Kyle Weng

IDS 126

Final Project: What contributes to an FRC match win?

Introduction

My favorite part of my high school experience was the time I spent participating in the FIRST Robotics Competition (FRC) via my school's robotics team. Each season, FIRST (the organization behind the competition) unveils a thematic "game," in which robots would have to perform a variety of tasks. Teams have six weeks* from the game reveal in early January (here is an example of a "kickoff" video) to late February to design, build, and program a robot to the best of their ability. After that, there are six to seven weeks of competition in which teams travel to regional competitions, where there are various ways to qualify to the world championships. (A few states in the US have a district qualifier system: teams accumulate district points from regional performances and then participate in a playoff after the regular season to determine who advances to worlds.)

In the typical competition, each team plays about ten times. Each match is played by two alliances of three teams each. Matches typically begin with an "autonomous" period during which robots operate only with pre-written code, followed by a "teleop" period during which human players control the robots. Each game is different, but at a high level, robots complete as many objectives as possible during the two to two and a half minutes for each match. Here's a video of the final match of one of the 2019 worlds championships.

Background – The 2018 FRC Game

The game varies from year to year, so as an overview to this section, I recommend watching the 2018 FRC game animation to better understand and visualize the objectives I'll be explaining shortly. There are many things robots could do in a 2018 FRC match, but I'll be focusing on a few I considered important in determining an alliance's margin of victory (or loss):

1. During the autonomous period, robots could move past a predetermined line on the field to gain points for their alliance. If all three robots in an alliance did so, they also gained a "ranking point" (the primary metric used to determine preliminary seeding of teams). The terminology I use in this report, "autorun", refers to the ability of any given team to cross that line autonomously.
2. During the teleop period, robots could carry and deliver "power cubes" to their alliance's side of the field; alliances could store those cubes to use them for "power-ups" during the match, which granted them various score bonuses (usually for a duration).
3. Near the end of the match ("endgame"), robots could climb a central structure to score additional points for their team. If all three robots in an alliance did so (whether by climbing onto that structure, climbing onto another robot that had already climbed, or using a "levitation" power-up without actually climbing), they also gained a ranking point.

Questions Explored

The primary question I explored was "What contributes to an FRC match win?". This question can be examined with respect to the teams on an alliance (that is, "How much does the presence of a particular team contribute to their alliance's overall score?") or the objectives achieved by each alliance during a match ("How much does achieving a particular objective contribute to an alliance's overall score?"). During my initial work, I tried to answer the first question, but for reasons I'll explain in the next section, I was unable to do so. Then, I turned to answering the second question, which proved to be much easier with the data.

Data (and Regression Implications)

The data was obtained manually via a Python library that interfaces with the API of the website storing the data. The Python code that was used to generate the CSV files that were then analyzed in R will also be included in this report.

I ran into very few measurement issues while examining my data. During my initial exploratory analysis, I focused on a single team (out of curiosity, I used my old high school team, FRC team 1160) in order to see if I could find any trends in that team's performance over the course of a year. However, I noticed that one of the competitions attended was an "off-season", non-official competition that didn't report all of the metrics I wanted (namely, whether or not robots crossed the auto line during the autonomous period). Since there was no reasonable way to impute that data, I decided against the time-intensive alternative (filling in the blanks myself by searching volunteer recordings of matches and then recording the missing data from those videos) and restricted the scope of my work to just official competitions, which weren't missing any data.

However, by far the biggest problem I had in the course of my work was the lack of overall data. This wasn't caused due to any data collection issues, but rather due to the small number of matches that most FRC teams play during any given regional competition or even any given competition season. To give a sense of perspective, the 2018 Las Vegas regional had 104 matches played between 44 teams. These are fairly average numbers. Now consider a model that attempted to predict an alliance's score based off which teams were participating in that alliance. While this model would hypothetically be able to directly answer the primary question (How much does a team contribute to its alliance's score?), it would have accurately predict 44 weights from 104 data points. At most regional competitions, teams are guaranteed eight to ten preliminary matches; even then, the model would still have to account for the varying combinations of alliances for those matches. The sheer number of interaction terms and individual weights would result in any model overfitting to the noise in any set of regional competition data. Even if you were to train the model on data from additional regional competitions, the core issue still holds: for perspective, around 4,000 teams participate in FRC; in 2018, about 17,000 matches were played. As you add more match data, you invariably add more teams you should account for in your model.

