# How The Washington Post Estimates Outstanding Votes for the 2020 Presidential Election

John Cherian, Lenny Bronner

October 2020

## 1 Introduction

On November 3rd, 2020, Americans will be closely watching The Washington Post as results pour in from across the country.[1] Making sense of election results as they arrive is no simple task. Careful observers of previous elections recognize that early leads do not necessarily translate to ultimate victories. The same patterns often seem to recur: In some states, a Democratic-leaning early vote is buried by a tidal wave of Republican election day votes. In others, a Republican lead is slowly whittled down as slower-counting and Democratic-leaning metropolitan counties report their tallies. To address this, The Washington Post, in collaboration with our friends at Decision Desk HQ/0ptimus Analytics, has developed a election night model that will make sense of live vote totals by delivering *reliable* inferences regarding what those early votes imply about the final results. For this election in particular, we think that our model is vitally important. It bridges the gap between what the results are showing at any given moment and the likely true underlying result.

Based on the votes reported, our model will estimate three quantities for each state: The overall turnout, the number of Democratic votes, and the number of Republican votes. For each of these outputs, we want to display the uncertainty in the prediction via a prediction interval. Accurate estimation of these intervals is critically important. An overly confident forecast could mislead readers relying on these figures, while being too conservative runs the risk of not being honest about what we know.

We'd like to provide a technical overview of how and why this model can provide forecasts and prediction intervals for unseen votes in each state.

## 2 A Model, Isn't It?[2]

Our model for the 2020 election is defined at the county level. We rely on reported results from a partial set of counties in the United States to infer results from unseen counties. While regular readers of The Washington Post's Engineering Blog[3] might recall that previous election models have modeled voters at the precinct level, we do not have access to the data that would enable a similarly granular model of voting for the entire United States. And, unlike other prominent election forecasting models, our model does not rely on polls. In fact, our model takes no inputs besides previous election results and the demographic makeup of the United States.

Additionally, some areas of the United States return results at the level of townships or state legislative districts. Others return county-level results but the reporting units are called something other than counties, e.g., in Louisiana, results are returned by parishes. For our purposes, we will refer to all of these reporting units as "counties" even though they are a mix of reporting unit levels. Critically, they are larger than precincts but smaller than states, and that is sufficient to unify them for our purposes.

[1] Due to the popularity of mail-in voting, "trickle in" may be a more appropriate choice of words.

[2] We are running out of model-related puns.

[3] We appreciate each and every one of you.

We'll now establish some notation that we'll use for the remainder of this post.

$$T_i^{(YY)} := \text{Number of votes cast in the 20YY election in county } i$$
$$D_i^{(YY)} := \text{Number of votes cast in the 20YY election in county } i \text{ for the Democratic candidate}$$
$$R_i^{(YY)} := \text{Number of votes cast in the 20YY election in county } i \text{ for the Republican candidate}$$
$$X_i := \text{Vector of demographic covariates for county } i$$

For the remainder of this write up, we will focus on the problem of predicting turnout or expressed in the notation above $\{T_i^{(20)}\}_{i=1}^n$. We'll assume also that counties are indexed by the order in which they arrive, i.e. $i = 1$ denotes the first county to report fully. The approach we describe below is identical for the other two variables we are interested in forecasting; just replace $T$ with $D$ and $R$.

On election night, before we see any results, our best guess for the 2020 election turnout in any given county is the turnout in that county from 2016. You may think that we should be able to do better than just using the previous election's turnout to predict turnout this year. There is evidence abound that turnout will be higher this year than in 2016, so why not scale 2016 turnout by some amount to account for that? In short, we did not find a good systematic way of doing this across all counties in the United States. Do we expect turnout to be up 5%? 7%? 10%? or maybe even more than that? Back-testing using elections since 1992 shows us that using the last presidential election turnout is a good place to start.

Since our model forecasts turnout once results start coming in, the quantity we are interested in predicting on election night can be expressed as the difference between the 2020 turnout that we have observed and the 2016 turnout. We'll refer to this as $r_i := T_i^{(20)} - T_i^{(16)}$. Then, the vector $r$, which has $n$ entries — one for each county in the United States that has yet to report — is the quantity we are interested in forecasting.

