Joe Biden projected to win Popular Vote in 2020 US Election with 51% of Vote

Data given with 4 percent margin of error

Alen Mitrovski, Xiaoyan Yang, Matthew Wankiewicz

November 2nd, 2020

Abstract

It seems like everyone has been waiting for the 2020 election, almost as soon as Donald Trump won in 2016. In our report, we use a logistic regression model along with multilevel regression with post-stratification in order to predict who will win the election. According to our model, we predict that Joe Biden will win the popular vote over Donald Trump, 51% to 49%. These results are promising for the people of the United States, as they may be getting the change in President they need.

keywords: Forecasting, US 2020 Election, Trump, Biden, multilevel regression with post-stratification;

1 Introduction

2 Data

We have used R (R Core Team [2019]), specifically Tidyverse (Wickham et al. [2019]) for data analysis Data is from ACS (Ruggles et al. [2020]) and from Voter Study Group (Tausanovitch and Vavreck [2020]).

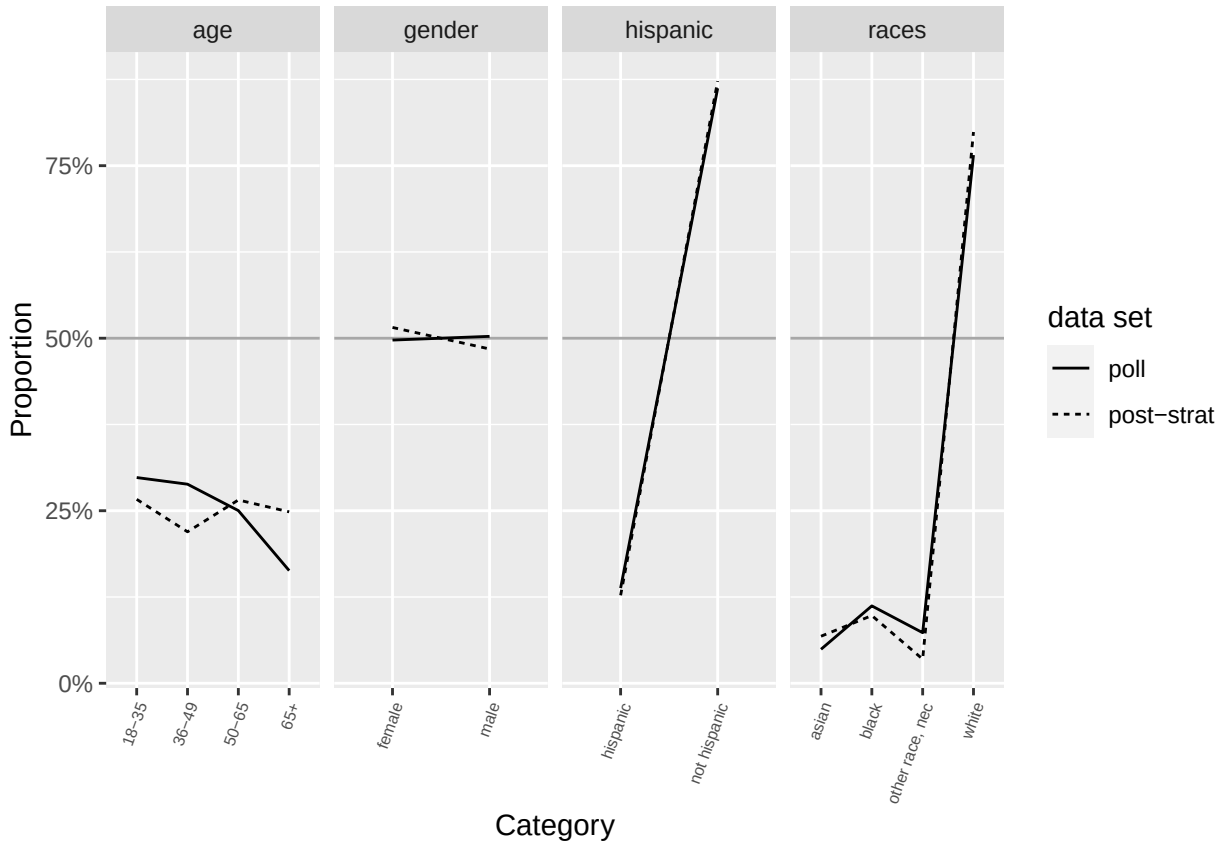

Figure 1: Demographics of Sample and Population

Figure 1 show us the voter demographics from the VSG data (Tausanovitch and Vavreck [2020]) vs the ACS data (Ruggles et al. [2020]).

