Supplementary Information for

Electoral College Bias and the 2020 Presidential Election

Robert S. Erikson, Karl Sigman, Linan Yao

Robert S. Erikson.
E-mail: rse14@columbia.edu

This PDF file includes:
- Supplementary text
- Figs. S1 to S2
- Tables S1 to S2
The statistical model for our simulations relies on the following data generating function (DGF) for generating sample simulated values of state presidential outcomes. The baseline model is an autoregressive model of order 1 (AR1), based on the OLS equation where the data are state two-party vote divisions \(Y_t\) in all presidential elections 1984-2016 as the dependent variable and state two-party vote divisions in the prior election \(Y_{t-4}\) plus year dummies as the independent variables.

The key to helping us use it for predicting what would happen in the Electoral College in year \(t\) if the national popular vote were a particular value of our choice is that we actually do know what the popular vote was for each election year \(t \leq 2016\). So we use that to adjust the \(\alpha\) in Equation 1 for each sample-path so that we get the correct prediction of state votes that would lead to this known popular vote. We illustrate this precisely next.

We re-write Equation 1 in the main text by introducing states, indexed by \(1 \leq i \leq 51\), and simplify notation \(D_t(i) = \text{Dem}(t,i)\) to obtain

\[
D_t(i) = \alpha + \beta D_{t-4}(i) + U_t(i), \tag{S1}
\]

where the \(\{U_t(i) : 1 \leq i \leq 51\}\) are simulated independent copies of \(u\). Next we introduce the known weights \(\{W_t(i) : 1 \leq i \leq 51\}\), \(( \sum_{i=1}^{51} W_t(i) = 1)\) that yield the (known) national popular vote in year \(t\), \(NP(t)\), from the state votes to force the desired equality

\[
NP(t) = \sum_{i=1}^{51} W_t(i)D_t(i)
\]

\[
= \alpha \sum_{i=1}^{51} W_t(i) + \beta \sum_{i=1}^{51} W_t(i)D_{t-4}(i) + \sum_{i=1}^{51} W_t(i)U_t(i)
\]

\[
= \alpha + \beta \sum_{i=1}^{51} W_t(i)D_{t-4}(i) + \sum_{i=1}^{51} W_t(i)U_t(i).
\]

Then we solve for the value of \(\alpha\), denoted by \(\hat{\alpha}\) for which the equality does hold:

\[
\hat{\alpha} = NP(t) - \left( \beta \sum_{i=1}^{51} W_t(i)D_{t-4}(i) + \sum_{i=1}^{51} W_t(i)U_t(i) \right). \tag{S2}
\]

Then we extract back the desired values of \(\{D_t(i) : 1 \leq i \leq 51\}\) by utilizing this value of \(\alpha\) in Equation S1

\[
D_t(i) = \hat{\alpha} + \beta D_{t-4}(i) + U_t(i). \tag{S3}
\]

Using these 51 state vote values then allows us to compute a predicted copy of the Electoral College vote. That gives us one sample-path of the election. We proceed to do this independently 10,000 times so as to obtain 10,000 independent sample paths of the election.

This model implies an underlying process that does not vary by election year rather than with unique parameters for each election. It implies also that history does not matter, other than the previous election. Simulating from this model also implies that the estimated parameters are known with certainty rather than with error.

Obviously we could make modeling more complicated. We could take into account state voting in Year \(t - 8\). We could use the estimate of \(\beta\) from the most recent election rather than from the pooled data. We could incorporate the sampling error of \(\beta\) rather than treat the estimated \(\beta\) as true. We can consider variation in the estimated standard deviation of the errors, sigma (\(\sigma\)), due to other factors besides the states’ most recent presidential division mattering. The potential advantage of any of these adaptations would be greater precision for our simulation. A potential danger would be the unwitting incorporation of an overfit model on a shaky empirical foundation.

In this appendix, we show that our estimates for 2020 are robust to using four alternative models.

