Problem Set 5

Ilayda Koca

2022-10-05

Getting Set Up

If you haven't already, create a folder for this course, and then a subfolder within for the second lecture `Topic7_ConditionalVariation`, and two additional subfolders within `code` and `data`.

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Change the title to `"DS1000: Problem Set 5"` and the author to your full name. Save this file as `[LAST NAME]_ps5.Rmd` to your `code` folder.

If you haven't already, download the `Pres2020_PV.Rds` and `Pres2020_StatePolls.Rds` file from the course github page (https://github.com/jbisbee1/DSCI1000/blob/main/Lectures/Topic7_ConditionalVariation/data/) and save it to your `data` folder. Then require `tidyverse` and load the `Pres2020_PV.Rds` data to `pres`.

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.0      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

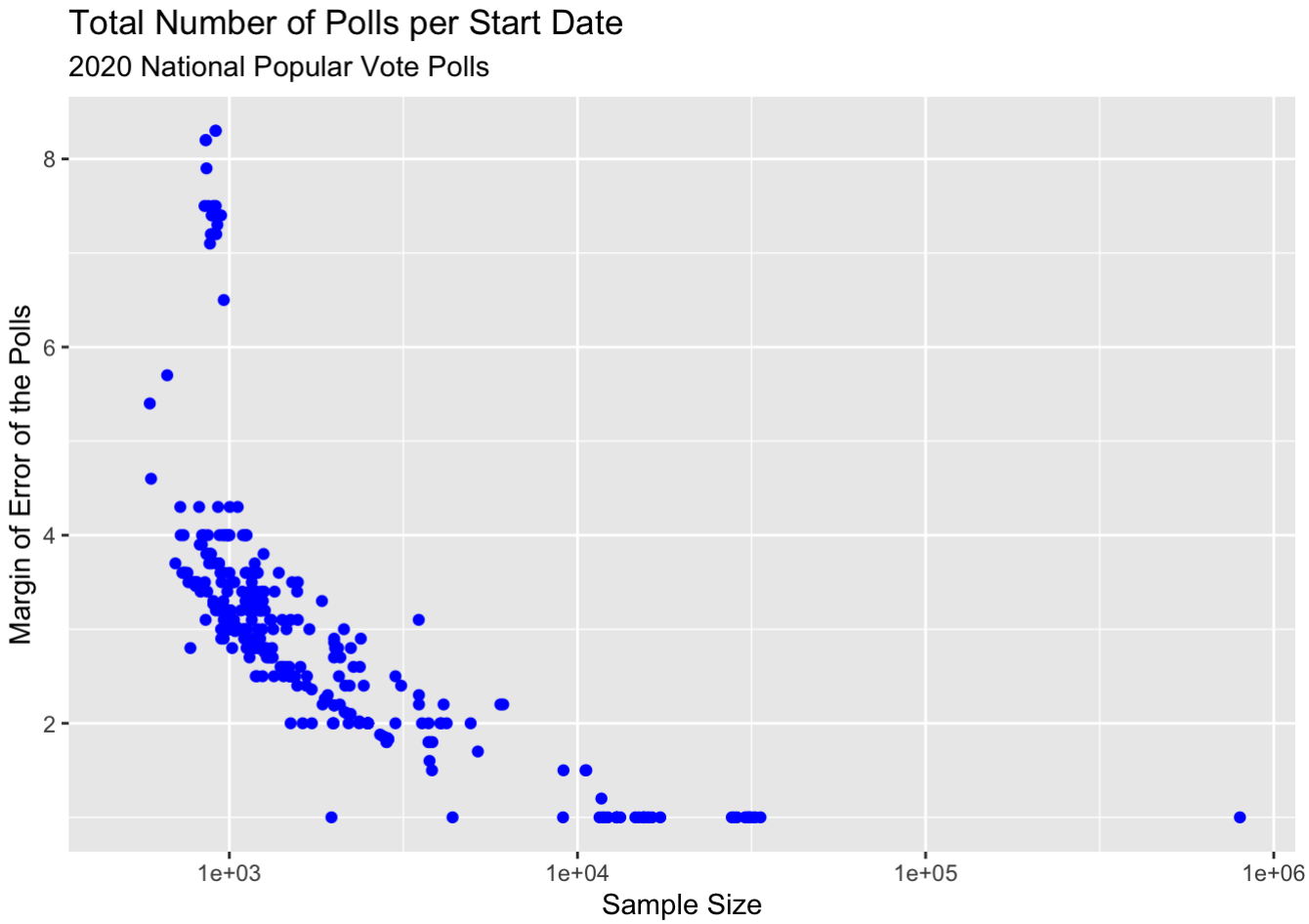
```
pres <- readRDS('../data/Pres2020_PV.Rds')
```

Question 1 [3 points]

