Now, we're ready to actually start building models.

So as usual, the first thing we're going to do is split our data into a training and a testing set.

And for this problem, we're actually going to train on data from the 2004 and 2008 elections, and we're going to test on data from the 2012 presidential election.

So to do that, we'll create a data frame called Train, using the subset function that breaks down the original polling data frame and only stores the observations when either the Year was 2004 or when the Year was 2008.

And to obtain the testing set, we're going to use subset to create a data frame called Test that saves the observations in polling where the year was 2012.

So now that we've broken it down into a training and a testing set, we want to understand the prediction of our baseline model against which we want to compare a later logistic regression model.

So to do that, we'll look at the breakdown of the dependent variable in the training set using the table function.

What we can see here is that in 47 of the 100 training observations, the Democrat won the state, and in 53 of the observations, the Republican won the state.

So our simple baseline model is always going to predict the more common outcome, which is that the Republican is going to win the state.

And we see that the simple baseline model will have accuracy of 53% on the training set.

Now, unfortunately, this is a pretty weak model.

It always predicts Republican, even for a very landslide Democratic state, where the Democrat was polling by 15% or 20% ahead of the Republican.

So nobody would really consider this to be a credible model.

So we need to think of a smarter baseline model against which we can compare our logistic regression models that we're going to develop later.

So a reasonable smart baseline would be to just take one of the polls-- in our case, we'll take Rasmussen-- and make a prediction based on who poll said was winning in the state.

So for instance, if the Republican is polling ahead, the Rasmussen smart baseline would just pick the Republican to be the winner.

If the Democrat was ahead, it would pick the Democrat.

And if they were tied, the model would not know which one to select.

So to compute this smart baseline, we're going to use a new function called the sign function.

And what this function does is, if it's passed a positive number, it returns the value 1.

If it's passed a negative number, it returns negative 1.

And if it's passed 0, it returns 0.

So if we passed the Rasmussen variable into sign, whenever the Republican was winning the state, meaning Rasmussen is positive, it's going to return a 1.

So for instance, if the value 20 is passed, meaning the Republican is polling 20 ahead, it returns 1.

So 1 signifies that the Republican is predicted to win.

If the Democrat is leading in the Rasmussen poll, it'll take on a negative value.

So if we took for instance the sign of -10, we get -1.

So -1 means this smart baseline is predicting that the Democrat won the state.

And finally, if we took the sign of 0, meaning that the Rasmussen poll had a tie, it returns 0, saying that the model is inconclusive about who's going to win the state.

So now, we're ready to actually compute this prediction for all of our training set.

And we can take a look at the breakdown of that using the table function applied to the sign of the training set's Rasmussen variable.

And what we can see is that in 56 of the 100 training set observations, the smart baseline predicted that the Republican was going to win.

In 42 instances, it predicted the Democrat.

And in two instances, it was inconclusive.

So what we really want to do is to see the breakdown of how the smart baseline model does, compared to the actual result -- who actually won the state.

So we want to again use the table function, but this time, we want to compare the training set's outcome against the sign of the polling data.

So in this table, the rows are the true outcome -- 1 is for Republican, 0 is for Democrat -- and the columns are the smart baseline predictions, -1, 0, or 1.

What we can see is in the top left corner over here, we have 42 observations where the Rasmussen smart baseline predicted the Democrat would win, and the Democrat actually did win.

There were 52 observations where the smart baseline predicted the Republican would win, and the Republican actually did win.

Again, there were those two inconclusive observations.

And finally, there were four mistakes.

There were four times where the smart baseline model predicted that the Republican would win, but actually the Democrat won the state.

So as we can see, this model, with four mistakes and two inconclusive results out of the 100 training set observations is doing much, much better than the naive baseline, which simply was always predicting the Republican would win and made 47 mistakes on the same data.

So we see that this is a much more reasonable baseline model to carry forward, against which we can compare our logistic regression-based approach.