A Simplified Bayesian Model for Forecasting the U.S. Presidential Election

Jonathan Larkin* and Shi Lin Sun†
(Dated: April 16, 2022)

Elections happen every four years in the United States, with the most recent one being in 2020. In this paper, we tried to construct a model to simulate the outcome of an election happening in 2020, between Donald Trump and Joe Biden. Using historical election results and polling data, we analyzed them with statistical tools such as Bayes' Inference and Markov Chain Monte Carlo to help us infer the fictitious election results. Our model favored Trump over Biden for the Electoral College votes, with both of them under 50% for the popular votes. This result was due to a small error when running an MCMC process; the initial position of the walkers were too far away from the peak of the posterior distribution. Given the expense of MCMC computations associated with our model and only a small number of iterations due to time constraints, the success of our model is still uncertain.

I. INTRODUCTION

In the United States, the president is elected every four years. Generally, the race is between two candidates, each chosen to represent the Democratic and Republican Parties. Each vote cast by eligible voters is recorded and tabulated to form the so-called *Popular Vote*. The candidate who receives the plurality of popular votes (and, barring the significant presence of a third-party candidate, the majority) is said to have won the popular vote. However, this does not necessarily mean that this candidate wins the election. Instead, the winner is determined based on the candidate who receives a majority of votes ($\geqslant 270$) in the electoral college. Each state (50 + 1 states including D.C.) is allocated a particular number of electoral college votes from a total of 538 votes, and the candidate who wins the most votes in each state is awarded all votes in the electoral college allocated to that state. While there is a possibility of a tie—where each candidate wins precisely 269 electoral votes—this has only occurred thrice in U.S. history, and all in the early 1800s[1].

The goal of this project is to propose a model which will allow for the forecasting of any U.S. presidential election, as long as the forecaster has access to sufficient, current polling data. Moreover, in theory, this model can also be adapted to other election scenarios so long as the core assumptions remain valid (see II A), and differences between the electoral college and the desired election system are accounted for. To demonstrate the capabilities and potential shortcomings of this model, we will apply it to the 2020 U.S. Presidential Election between Joe Biden and Donald Trump. This will allow us to compare our results with a variety of other models–both more complex and of comparable simplicity–and gauge the effect of our core assumptions (III).

* jonathan.larkin@mail.mcgill.ca; Physics Department, McGill University
† shi.l.sun@mail.mcgill.ca; Physics Department, McGill University

II. METHODS

Our presidential forecast includes two components: a model for determining the probability that a particular candidate will win a majority of votes in a given state (and thus their entire share of electoral college votes), and a method of using these probabilities to predict the outcome of the overall election. For the former, we have adopted an approach which utilizes Bayes' Theorem and so-called Bayesian Inference to incorporate both polling data from the relevant election cycle as well as results from historical elections to predicatively model the state-by-state outcomes. For the latter, we utilize a Monte Carlo sampling process, where points are drawn from the probability distributions generated by our Bayesian Inference.

Together, these steps allow us to forecast the likely winner of a presidential election in the electoral college, the likely winner of the popular vote, and other interesting situations such as the chance of a tie in the electoral college (occurred three times)[1], of a landslide victory/defeat by popular vote (notably between Richard Nixon and George McGovern), and of one candidate winning the popular while the other wins the electoral college vote and thereby the election (occured five times, most recently in the 2016 election between Hillary Clinton and Donald Trump)[2].

A. Model Assumptions

In order to construct our model, we made a number of clarifying assumptions. The first concerns polling data. We assumed that all polls available for a given election cycle for a particular state (in this case, conducted between 2018 and 2020) are equally valid and comparable data points. Essentially, this means that voters who have made up their mind early in the cycle (non-undecideds) will not change their mind later–potentially years later–into the cycle. Moreover, for each individual state, polls are taken to be normally distributed. Thus, when considered jointly, polling data for all 51 states is thought to be drawn from a multivariate normal distri-

Table I. Sample form of data set for 1976 - 2016 U.S. historical election outcomes. [3]

Year	State	⋯	Votes	Total Votes	⋯	Party
1976	ALABAMA	...	659710	1182850	...	DEMOCRAT
1976	ALABAMA	...	504070	1182850	...	REPUBLICAN
1976	ALABAMA	...	9198	1182850	...	OTHER
⋮	⋮	⋱	⋮	⋮	⋱	⋮
2000	WASHINGTON	...	13135	2487433	...	LIBERTARIAN
⋮	⋮	⋱	⋮	⋮	⋱	⋮