- In **Model 1**, we model the state Democratic vote as a function of the Democratic vote in the previous election in year \(t - 4\), based on the pooled equation for 1984-2016. For the prediction of each election from 1984 to 2020, we modify the data generating function by eliminating data from the election at hand. We use the rest of the observations pooled together. This the baseline specification that we use for the body of this paper.

- In **Model 2**, we model the state Democratic vote as a function of the Democratic vote in the previous 2 elections in years \(t - 4\) and \(t - 8\), based on the pooled equation for 1988-2016. For the prediction of each election from 1984 to 2020, we modify the data generating function by eliminating data from the election at hand. We use the rest of the observations pooled together.

- In **Model 3**, we model the state Democratic vote as a function of the Democratic vote in year \(t - 4\), based on the regression equation for the previous election at \(t - 4\) in Table S1b. In other words we use the equation for one election to model the next election rather than relying on pooled data.
In *Model 4*, we incorporate the sampling error of $\beta$ rather than treat the estimated $\beta$ as true. We base this method on results from *Model 1*, but we add randomly-drawn errors to its $\beta$ according to its standard error. We scale the states’ Democratic votes in 2016 by subtracting their mean and we generate the predicted vote in 2020 using the scaled 2016 vote times the new $\beta$ plus an intercept that makes the Democratic vote 51.1% nationally. As usual, we simulate this process 10,000 times, each time with state-specific shocks that sum to the national average of 51.1%.

As background, Table S1 shows the OLS equations for the alternative models, *Model 1* - *Model 4*. In Table A1a, we present OLS results by pooling all observations (with year fixed effects as dummy variables), with independent variables as either lagged one election year (*Model 1*, *Model 4*) or lagged both one and two election years (*Model 2*). In Table A1b, we model each year separately, using one election to predict the next, (*Model 3*).

Table S1a shows that the importance of the vote two elections past pales to insignificance as a predictor of state voting, once the vote in the most recent election is in the equation. There is little reason to consider the vote in 2012 when predicting the vote in 2020. Table S1b shows, however, that the equation does vary by year both in terms of $\beta$ and $\sigma$. As a solution, we could model this variation, considering the mean $\beta$, its observed variance over the sampled elections, and also modeling its standard error.

Next, we run simulations of 2020 via this set of alternative model specifications. Here, we set the projected 2020 national popular vote to be a constant, 51.1% Democratic, to match the Democrat’s performance in 2016 and simulate the Electoral College. Figure S1 presents the predicted 2020 Electoral College outcomes, generated using our four different model specifications, *Model 1* - *Model 4*.

As we can see in Figure S1, the simulations based on four different model specifications produces similar distributions of predicted 2020 Electoral College outcomes. The probability of the Democrat winning is close across four models as well, which is 46.14% for *Model 1*, 43.10% for *Model 2*, 44.88% for *Model 3*, 44.35% for *Model 4*.

In Table S2, we present the mean and the standard deviation of the predicted 2020 Electoral College outcomes generated from each model. We can see that both the mean and the standard deviation are very consistent across models.

Finally, we can also consider the variability of $\sigma$, the standard deviation of the error in predicting the state vote from the state vote in the previous election. As one extreme test, we double the $\sigma$ for 2020 from *Model 1*. As the range of simulated Electoral Vote tallies widens, Biden’s probability of winning, given a national vote of 51.10%, becomes 58.45%. As the test at the opposite extreme, we can cut $\sigma$ in half. As the range of Electoral Vote outcomes constricts, Biden’s probability of winning, given the same national vote, is 29.19%. We present the Electoral College results given a national vote of 51.10% for Biden simulated with varying $\sigma$ in Figure S2. The probability that Biden wins, given a national vote of 51.10%, is a non-decreasing function of $\sigma$, as is illustrated here by doubling and halving $\sigma$. In fact, in the case when NP(2020) is set to 51.10%, and no shocks (i.e., $\sigma$ tends to 0), then the outcome (using the Equation S2–S4 methods) is deterministically yielding a repeat of 2016; Biden loses with certainty. In Figure S3, we show results from an expanded range of national popular votes. A bigger $\sigma$ also diminishes the asymmetry of the predicted partisan advantage in the Electoral College system.
### Table S1. Regression results