This approach does require that some of the 4,000+ counties that we are tracking have reported 100% of their votes. Having some number of counties at 100% reporting is the only way to make sure that the comparison to the 2016 election is fair. Given this constraint — and the fact that our model needs a few dozen reporting units to be fully reported to start running, and a few hundred more for the uncertainty estimates be well calibrated — it is possible though not likely that we will not be able to activate our model on election night.

Our model is premised on the assumption that partially observing $r$ counties narrows down the range of possibilities of the unobserved entries of that vector. In other words, we assume that the county-by-county differences between the 2020 and 2016 turnout are correlated. If we assumed every county's change from 2016 was independent, observing results from any set of counties would not help us predict turnout in counties that have yet to report any votes. Luckily for us, election results are not quite so arbitrary. For example, the change between 2016 and 2020 turnout in Detroit, MI is likely to be fairly similar to the change between 2016 and 2020 turnout in Milwaukee, WI. Our goal then is to build a model that uses final results from one county to help us understand what's happening in other demographically similar counties throughout America.

# 3 Close Encounters of an Inferred Kind

Before we dive into our model, we would like to discuss some related work.

## 3.1 Our 2019 Virginia Model

Quantitatively estimating the correlation structure that determines how county turnouts are interrelated is a difficult task. Some of you might remember that The Washington Post has done something like this before. During the 2019 general election in Virginia, we also estimated the total number of votes cast. We did so by assuming that the turnout in the election came from a normal distribution with a covariance matrix estimated using previous election results. Though the approach we will take for the 2020 election still relies on the assumption that certain counties are likely to have similar turnout, we have changed our approach

from 2019 somewhat because the model we built for Virginia was computationally infeasible to scale to the level of a national election.

To predict the general election results on November 3rd, 2020 using our 2019 Virginia model, we would have to quantify the relationship between every pair of counties in the United States.[4] For those of you keeping track at home, that's over 20,000,000 parameters to estimate! To make matters worse, we only have a handful of elections to estimate these correlations from. Statisticians call this a $p \gg n$ problem as the number of correlation parameters ($p$) we have to fit greatly exceeds the number of data points ($n$). Directly estimating a 20,000,000 parameter covariance matrix from the historical data will not work.

The final nail in the coffin for our 2019 Virginia model was our observation that historical correlations are less persistent than one might imagine. Two counties that exhibit a similar change in turnout between the 2008 and 2012 elections are only slightly more likely to have similar changes in turnout between 2012 and 2016. This observation implies that even if one was somehow able to estimate historical correlations between counties, they may not be particularly predictive for the upcoming election.

### 3.2 Mining for Correlations

A county-level map of vote share says a lot about the politics of America. Vast seas of red rural and exurban counties surround blue islands of metropolitan areas. Critically, it should be apparent that the map is not colored at random. If one was presented with a map where one county had not been colored with its true outcome, it wouldn't take a lot of thinking to come up with a reasonable guess — one could just look at how the neighboring counties voted!

We can make our "use the county's neighbors to guess the county's outcome" heuristic rigorous by applying a method known as kriging — AKA "Gaussian Process regression." Kriging was originally invented to answer the question of how much mineral content might be found in a particular soil layer given measurements at neighboring samples. Here we replace "mineral content" with the change in turnout, $r_i$, and "soil layer" with county. The math behind kriging shows that the "use a weighted combination of the neighbors" approach is optimal when correlations can be estimated as a function of the *distance* between the two counties. The weights we use on each neighbor's outcome are a function of the estimated correlation with that neighbor. Prior published work on night-of election forecasting relies on this principle — Pavia, Larraz, and Montero used a type of kriging to predict vote shares and totals in Spanish regional elections. [5]

We tried this approach with "demographically aware" kriging, that is basing the "distance" between two counties not on physical distance but on demographic similarity. This approach did not work as well as we hoped. Beyond predictive accuracy, we were also dissatisfied with our ability to quantify our uncertainty about the kriging predictions. Under the hood, kriging assumes that voter turnout follows a jointly normal distribution. The symmetric prediction intervals that result often appeared to poorly cover the true outcome in our tests. More generally, we are skeptical of making parametric assumptions regarding the distribution of our random vector $r$.