But what if you were to lower the number of predictors? Let's say I only wanted to examine the performance of five teams at the 2018 Las Vegas regional. At first blush, this seems much more doable—you will have a much better time fitting a model with five predictors to 104 data points. And to

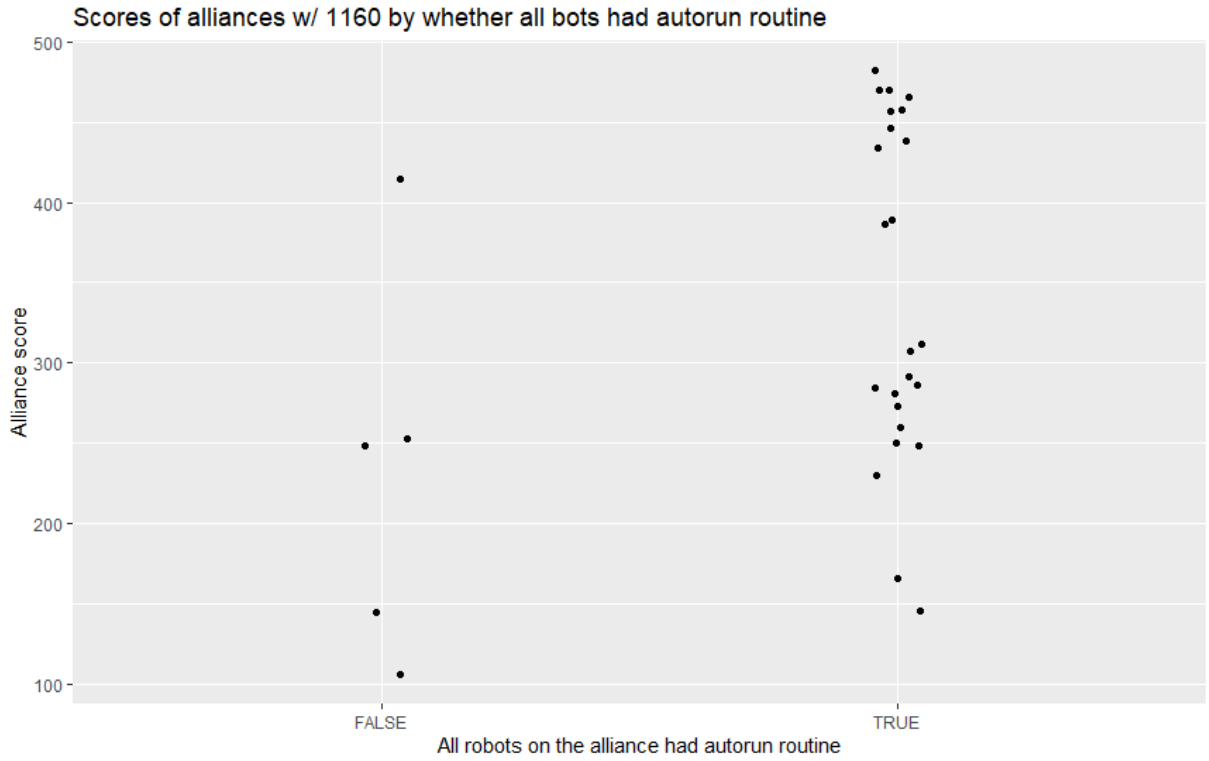
an extent, this tactic does yield some interesting results: I found that more often than not, the relative weights of the predictors are correlated with the preliminary seeding of the corresponding teams—that is, predictors with higher weights corresponded to teams with higher rankings. However, I found that these models had much worse predictive performance, which makes sense: if you're only examining five teams, then you're virtually guaranteed in every match to be unable to explain some portion of the final score. What if only two of the five teams you're examining are on an alliance for a particular match? What if none of the relevant teams are participating in a match? Those models had incredibly high errors because of their lack of predictive power. If the matches were only played by various combinations of those teams, then the models would better fit the data.