Table 1: (#tab:polling props)Who decided voters plan to support (Polling Data)

Candidate	Number of Respondents	Proportion (%)
Donald Trump	2481	48
Joe Biden	2719	52

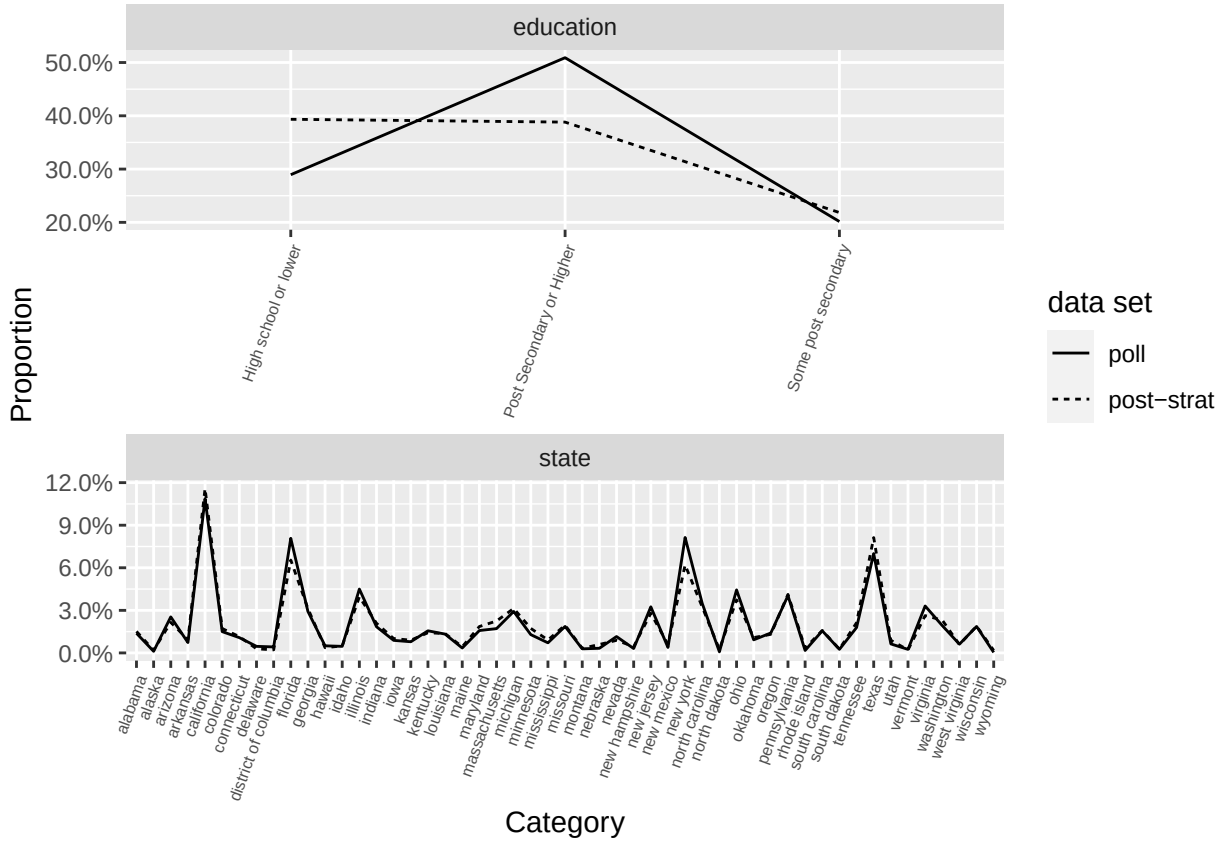

Figure 2: More Demographics of Sample and Population

Figure 2 shows us more of the voter demographics from the VSG data (Tausanovitch and Vavreck [2020]) vs the ACS data (Ruggles et al. [2020]).

Table @ref(tab:polling props) shows the proportion of decided voters who plan to vote for Donald Trump or Joe Biden. The data used to create this table is from the Voter Study Group (Tausanovitch and Vavreck [2020]).

3 Model

For our analysis, we plan to use multilevel regression with post-stratification. Multilevel regression with post-stratification (MRP) is a type of analysis where we fit a model using a smaller data set, in this case our polling data and then use the results of the model to apply it to a larger population.

The main steps for MRP are: Find the data set you want to use to create your model. For our scenario, we used the polling data from the Voter Study Group (Tausanovitch and Vavreck [2020]). Next, you must create a model using your smaller sample. We used logistic regression and the data used was the polling data. Our

equation takes the form of equation (1), as seen below. Once you have your model, you must apply it to your larger data set to give an idea of the population. For our report, we used the Census data from IPUMS (Ruggles et al. [2020]).

MRP is extremely useful not only because of how simple it is but also because it allows us to estimate preferences of a population using individual responses from surveys. Also, as Kennedy and Gelman discuss in their paper "Know your population and know your model: Using model-based regression and post-stratification to generalize findings beyond the observed sample", MRP works best when you have a population and variables of interest and want to apply these variables using two data sets with different characteristics (Kennedy and Gelman). Also, in relation to surveys, since you only use one survey to create your model, you don't encounter issues with having to have multiple surveys for each region/state and lets under-sampled populations to be represented through post-stratification.

MRP does encounter some weaknesses as well. Once again we can look to Kennedy and Gelman's paper where they say that MRP is dependent on how accurate the model is (Kennedy and Gelman). If the generated model makes incorrect predictions or assumptions, your post-stratification will be incorrect and will hurt your results.

To predict whether or not a person plans to vote for Joe Biden or Donald Trump, we plan to build a logistic regression model using data from the Voter Study Group (Tausanovitch and Vavreck [2020]) and then post-stratify it using Census Data (Ruggles et al. [2020]). Since logistic regression only works for binary response variables, we created a variable called `vote_biden` which returns a 1 is the respondent plans to vote for Joe Biden and a 0 if they plan to vote for Donald Trump.