Table II. Sample form of data set for polling data taken from 2020 U.S. presidential election cycle. []

Candidate	Date	Approval	State	Cycle[a]
Joseph R. Biden	11/3/20	37.82732	Alabama	2020
Donald Trump	11/3/20	57.36126	Alabama	2020
⋮	⋮	⋮	⋮	⋮
Joseph R. Biden	3/1/20	45.54288	Wisconsin	2020
Donald Trump	3/1/20	46.467	Wisconsin	2020

[a] Information not relevant for analysis and thus not included in array.

bution. The presence of a covariance matrix in the distribution will account, at least somewhat, for correlations between the voting behavior of each state when determining and drawing from probability distributions for each state. However, forming this covariance matrix raises some practical concerns, given that the size of polling data sets (sometimes referred to here as *polling observations*) varies for each state. To "make up" for this polling deficit, our model simulates "missing" polling observations by drawing from a normal distribution with mean and variance given by the set of all unsimulated polls for that state. This multivariate normal distribution will form the likelihood PDF used in our Bayesian Inference (II C).

The second major assumption concerns data from historical elections. In our forecast, we will only be considering outcomes from U.S. elections held between 1976 and 2016 inclusive. This was chosen largely due to the availability of this particular data set, as well as separate concerns over the use of older elections in our model. Furthermore, these outcomes are presumed to be normally distributed for each state. Thus, we yet again consider the historical outcomes from each state jointly via a multivariate normal distribution, with mean and covariance determined the usual way from this data set (see II B). This multivariate normal distribution will from the prior PDF used in our Bayesian Inference (II C).

Finally, we assume that a candidate's chance of winning a state is given by a normal distribution. Again, the distribution which describes the chance of victory in all states, jointly, is given by a multivariate normal distribution, whose mean vector is obtained from the posterior of our Bayesian inference, following a Markov Chain Monte Carlo (MCMC) process. The posterior covariance is assumed to be the likelihood covariance, thereby assigning more relevance to polling data than historical election results in our model.

B. Format of Datasets

Our model utilizes two data sets in order to form the Bayesian likelihood and prior functions: the historical election results from 1976 to 2016 made available from the *MIT Election Data and Science Lab* [3] and polling data from 2020 agglomerated by *FiveThirtyEight* [4]. Both data sets were readily available in *.CSV* file format, which made ingesting into Python–our data analysis tool–more convenient. Samples from each data set in their native format are provided in Tables I and II.

For the historical data set (Table I), the only columns we utilize are Year, State, Votes, Total Votes, and Party. Although various third party candidates–noted under Party as "OTHER" and "LIBERTARIAN"–these are not considered for our analysis. By dividing Votes by Total Votes for each state in each year, we ultimately yield two multidimensional (51 x 11) arrays for each party, where each row represents a particular state, and each column a different election year.

For the polling data set, we again obtain two multidimensional arrays (251 x 51)–one for each party. However, this time, each row represents a set of 51 unique polling observations, where each column of polling observations corresponds to a particular state. There are 251 rows, because the state with the most polls conducted (Florida) had 251 total polls.