There's one last question worth discussing—what relationship should I look for? What the data lacked in the raw number of observations, it made up for in the sheer details of each observation. If I wanted to, I could make a "perfect" regression by fitting the margin of victory to the score difference during the autonomous period, the score difference during the teleop period, and the score difference during the endgame period. Since each alliance's final score is the sum of its autonomous score, the teleop score, and endgame score (before adjusting for referee fouls, which are also provided in the data), calculating the difference in final scores would be trivially easy; there would be no need for any sort of regression. Thus, I approached this question not from the perspective of someone with a (practically) omniscient view of every match, but rather from the perspective of a match spectator or high school team member—someone who recognizes general trends in what winning alliances do but isn't meticulously tracking every possible statistic during every single match. This is also a more realistic approach—teams commonly scout each other and attempt to create their own assessments and predictions of which teams will perform well based on their robot capabilities and match performances. They will track statistics relating to whether an objective was scored, and if so, how many times. This data is also useful in assessing which objectives are critical to winning a match, as the average robot cannot do everything in the average match.

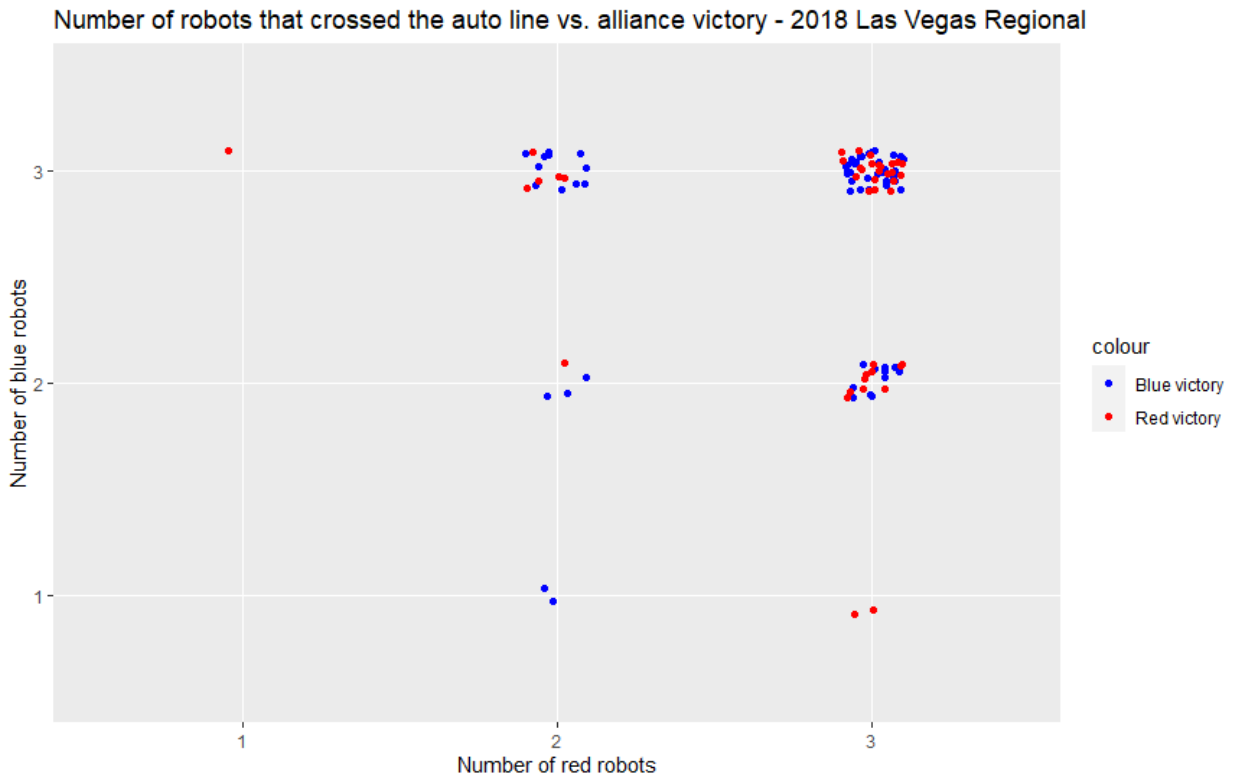
To fit my model, I used competition data from the 2018 Las Vegas regional. To check the performance of my model, I also used competition data from the 2018 Aerospace Valley regional and the Hopper Division of the 2018 Houston world championships. This data is available at thebluealliance.com.