The logistic regression model takes the form of:

$$log(\frac{\hat{p}}{1-\hat{p}}) = \beta_0 + \beta_1 x_{sex} + \beta_2 x_{agegroup} + \beta_3 x_{race} + \beta_4 x_{state} + \beta_5 x_{income} + \beta_6 x_{hispanic} \tag{1}$$

Once we have our regression model, we will use the `predict` function in R (R Core Team [2019]), to apply our model to the Census data (Ruggles et al. [2020]). We do this by grouping the Census data by the demographics we plan to analyze (sex, race, age, education level, hispanic or not, state), and then applying the model to each of those groups. After applying the model, we will receive probabilities that a person in that group will vote for Joe Biden. Once the predictions are complete, we can use them to find out who will win the popular vote or how many electoral colleges a candidate will win. We also can use a 95 percent confidence interval, which means that we are 95 percent certain that true value for the population (in this case popular vote percentage), is within the range we obtain. From this we can also conclude that our results will be accurate between +/- 4%.

In equation (1), each β represents a coefficient that the regression model will compute for us. As for our variables, we have chosen to use sex, age, race, income, state, and whether the respondent is hispanic. We decided to use the first 3 because they are generally strong predictors of which candidate a person would support, such as how some states tend to vote republican year after year while some states flip between democratic and republican almost every election. Next, we decided to choose income, because Joe Biden has made claims to increase taxes on the rich, which may influence their support for him. Lastly, we wanted to focus on whether the respondent was hispanic and if so, where they were from. This variable was important for our predictions because we know how poorly Donald Trump has spoken of hispanic people and we believe they could have a strong impact on the election.

The output of the logistic regression model will give us a probability of whether or not a voter plans to vote for Joe Biden or not. In order to find this probability, we take the sum of the right side of equation (1) and plug it into the equation below:

$$\frac{e^{sum}}{1+e^{sum}} \tag{2}$$

Equation (2) is just a manipulation of equation (1), where e is the exponential equation and *sum* is the sum of the right side of equation (1). We see that as the sum of the right side increases, the probability that a person will vote for Joe Biden increases as well. We are running our regression model using the `glm()` function in R (R Core Team [2019]). The decision to run this model over other models like linear regression was made by the fact that we were predicting a binary variable about a voter's decision. Since there are only two possible options our data will likely follow an S shape and a straight line equation will not be helpful to model this relationship. Another strength present for logistic regression is that when combined with post-stratification it allows us to take information from under-represented populations and it allows their views to be accounted for more greatly. For example, our polling data (Tausanovitch and Vavreck [2020]), includes only 2 observations from Wyoming, but using multilevel regression with post-stratification, we can have that expanded to over 3000 people!

Our model does have some weaknesses, since the output must be binary, we cannot account for other candidates or a person deciding not to vote. This issue isn't too large because our main goal is to determine which of the two main candidates will be chosen by the people of America. Another weakness we do encounter with our model and multi level regression with post-stratification is that it has a strong dependence on the survey data. This is a weakness because if the survey has any gaps or there are any tweaks we need to make, it can change the course of results.

Table 2, shows the estimates for the coefficients that will fit into our logistic regression equation. These coefficients will fit into Equation (1), and were calculated using data from the Voter Study Group (Tausanovitch and Vavreck [2020]). The table is made using `kable` from `knitr` (Xie [2020]) and is formatted using `kableExtra` (Zhu [2020])

