Modeling the Georgia Senate Runoff Elections
Model Methodology
Open Model Project
https://openmodelproject.org/

Our Philosophy

Our goal is to forecast elections accurately, but also in a completely open source and transparent way. We also want to be clear about what our assumptions are and allow the user to change them if desired and see how the change in assumptions affect the model.

Table of Contents

Our Method

Step 1: Collect Polls

The first step is to get a list of polls of the Georgia Senate runoff elections. We use 538's list of polls from https://projects.fivethirtyeight.com/polls/georgia/ and list them in our spreadsheet with the first day that polling was conducted on (Column A of the polls list), the day the poll was listed on 538 (Column B), the name of the pollster and a link to the poll (Column C), the expected margin of victory for Raphael Warnock over Kelly Loeffler (with a negative number if Loeffler is ahead) (Column D), the expected margin of victory for Jon Ossoff over David Perdue (with a negative number if Perdue is ahead) (Column E), and the number of undecided voters (Column F).

However, there are some polls we did not include that 538 does include. We decided to exclude any poll that did not sample all eligible voters (e.g., the 14 December Lake Research poll of only female voters, the 30 November Fabrizio Ward poll of only voters 65+).

We also excluded polls that (a) are from partisan pollsters that (b) did not poll prior to 2020 and that (c) were not accurate in 2020 (e.g., the 10 Nov VCreek / AMG poll).

We included polls from Trafalgar despite noted issues with their past results as their errors are at least well known enough to adjust for (see below) and we believe there is a signal there after this noise is adjusted for. However, users can use the custom menu provided to tweak the model assumptions and exclude Trafalgar polls if desired (set "Include Trafalgar?" to FALSE).

Step 2: Adjust Polls for Historical Error

Polls have been wrong historically - most notably, missing the election of Donald Trump in 2016. Any good model should try to adjust for this historical error. Luckily for us, there was just recently an election where we could measure pollster accuracy. We assume that the best guess is that they will repeat this error again and be just as off at polling the runoff as they were at polling the initial 2020 November election. We therefore adjust for the error in the Jon Ossoff vs. David Perdue race if available, and otherwise adjust for the error in the Georgia Biden margin if available, and otherwise adjust for the average error of all Georgia runoff polls in our database if neither is available (new pollster).

The data for 2020 poll error is in Column H of the polls list (e.g., “Polls - 24 Dec” tab) and the adjustment made to the raw poll result is in Column I. The final poll result will be the raw result plus the sum of all the adjustments.

This seems like a safe guess as we have not yet seen any indication that pollsters are updating their methodologies in the short time between the November election and now. However, users can use the custom menu provided to tweak the model assumptions and remove this adjustment if desired (set “Historical error adj?" to “None”) or, if desired, change the exact adjustment on a per-polster basis in a copy of the spreadsheet.

Step 3: [Experimental] Adjust Polls for COVID

There is some empirical indication that polling errors in 2020 were correlated with COVID rates, where COVID rates in the state two weeks prior to the election explained nearly 40% of the error in a state’s polling. The theory here is that democratsare more likely to self-isolate from COVID and more likely to be in urban centers with lockdowns, and therefore more likely to respond to polls, and thus more COVID makes polls more likely to oversample Democrats.

The model produced in the paper suggests the polls will be off an additional 0.17 percentage points for every additional increase in the “COVID cases per 100,000” rate of a state. We assume this trend will continue as COVID in Georgia continues to increase. However, there are some reasons to think this effect may be nonexistent or overstated[1] or not actually linear[2] and therefore we only apply 1/4th the strength in our model, assuming the polls will be off an additional 0.0425 percentage points for every additional increase in the “COVID cases per 100,000” rate of Georgia after the election.

The COVID rate (as new COVID cases per 100K people) is taken from the first day of poll collection and is in Column J of the polls list (e.g., “Polls - 24 Dec” tab) and the adjustment made to the raw poll result is in Column K. The final poll result will be the raw result plus the sum of all the adjustments.