*I was originally going to attach my R and Python code to this report, but this results in strange font width issues in the final pdf. Instead, **I'll upload everything**—data and code—**to a Github repository**, which you can access here.

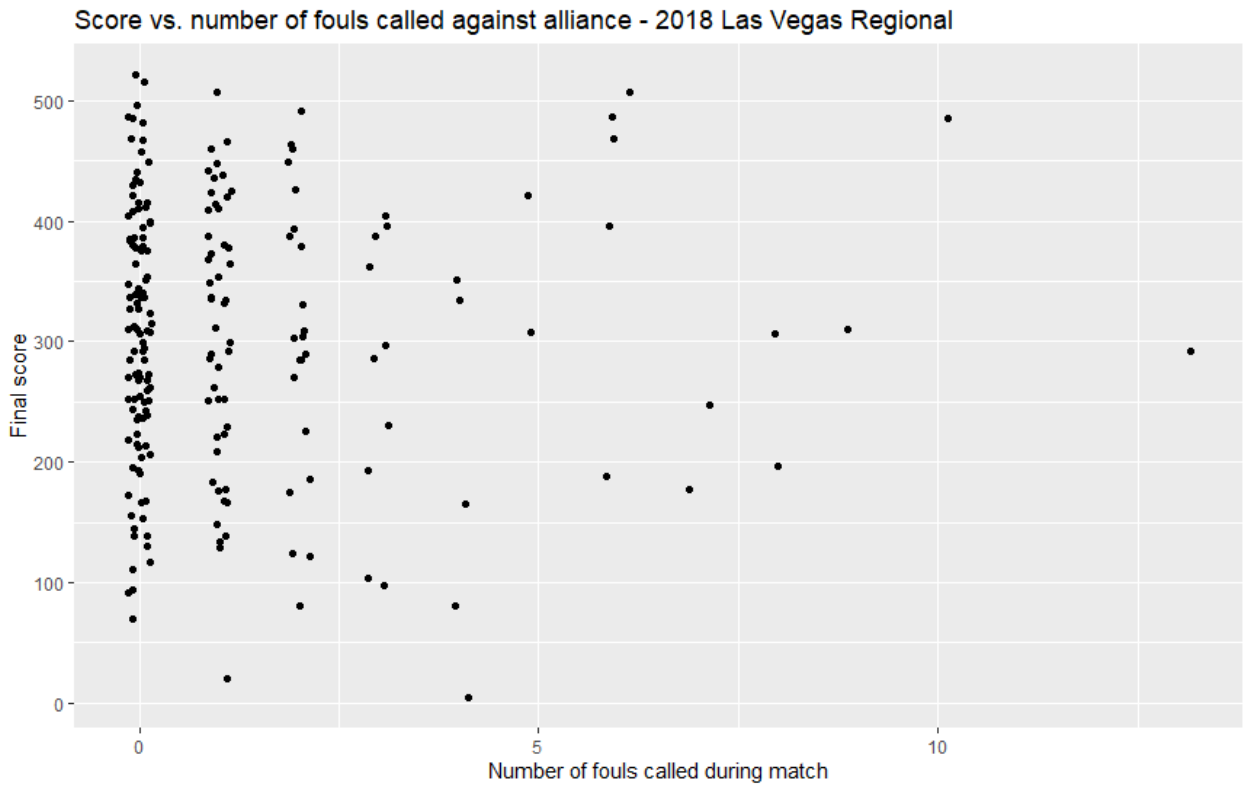
Preliminary Graphs



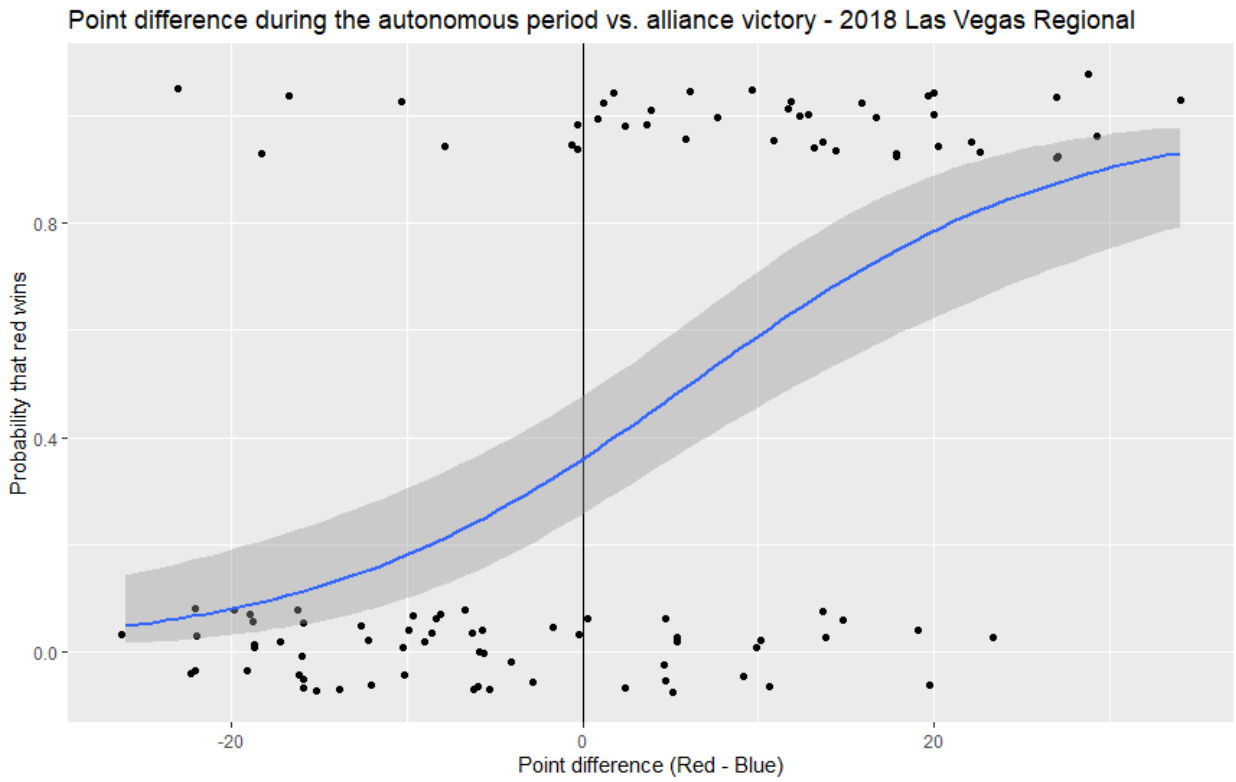
The above graph of the distribution of alliance scores by whether all robots had an autonomous routine to cross the auto line implied that there might be a correlation between the two variables, as there was a higher frequency of scores at higher values when all of the robots crossed the auto line. This graph also shows the dangers of focusing on only one team (this graph examines matches where team 1160 played)—there isn't enough data to draw conclusions. I repeated the same graph but with a few modifications: instead of tracking alliance score, I tracked match victor, and instead of tracking whether all robots crossed the auto line, I tracked how many robots (from each side) crossed the line:



This graph shows that in most matches, all six robots successfully cross the auto line. In matches where that doesn't happen, there isn't enough data to safely claim that crossing the auto line directly affects which alliance wins. This is expected: there are many more factors in a match that contribute to the final scorelines on both sides. However, I decided to include this as a predictor variable in my final regression because it was the easiest autonomous period statistic to track (from the perspective of a match spectator) and because it still carried importance (see model explanation).



Next, I explored whether there was a correlation between the number of fouls called against an alliance during a match and that alliance's final score. There was no considerable correlation, implying that I probably didn't have to worry about the impact of referee fouls on final scorelines. There was no considerable benefit to playing dirty or not dirty; the referees seemed to be good at making appropriate calls to balance scores as necessary. Thus, I excluded the number of fouls from my list of predictors in my final regression.