What is the relationship between the margin of error (`MoE`) and the sample size (`SampleSize`)? Choose the appropriate plot and describe your answer in a few sentences. Make sure to label the figure thoroughly. EXTRA CREDIT: make a sensible decision about how to handle the skew in the data and defend your choice.

```
g1 <- pres %>%
  group_by(MoE) %>%
  ggplot() +
  geom_point(aes(x = SampleSize, y= MoE), color = "blue") +
  labs(title = "Total Number of Polls per Start Date",
       subtitle = "2020 National Popular Vote Polls",
       x = "Sample Size",
       y = "Margin of Error of the Polls") +
  scale_x_log10()
g1
```

```
## Warning: Removed 150 rows containing missing values (geom_point).
```



- According to the graph, there is a negative correlation between Sample Size and Margin of Error in polling data. The greater the Sample Size is, the lower the Margin of Error. We removed the outlier data for better visualization purposes and focused on a certain domain. This does not undermine the conclusion drawn in the analysis since the removed data point was also in line with the conclusion made (having large sample size with low margin of error).

Question 2 [3 points]

Calculate the weighted average of Biden and Trump's support via the `weighted.mean()` function. Try using both the sample size (`SampleSize`) as weights or the inverse of the margin of error (`1/MoE`). Compare these estimates to the un-weighted estimate. Which is more accurate? NB: you will have to make a decision about how to handle polls that don't report their margin of error or sample size. Describe and justify your decision.

```
g2 <- pres %>%
  mutate(SampleSize = ifelse(is.na(SampleSize),mean(SampleSize,na.rm=T),SampleSize))
 %>%
  summarize(WA_Trump_SS = weighted.mean(x=pres$Trump, w=pres$SampleSize, na.rm=T))

g3 <- pres %>%
  mutate(SampleSize = ifelse(is.na(SampleSize),mean(SampleSize,na.rm=T),SampleSize))
 %>%
  summarize(WA_Biden_SS = weighted.mean(x=pres$Biden, w=pres$SampleSize, na.rm=T))

g4 <- pres %>%
  mutate(MoE_new = ifelse(is.na(pres$MoE),mean(pres$MoE,na.rm=T),pres$MoE)) %>%
  summarize(WA_Trump_MoE = weighted.mean(x=pres$Trump, w=(1/MoE_new), na.rm=T))

g5 <- pres %>%
  mutate(MoE_new = ifelse(is.na(pres$MoE),mean(pres$MoE,na.rm=T),pres$MoE)) %>%
  summarize(WA_Biden_MoE = weighted.mean(x=pres$Biden, w=(1/MoE_new), na.rm=T))

g2
```

```
## # A tibble: 1 × 1
##   WA_Trump_SS
##         <dbl>
## 1        43.1
```

```
g3
```

```
## # A tibble: 1 × 1
##   WA_Biden_SS
##         <dbl>
## 1        50.2
```

```
g4
```

```
## # A tibble: 1 × 1
##   WA_Trump_MoE
##          <dbl>
## 1         41.5
```

```
g5
```

```
## # A tibble: 1 × 1
##   WA_Biden_MoE
##          <dbl>
## 1         49.3
```

```
g6 <- pres %>%
  summarize(UW_Trump = mean(x=pres$Trump, na.rm=T), UW_Biden = mean(x=pres$Biden, na.
rm=T))

g6
```

```
## # A tibble: 1 × 2
##   UW_Trump UW_Biden
##      <dbl>    <dbl>
## 1     41.2     49.3
```

- The actual support for Biden was 51% and the actual support for Trump was 47% in the 2020 presidential elections. The un- weighted average support for Biden was 49.25% and the un-weighted average support for Trump was 41.23% according to the poll results. The most accurate average support for Biden and Trump is given by the weighted average, weighted by sample size, as the sample size-weighted average support were 50.15% and 43.06% respectively.