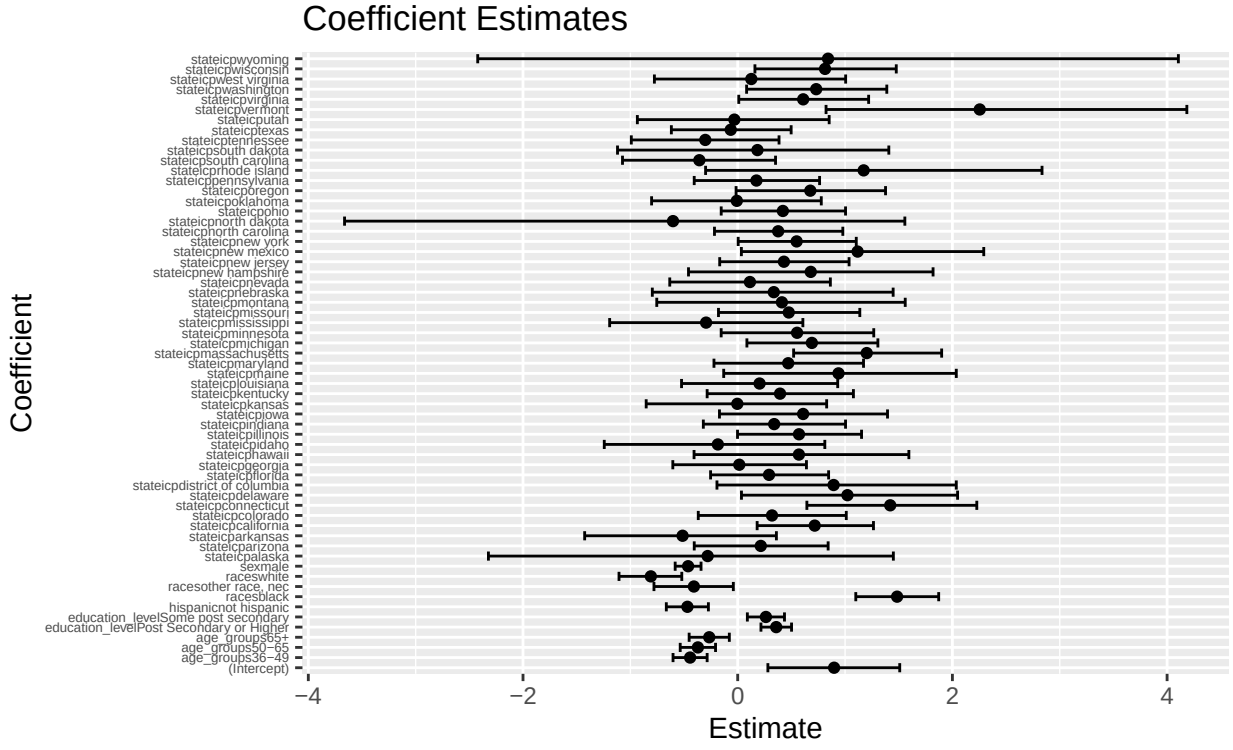

Figure 3: Coefficient Estimates

Figure 3 shows us the coefficients that would fit into equation (1) using the polling data (Tausanovitch and Vavreck [2020]). We also have error bars present, which show the upper and lower estimates for the coefficients. What we have to look out for in this scenario is that coefficients with negative values would mean that the person is more likely to vote for Donald Trump (with that characteristic) and positive values

Table 2: Coefficients from the Model

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	0.8980072	0.3126492	2.8722518	0.0040756	0.2811482	1.5092669
sexmale	-0.4619733	0.0614678	-7.5156968	0.0000000	-0.5825849	-0.3416100
age_groups36-49	-0.4433787	0.0810235	-5.4722267	0.0000000	-0.6023717	-0.2847185
age_groups50-65	-0.3709067	0.0839278	-4.4193534	0.0000099	-0.5355720	-0.2065317
age_groups65+	-0.2650972	0.0954808	-2.7764449	0.0054957	-0.4524197	-0.0780729
racesblack	1.4848649	0.1967429	7.5472352	0.0000000	1.1000293	1.8722214
racesother race, nec	-0.4082380	0.1884919	-2.1658125	0.0303255	-0.7800952	-0.0406482
raceswhite	-0.8091757	0.1489996	-5.4307243	0.0000001	-1.1057163	-0.5208546
stateicpalaska	-0.2797864	0.9132895	-0.3063502	0.7593381	-2.3221840	1.4499433
stateicparizona	0.2150895	0.3175737	0.6772900	0.4982220	-0.4050470	0.8423335
stateicparkansas	-0.5136530	0.4533527	-1.1330098	0.2572102	-1.4256850	0.3603696
stateicpcalifornia	0.7171878	0.2756199	2.6020894	0.0092658	0.1809687	1.2646193
stateicpcolorado	0.3196581	0.3509959	0.9107174	0.3624443	-0.3674600	1.0111726
stateicpconnecticut	1.4206903	0.4024846	3.5298003	0.0004159	0.6444034	2.2272548
stateicpdelaware	1.0223226	0.5096224	2.0060394	0.0448520	0.0349976	2.0485310
stateicpdistrict of columbia	0.8934468	0.5630684	1.5867464	0.1125701	-0.1932745	2.0361631
stateicpflorida	0.2917968	0.2796465	1.0434486	0.2967406	-0.2525567	0.8467523
stateicpgeorgia	0.0145529	0.3169457	0.0459162	0.9633770	-0.6044370	0.6403815
stateicphawaii	0.5701168	0.5070301	1.1244241	0.2608331	-0.4067535	1.5945679
stateicpidaho	-0.1849658	0.5193924	-0.3561196	0.7217510	-1.2431768	0.8115242
stateicpillinois	0.5709498	0.2940343	1.9417793	0.0521638	-0.0017038	1.1536467
stateicpindiana	0.3397257	0.3373038	1.0071802	0.3138482	-0.3203070	1.0044955
stateicpiowa	0.6094353	0.3982495	1.5302852	0.1259462	-0.1700387	1.3956818
stateicpkansas	-0.0036380	0.4273321	-0.0085133	0.9932075	-0.8526640	0.8287237
stateicpkentucky	0.3944897	0.3468740	1.1372710	0.2554250	-0.2847029	1.0777918
stateicplouisiana	0.2037260	0.3703378	0.5501086	0.5822449	-0.5232388	0.9314653
stateicpmaine	0.9391237	0.5468677	1.7172777	0.0859284	-0.1304090	2.0362073