This COVID adjustment is experimental and has never been done before, so it may be very wrongheaded. Therefore, it is disabled by default, but users can use the custom menu provided to tweak the model assumptions and remove this adjustment if desired (set “COVID Adjust?" to FALSE).

[1] The effect may be lower now that people are more used to COVID and less likely to self-isolate in response to COVID case increases.
[2] There may be a threshold effect as lockdowns are introduced, but not much effect beyond that.

Step 4: [Experimental] Adjust Polls for Undecideds

Every poll currently has some level of undecided voters. This increases uncertainty (see below), but if we think undecided voters might be more likely to break a particular way, it also means we may want to adjust the polls. In 2016 and 2020, it appears that undecided voters may have been more likely to break toward the Republicans (although the magnitude and, infact, reality is uncertain) and this may occur again in the Georgia runoff.

We are not sold on this assumption so our default is currently to not enable this. However, users can use the custom menu provided to tweak the model assumptions and add this adjustment if desired - set “Undecideds break" to “Break evenly” to assume undecideds break evenly (current default), or set to “Republican” to assume undecideds split 70% towards Republicans, or set to “Democrats” to assume undecideds split 70% towards Democrats.

The number of undecideds is in Column F of the polls list (e.g., “Polls - 24 Dec” tab) and the adjustment made to the raw poll result is in Column G. The final poll result will be the raw result plus the sum of all the adjustments. For example, the 16 Nov InsiderAdvantage poll has a raw result of Warnock +1, but then is adjusted -3 points (to Loeffler +2) because of InsiderAdvantage’s 3 point error in the 2020 general election. If the COVID adjustment were used, the poll would be adjusted an additional -0.3 points (to Loeffler +2.3) due to increased COVID rates. If “Breaks Republican” was selected for the undecideds in addition to the COVID adjustment, the poll would also be adjusted an additional -1.05 points (to Loeffler +3.3).

Step 5: Weigh Polls Based on Recency

Even after adjustments, not all polls are equal, so we also aim to weigh polls. The first weight is just to weigh polls by how recent they are. Our recency adjustment is

```
adjustment = (1 / (days since first day of polling relative to
youngest poll + 1)^0.2) * 0.1 + max((30 - days since first day of
polling relative to youngest poll) / 30), 0) * 0.1

if days since first day of polling relative to youngest poll > 30,
then adjustment = 0
```

This gives all polls some weight, but emphasizes more recent polls:

Age of poll	**Weight**
Youngest poll	1.0

1 day older than youngest poll	0.948
1 week older than youngest poll	0.703
3 weeks older than youngest poll	0.242
4 weeks older than youngest poll	0.078
5 weeks older than youngest poll	0

The recency of the poll in number of days between the model time and the first day of polling for the poll is given in Column P of the polls list (e.g., the “Polls - 24 Dec” tab would calculate recency as of 24 Dec) and the weighting made to the poll is in Column P. The final weight will be the product of all weights.

Step 6: Weigh Polls Based on Sample Size

We also weigh polls by how many people are in the sample, assuming that larger polls are more likely to be accurate, as they have a lower margin of error. However, we should note that this weight is not as simple as it seems, given that the vast majority of true error in a poll does not come from the sample size but rather from the sampling methodology. Additionally, there may be adverse correlation effects as better sampling methodologies requiring high costs per participant recruited and thus lower sample sizes and lower quality pollsters trying to look more impressive by having larger sample sizes.