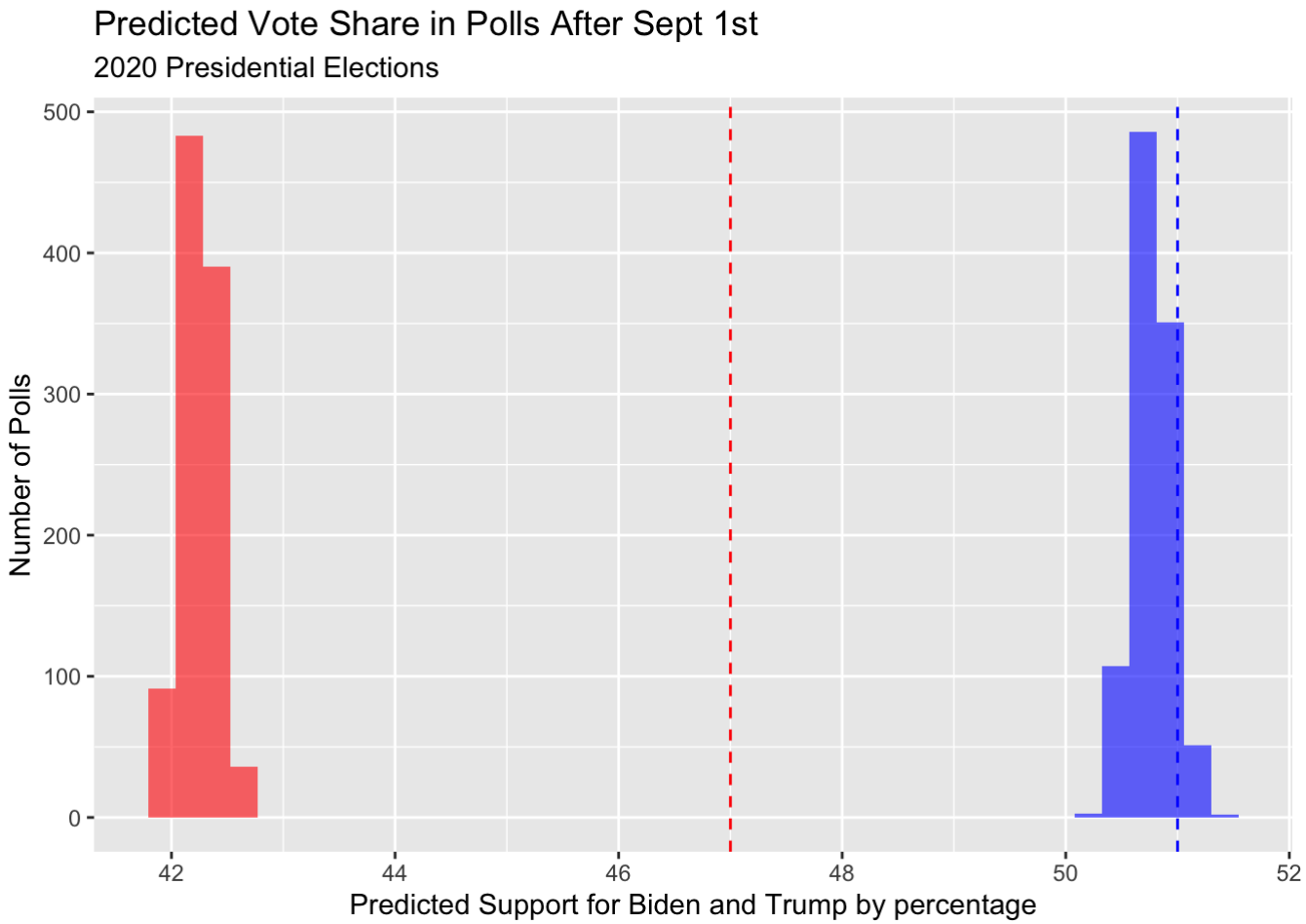
Question 3 [4 points]

Did national polls fielded on or after September 1st over-estimate Biden's support and underestimate Trump's support? Answer the question using bootstrap simulations that randomly sample every poll with replacement (`sample_n(size = nrow(.), replace = T)`) to express your confidence in your conclusion. Save your bootstrapped results to an object named `bs_Nation_Vanilla` . Plot the simulated results with two histograms on the same plot, using red for Trump and blue for Biden, and overlay vertical lines showing their true vote shares (red dashed lines for Trump and blue dashed lines for Biden). Make sure to label your figure thoroughly.

```
pres_sept <- pres %>%
  mutate(StartDate = as.Date(StartDate, format='%m/%d/%Y')) %>%
  filter(StartDate >= as.Date("2020-09-01"))

set.seed(123)

bs_Nation_Vanilla  <- NULL
for(i in 1:1000) {
  bs_Nation_Vanilla <- pres_sept %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarise(avg_biden = mean(Biden, na.rm=T), avg_trump = mean(Trump, na.rm=T)) %>%
    bind_rows(bs_Nation_Vanilla)
}

g7 <- bs_Nation_Vanilla %>%
  summarize(conf_Biden = mean(bs_Nation_Vanilla$avg_biden > 51, na.rm=T), conf_Trump
 = mean(bs_Nation_Vanilla$avg_trump < 47, na.rm=T))

g7
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1      0.084          1
```

```
g8 <- bs_Nation_Vanilla %>%
  ggplot(aes(x = bs_Nation_Vanilla$StartDate)) +
  geom_histogram(aes(x=avg_biden),bins=40,fill='blue',alpha=.6) +
  geom_histogram(aes(x=avg_trump),bins=40,fill='red',alpha=.6) +
  geom_vline(xintercept = 47, linetype='dashed', color= 'red') +
  geom_vline(xintercept = 51, linetype='dashed', color= 'blue') +
  labs(title = "Predicted Vote Share in Polls After Sept 1st",
       subtitle = "2020 Presidential Elections",
       x = "Predicted Support for Biden and Trump by percentage",
       y = "Number of Polls")

g8
```



- The confidence level for Biden being over-predicted is 8.4% while the confidence level for Trump being under-predicted is 100%.

