2018 U.S. Midterm Popular Vote Estimation

Matt Klapman

11/30/2018

Introduction

The 2018 United States midterm elections took place on Nov. 4, 2018. The goal of this project was to analyze polling data related to these elections, attempting to estimate the national popular vote and quantify some type of error along with it. In the past, polling has been an important aspect of election forecasting in political science. But, in the 2016 U.S. presidential election, some doubt was cast on the accuracy of polling. This was due to the perceived failure of polls and polling aggregators like the website *FiveThirtyEight* by the media and the populous. While *FiveThirtyEight* pointed out in their 11-part series **The Real Story of 2016** that the error in 2016 was mostly within the realm of normal polling error and the story around the "failure" of the polls centered around a predominant misunderstanding of probability [1], I thought it would be interesting to take a stab at predicting this year's election using only polls.

I was able to find **polling data** from the website *Real Clear Politics (RCP)* [2]. *RCP* had been recording and aggregating polls for this year's election from March 2017 until just days before the election. This data was presented as a table on their website, so it took some work to upload into R. There were a total of 309 polls taken in this time span by a range of entities such as news organizations, non-profit research firms, and partisan political organizations. The polls contained information about the polling organization, date of the poll, sample size, sample type, and the results for the Democratic Party and the Republican Party in a "generic ballot" poll. A "generic ballot" poll just asks a surveyee which party they would vote for in the upcomming election, assigning no politician to the question. I did not have access to information about those who chose neither of the parties.

Methods

I had three main strategies for estimating the difference in popular vote proportion. Each of these strategies involved taking some type of mean (weighted or not) and then some type of bootstrapping in order to quantify error. These three methods were a naive method(1), a weighting method(2), a subsetting method(3), and a relative support moethod(4).

1 Naive Estimate

My initial strategy for estimating the difference in popular vote proportion was the naive estimate of equally weighting the polls and taking a standard mean. After taking the standard mean, I performed a bootstrap analysis on this estimate to attempt to quantify error. Each poll in the sample had an equal probability of being selected in the bootstrap sample. I chose this method because if we are assuming each poll to be *representative of the population of voters* (which I will discuss the validity of later), then we can think of each poll as being an iteration of the election on Nov. 4th.

I found my estimated difference in proportions to be Dem +7.32% with a 95% bootstrapped confidence interval of (Dem +7.01%, Dem +7.65%).

2 Weighted Estimate

My next strategy was to implement some type of weighting scheme to account for the general principle that polls taken closer to the election should be more accurate than polls taken much earlier than the election. I created a probability vector in which each poll was weighted by the date it was performed (specifically the closing date). For each date, I calculated it's numerical value as the number of days between it and January 1, 2017. Then, the weight for any particular poll was calculated as

$$p_i = \frac{n_i}{\sum\limits_{i=1}^{309} n_i}$$

where n_i is the numerical version of the date described above. These probabilities were designed in such a way that they sum to 1 in order to be able to use them as weights in the calculation of a weighted mean. The weighted mean difference was found to be Dem +7.2% with a 95% bootstrap confidence interval of (Dem +6.86%, Dem +7.6%) calculated using the weighted probabilities.

3 Subsetting Method

The final method I used relied on the assumption (or lack there of) of each poll being a representative sample. If a sample was taken of registered voters or just adults, it may not have been as representative as a sample of "likely voters," where a sample of likely voters is determined by the researchers behind the poll. So, for my third method I decided to estimate using just the likely voter samples. I also used the same weighting scheme as in method 2 for this subset of data, weighting polls higher the closer they were to election day. With this method I ended up with an estimate of Dem +5.98% with a 95% bootstrap confidence interval of (Dem +5.23%, Dem +6.74%).