We weigh polls based on sample size by the square root of the number of participants, to account for diminishing returns. The equation is `sqrt(N) / sqrt(max N among all polls).`

Sample size of poll	**Weight**
N=500 (largest poll N=1500)	sqrt(500)/sqrt(1500) = **0.577**
N=500 (largest poll N=4000)	sqrt(500)/sqrt(4000) = **0.354**
N=1000 (largest poll N=1500)	sqrt(1000)/sqrt(1500) = **0.816**
N=1500 (largest poll N=1500)	sqrt(1500)/sqrt(1500) = **1.0**

The sample size of the poll in number of days between the model time and the first day of polling for the poll is given in Column Q of the polls list (e.g., the “Polls - 24 Dec” tab) and the weighting made to the poll is in Column S. Column R shows the square root of the sample size. The final weight will be the product of all weights.

Step 7: Weigh Polls Based on Historical Quality

The best polls are not more accurate because they are more recent or have larger sample sizes, but because they have a higher quality methodology. We thus want to weigh polls based on their quality. We do not make any assumptions about which methods are better, however, as this changes over time and also changes between pollsters (even those using the same methods). We instead look at the historical results of pollsters using a database of all polls conducted 2000-2018[3].

We look at a measure of forecast accuracy called the mean arctangent absolute percentage error (MAAPE) to see how good pollsters have been at forecasting the results of races. A lower MAAPE score indicates superior accuracy.

We look at the entire database (2000-2018) plus the "Trump era" years (2016 and 2018) with a 25%-75% tilt toward the "Trump era" given that Donald Trump is still President and we expect our current Georgia runoff polling to be more like 2016+2018 polling than the polling prior to 2016.

The equation is:

```
Blended MAAPE = 0.75*(2016+2018 MAAPE) + 0.25*(2000-2018 MAAPE)

Weight = 1/((Blended MAAPE - Average Blended MAAPE of all GA runoff
polls) - (minimum of {Blended MAAPE - Average Blended MAAPE of all GA
runoff polls} for all polls) + 1) * 0.7 + 0.3
```

Therefore the lowest quality weight a poll can get is 0.6, so as to not have certain polls dominate over other polls. Consider this hypothetical list of pollsters and see how they would be quality weighted.

Pollster	**2000-2018 MAAPE**	**2016+2018 MAAPE**	**Blended MAAPE**	**Quality Weight**
SurveyUSA	53.8	67.4	64.0	1/((64.0 - 66.1) - (61.2 - 66.1) + 1) * 0.4 + 0.6 = **0.705**
Trafalgar	81.4	72.2	74.5	1/((74.5 - 66.1) - (61.2 - 66.1) + 1) * 0.4 + 0.6 = **0.628**
Rasmussen	59.2	61.9	61.2	1/((61.2 - 66.1) - (61.2 - 66.1) + 1) * 0.4 + 0.6 = **1.0**
Emerson College	63.0	61.5	61.9	1/((61.9 - 66.1) - (61.2 - 66.1) + 1) * 0.4 +

[3] We do not yet have MAAPE data for 2020 but will add it as soon as it is available.

				0.6 = **0.835**
Monmouth	49.8	78.0	71.0	1/((71.0 - 66.1) - (61.2 - 66.1) + 1) * 0.4 + 0.6 = **0.637**
Siena College	60.1	64.9	63.7	1/((63.7 - 66.1) - (61.2 - 66.1) + 1) * 0.4 + 0.6 = **0.714**
Average of these[4]	61.2	67.7	66.1	

Polls with no track record in our dataset are given the track record of a bottom 25th percentile poll.

The “2000-2018 Avg. MAAPE” for the pollster is given in Column T of the polls list (e.g., the “Polls - 24 Dec” tab). The “2016+2018 Avg. MAAPE” for the pollster is given in Column U. The “Blended MAAPE” for the pollster is given in Column V. The weighting made to the poll is in Column X. Column W shows the difference between the blended MAAPE and the average blended MAAPE of all polls. The final weight will be the product of all weights.