Question 4 [4 points]

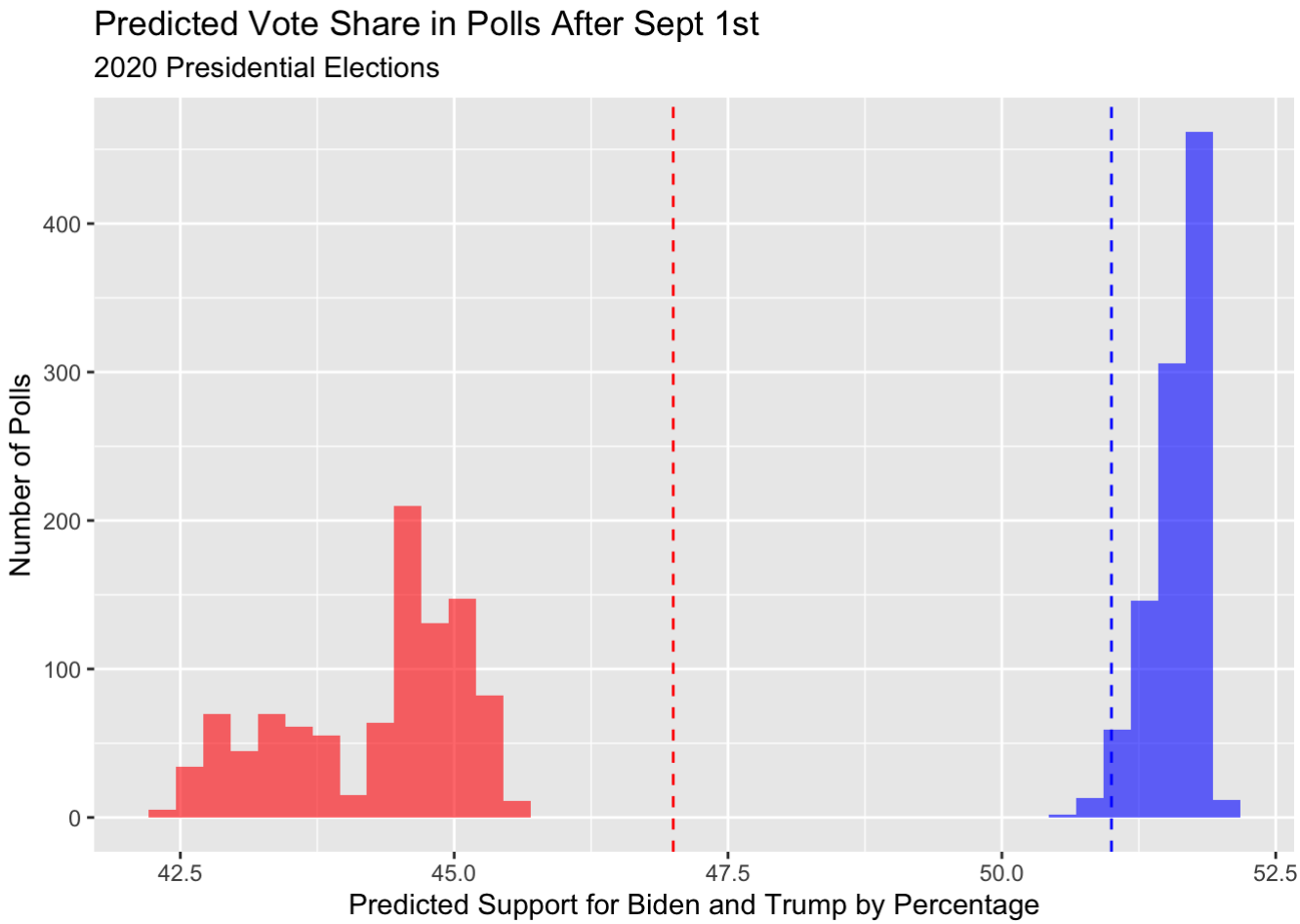
Re-estimate the preceding question, but this time calculate the weighted average of support for Biden and Trump using the best weights as identified in Question 2 above. Save your bootstrapped results to an object named `bs_Nation_Wgt` . Does your *confidence* change from the previous question? Which set of results are