## 4 Our 2020 General Election Model

While the previous work we've discussed tries to estimate correlations between counties, we sidestep that difficult problem by building a model to directly predict $r$. Namely, we assume the following model holds:

$$\tilde{r}_i = f_\beta(X_i) + \epsilon_i \tag{1}$$

$\tilde{r}_i$ is the normalized vector of changes in turnout (i.e. $r_i / T_i^{(2016)}$). $X_i$ is the vector of normalized demographic covariates. $\beta$ are the parameters of our regression model, and $\epsilon_i$ is an unknown error term. Unlike the previous modeling approaches we described, we make no parametric assumptions about the distribution of any quantities here other than that the $(X_i, r_i)$ should be "exchangeable" (more on this later).

[4] Recall: In New England results are reported on township levels, in Louisiana on a Parish level and in Alaska results are tallied at the level of state legislative districts.

[5] https://www.tandfonline.com/doi/abs/10.1198/016214507000001427

By assuming that turnout can be modeled as a function of the demographic covariates, the correlations are implicitly represented by the parameters that we estimate. For instance, if we use the percentage of the population that is Hispanic to predict $\tilde{r}$, we are implicitly claiming that counties with similar Hispanic populations are more likely to be correlated. And the same is true for any other demographic factor that we include in our model.

For those of you who want a more rigorous understanding of how our model works, we'd suggest reading the paper by Romano et al. that introduces the method being used here.[6] Here we present a slightly less rigorous discussion of the methods we used and the heuristics we developed.

## 4.1 Quantile Regression

We use quantile regression to estimate the parameters of our model. This allows us to directly estimate the median as well as lower and upper bounds of our prediction interval without making any distributional assumptions.

Let's start off by describing a kind of regression that everyone knows and loves, least-squares (LS). We find the least-squares regression model $f_\beta(\cdot)$ by choosing parameters $\beta$ such that $\sum_{i=1}^{n}(y_i - f_\beta(x_i))^2$ is minimized. What's perhaps less well known is that the resulting model $f_\beta(x_i)$ also represents our best guess for the mean of $y_i$ conditioned on $x_i$. A small change to the least-squares objective allows us to instead estimate the conditional *median* of $y_i$ given $x_i$. Namely, instead of optimizing the squared error expression we describe above, we select $\beta$ to minimize $\sum_{i=1}^{n}|y_i - f_\beta(x_i)|$.

We prefer the estimated median to the more standard least-squares solution (i.e. the estimated mean) because we are concerned about overfitting to outlier counties. We fully anticipate that there exist certain counties in which local circumstances such as unusual vote-by-mail laws or excitement about downballot races drive unusual voting patterns. These unique counties should not exert excessive influence on our predictions elsewhere though. Though the choice of model in Eq. 1 is linear, nothing about quantile regression limits us to simple linear models. In other problem settings, we might choose $f_\beta(x)$ to be a deep neural network or a decision tree. But since we would like to produce useful predictions even when only a few counties are reporting results, it is imperative that we use a choice of $f_\beta(x)$ that requires minimal training data.

As the name might suggest, quantile regression is useful for predicting quantiles other than the median. The key idea is that if minimizing the absolute value loss returns the median (aka the 0.5-quantile), then minimizing a rotated version of the absolute value loss will yield estimates of any other quantile. If we rotate the function enough we could get the 0.95-quantile and the 0.05-quantile of our prediction, giving us a 90% prediction interval.

Although we'd love to leave the model here and be done, we can't really fit any quantile regression model to our county data set and conclude that we now know the true quantiles of the turnout shift $r$ conditioned on each county's demographics $X$. This statement is only valid when our model relating $X$ to $r$ is correct, i.e., turnout shifts are actually a function of the racial, age, gender, education and income data we have obtained. More crucially, this statement is only valid when we have collected infinitely many data points. Back in the real world, we know that our model is just an approximation to the complicated real world process that results in a particular observed turnout. The outputs of our quantile regression are ultimately just educated guesses.