Step 8: Weigh Polls Based on Duplicate Polling

Some pollsters are more prolific than others, and we don’t want their polls dominating the average. Therefore we re-weigh duplicate polls according to how many times they show up in our polling. We offer four such ways to do this:

- **No adjustment:** Do not adjust polls based on duplication. That is, if Trafalgar polls a race four times, all four of their polls will be counted for 1.
- **Linear no skew:** Adjust polls to smooth to 1. That is, if Trafalgar polls a race four times, all four of their polls will be counted by 0.25x.
- **Linear skew recent (default):** Adjust polls to smooth to 1, but count recent polls for more. That is, if Trafalgar polls a race four times, their first poll will get 0.1x weight, their second poll will get 0.2x weight, their third poll will get 0.3x weight and their fourth poll will get 0.4x weight.
- **Double no skew:** Adjust polls to smooth to 2, allowing a pollster to only be double counted at max. That is, if Trafalgar polls a race four times, all four of their polls will be counted by 0.5x.
- **Double skew recent:** Adjust polls to smooth to 2, allowing a pollster to only be double counted at max, but count recent polls for more. That is, if Trafalgar polls a race four times, their first poll will get 0.2x weight, their second poll will get 0.4x weight, their third poll will get 0.6x weight and their fourth poll will get 0.8x weight.

[4] Average of this hypothetical list of polls, as an example for calculation purposes

Users can change this using the menu to change the model assumptions and modifying “Penalize duplicates?” to their choice.

Step 9: Combine Weights

We make our final poll weight based on the product of all the weights: `Final Weight = Recency Weight * Sample Size Weight * Quality Weight * Duplication Weight`.

The above is the default, however users can use the custom menu provided to tweak the model assumptions and change how weighting works by changing “Weigh polls?”:

- **Simple average:** Weigh all polls equally (though still apply duplication weight by default) - you may want this if you don’t trust poll weighting and/or just want to see the simple polling average
- **Recency:** Weigh polls only according to their recency weight (and duplication weight) - you may want this if you don’t trust our quality scores and don’t think sample size is an important factor for poll quality
- **Recency + Size:** Weigh polls only according to recency and sample size (and duplication weight) - you may want this if you don’t trust our quality scores
- **Recency + Quality:** Weigh polls only according to recency and quality score (and duplication weight) - you may want this if you don’t think sample size is an important factor for poll quality
- **Recency + Quality + Size (default):** Weigh polls according to recency, quality score, and sample size (and duplication weight)

Whether or not to apply the duplication weight can be changed by using the menu to change the model assumptions and modifying “Penalize duplicates?” (see above).

Step 10: Calculate Uncertainty and Create Race Odds from Polls

Now that we have adjusted and weighed the polls, we have a final polling margin. However, while we think this margin is the best guess for where the election margin will actually end up by the time the election is over, we surely don’t expect the election will be precisely our forecasted margin, as we may well be wrong (even if only a little bit). We therefore need to account for uncertainty, and this uncertainty can also tell us the odds of Democrats capturing the Senate.

We assume the races follow a Student’s t distribution with degrees of freedom varying based on time between the model and election day, the number of undecideds in the race, the sample size of all of the polls, and the average polling error of polls. This is calculated in the “Polls -> Odds” tab.

- **time between the model and election day** - using historical data collected by 538, we see that polls can change over a two month period between the general and runoff election by up to 10 points in either direction, implying a standard deviation of 3.53 but a mean of 0. This error then linearly decays as we get closer to the election.
- **The number of undecideds in the race** - undecideds can break disproportionately toward one candidate, as happened in the 2016 and 2020 Presidential elections. With more voters undecided, there is a larger chance of this. We assume a 5% chance that all the undecideds break toward one particular candidate and assume this is t-distributed, thus giving a standard deviation of 0.305 per percentage point of undecided voters[5]. We assume here that this has a mean of 0, but this can be changed with the model assumptions (see above).
- **The sample size of all of the polls** - Even completely representative polls still have a raw statistical margin of error that declines with increasing sample size. We model this by taking all the polls and totaling all their sample sizes and looking at the margin of error.
- **The average polling error of polls** - using historical data collected by 538 since 2000, we see that Senate polls can be off by up to 5 points in either direction, implying a standard deviation of 2.76 but a mean of 0.
- **Combined** - we then combine all these sources of uncertainty by adding up the different distributions