stateicpmaryland	0.4704849	0.3547184	1.3263617	0.1847199	-0.2211046	1.1719732
stateicpmassachusetts	1.2019368	0.3510437	3.4238951	0.0006173	0.5210273	1.8998570
stateicpmichigan	0.6908228	0.3106500	2.2237975	0.0261621	0.0855832	1.3058171
stateicpminnesota	0.5531700	0.3618500	1.5287273	0.1263321	-0.1539583	1.2674587
stateicpmississippi	-0.2945026	0.4578634	-0.6432107	0.5200874	-1.1928848	0.6069939
stateicpmissouri	0.4757005	0.3352768	1.4188291	0.1559488	-0.1792418	1.1375350
stateicpmontana	0.4116022	0.5823474	0.7067984	0.4796918	-0.7540098	1.5605872
stateicpnebraska	0.3361727	0.5657196	0.5942391	0.5523522	-0.7953709	1.4480437
stateicpnevada	0.1138363	0.3808846	0.2988735	0.7650365	-0.6330333	0.8632322
stateicpnew hampshire	0.6805873	0.5742896	1.1850941	0.2359802	-0.4582967	1.8199862
stateicpnew jersey	0.4305098	0.3068634	1.4029365	0.1606358	-0.1677030	1.0377356
stateicpnew mexico	1.1169726	0.5688036	1.9637230	0.0495622	0.0341497	2.2920748
stateicpnew york	0.5489828	0.2798485	1.9617146	0.0497957	0.0043994	1.1044931
stateicpnorth carolina	0.3769283	0.3043149	1.2386125	0.2154890	-0.2163184	0.9791604
stateicpnorth dakota	-0.6028355	1.2029598	-0.5011269	0.6162818	-3.6624093	1.5571343
stateicpohio	0.4205613	0.2948223	1.4264906	0.1537268	-0.1539062	1.0045259
stateicpoklahoma	-0.0082254	0.4019938	-0.0204615	0.9836752	-0.8020673	0.7784033
stateicporegon	0.6762432	0.3544891	1.9076559	0.0564357	-0.0152077	1.3771460
stateicppennsylvania	0.1745544	0.2972761	0.5871793	0.5570833	-0.4051598	0.7628937
stateicprhode island	1.1732738	0.7752391	1.5134348	0.1301693	-0.2992149	2.8349608
stateicpsouth carolina	-0.3574239	0.3629406	-0.9848000	0.3247223	-1.0732722	0.3525165
stateicpsouth dakota	0.1840415	0.6330361	0.2907282	0.7712592	-1.1196390	1.4078076
stateicptennessee	-0.3020013	0.3503869	-0.8619082	0.3887380	-0.9912078	0.3850810
stateicptexas	-0.0649951	0.2838591	-0.2289698	0.8188924	-0.6179683	0.4977429
stateicputah	-0.0312335	0.4535913	-0.0688583	0.9451024	-0.9346965	0.8520835
stateicpvermont	2.2547412	0.8171449	2.7592918	0.0057927	0.8231091	4.1840250
stateicpvirginia	0.6100694	0.3079872	1.9808268	0.0476107	0.0099348	1.2197413
stateicpwashington	0.7311564	0.3325406	2.1986981	0.0278994	0.0828534	1.3888521
stateicpwest virginia	0.1268231	0.4520245	0.2805668	0.7790427	-0.7761741	1.0056192
stateicpwisconsin	0.8136348	0.3352005	2.4273077	0.0152114	0.1605671	1.4770183
stateicpwyoming	0.8415398	1.4394702	0.5846177	0.5588048	-2.4218041	4.1050804
education_levelPost Secondary or Higher	0.3587122	0.0726763	4.9357504	0.0000008	0.2165005	0.5014293
education_levelSome post secondary	0.2626752	0.0881652	2.9793539	0.0028886	0.0900105	0.4356705
hispanicnot hispanic	-0.4683979	0.0999464	-4.6864901	0.0000028	-0.6650514	-0.2731151