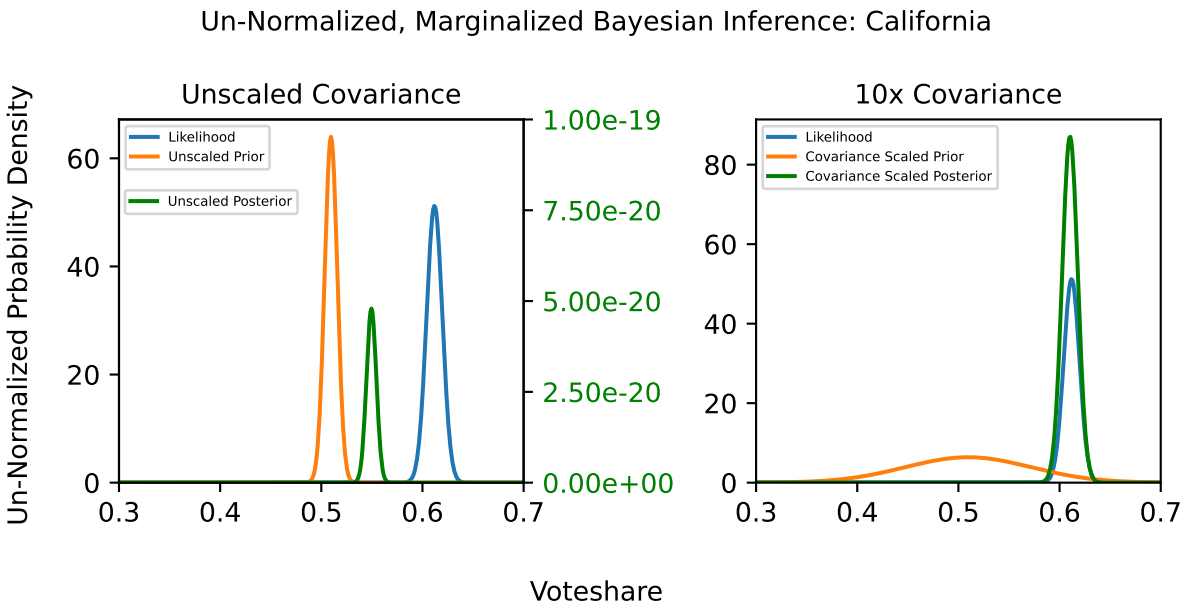

Figure 1. Marginalized distributions for the covariance. The multivariate Gaussian is marginalized over all states except California. The figure on the left is the raw distributions while on the figure on the left uses a covariance matrix with each entry multiplied by 10x in order to increase the width of the prior distributions, thereby decreasing their contribution to the resulting posterior.

C. Model

Presented below is Bayes' Theorem:

$$p(\theta|\vec{x}) \sim p(\vec{x}|\theta) \times p(\theta) \tag{1}$$

where θ is a parameter or set of parameters we are interested in modeling, and $\vec{x}$ is a set of data points from which we hope to infer θ. The functions $p(\vec{x}|\theta)$, $p(\theta)$, and $p(\theta|\vec{x})$ are known as the likelihood, prior, and posterior probability density functions (PDF), respectively. In essence, Bayes' Theorem provides a method of updating one's prior belief (the prior) about some parameter(s) by taking data (the likelihood), in order to generate an updated set of assumptions (the posterior).

In our model, the likelihood takes the form:

$$\begin{aligned} marginal\ likelihood &= p(\vec{y}_i|\vec{\mu},\mathbf{\Sigma}) \sim \mathcal{N}_{51}(\vec{y}_i;\vec{\mu},\mathbf{\Sigma}) \\ \Rightarrow likelihood &\sim \prod_{i=1}^{251} \mathcal{N}_{51}(\vec{y}_i;\vec{\mu},\mathbf{\Sigma}) \end{aligned} \tag{2}$$

where $\vec{y}_i$ is the ith row of the (251 x 51) polling observation array (see II B), $\vec{\mu}$ the mean parameter vector whose posterior distribution we are trying to compute, and $\mathbf{\Sigma}$ the polling data covariance matrix.

Our prior takes the form:

$$prior = p(\vec{\mu}) \sim \mathcal{N}_{51}(\vec{\mu};\vec{\mu}_0,\mathbf{\Sigma}_0) \tag{3}$$

where, again, $\vec{\mu}$ is the mean parameter vector, $\vec{\mu}_0$ the known mean parameter vector calculated from the historical election data, and $\mathbf{\Sigma}_0$ the covariance matrix derived from the historical election data.

Thus, we achieve a posterior of the form:

$$posterior \sim \mathcal{N}_{51}(\vec{\mu},\mathbf{\Sigma}) \tag{4}$$

Figure II B shows a sample prior, likelihood, and posterior distribution for Joe Biden in California, where the 51 dimensional multivariate normal distributions have been marginalized to 1 dimensional gaussians.

In our second simulation (10x covariance in Fig. II B), each entry of the historical election covariance matrix, $\mathbf{\Sigma}_0$, was multiplied by 10 in order to increase the width of the prior distribution, thus increasing our model's emphasis on current polling data in determining the likely outcomes for each state.

