintercept = .5, linetype = 'dashed', alpha = .5)
```

Visualizing Data

```
ggplotly(gg, tooltip = 'text')
```

Comparing Probability of Winning Measures



Translate to EC Votes

- Start with a single state to get the idea
- **Expected Value** (EV...could also be Expected Votes): Probability * Value

```
stateProbs %>%
  filter(State == 'PA') %>%
  mutate(BEV1 = BidenProb1 * EV,
        BEV2 = BidenProb2 * EV,
        BEV3 = BidenProb3 * EV) %>%
  select(EV,matches('^B.*\\d'))
```

```
## # A tibble: 1 × 7
##       EV BidenProb1 BidenProb2 BidenProb3   BEV1   BEV2   BEV3
##   <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl> <dbl>
## 1     20     0.916     0.499     0.529  18.3   9.98  10.6
```

- Probability matters! Biden could get between 18.3 and 9.98 votes in PA!

Translate to EC Votes

- Do this for every state

```
stateProbs %>%
  mutate(BidenEV1 = BidenProb1 * EV,
        BidenEV2 = BidenProb2 * EV,
        BidenEV3 = BidenProb3 * EV,
        TrumpEV1 = EV - BidenEV1,
        TrumpEV2 = EV - BidenEV2,
        TrumpEV3 = EV - BidenEV3) %>%
  summarise_at(vars(matches('EV\\d')), sum) # Fancy shortcut!
```

```
## # A tibble: 1 × 6
##   BidenEV1 BidenEV2 BidenEV3 TrumpEV1 TrumpEV2 TrumpEV3
##     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1     345.     272.     289.     190.     263.     246.
```

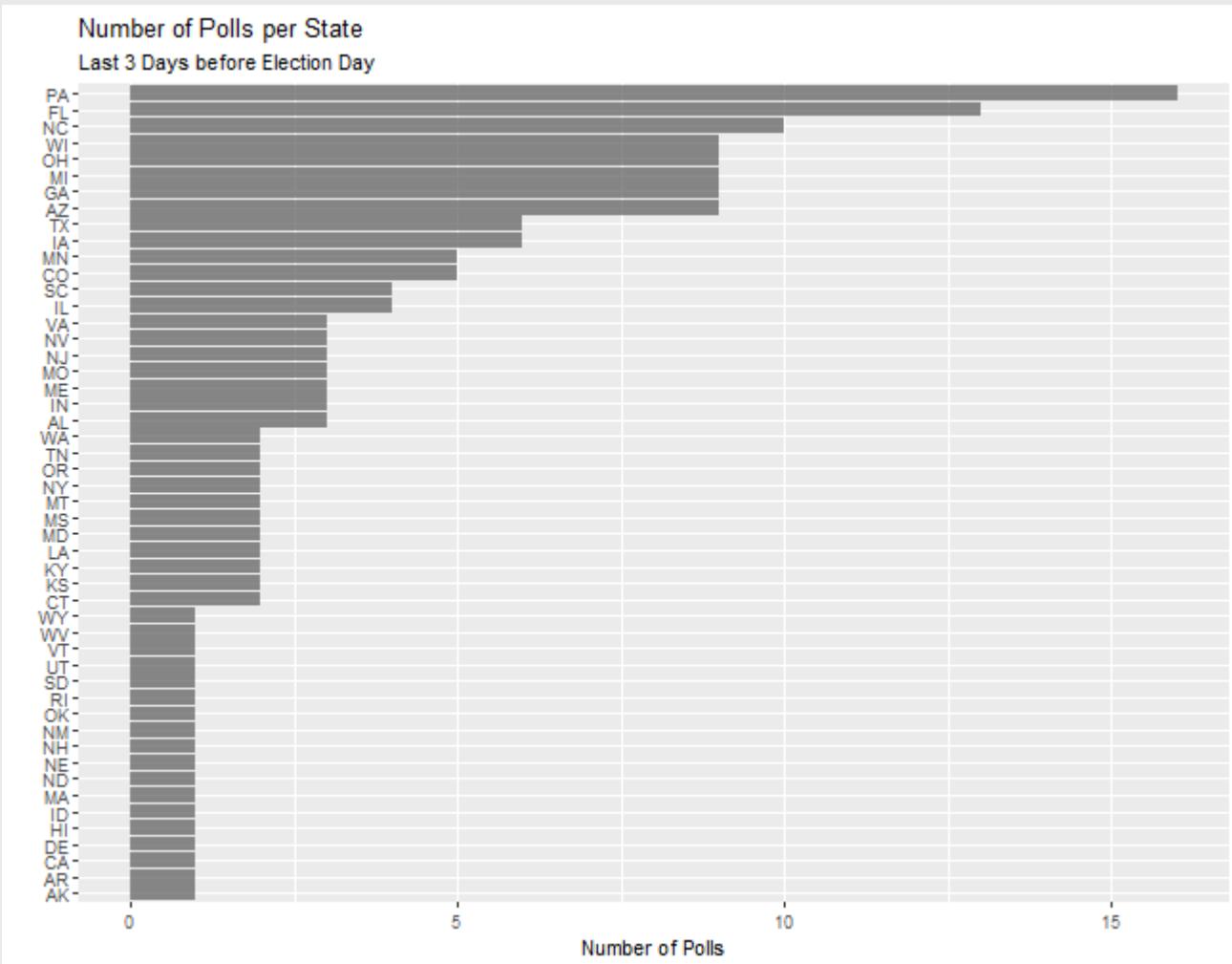
- Overall EC votes *always* favored Biden (always above 270)
- But none of our estimates are close to the true outcome of 306 to 232

Science Questions

- We know that **national** polls changed closer to election
- Can we do better if we look only at **state** polls close to election day?
- **Data** limitations!

```
pollsPerState <- statePoll %>%
  filter(DaysToED <= 3) %>%
  count(State) %>%
  ggplot(aes(x = n,y = reorder(State,n))) +
  geom_bar(stat = 'identity',alpha = .7) +
  labs(title = 'Number of Polls per State',
       subtitle = 'Last 3 Days before Election Day',
       x = 'Number of Polls',
       y = '')
```

Data Limitations



Do better with recent polls?

```
statePoll %>%  
  filter(DaysToED <= 3) %>%  
  group_by(StateName, State) %>%  
  summarise(BidenProb1 = mean(Biden > Trump),  
            BidenProb2 = mean(Biden),  
            BidenProb3 = mean(Biden2W),  
            EV = mean(EV), .groups = 'drop') %>%  
  mutate(BidenEV1 = BidenProb1 * EV,  
        BidenEV2 = BidenProb2 * EV,  
        BidenEV3 = BidenProb3 * EV,  
        TrumpEV1 = EV - BidenEV1,  
        TrumpEV2 = EV - BidenEV2,  
        TrumpEV3 = EV - BidenEV3) %>%  
  summarise_at(vars(matches('EV\\d')), sum)
```

```
## # A tibble: 1 × 6  
##   BidenEV1 BidenEV2 BidenEV3 TrumpEV1 TrumpEV2 TrumpEV3  
##     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  
## 1     334.     277.     285.     201.     258.     250.
```

Conclusion

- Still not nailing the true votes!
 - Not even within the distributions of the bootstrapped estimates!
- Why?
 1. Polls have their own uncertainty (could weight by total respondents?)
 2. Polls might not be representative of voters
 3. Polls might not be independent of each other

Quiz & Homework

- Go to Brightspace and take the **8th** quiz
 - The password to take the quiz is #####
- **Homework:**
 1. Work through Data_Wrangling_Part2_hw.Rmd
 2. Finish Problem Set 3 (Brightspace)