This is a real problem for our prediction interval. If our model isn't good enough or we don't have enough data, then an interval stretching from the estimated 0.05-quantile to the 0.95-quantile for $r_i$ is unlikely to contain the true turnout shift 90% of the time. When trying to accurately illustrate the uncertainty of our model, an error such as this can be catastrophic. To fix this problem, we're going to turn to a relatively new idea in statistics: Conformal prediction.

[6] http://papers.nips.cc/paper/8613-conformalized-quantile-regression.pdf

## 4.2 Conformal Prediction

For a detailed introduction to conformal prediction, we'd encourage you to check out the excellent Journal of Machine Learning Research tutorial by Shafer and Vovk.[7] We won't explain why conformal prediction works here. Instead, we will focus on the assumptions and limitations of conformal methods.

Conformal methods are best thought of as wrappers that you can put around any black box prediction method to produce valid prediction intervals. The method we use, "conformalized quantile regression," wraps a quantile regression model that outputs a guess for the bounds of a $100 \cdot \alpha\%$ prediction interval for $\tilde{r}_i$. The conformalized quantile regression then produces a prediction interval that is guaranteed to include the true $\tilde{r}_j$ with probability $\alpha$ for unseen $(X_j, \tilde{r}_j)$.

As we have mentioned, one of the major benefits of our model is that we avoid making distributional assumptions about the data generating process. Quantile regression and conformalization work without them. There is however one assumption that we cannot do without: Our data must be *exchangeable*. In particular, conformalized quantile regression assumes that the $(X_1, \tilde{r}_1), \ldots, (X_n, \tilde{r}_n)$ are exchangeable samples.

What does it mean for random variables to be exchangeable? Intuitively, this means that if we observe $N$ samples, $X_1, \ldots, X_N$, any ordering of those samples is equally likely. More rigorously, the joint distribution of $X_1, \ldots, X_N$ is invariant to permutations of the indices. Let $\{T_1, T_2, T_3\}$ be three random variables. Then, the $T_i$ are exchangeable if shuffling the values of the 3 random variables leaves the joint distribution is unchanged. For instance,

$$P(T_1 = t_1, T_2 = t_2, T_3 = t_3) = P(T_1 = t_2, T_2 = t_1, T_3 = t_3).$$

You may have seen a related statistical assumption before, for example, that the random variables are independent and identically distributed (i.i.d.). It is easy to see that i.i.d. random variables are exchangeable. But exchangeability is a weak enough assumption to cover cases that are far from i.i.d. Imagine we have an urn filled with balls labeled from 1 to 10. Let's define $S_1, \ldots, S_{10}$ as the labels of the balls we select by removing balls randomly one-by-one from this urn. Then, $S_1, \ldots, S_{10}$ are exchangeable! Any ordering of the balls is equally likely.

However, the balls-in-an-urn process isn't i.i.d. Drawing $S_{10}$ conditioned on $S_1, \ldots, S_9$ isn't even random! Given the other balls, we can know exactly what the last ball will be. Conveniently, drawing balls from an urn corresponds to an idealized vote reporting process. Counties record and report their results in some random order and we get to observe them in a sequence. Does this mean that counties reporting results are exchangeable and we are in the clear to use conformal prediction?

Unfortunately, not quite. If you've seen election results come in on election day, you may realize that this idealized "balls from an urn" model is very inaccurate. In many parts of the US, rural counties report earlier than urban counties since there are often fewer votes to count in polling places. Also, states on the east coast are guaranteed to report earlier than states on the west coast because New Hampshire can start counting votes when California is still voting. New Hampshire might be entirely finished reporting before California has even closed their polls. What does this mean for our model? Thankfully, the situation isn't as dire as it may seem. Because we are predicting the normalized residual $\tilde{r}_i = (T_i^{(2020)} - T_i^{(2016)})/T_i^{(2016)}$, we don't expect the response variable to be vastly different for rural and urban precincts. Our normalization of the response variable ensures that our samples are "close to" exchangeable. We also won't even try to forecast election results for states that are still voting. We simply cannot trust that the county results we observe in New Hampshire tell us enough about what might be happening in California.