Finally, the actual outcomes in the overall election–determined by simulating the results in each of the 51 states–was determined by drawing 100,000 simulations from a Monte Carlo process. For a given run, the distribution of probabilities across the states was drawn from our multivariate normal posterior distribution (Eq. 4), and in each state a weighted coin flipped with weights given by the drawn probabilities. The popular vote results were made by multiplying the voter turnout from each state in the 2016 election by the drawn probabilities, before summing. The electoral college results were made by multiplying the binary result of each weighted coin toss by that states' designated electoral college votes,

before summing. All subsidiary results (such as landslide win/loss, etc.) were tabulated from these two processes.

III. RESULTS

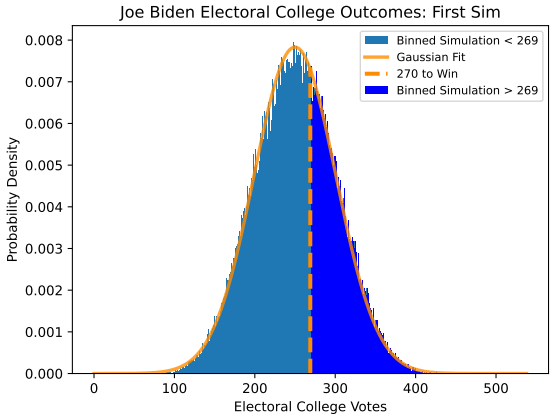

Figure 2. Histogram of the Electoral College Outcomes for Joe Biden in the first simulation. The mean of the fitted Gaussian is 250, with a standard deviation of 7.14. The chance of a tie in the electoral college is 0.737%. The chance of victory is 35.35%.

Currently, there exists a variety of models attempting to predict the 2020 electoral results. As expected, some models are more sophisticated than others and take into account several variables that can potentially influence the outcome. On the simpler end of the spectrum, the only factor used to infer the electoral results is the polling data. One such example is YouGov's model [5]. In their case, instead of using the polling data, they opted for conducting interviews. With the data collected, they performed a Multilevel Regression and Post-Stratification (MRP) in order to obtain an estimate of the result in each state. With this technique, they estimated that Biden would win the election with 364 electoral college votes and a 53.2% popular vote. Trump would get 174 electoral college votes and a 44.3% popular vote.

On the other end of the spectrum, not only do the more complicated models consider more variables, they also have a more sophisticated method of inferring the results. In the case of "The Economist", they combined two techniques in order to generate a more accurate model: the "elastic-net regularisation" and "leave-one-out cross-validation" [6]. The regularisation process simplifies the model. It makes predictor variables less impactful and completely removes variables that have little significance. To determine how much of the regularisation process they used the "leave-one-out cross-validation" technique. They separated the data set into multiple pieces, developed a model with respect to some of the pieces and tested the performance on the rest. In terms of variables, The Economist used a similar approach to Abramowitz's method, which penalizes parties that remained in power for at least two consecutive years. The economy was also taking into consideration, with data on real disposable income and the stock market. After the two steps, a single number was produced as an estimate of the probability of a party winning the election. The estimate was then fed into a Beta distribution, which served as the likelihood function of the Bayes' theorem. The prior function came from historical polling data. Naturally, the posterior distribution was the forecast that The Economist wanted. At the end, the model predicted an almost guaranteed win for Biden, with a 97% chance of winning the electoral college (3% for Trump) [7]. The predicted range of electoral college votes for Biden was 259-415, compared to a range of 123-279 for Trump.

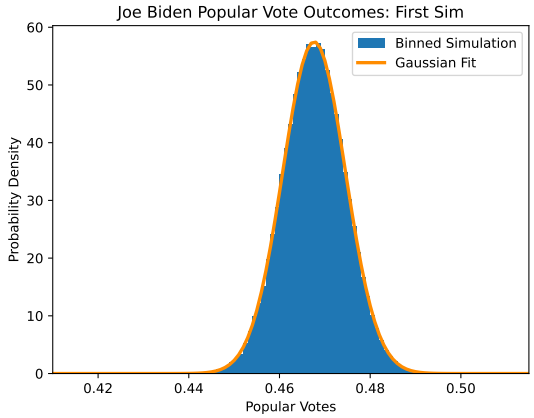

Figure 3. Historogram of the Popular Vote outcome for Joe Biden in the first simulation. The mean of the fitted Gaussian is 46.8%, with a standard deviation of 8.33. Chance of landslide win is 0%, and of a landslide loss is 0.584%.