mean the person is more likely to vote for Joe Biden. Table 2 shows a numerical view of Figure 3, along with p-values.

Using the outputs of the logisitic regression model, we can get an equation that follows the form of (1), but with the β values filled out. This equation is difficult to write out because of the many variables, but in short, if the person's characteristic fits in with a certain variable, it is used in the equation. Then the equation is summed up and the probability is found using equation (2).

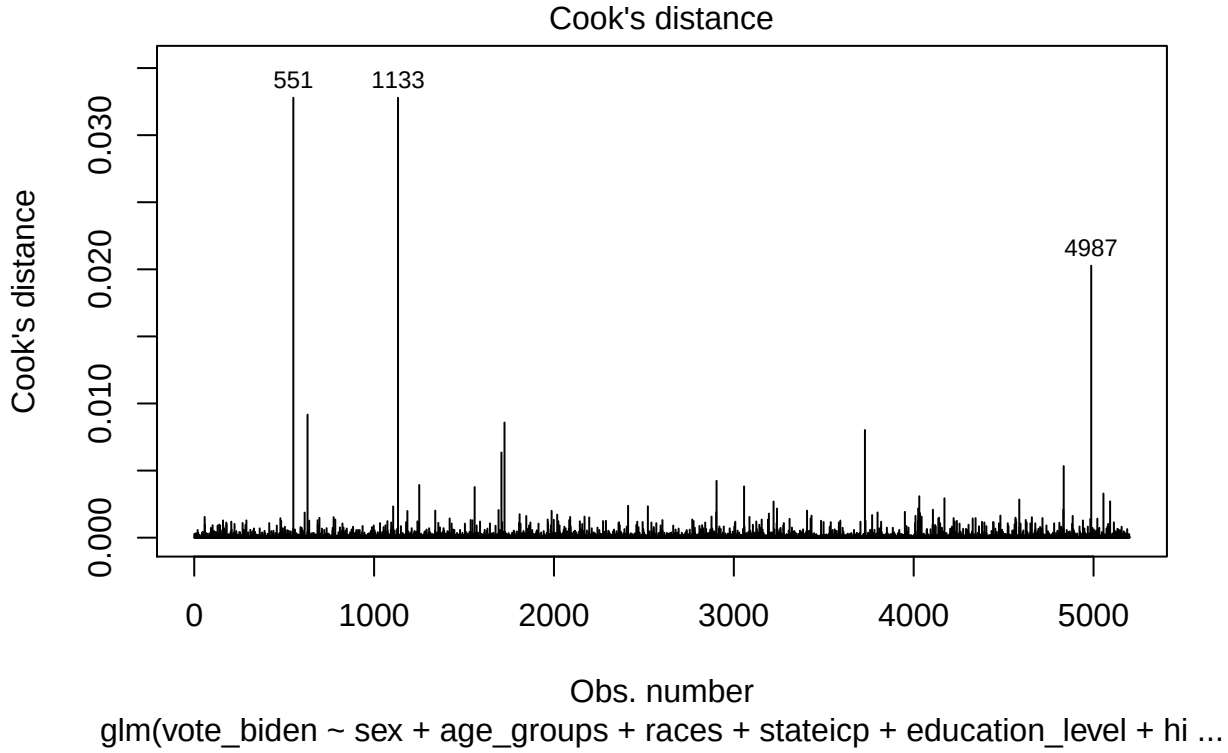

Figure 4: Cook's Distance Plot for Model

Figure 4 shows the cooks distance for observations in our polling data set. Cook's distance is useful for checking if a model is working correctly because it tells us how far a point is from the predicted value of it, telling us which points negatively impact our model. In our scenario, the distance between values would be the true value using our `glm` model vs what the `predict.glm` function returned. Figure 4 takes all observations of our polling data (Tausanovitch and Vavreck [2020]), and calculates the Cook's distance for it, showing which points can hurt the results of our model. We find that out of the 5200 obsevations, there aren't too many points that deviate from what we predict. This makes sense for our model because in the real world, there are some people who will support Donald Trump or Joe Biden, even if they don't "fit" the usual voter demographic for the candidate. Since the number of observations with large Cook's Distances are low, we can conclude that our model is fairly strong.