4 Relative Support

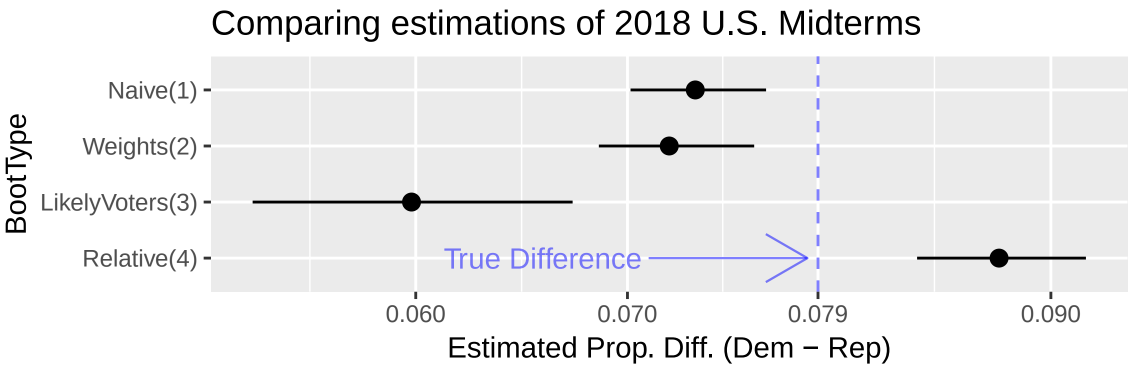
One common method in the polling field is to use some type of "relative support" metric to quantify the support for a given party. This method basically forgoes any third party support and temporarilly ignores undecided voters. This can be especially helpful for early polls that may have a good amount of undecided voters. The assumption here is that undecided voters would mostly be split in half. So, the relative support in my data was calculated to be

$$p_{rel} = \frac{p_D}{p_D + p_R}$$

where p_D is the proportion of surveyees claiming support for Democrats and p_R is the proportion of surveyees claiming support for Republicans. With this method, I found the estimated difference in support to be Dem +8.76% with a 95% bootstrap confidence interval of (Dem +8.37%, Dem +9.17%).

Results

The true result of the 2018 U.S. midterm elections ended up being a Democratic advantage of about 7.9%. We can compare this eventual result with the results of my bootstrapping in the following plot (with the true difference marked in blue).



It's obvious to see from that graph that my estimates and subsequent bootstrap confidence intervals severely underestimated the actual result from the election, except for my relative support estimate which severely *over*estimated the actual result.

Discussion

In the end, using just the polling data did not provide a completely accurate result. I think that this was mainly due to the assumptions I had to make about the polls and those who ran them. The assumption that each poll was represtentative of all registered voters probably was not valid. There is a good amount of polling error that I did not account for. Though, since I didn't know the methodology behind each and every poll, I didn't have adequate information to account for that like a more sophisticated poll aggregator might. If I had information on the pollsters and their respective historical biases, I could create weights based on that to incorporate into my bootstrap.

Another anomaly in 2018 was that this election resulted in the highest voter turn out in over 100 years in the U.S. We also know that though both parties increased turn out, the Democratic party was especially energized due to their poor results in 2016. This could have caused polling error as the polls likely underestimated just how many people would show up on election day.

I think that future work on this data could involve some type of accounting for the accuracy of the polls. My estimates using the polls were not accurate, including both over and under estimation. There is plenty of polling error that needs to be accounted for in the analysis of the polls. One way to account for more error could be to do some type of bootstrap prediction interval. I'm not sure if there is a way to do a prediction interval with bootstrapping, considering the distribution you get from the bootstrap procedure is centered around taking means. This project, though, was just an introduction into the ideas of estimating and predicting election results. Overall, I think it was a good experience learning some different methods for election result estimation and learning how to use Github and Rstudio cloud to organize a large scale project.

Resources

[1] Silver, Nate. "The Real Story Of 2016." FiveThirtyEight, FiveThirtyEight, 12 June 2017, fivethirtyeight.com/features/the-real-story-of-2016/.