Our results come from running two sets of election simulations: one with prior covariance determined from the historical election data (*First Sim*), and another with each entry of this matrix multiplied by 10 in an attempt to correct shortcomings clear to us after the first simulation (*Second Sim*).

From our figures, it is clear that the Democrats are most likely to lose both the Electoral College and the popular votes by a small margin. In the first simulation, Joe Biden is expected to win 250 Electoral College votes, with an error of 7.14. On the second simulation, the number goes to 252 with an error of 7.12. On the other hand, Trump is favored to win the Electoral College, with 288 votes on the first simulation and 285 on the second. The error is 7.13 for both simulations. The graphs can be found in the Appendix section. Regarding the popular votes, both candidate are expected to be tied at 47%, with an identical error of 8.34%.

The main reason why our results are skewed is found in the MCMC step. When running our simulation, we

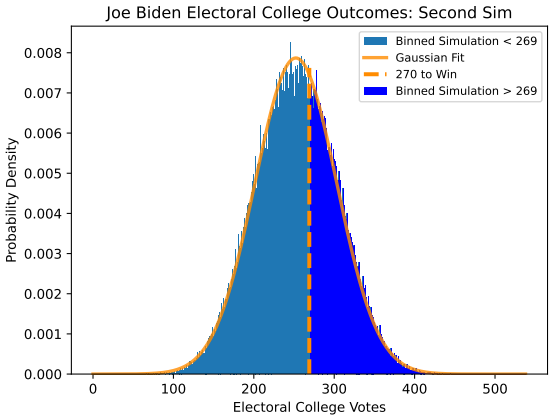

Figure 4. Histogram of the Electoral College Outcomes for Joe Biden in the second simulation. The mean of the fitted Gaussian is 252, with a standard deviation of 7.12. The chance of a tie in the electoral college is 0.718%. The chance of victory is 37.12%.

made a mistake when choosing the initial positions of the walkers; they are too far away from the peak of the posterior function. 9000 iterations is not enough for the walkers to find the right value, which leads to the MCMC spitting out the wrong values.

It is clear another computation is in order–in particular where the initial position lies somewhere close to the peaks of the marginalized likelihood distributions–which would allow us to better test the success/failure of our model.

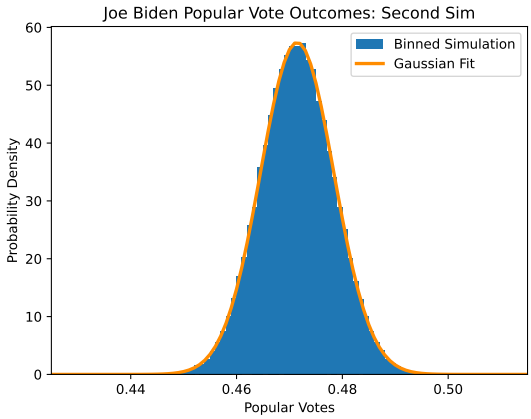

Figure 5. Histogram of the Popular Vote outcome for Joe Biden in the second simulation. The mean of the fitted Gaussian is 47.1%, with a standard deviation of 8.34.

IV. CONCLUSION

The success of our model of forecasting presidential election results was inconclusive. The main reason our model did not work well is when performing the MCMC, the given initial position of the walkers was too far away from the peak of the posterior distribution. On top of that, we did not run enough iterations for the MCMC to get to the right value, which skewed our results. Our model favored a Trump win, which clearly did not happen in the 2020 election. However, given enough time to run the MCMC again with more reasonable initial positions for the walkers, we believe that our model would at least be comparable to the already existing models. Certainly, however, it is clear that our model is quite computationally expensive–one potential downside if speedy forecasting is in order.

Although this is a good start given the scope of our project, there are more factors that we could consider in order to reflect the reality better. As a future project, we could add a weighting system that emphasizes the more recent data points. The model would also certainly benefit from the addition of non-political factors such as the economy, since they do often times have a lot of influence on the electoral results.

V. APPENDIX

This section contains the histogram figures of the electoral outcome for the Republican party. The mean and the standard deviation of the fitted Gaussian are in the corresponding figure description.