4 Results

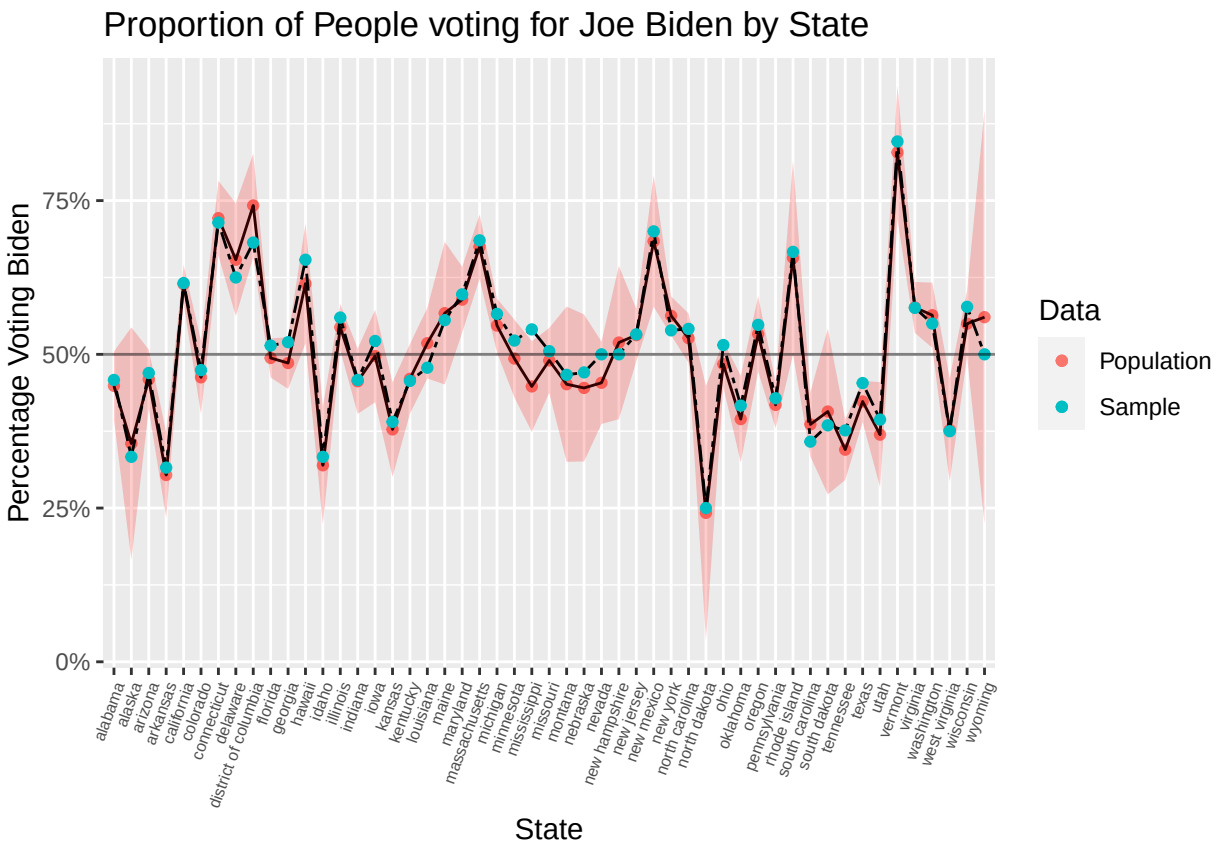

Figure 5: Proportion of each State Voting for Biden

Figure 5 shows us the proportion of respondents voting for Joe Biden broken down by state, along with the predicted proportions using MRP. We have also included the errors for each predicted value as well. We can see that the predicted proportions are fairly close to what the polling data showed. But, we also have to acknowledge the errors present, we can see that many states' errors cross over the 50% line which could be pivotal for each candidate because of the winner take all nature of the electoral college. We can see that some states like Florida and Georgia have small errors but there are some states who have very large ones like North Dakota and Wyoming.

Table 3: Joe Biden Voting Result Estimates

	Lower Estimate	Mean Estimate	Upper Estimate
Number of Colleges	191.0	260	423.0
Proportion of Vote (%)	46.1	51	55.9

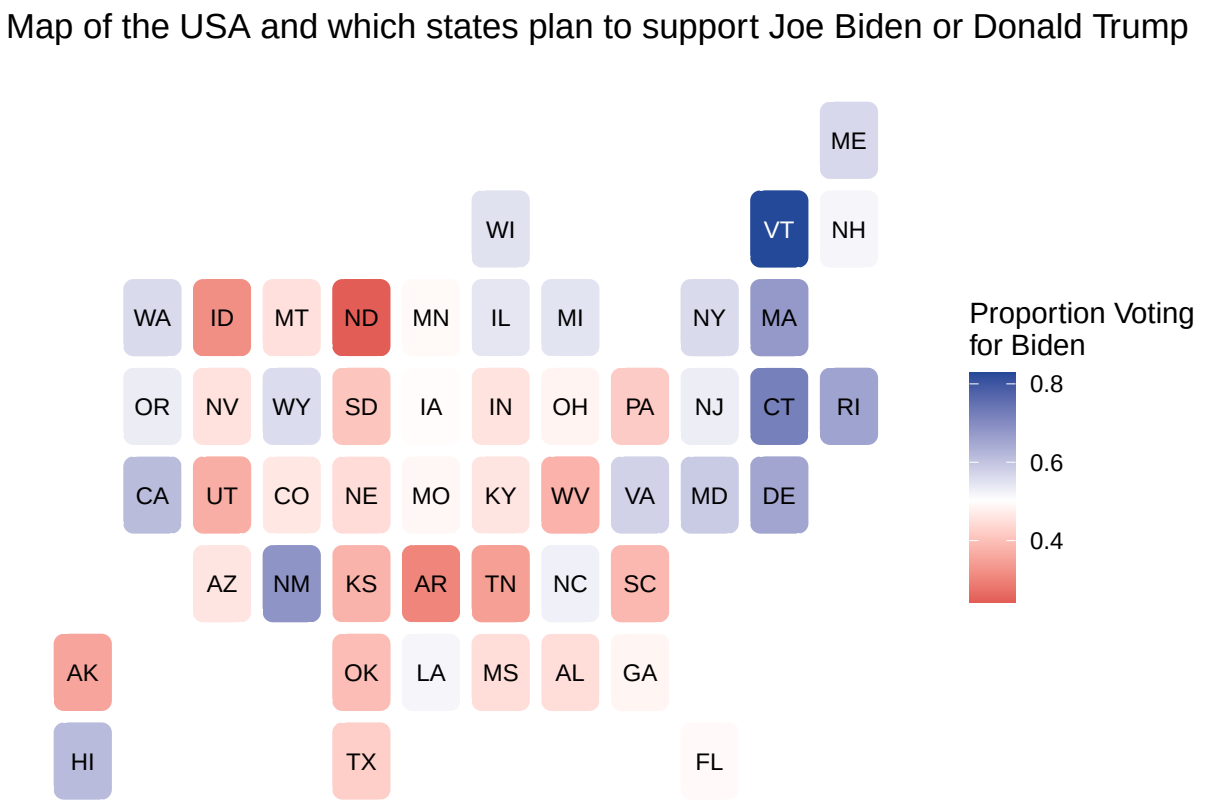

Figure 6: Proportion of Voters from Each State Voting Joe Biden

Figure 6 is an amazing view of all the states and which candidate they are leaning towards voting for using `statebins` (Rudis [2020]). We see that states like Vermont, Conneticut and California are some of the more "blue" states, meaning they plan to vote for Joe Biden, while North Dakota, Arkansas and Idaho are the "red" states, planning to vote for Donald Trump. The states that are more white in colour can be the most important for the race when it comes to deciding an actual winner through the electoral college. We see that Florida, Louisiana and New Hampshire are some of the more undecided states, and a switch in these states, can influence the election greatly.

Table 3 shows the lower, mean and upper predictions for the results of the election for Joe Biden. We see that on the lower end, Biden can expect to get 46.1% of the popular vote while only getting 191 electoral colleges. We also see that our middle estimate says Biden will get 51% of the popular vote while still losing the election by getting only 260 colleges. Lastly, on the upper estimate for Biden's results, he can get 55.9% of the popular vote while getting 423 colleges! This truly shows how close this election is, as we can see in Figure 5, many states are hovering around the 50% mark, which shows the colleges can go either way. The number of electoral colleges by state were found on the Britannica Encylopedia's website (of Encyclopaedia Britannica).