more accurate?

```
pres_sept <- pres %>%
  mutate(StartDate = as.Date(StartDate, format='%m/%d/%Y')) %>%
  filter(StartDate >= as.Date("2020-09-01"))

set.seed(123)

bs_Nation_Wgt  <- NULL
for(i in 1:1000) {
  bs_Nation_Wgt <- pres_sept %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarize(w_avg_trump = weighted.mean(Trump, w=SampleSize, na.rm=T),
              w_avg_biden = weighted.mean(Biden, w=SampleSize, na.rm=T)) %>%
    bind_rows(bs_Nation_Wgt)
}

g9 <- bs_Nation_Wgt %>%
  summarize(conf_Biden = mean(bs_Nation_Wgt$w_avg_biden > 51, na.rm=T), conf_Trump =
 mean(bs_Nation_Wgt$w_avg_trump < 47, na.rm=T))

g9
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1      0.979          1
```

```
g10 <- bs_Nation_Wgt %>%
  ggplot(aes(x = bs_Nation_Wgt$StartDate)) +
  geom_histogram(aes(x=w_avg_biden),bins=40,fill='blue',alpha=.6) +
  geom_histogram(aes(x=w_avg_trump),bins=40,fill='red',alpha=.6) +
  geom_vline(xintercept = 47, linetype='dashed', color= 'red') +
  geom_vline(xintercept = 51, linetype='dashed', color= 'blue') +
  labs(title = "Predicted Vote Share in Polls After Sept 1st",
       subtitle = "2020 Presidential Elections",
       x = "Predicted Support for Biden and Trump by Percentage",
       y = "Number of Polls")

g10
```



- The confidence level for Biden being over-predicted is 97.9% while the confidence level for Trump being under-predicted is 100%. Compared to the previous un-weighted set of results, weighted results are more accurate. In the raw data, the underestimate on Trump is something like 5.7 percentage points, and the underestimate on Biden is 1.7 percentage points, whereas the in the weighted data, the underestimate on Trump is 3.9 percentage points, and the underestimate on Biden is 1.0 percentage points.

Question 5 [3 points]

Can we do better with aggregating state polls? Load the [`Pres2020_StatePolls.Rds`] to an object called `state` . Then, aggregating over all states, take the average popular vote share for Biden and Trump and compare this estimate to their true values. Do the raw state polls do better or worse than the national polls? (NB: your point of comparison should still be the `DemCertVote` and `RepCertVote` from the `pres` data, **not** the `BidenCertVote` and `TrumpCertVote` from the state data.) What if you weighted the state polls by `SampleSize` instead?

```
state <- readRDS('../data/Pres2020_StatePolls.Rds')

p1 <- state %>%
  filter(!is.na(SampleSize)) %>%
  group_by(State) %>%
  summarize(mBiden = mean(Biden, na.rm = T),
            wmBidenSS = weighted.mean(Biden, w=SampleSize, na.rm = T),
            mTrump = mean(Trump, na.rm = T),
            wmTrumpSS = weighted.mean(Trump, w=SampleSize, na.rm = T))
p2 <- p1 %>%
  summarize(State_Biden_UnWeighted = mean(mBiden),
            State_Trump_UnWeighted = mean(mTrump),
            Nation_Trump_UnWeighted = mean(x=pres$Trump, na.rm=T),
            Nation_Biden_UnWeighted = mean(x=pres$Biden, na.rm=T),
            State_Biden_Weighted = mean(wmBidenSS),
            State_Trump_Weighted = mean(wmTrumpSS),
            Nation_Biden_Weighted = weighted.mean(x=pres$Biden, w=pres$SampleSize, n
a.rm=T),
            Nation_Trump_Weighted = weighted.mean(x=pres$Trump, w=pres$SampleSize, n
a.rm=T))
p2
```

```
## # A tibble: 1 × 8
##   State_Biden_UnWeighted State…¹ Natio…² Natio…³ State…⁴ State…⁵ Natio…⁶ Natio…⁷
##                    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1                   48.9    46.0    41.2    49.3    49.6    46.4    50.2    43.1
## # … with abbreviated variable names ¹State_Trump_UnWeighted,
## #   ²Nation_Trump_UnWeighted, ³Nation_Biden_UnWeighted, ⁴State_Biden_Weighted,
## #   ⁵State_Trump_Weighted, ⁶Nation_Biden_Weighted, ⁷Nation_Trump_Weighted
```

- The state-level prediction for Biden (48.9%) is less accurate than the nation-level predicted support for Biden (49.25%). For Trump, the state-level prediction of support is more accurate than the nation-level prediction of support (41.2%). When we consider weighted data, Biden's predicted support is more accurate at the nation level again, and Trump's support is more accurate at the state level again. For both of the candidates, weighted results are more accurate than the un-weighted prediction of support.