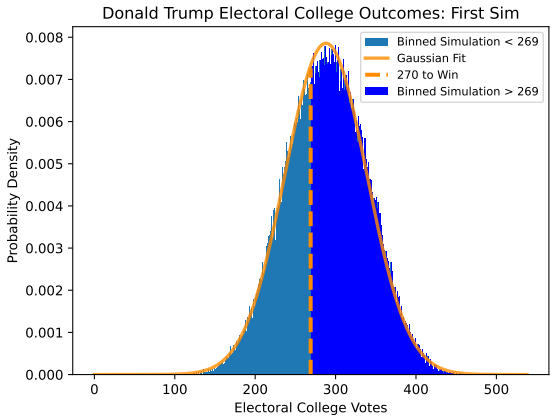

Figure 6. Histogram of the Electoral College Outcomes for Donald Trump in the first simulation. The mean of the fitted Gaussian is 288, with a standard deviation of 7.13.

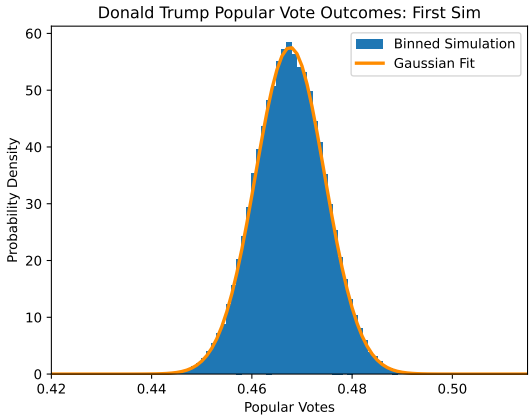

Figure 7. Historogram of the Popular Vote outcome for Donald Trump in the first simulation. The mean of the fitted Gaussian is 46.8%, with a standard deviation of 8.33.

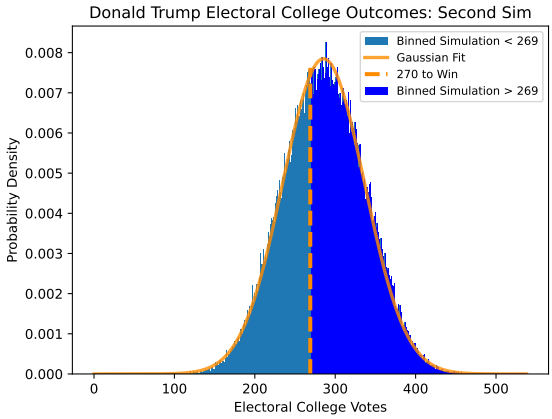

Figure 8. Histogram of the Electoral College Outcomes for Donald Trump in the first simulation. The mean of the fitted Gaussian is 285, with a standard deviation of 7.13.

[1] D. Roos, What happens if there's a tie in a us presidential election? (2020).
[2] D. Roos, 5 presidents who lost the popular vote but won the election (2020).
[3] MIT Election Data And Science Lab, U.S. President 1976–2020 (2017).
[4] FiveThirtyEight, Biden is favored to win the election (2022).
[5] YouGovAmerica, 2020 presidential election model (2022).
[6] T. Economist, How the economist presidential forecast works (2022).
[7] T. Economist, Forecasting the us elections (2022).
[8] T. Economist, Meet our us 2020 election-forecasting model (2022).
[9] B. Walker, Us 2020 presidential election forecast model: will donald trump or joe biden win? (2022).
[10] B. Y. Mark Zandi, Dan White, 2020 presidential election model (2019).
[11] Bloomberg, 2020 presidential election results (2022).
[12] Bloomberg, Explaining the bloomberg news 2020 election turnout model (2022).

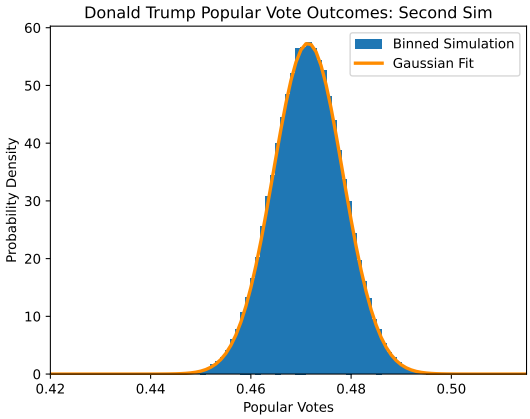

Figure 9. Historogram of the Popular Vote outcome for Donald Trump in the first simulation. The mean of the fitted Gaussian is 47.1%, with a standard deviation 8.34.