Still, because of increased mail in voting and extended return dates we expect many counties not to report at 100% for a very long time. Because of state differences in vote-by-mail laws, we may have large numbers of counties for which we have no representative observed in the training set. This remains an issue, but we won't have a good handle on how bad the situation is until election night. We'll have more on this topic in Section 5.

One more thing to keep in mind is that the prediction intervals we get through conformal prediction aren't exactly the prediction intervals you were probably expecting. Instead of providing *conditional coverage*, they provide *marginal coverage*.

[7] https://jmlr.csail.mit.edu/papers/volume9/shafer08a/shafer08a.pdf

A 95% prediction interval with marginal coverage will include the true turnout value for 95% of counties over which the model is applied. That sounds pretty good! But there's a problem. For a fixed set of characteristics $X$, the model isn't guaranteed to be correct 95% of the time. The coverage of this interval is only guaranteed marginally. Imagine that there are 100 counties, 95 of them being majority White and 5 of them being majority non-White. A model with marginal coverage could achieve this coverage by covering all 95 majority White counties 100% of the time and never covering the majority non-White ones.

By contrast, a 95% prediction interval with conditional coverage will, for any given set of characteristics $X$, include the true turnout 95% of the time. Note that an interval with 95% conditional coverage will also have 95% marginal coverage. Unfortunately, guaranteeing conditional coverage is provably impossible without making strong distributional assumptions, and that's exactly the kind of assumption we didn't feel comfortable making. In the next subsection, we'll talk about how this problem affects our forecasts, and how we get around that.

So now that we have county predictions and county prediction intervals[8], it sounds like we've got nothing left to do? Unfortunately, this is not the case. Most Americans would not be particularly interested in having merely the turnout predictions for any *county*. With few exceptions, electoral votes are cast by *states*, so our model needs to produce valid state-level prediction intervals.

#### 4.2.1 State Prediction Intervals

To get our state-level prediction intervals, we aggregate all of our county-level prediction intervals within the state into one larger interval. This is similar to how we aggregate all the county-level point estimates to get our state-level turnout predictions.

The marginal coverage of conformal prediction intervals poses a threat to the validity of these state-level predictions. Imagine that we have 100 counties that we are trying to predict turnout for and 95 of them are small townships in the New England, while 5 of them are larger counties near Boston, Providence, and New Haven. As we explained above, prediction interval with 95% marginal coverage could cover the true turnout for the townships only, but we know that each urban county is, individually, far more important for quantifying state-level results because it contributes more voters to the total we are trying to estimate. By contrast, note that if these intervals had conditional coverage, the intervals would include the true turnout 95% of the time regardless of which type of county we were predicting turnout for.

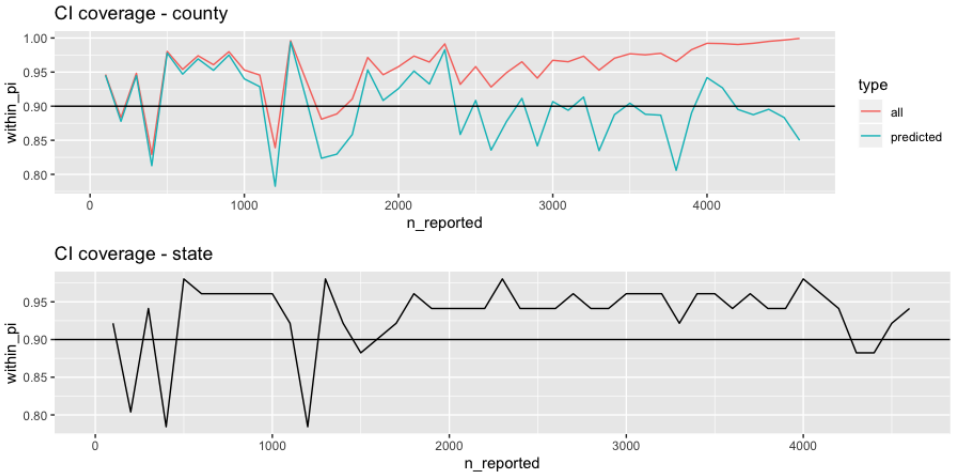
Luckily, there are a couple factors that help us avoid this pitfall. First, given enough data — as well as a well-specified $f_\beta(x)$ — we know that quantile regression produces prediction intervals with conditional coverage. Still, we can never be sure of either of those assumptions, so we modified the conformalization procedure to increase the chance of coverage for counties with more voters in them. Instead of only providing 95% marginal coverage, we also guarantee that on average 95% of voters will be members of counties for which our prediction intervals include the true turnout. We practically implement this by upweighting counties that had larger turnout in the previous elections. In doing so, we ensure that our model's prediction intervals are not overfit to small counties that are less important for predicting state-level turnout.