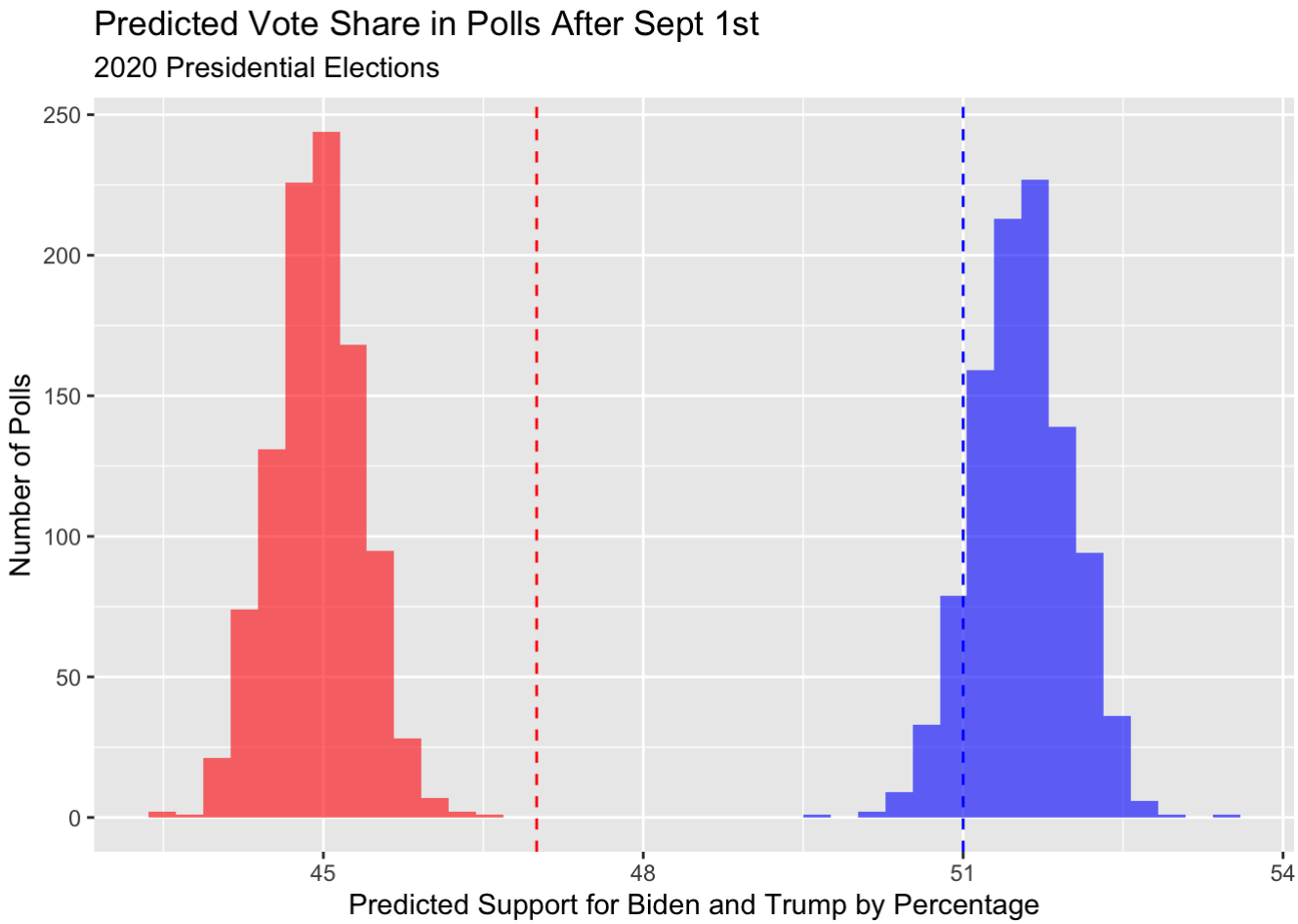
Question 6 [3 points]

Redo the analyses for question 4 using the state polls fielded after September 1st and weighting by `SampleSize` (again, make a choice about how to deal with the missing weights). Save your bootstrap results to an object named `bs_State_Wgt`. How confident are you that state polls overestimate Biden's actual popular vote (`DemCertVote` from the `pres` data)? How confident are you that state polls underestimate Trump's actual popular vote (`RepCertVote` from the `pres` data)? Based on this analysis, which set of polls would you prefer to use: national (i.e., the `pres` data) or state (i.e., the `state` data)?

```
afterSepfirst <- state %>%
  mutate(date = as.Date(StartDate,format = '%m/%d/%Y')) %>%
  filter(date >= as.Date("2020-09-01"))

set.seed(123)

bs_State_Wgt <- NULL
for(i in 1:1000){
  bs_State_Wgt <- afterSepfirst %>%
  sample_n(size = nrow(.),replace=T) %>%
  mutate(SampleSize = ifelse(is.na(SampleSize),mean(SampleSize,na.rm=T),SampleSize))
 %>%
  summarize(Biden_wSmean = weighted.mean(Biden,w = SampleSize,na.rm=T),
            Trump_wSmean = weighted.mean(Trump,w = SampleSize,na.rm=T)) %>%
  bind_rows(bs_State_Wgt)
}

bs_State_Wgt %>%
  summarise(WsconfBiden = mean(Biden_wSmean> 51, na.rm=T), WsconfTrump = mean(Trump_w
Smean< 47, na.rm=T))
```

```
## # A tibble: 1 × 2
##   WsconfBiden WsconfTrump
##         <dbl>       <dbl>
## 1       0.892           1
```

```
g12 <- bs_State_Wgt %>%
  ggplot(aes(x = bs_State_Wgt$date)) +
  geom_histogram(aes(x=Biden_wSmean),bins=40,fill='blue',alpha=.6) +
  geom_histogram(aes(x=Trump_wSmean),bins=40,fill='red',alpha=.6) +
  geom_vline(xintercept = 47, linetype='dashed', color= 'red') +
  geom_vline(xintercept = 51, linetype='dashed', color= 'blue') +
  labs(title = "Predicted Vote Share in Polls After Sept 1st",
       subtitle = "2020 Presidential Elections",
       x = "Predicted Support for Biden and Trump by Percentage",
       y = "Number of Polls")

g12
```