#### (a) Regression results using pooled data for election years 1984-2012 but not the year in question

| Year to Predict | 2020 | 2016 | 2012 | 2008 | 2004 |
|-----------------|------|------|------|------|------|
|                 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 |
| Lag t-4         | 0.9755 | 0.9984 | 0.9666 | 0.9924 | 0.9629 | 1.0242 | 0.9709 | 0.9824 | 0.9801 | 1.0321 |
|                 | (0.0164) | (0.0475) | (0.0178) | (0.0482) | (0.0183) | (0.0533) | (0.0174) | (0.0478) | (0.0182) | (0.0525) |
| Lag t-8         | 0.0034 | 0.0051 | -0.0353 | 0.0205 | -0.0224 |
|                 | (0.0489) | (0.0492) | (0.0545) | (0.0498) | (0.053) |
| R Squared       | 0.9022 | 0.9097 | 0.8995 | 0.9074 | 0.8933 | 0.9011 | 0.9024 | 0.9124 | 0.8967 | 0.9039 |
| Adj R Squared   | 0.9002 | 0.9076 | 0.8975 | 0.9052 | 0.8911 | 0.8978 | 0.9005 | 0.91 | 0.8946 | 0.9017 |
| Sigma           | 0.0348 | 0.0327 | 0.0345 | 0.0321 | 0.0359 | 0.0338 | 0.0344 | 0.032 | 0.0359 | 0.0337 |

#### (b) Regression results using one election to model the next (Model 3)

| Year to Predict | 2000 | 1996 | 1992 | 1988 | 1984 |
|-----------------|------|------|------|------|------|
|                 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 |
| Lag t-4         | 0.9599 | 0.9802 | 0.9861 | 0.9898 | 0.9808 | 1.0642 | 0.9876 | 0.9011 | 1.0015 |
|                 | (0.0173) | (0.0504) | (0.0172) | (0.0515) | (0.0168) | (0.0484) | (0.0175) | (0.0545) | (0.0164) |
| Lag t-8         | 0.0073 | 0.0062 | -0.0604 | 0.1296 |
|                 | (0.0513) | (0.0531) | (0.0505) | (0.0573) |
| R Squared       | 0.9026 | 0.9102 | 0.9019 | 0.9113 | 0.9083 | 0.9192 | 0.9042 | 0.9138 | 0.9097 |
| Adj R Squared   | 0.9006 | 0.9081 | 0.9 | 0.9093 | 0.9064 | 0.9173 | 0.9023 | 0.9118 | 0.9079 |
| Sigma           | 0.0348 | 0.0326 | 0.0349 | 0.0327 | 0.0343 | 0.0317 | 0.0353 | 0.0326 | 0.0327 |

1. The results shown for 2020, Model 1, is from Equation 1.
2. We ignore the intercepts and the year fixed effects in this table. The intercepts are ignored because they are essentially arbitrary given the year fixed effects.

---

1. For model 3 applied to predict 2020, the 2016 equation is used.
Table S2. Mean and standard deviation of 2020 Electoral College outcomes via four models

|                  | Model 1     | Model 2     | Model 3     | Model 4     |
|------------------|-------------|-------------|-------------|-------------|
| Mean             | 267.7561    | 265.9649    | 266.9099    | 267.1294    |
| Standard Deviation | 23.7872    | 23.2733    | 23.8343    | 24.0087    |

1 The outcomes are measured as Democratic electoral votes, assuming a 2020 popular vote of 51.1% Democratic, 48.9% Republican, the same as in 2016.
Fig. S1. Simulated Electoral votes, 2020, with varying $\sigma$. Here, we assume that the popular vote is 51.1% Democrat and 48.9% Republican in 2020.
Fig. S2. Probable Electoral College winner, 2020, as function of the national (two-party) popular vote, with varying $\sigma$. 

(a) Halve the $\sigma$

(b) Original $\sigma$

(c) Double the $\sigma$