With this modification, we believe that we can reasonably approximate the output of our algorithm as providing 95% conditional coverage for county-level turnout. But we still haven't answered the question of how to produce state-level results. If the error of our model is correlated between counties, our prediction intervals at the state level need to be larger. We spent the first part of this write-up trying to convince you that we don't want to be in the business of estimating correlations between counties, so instead, we'll focus on developing an approach to aggregating county prediction intervals that is valid regardless of how large the between-county correlations may look.

Fortunately for us, the procedure is simple. We can sum the bounds of the county level prediction intervals to get our state prediction intervals. Why is it that simple? We have a longer discussion in the appendix, but note that summing the intervals reproduces the worst case, i.e. perfect, correlation between random variables.

[8]The full algorithm for constructing county-level prediction intervals can be found in the Appendix.

Under certain assumptions that we detail in the next section, we claim that the state level prediction intervals will include the true turnout for at least 90% of the states. If you don't believe the math, trust our experiments:




As can be seen in figure 4.2.1, which reflects a simulated version of the 2016 election, we very quickly achieve stable 90% coverage for the county level prediction intervals. For states we have a bit more variability, but eventually converge at 90%. Our tests show accurate prediction interval coverage for simulations of reporting orders far more extreme than we expect on election night, i.e. all rural counties followed by a few urban ones.

# 5 Limitations and Future Work

As we have discussed above, our model is built on certain assumptions, such as exchangeability. Unsurprisingly, there are other assumptions that we want to discuss.

Underlying the basics of the model is the assumptions that votes are counted fairly. This is true for both in-person and mail-in votes. The fact that states could have different vote-by-mail rejection rates could be a real problem for the model, since those will influence turnout numbers in unexpected ways.

We can't run our model without a steady stream of results coming in on election night. Because of COVID-19 and the expansion of absentee voting, many counties may not fully report their results for weeks. On election night, it is entirely plausible that we will only be able to train this model on counties from a small handful of states. In that case, we will likely only display prediction intervals for states from which we have at least some complete county returns. In this way we can avoid state-specific counting issues causing problems with our prediction intervals.

More perniciously, however, our model relies on knowing when a county has finished reporting. In a normal election we could use "precincts reporting" as our signal for when a county was nearly done. If a county reached 100% of precincts reporting, we knew that only the absentee votes were left outstanding, and those usually made up a relatively small fraction of the vote. Given the substantial amount of mail-in voting in the 2020 cycle, it has become a lot trickier to know when a county has finished reporting results. We rely on communication with our election results provider to know whether we can include a county in our model. This means that if we cannot trust that enough counties have actually finished reporting on election night, we may never decide to turn our model on.

States and counties with different vote-by-mail regulations may exhibit unusually variable turnout and party vote-share patterns. This would invalidate the primary assumption of our model, that counties with similar demographics will vote similarly. The model that we run on election night will use an augmented $\tilde{X}_i$ that includes additional features that identify the extent of vote-by-mail availability in county $i$.