[2] "2018 Generic Congressional Vote" Polls, RealClearPolitics,

www.realclearpolitics.com/epolls/other/2018_generic_congressional_vote-6185.html.

Apendix (R codes)

```r
knitr::opts_chunk$set(echo = FALSE, fig.height = 2, fig.width = 6)
library(tidyverse)
library(here)
rcp_midterms <- read_csv(here("DATA","RCP_midterms.csv"))
rcp_midterms <- rcp_midterms %>% mutate(prop_diff = (Dem - Rep)/100, ord = 1:length(prop_diff),
                                        Dates = as.Date(Dates, "%m/%d/%Y",
                                                        origin = as.Date("01/01/2017","%m/%d/%Y")),
                                        #tr_num_dates = trunc(as.numeric(Dates) - 17167, prec = -1)/10
                                        num_dates = as.numeric(Dates) - 17167,
                                        prob = num_dates/sum(num_dates),
                                        rel_Dem = Dem/(Dem + Rep),
                                        rel_Rep = Rep/(Dem + Rep),
                                        rel_diff = rel_Dem - rel_Rep,
                                        prob_full = double(length(prob)))

rcp_midterms$prob_full[rcp_midterms$SampType == "LV"] <-
  rcp_midterms$prob[rcp_midterms$SampType == "LV"] + 1/3/309

rcp_midterms$prob_full[rcp_midterms$SampType == "RV"] <-
  rcp_midterms$prob[rcp_midterms$SampType == "RV"] + 2/9/309

rcp_midterms$prob_full[rcp_midterms$SampType == "A"] <-
  rcp_midterms$prob[rcp_midterms$SampType == "A"] + 1/9/309

library(rsample)
devtools::load_all()


# 1 Naive
naive_est <- mean(rcp_midterms$prop_diff)
ci_naive <- with(rcp_midterms, boot_ci_propdiff(prop_diff, ntimes = 100))

# 2 Weighted
ci_weight <- with(rcp_midterms, boot_ci_propdiff(prop_diff, prob = prob, ntimes = 100))
w_est <- with(rcp_midterms, weighted.mean(prop_diff, w = prob)) # weighted estimate of proportion.

# 3 LikelyVoters
rcp_lv <- filter(rcp_midterms, SampType == "LV")
lv_est <- with(rcp_lv, mean(prop_diff))
ci_lv <- with(rcp_lv, boot_ci_propdiff(prop_diff, ntimes = 100))

# 4 Relative Support
ci_relative <- with(rcp_midterms, boot_ci_propdiff(rel_diff, ntimes = 100))
rel_est <- with(rcp_midterms, mean(rel_diff))

# Results
true_diff <- 0.079
confints <- as.tibble(rbind(ci_naive, ci_weight, ci_relative, ci_lv)) %>%
  add_column(BootType = c("Naive", "Weights", "Relative", "LikelyVoters")) %>%
  add_column(Est = c(naive_est, w_est, rel_est, lv_est))

ggplot(data = confints) +
  geom_hline(aes(yintercept = true_diff), linetype = "dashed", color = "blue", alpha = .5) +
  geom_pointrange(aes(x = BootType, y = Est, ymin = `2.5%`, ymax = `97.5%`)) +
  annotate("segment", x = "Relative", y = .071, xend = "Relative", yend = true_diff-.0005,
           arrow = arrow(), size=.4, alpha=.5, color = "blue") +
  annotate("text", x = "Relative", y = 0.066, label = "True Difference", color = "blue", alpha = .5)+
  scale_y_continuous(breaks = c(.05, .06, .07, true_diff, .09)) +
  scale_x_discrete(limits = c("Relative", "LikelyVoters", "Weights", "Naive"),
                   label = c("Relative(4)", "LikelyVoters(3)", "Weights(2)", "Naive(1)")) +
  coord_flip() +
  labs(y = "Estimated Prop. Diff. (Dem - Rep)", title = "Comparing estimations of 2018 U.S. Midterms")
```