- According to the state-level weighted average of predicted support for Biden and Trump, the confidence level for Biden being over-predicted is %89.2% while the confidence level for Trump being under-predicted is 100%. Compared to the previous national-level set of results, weighted results provide better estimates since the magnitude of error is lower in the state-level results. This can be concluded because the confidence level Biden over-estimation is lower, therefore closer to the actual results, in the state-level results.

Question 7 [3 extra credit points]

EXTRA CREDIT: Run the bootstrapped analysis on the state poll data over time. Specifically, start with the full data and then restrict attention to 90 days prior to the election, 60 days prior to the election, 30 days prior to the election, and 14 days prior to the election. Plot the results with `geom_violin()` for Trump (in red) and Biden (in blue) for each subset of the data, overlaying dashed horizontal lines (again in red and blue) depicting the true support for each candidate. When do the state polls begin to converge on the true popular vote share? Provide a theory for why this might be the case.

```
days_prior90 <- state %>%
  filter((DaysToED <= 90))

set.seed(123)

bs_State_prior90  <- NULL
for(i in 1:1000) {
  bs_State_prior90 <- days_prior90 %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarise(avg_biden = mean(Biden, na.rm=T), avg_trump = mean(Trump, na.rm=T)) %>%
    bind_rows(bs_State_prior90)
}

bs_State_prior90 %>%
  summarize(conf_Biden = mean(bs_State_prior90$avg_biden > 51, na.rm=T),
            conf_Trump = mean(bs_State_prior90$avg_trump < 47, na.rm=T))
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1          0          1
```

```
days_prior60 <- state %>%
  filter((DaysToED <= 60))

set.seed(123)

bs_State_prior60  <- NULL
for(i in 1:1000) {
  bs_State_prior60 <- days_prior60 %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarise(avg_biden = mean(Biden, na.rm=T), avg_trump = mean(Trump, na.rm=T)) %>%
    bind_rows(bs_State_prior60)
}

bs_State_prior60 %>%
  summarize(conf_Biden = mean(bs_State_prior60$avg_biden > 51, na.rm=T),
            conf_Trump = mean(bs_State_prior60$avg_trump < 47, na.rm=T))
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1          0          1
```