Beyond the limitations of our work, there are a few things that we hope to incorporate in the future. Integrating the conformal prediction methodology we've described here with kriging would allow us to make better use of the spatial correlations that we know to exist among neighboring counties. Incorporating polling data and other contemporaneous measures of voter sentiment, e.g. local unemployment rates, is likely to improve our model's predictive accuracy. Finally, we also think there is potential work to be done with pre-night prediction. For this election we are using 2016 turnout as our pre-night estimate, but constructing a likely voter model could give us a better starting point.

# 6 Conclusion

The Washington Post will be estimating both turnout and partisan vote share on election night. This election is likely to be exceptional in many ways, and we want to be able to provide our readers with as much context and information as we can. Raw vote counts do not tell the full story while results are still outstanding. And this is even more true in elections where a significant fraction of the vote may not be counted or reported immediately on election night. We started the endeavor of adding expected turnout predictions to our election results for the Virginia election in 2019. Since that experiment, we have iterated on our approach and have written about what we have tried and learned along the way.

As always, please reach out with any suggestions or if you spot any mistakes. We'd love to hear from you.

*This model was created in cooperation with Decision Desk HQ/0ptimus Analytics. Special thanks to Alexander Podkul, Alex Alduncin, Matt Shor, Kiel Williams, Scott Tranter and Drew McCoy*

*Furthermore, Jessica Eng was integral to this work. We would not be where we are today without her*

*We want to thank everyone at The Washington Post that made this work possible. That includes, but is not limited to: Peter Andringa, Jason Bernert, Jeremy Bowers, David Byler, Reuben Fischer-Baum, Simon Glenn-Gregg, Shana Hadi, Jason Holt, Aditya Jain, Teddy Kentor, Emily Liu, Anthony Pesce, Erik Reyna, Terri Rupar, Ashlyn Still and Susan Tyler.*

*Finally, we want to thank Stephen Bates, Laura Bronner, Akosua Busia, Emmanuel Candés, Benjamin Cherian, Nick Diakopoulos, Drew Dimmery, Jessica Hullman, Idrees Kahloon, Kabir Khanna, G. Elliott Morris, Hunter Nisonoff, Daniel Richman, and Yaniv Romano for their help, advice and support along the way.*

# A Appendix

## A.1 County Prediction Intervals

The following outlines our procedure for generating a 90% prediction interval for the 2020 turnout in unseen county $k$ using split conformal prediction.

1. Observe county results $\tilde{r}_1, \ldots, \tilde{r}_m$
2. Randomly split $\tilde{r}_1, \ldots, \tilde{r}_m$ into a training set of size $p$ and calibration set of size $q$ where $p \approx 0.9m$.
3. Using the training set, fit two quantile regression models, $f_\alpha(\cdot)$ and $f_\beta(\cdot)$, to predict the 0.05 and 0.95-quantiles of $\tilde{r}$, respectively.
4. Compute conformity scores $E_l := \max(f_\alpha(X_l) - \tilde{r}_l, \tilde{r}_l - f_\beta(X_l))$ for $(X_l, \tilde{r}_l)$ in the calibration set.
5. Let $C$ be the $90\left(1 + \frac{1}{q}\right)$-th percentile of the conformity scores $\{E_l\}_{l=1}^{q}$ where we assume that $P(E_l) = \frac{1}{q}$ for any $l$.

6. Output $\left( \underbrace{T_k^{(16)}(f_\alpha(X_k) - C) + T_k^{(16)}}_{\hat{T}_k^{(20)}(0.05)}, \underbrace{T_k^{(16)}(f_\beta(X_k) + C) + T_k^{(16)}}_{\hat{T}_k^{(20)}(0.95)} \right)$

The prediction intervals we generate with the procedure will be perfect, i.e. they cover the true turnout for exactly 90% of the unseen counties, when the ordering of the reported counties is random. This assumption is not met in a presidential election since Alaskan counties will still be voting while most of other counties have begun to report results. Still, empirically, deviations from this assumption have minimal effect on the accuracy of these prediction intervals once 20-25% of the counties nationally have reported.

## A.2 State Prediction Intervals

Why does summing the county prediction intervals work?