For one hypothetical example:

Days until election	**Undecideds**	**Sample Size**	**Polling Error**	**Combined SD**
0 (SD = 0)	1% (SD = 0.31)	20,000 (SD = 0.22)	SD = 2.76	2.78
1 (SD = 0.06)	1% (SD = 0.31)	19,000 (SD = 0.22)	SD = 2.76	2.78
3 (SD = 0.18)	2% (SD = 0.61)	16,000 (SD = 0.24)	SD = 2.76	2.84
7 (SD = 0.41)	3% (SD = 0.92)	14,000 (SD = 0.26)	SD = 2.76	2.95
10 (SD = 0.59)	3% (SD = 0.92)	12,000 (SD = 0.28)	SD = 2.76	2.98
30 (SD = 1.76)	3% (SD = 0.92)	8000 (SD = 0.34)	SD = 2.76	3.42
60 (SD = 3.53)	5% (SD = 1.53)	1000 (SD = 0.96)	SD = 2.76	4.83

And then with a standard deviation and a mean (the modeled margin of error), we can calculate the odds of winning an election. Comebacks are more likely with higher “Combined SD”:

[5] A normal distribution with a 5% chance of -1 and a 5% chance of 1 has a standard deviation of 0.69 and a mean of 0. We divide this standard deviation in half given that we are expecting undecideds to break evenly by default.

Margin	**Combined SD**	**Odds of winning**
-6	2.78	2%
-3	2.78	14%
-1	2.78	36%
0	2.78	50%
+1	2.78	64%
-6	4.83	11%
-3	4.83	27%
-1	4.83	42%
0	4.83	50%
+1	4.83	58%

We use a Student's t distribution to account for "fat tails", allowing for extreme events to be more likely. This makes the model more conservative. Users can use the custom menu provided to tweak the model assumptions and change this to use a normal distribution instead by changing "Distribution type" to "Normal".

As of 24 December, using all the default values, our model thinks there is a 34% chance that Ossoff is the winner and a 42% chance that Warnock is the winner with the default assumptions.

Step 11: Calculate Democrat Senate Odds from Race Odds

The main reason we are paying attention to these double Georgia Senate runoff elections is that they determine control of the Senate. If Democrats win both races, they win control of the Senate (with the 50-50 tie broken by Democrat Vice President Elect Kamala Harris). If the Republicans win one out of two, they win control.

It might be tempting to think that if Ossoff has a 34% chance and Warnock has a 42% chance, that the chance of them both winning is, `34% * 42%` or ~14%. However, this would mean that both races are independent events… given that most people vote Democrat or Republican regardless of the candidate, this is not the case. We, therefore, have to see how much these races are correlated with each other.

We take this "Warnock-Ossoff Correlation", and determine the odds of a D Senate using the equation `(Warnock-Ossoff Correlation)*min(Warnock win odds, Ossoff win odds) + (1-Warnock-Ossoff Correlation)*(Warnock win odds)*(Ossoff win odds)`

Warnock-Ossoff Correlation	**Odds of D Senate**
0	`0*min(0.42, 0.34) + 1*0.42*0.34` = **14%**
0.2	`0.2*min(0.42, 0.34) + 0.8*0.42*0.34` = **18%**
0.4	`0.4*min(0.42, 0.34) + 0.6*0.42*0.34` = **22%**
0.5	`0.5*min(0.42, 0.34) + 0.5*0.42*0.34` = **18%**
0.6	`0.6*min(0.42, 0.34) + 0.4*0.42*0.34` = **24%**
0.7	`0.7*min(0.42, 0.34) + 0.3*0.42*0.34` = **28%**
0.75	`0.75*min(0.42, 0.34) + 0.25*0.42*0.34` = **29%**
0.8	`0.8*min(0.42, 0.34) + 0.2*0.42*0.34` = **30%**
0.9	`0.9*min(0.42, 0.34) + 0.1*0.42*0.34` = **32%**
0.95	`0.95*min(0.42, 0.34) + 0.05*0.42*0.34` = **33%**
1.0	`1*min(0.42, 0.34) + 0*0.42*0.34` = **34%**