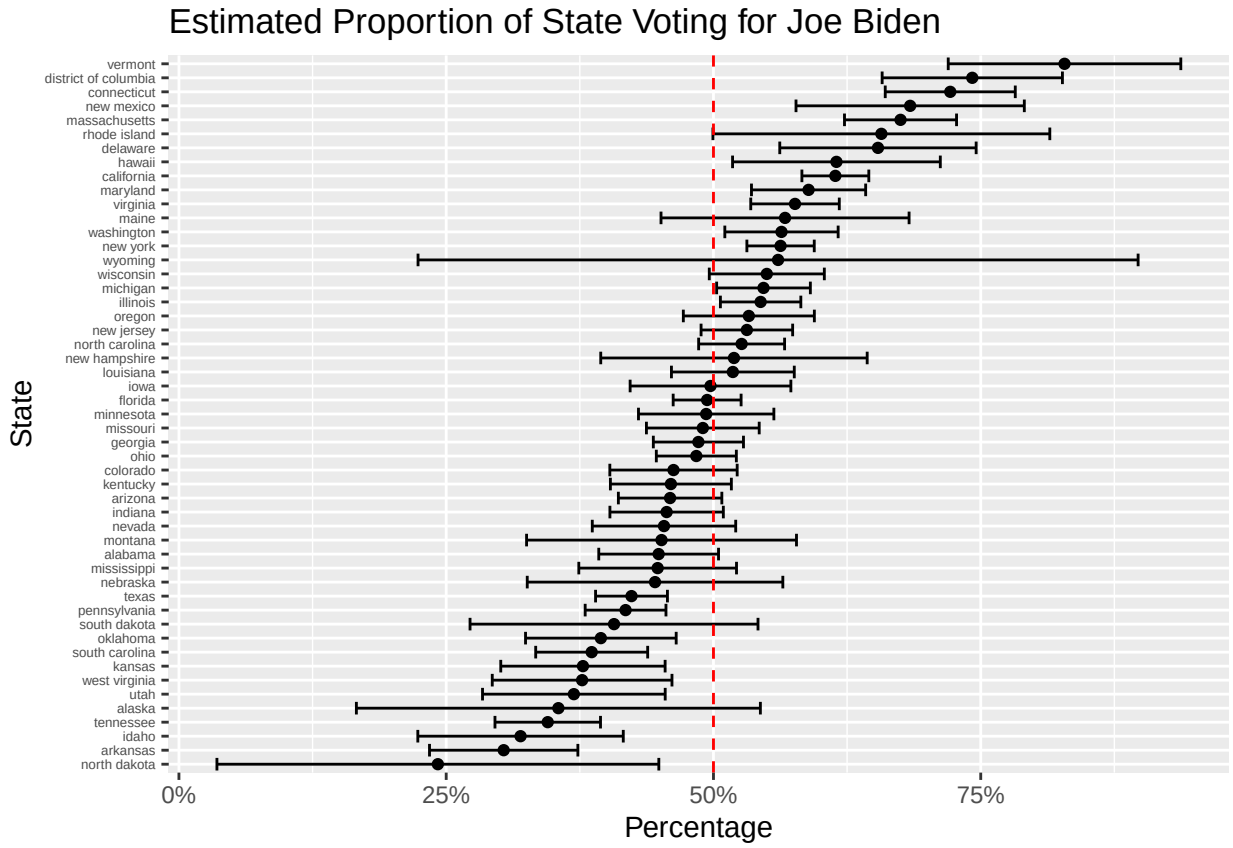

Figure 7: Proportion of Biden votes by state

Figure 7 is a different view of Figure 5, this time only focusing on the predictions for each state with the errors included. We see that many states have error bars overlapping the 50% line, showing that many states are a toss up, given the nature of the electoral college.

5 Discussion

6 Code

Code supporting this analysis can be found at: https://github.com/matthewwankiewicz/US_election_forecast

References

Lauren Kennedy and Andrew Gelman. Know your population and know your model: Using model-based regression and post-stratification to generalize findings beyond the observed sample. URL https://arxiv.org/pdf/1906.11323.pdf.

Editors of Encyclopaedia Britannica. United states electoral college votes by state. URL https://www.britannica.com/topic/United-States-Electoral-College-Votes-by-State-1787124.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Bob Rudis. *statebins: Create United States Uniform Cartogram Heatmaps*, 2020. URL https://CRAN.R-project.org/package=statebins. R package version 1.4.0.

Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas, and Matthew Sobek. *IPUMS USA: Version 10.0 [dataset].* Minneapolis, MN: IPUMS, 2020. URL https://doi.org/10.18128/D010.V10.0.

Chris Tausanovitch and Lynn Vavreck. *Democracy Fund + UCLA Nationscape.* October 10-17, 2019 (version 20200814), 2020. URL https://www.voterstudygroup.org/publication/nationscape-data-set.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2020. URL https://yihui.org/knitr/. R package version 1.30.

Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, 2020. URL https://CRAN.R-project.org/package=kableExtra. R package version 1.3.1.