Luckily for us, some other folks have studied this problem extensively. As it turns out, there is a lot of money to be made in accurately aggregating prediction intervals. Portfolio managers at large investment firms are interested in knowing how much risk they take on when they accumulate assets that each have some uncertainty associated with them. Formally, they use copulas to model a joint distribution over portfolio assets given a marginal distribution for each asset. These copulas encode some notion of dependency between each of the assets; we model the joint distribution between county-level turnouts using the maximally conservative comonotonic copula. The conditions under which this copula is maximally conservative take dozens of pages to properly discuss and enumerate, but the upshot of this choice is that we obtain the state-level prediction intervals by simply summing the bounds of the county-level prediction intervals. For example, if we predict turnout shifts between 1000 and 2000 votes for county 1 and 1300 and 2800 votes for county 2, the comonotonic copula model of dependence would suggest that the prediction interval for county 1 + county 2 is $(2300, 4800)$.

If you don't believe the arguments others have made for using the comonotonic copula, we also note that the comonotonic copula naturally reproduces the worst-case prediction interval for jointly normal county turnouts. Imagine that we did assume that county turnouts were sampled from a normal distribution, i.e. $\tilde{r}_1 \sim \mathcal{N}(0, 0.1^2)$ and $\tilde{r}_2 \sim \mathcal{N}(0, 0.2^2)$. Then, $Var(\tilde{r}_1 + \tilde{r}_2)$ is maximized when the correlation $\rho_{12} = 1$. The resulting 95% prediction interval, $0 \pm 1.96 \times (0.1 + 0.2)$ is then equal to the sum of the two prediction intervals for $\tilde{r}_1$ and $\tilde{r}_2$, $0 \pm 1.96 \times 0.1$ and $0 \pm 1.96 \times 0.2$ respectively.

The following steps then outline our procedure for generating a state-level prediction interval. We denote counties in the state that have already reported using the indices $a_1, \ldots, a_U$, and counties in the state that have yet to report using indices $b_1, \ldots, b_V$.

1. Observe county results $\tilde{r}_1, \ldots, \tilde{r}_m$
2. Randomly split $\tilde{r}_1, \ldots, \tilde{r}_m$ into a training set of size $p$ and calibration set of size $q$ where $p \approx 0.9m$.
3. Using the training set, fit two quantile regression models, $f_\alpha(\cdot)$ and $f_\beta(\cdot)$, to predict the 0.05 and 0.95-quantiles of $\tilde{r}$, respectively. We fit a weighted loss function where $w_i = T_i^{(12)}$.
4. Compute conformity scores $E_l := \max(f_\alpha(X_l) - \tilde{r}_l, \tilde{r}_l - f_\beta(X_l))$ for $(X_l, \tilde{r}_l)$ in the calibration set.
5. Let $C$ be the $90\left(1 + \frac{1}{q}\right)$-th percentile of the conformity scores $\{E_l\}_{l=1}^q$ where we assume that $P(E_l) = T_l^{(12)} / \sum_{l=1}^q T_l^{(12)}$ for any $l$

6. Output the following prediction interval:

$$\left( \underbrace{\sum_{u=1}^{U} T_{a_u}^{(20)}}_{\text{Observed}} + \underbrace{\sum_{v=1}^{V} T_{b_v}^{(16)}(f_\alpha(X_{b_v}) - C) + T_{b_v}^{(16)}}_{\text{Estimated lower bound}}, \right.$$

$$\left. \underbrace{\sum_{u=1}^{U} T_{a_u}^{(20)}}_{\text{Observed}} + \underbrace{\sum_{v=1}^{V} T_{b_v}^{(16)}(f_\beta(X_{b_v}) + C) + T_{b_v}^{(16)}}_{\text{Estimated upper bound}} \right)$$

After some time on election night, we may observe sufficiently many counties from a given state to construct a calibration set that exclusively contains counties from that state. We can take advantage of this surplus of data to provide guarantees on the accuracy of our prediction intervals for all counties in that state.

First, we compute another constant, $D$, in step 5 that is the $90(1 + 1/q_s)$-th percentile of the conformity scores $E_j$ calculated on this state-specific calibration set. Then, the correction we apply in part 6 to the quantile regression outputs becomes $\max(C, D)$.