This graph shows the relationship between alliance victories and score difference at the end of the autonomous period. The points to the right of the vertical line at x = 0 show matches where the red alliance had more points than the blue alliance after the autonomous period; the points to the left show matches where the blue alliance had more points. When examining data that corresponds to y = 1 (a red alliance victory), most of it is concentrated to the right of x = 0, which makes sense, as the red alliance has the upper hand in those scenarios. The same conclusion holds for the data that corresponds to a blue alliance victory, which is mostly concentrated to the left of x = 0. However, I chose not to include this predictor, as the autonomous period scores also take into account how long each alliance has ownership of certain objectives (besides previously discussed factors, ex. whether or not the robots crossed the auto line). These statistics are more difficult to manually track, as they are time-based. Additionally, I felt that the inclusion of the aforementioned auto line statistic captured enough information about an alliance's performance during the autonomous period.

Model Used

The model I ultimately settled on related the margin of victory to the following predictors:

1. The number of red teams that crossed the auto line during the autonomous period
2. The number of blue teams that crossed the auto line during the autonomous period
3. The number of power cubes that the blue alliance collected during the teleop period
4. The number of power cubes that the red alliance collected during the teleop period
5. The number of red teams that climbed during the endgame period
6. The number of blue teams that climbed during the endgame period

What these predictors share in common is that they are all easily countable by match spectators. None of these predictors are dependent on time (as mentioned before, the rest of the mechanics in this game are dependent on the time they are controlled by an alliance). Additionally, they all underscore important core features of the ideal robot:

1. Robots should be able to have a basic autonomous procedure. If they don't have the capability to autonomously move forward a few feet over fifteen seconds (a very simple task), then the overall quality of the robot should be questioned.
2. Robots should be able to repeat tasks consistently during the teleop period. The process of obtaining power cubes on the field and delivering them to their side of the field is an important example of this. It relies on their human drivers being skilled enough to navigate the field while interacting with other robots (especially those from the opposing alliance). In addition, the power cubes collected can be used by the alliance to activate beneficial power-ups, which can change the direction of any match.
3. Robots should be able to climb during the endgame period. In terms of point value, climbing is quite heavily weighted. Having one robot capable of climbing in an alliance of three was typically perceived as "good" during the early weeks of the 2018 season, but as the weeks progressed, alliances had to be able to have at least two climbing robots to remain competitive. (Consider that in many cases, only one robot could actually climb the central structure. For "double climbs" to occur, one robot usually had to hook onto the other climbing robot and then elevate itself. Unsurprisingly, such piggyback setups were frequently met with one robot falling to the ground under the weight of the additional bot, leading to many points lost and many sad teams.)

Essentially, robots should be able to consistently score points during all phases of a match.