It's hard to determine what this value is going to be. It's certainly not 0, but it's certainly not 1 either, given that some people do vote split ticket. We know *a priori* that from the history of double-barrelled Senate runoff elections, they almost always go to the same party - they haven't split since 1966, and they've only split 12.5% of the time over the past 100 years.

Looking at our polls, the current correlation between Ossoff results and Warnock results is 0.62, but this could be heavily influenced by noise in polling sampling and/or polling methodology.

Information source	**Result**
Past 100 years of elections	Correlation = 0.875
Polling	Correlation = 0.62

Average of information	*0.747*

Combining all of this, we currently expect a correlation of ~0.75 as our best guess. We make this the default, but users can use the custom menu provided to tweak the model assumptions and set “Warnock-Ossoff corr” to their desired value.

Step 12: Include the Election as a Prior

Even with poll margin adjustments and poll weighting, polls can still be off by a lot (see sources of uncertainty, above). One way to help reduce that error is to combine the estimated results from the model with the results from a prior. Many models use priors calculated from things such as incumbency, the partisan lean of the state, economic conditions, etc. However, we are lucky enough to have just had a recent election that we can use as our prior since we should put some weight on the hypothesis that nothing really has changed or will change over the two months from the election - that very few people are persuadable to change sides - and that maybe all the poll results we see are more noise than signal.

We, therefore, include the election as a prior. To begin, we assume a 50-50 blend between the election prior and the polls, heavily weighing the election prior in this model because the election was so recent.

Step 13: Ensemble with Prediction Markets

Another potential boost is to combine the results from this model with the results of the prediction markets. We use Predictit.org as our source. Prediction markets can include all sorts of information that we do not include, such as “wisdom of the crowds”, fundraising, and/or early turnout data. They also could include a lot of noise and nonsense. Historically, a combination of a model and the prediction markets have outperformed either models or markets at predicting in the 2016 and 2020 elections[6], so we attempt to leverage this.

However, users do not have to agree with our assumptions and can use the custom menu provided to tweak the model assumptions and set “Ensemble picker…” to their desired value:

- **All model:** Only use our estimated model margins and odds
- **Model + prior:** 50% results from our model, 50% results from the most recent election
- **Model + market:** 90% results from our model, 10% results from PredictIt
- **Model + prior + market (default):** 45% results from our model, 45% results from the most recent election, 10% results from PredictIt

[6] According to forthcoming but currently unpublished analysis. Further research is warranted here.

- **Custom:** Be able to set the three weights to whatever you want. You can also use this to give weight to the 538 polling average if you desire[7].

Step 14: Smooth Model Odds Over Batch Updates

Lastly, to avoid making the model too jumpy, we track each day's model update and average over multiple days. This prevents a particular outlier poll from coming out and jumping the average, as the model will wait for additional polls to confirm the trend before jumping into it. Without model smoothing (or the election priors, which also stabilizes things), odds for Warnock jump as high as 70% on 4 Dec before crashing down to 35% by 16 Dec. The smoothing helps even this out.

We currently use a model smoothing of 2 by default, which means averaging over the current and two prior model updates. However, the user can use the custom menu provided to tweak the model assumptions and set "Model Smoothing" to 0 for no smoothing, 1 to average the current and previous day, 2 to average the current and past two days, or 3 to average the current and past three days.

[7] We currently do not give any weight to the 538 polling average as we think our model ends up making all the same or better choices.