Now it's time to evaluate our models on the testing set.

So the first model we're going to want to look at is that smart baseline model that basically just took a look at the polling results from the Rasmussen poll and used those to determine who was predicted to win the election.

So it's very easy to compute the outcome for this simple baseline on the testing set.

We're going to want to table the testing set outcome variable, Republican, and we're going to compare that against the actual outcome of the smart baseline, which as you recall would be the sign of the testing set's Rasmussen variables.

And we can see that for these results, there are 18 times where the smart baseline predicted that the Democrat would win and it's correct, 21 where it predicted the Republican would win and was correct, two times when it was inconclusive, and four times where it predicted Republican but the Democrat actually won.

So that's four mistakes and two inconclusive results on the testing set.

So this is going to be what we're going to compare our logistic regression-based model against.

So we need to obtain final testing set prediction from our model.

So we selected mod2, which was the two variable model.

So we'll say, TestPrediction is equal to the predict of that model that we selected.

Now, since we're actually making testing set predictions, we'll pass in newdata = Test, and again, since we want probabilities to be returned, we're going to pass type="response".

And the moment of truth, we're finally going to table the test set Republican value against the test prediction being greater than or equal to 0.5, at least a 50% probability of the Republican winning.

And we see that for this particular case, in all but one of the 45 observations in the testing set, we're correct.

Now, we could have tried changing this threshold from 0.5 to other values and computed out an ROC curve, but that doesn't quite make as much sense in this setting where we're just trying to accurately predict the outcome of each state and we don't care more about one sort of error-- when we predicted Republican and it was actually Democrat-- than the other, where we predicted Democrat and it was actually Republican.

So in this particular case, we feel OK just using the cutoff of 0.5 to evaluate our model.

So let's take a look now at the mistake we made and see if we can understand what's going on.

So to actually pull out the mistake we made, we can just take a subset of the testing set and limit it to when we predicted true, but actually the Democrat won, which is the case when that one failed.

So this would be when TestPrediction is greater than or equal to 0.5, and it was not a Republican.

So Republican was equal to zero.

So here is that subset, which just has one observation since we made just one mistake.

So this was for the year 2012, the testing set year.

This was the state of Florida.

And looking through these predictor variables, we see why we made the mistake.

The Rasmussen poll gave the Republican a two percentage point lead, SurveyUSA called a tie, DiffCount said there were six more polls that predicted Republican than Democrat, and two thirds of the polls predicted the Republican was going to win.

But actually in this case, the Republican didn't win.

Barack Obama won the state of Florida in 2012 over Mitt Romney.

So the models here are not magic, and given this sort of data, it's pretty unsurprising that our model actually didn't get Florida correct in this case and made the mistake.

However, overall, it seems to be outperforming the smart baseline that we selected, and so we think that maybe this would be a nice model to use in the election prediction.