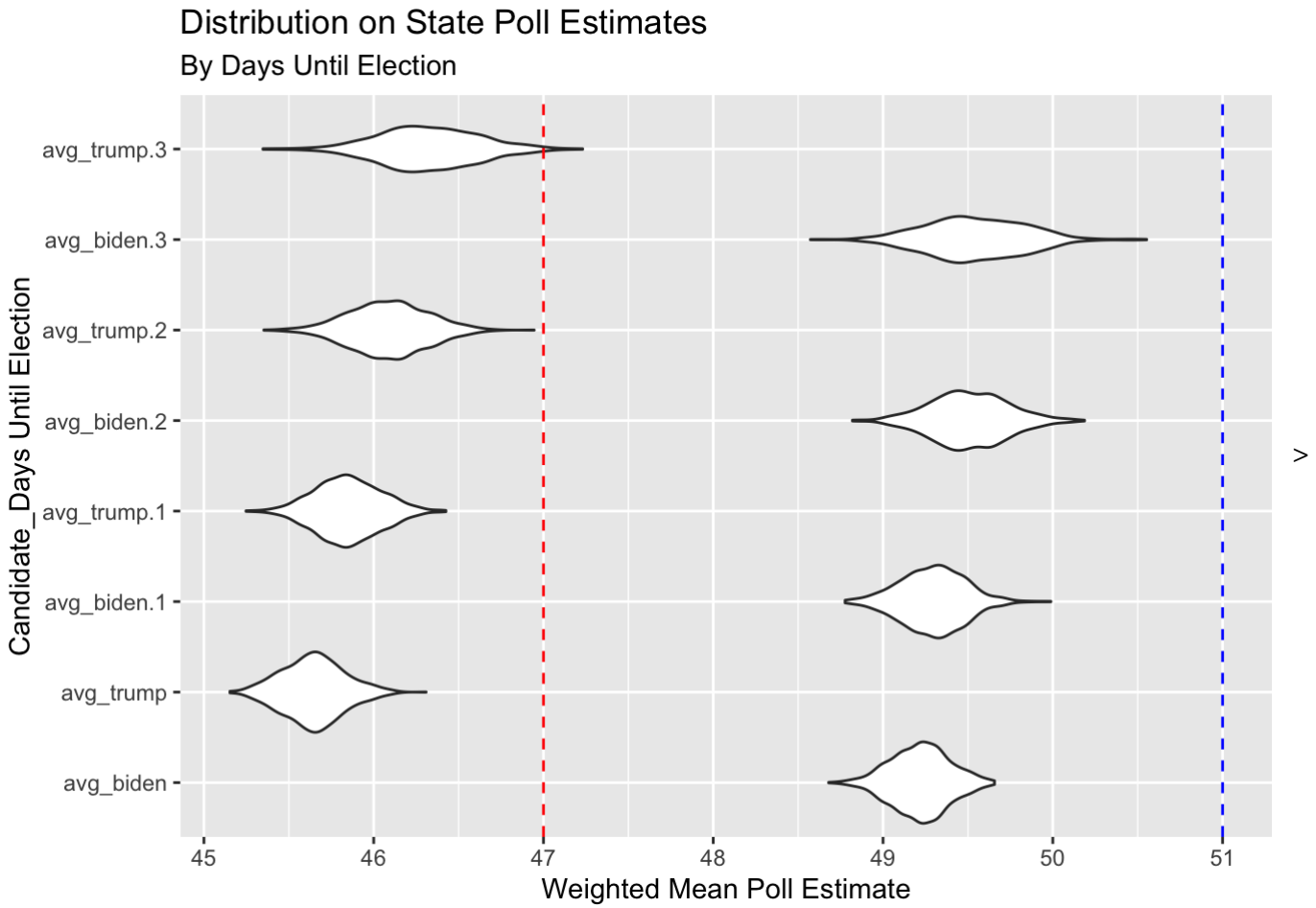
```
days_prior30 <- state %>%
  filter((DaysToED <= 30))

set.seed(123)

bs_State_prior30  <- NULL
for(i in 1:1000) {
  bs_State_prior30 <- days_prior30 %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarise(avg_biden = mean(Biden, na.rm=T), avg_trump = mean(Trump, na.rm=T)) %>%
    bind_rows(bs_State_prior30)
}

bs_State_prior30 %>%
  summarize(conf_Biden = mean(bs_State_prior30$avg_biden > 51, na.rm=T),
            conf_Trump = mean(bs_State_prior30$avg_trump < 47, na.rm=T))
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1          0          1
```

```
days_prior14 <- state %>%
  filter((DaysToED <= 14))

set.seed(123)

bs_State_prior14  <- NULL
for(i in 1:1000) {
  bs_State_prior14 <- days_prior14 %>%
    sample_n(size = nrow(.), replace = T) %>%
    summarise(avg_biden = mean(Biden, na.rm=T), avg_trump = mean(Trump, na.rm=T)) %>%
    bind_rows(bs_State_prior14)
}

bs_State_prior14 %>%
  summarize(conf_Biden = mean(bs_State_prior14$avg_biden > 51, na.rm=T),
            conf_Trump = mean(bs_State_prior14$avg_trump < 47, na.rm=T))
```

```
## # A tibble: 1 × 2
##   conf_Biden conf_Trump
##        <dbl>      <dbl>
## 1          0       0.99
```

```
VIOLIN <- NULL

VIOLIN <- cbind(bs_State_prior90, bs_State_prior60, bs_State_prior30, bs_State_prior1
4)

ECplt <- ggplot(stack(VIOLIN), aes(x = ind, y = values)) +
  geom_violin() +
  geom_hline(yintercept = 47, linetype = 'dashed',color = 'red') +
  geom_hline(yintercept = 51, linetype = 'dashed',color = 'blue') +
  coord_flip() +
  scale_fill_manual(values = c('Biden' = 'blue','Trump' = 'red')) +
  labs(title = 'Distribution on State Poll Estimates',
       subtitle = 'By Days Until Election',
       x = 'Candidate_Days Until Election',
       y = 'Weighted Mean Poll Estimate')

ECplt
```



- The poll results start converging to the actual results 14 days prior to the election date. This might be due to the fact that people's minds regarding their votes change over time such that Trump supporters 90 days ago might start supporting Biden 90 days later and visa-versa.