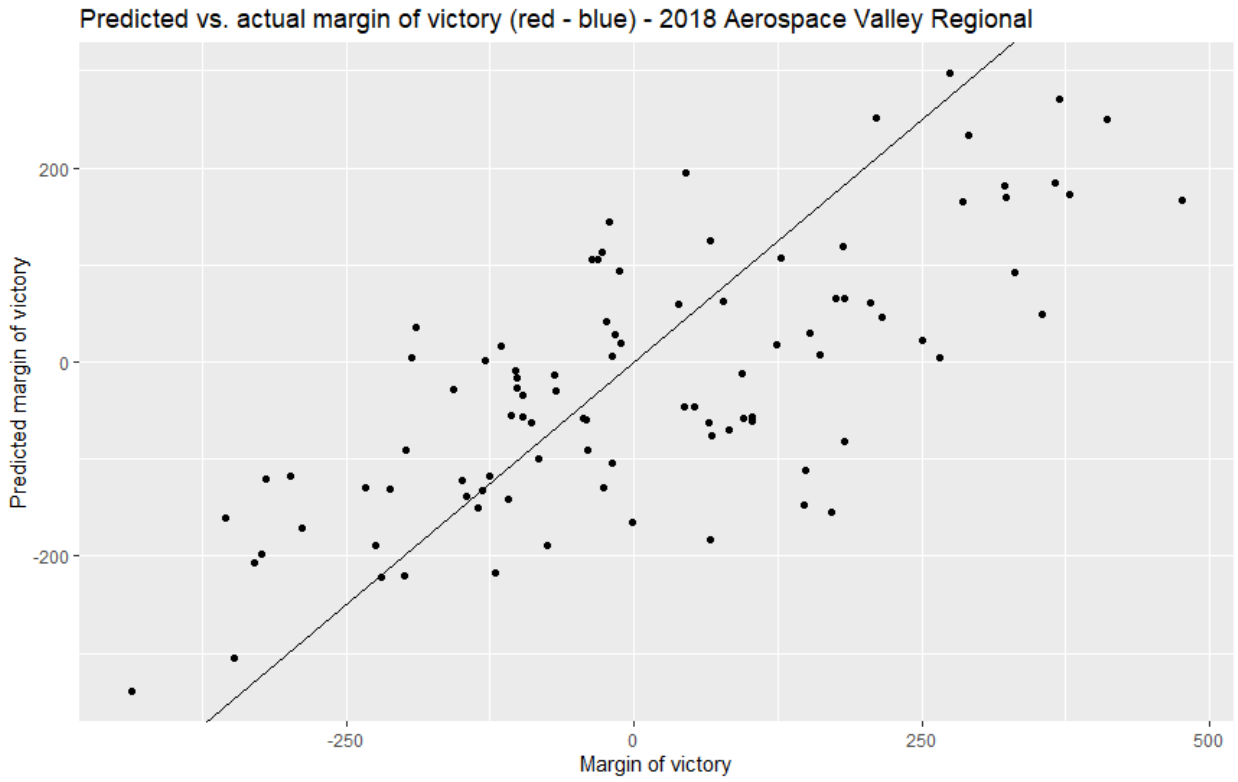
Results

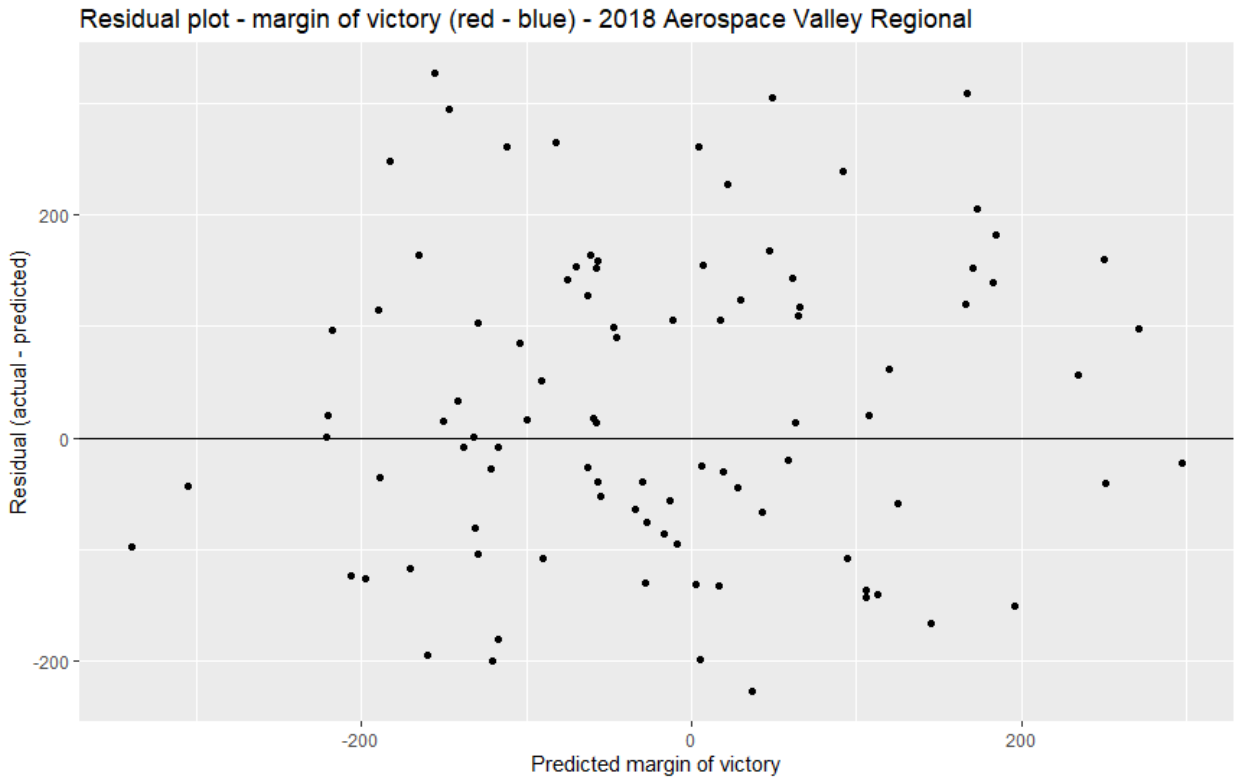
For each of the three datasets (Las Vegas regional, Aerospace Valley regional, Hopper division of the Houston Championships), there is:

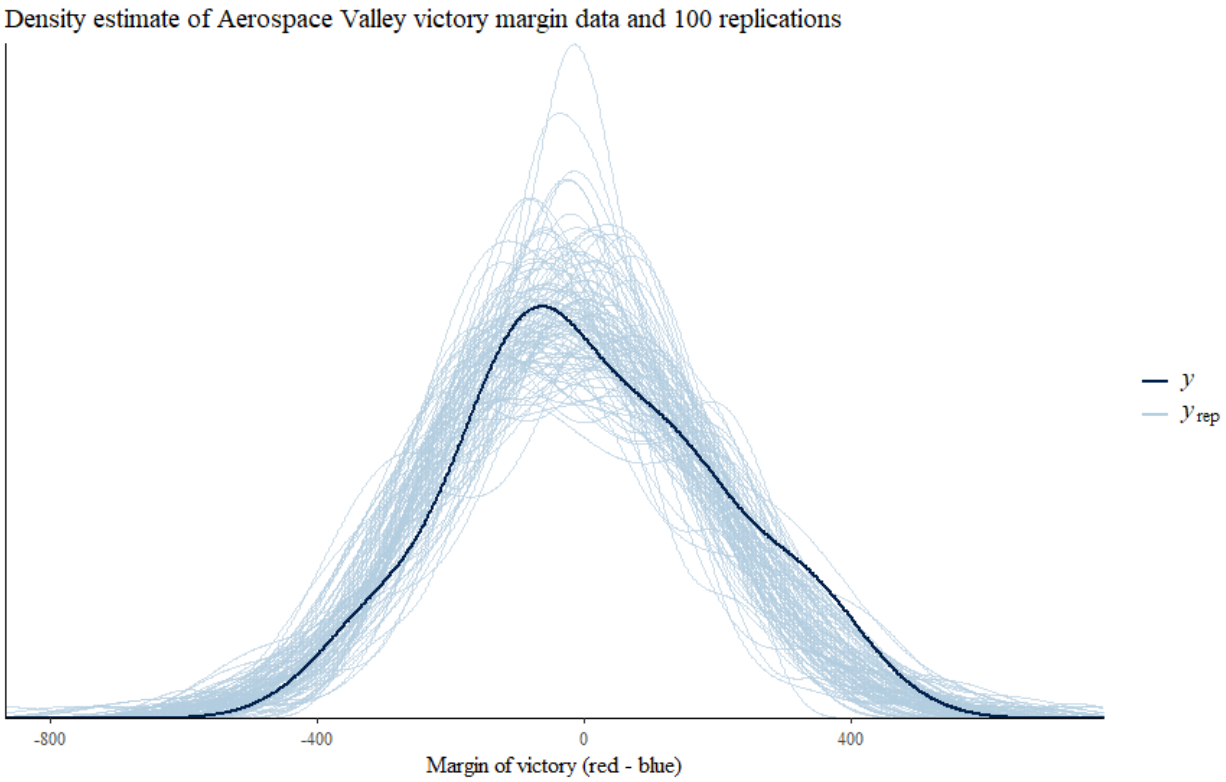
1. A graph showing the predicted vs. actual margin of victory
2. A graph showing the residual plot
3. A graph showing the actual distribution of the data and 100 replications of such (posterior predictive check)

Additionally, I defined an “accuracy” statistic, which calculates the proportion of observations for which the regression’s prediction of the victor (as measured by whether the margin is positive or negative) matches the original data.

Results – Aerospace Valley Data

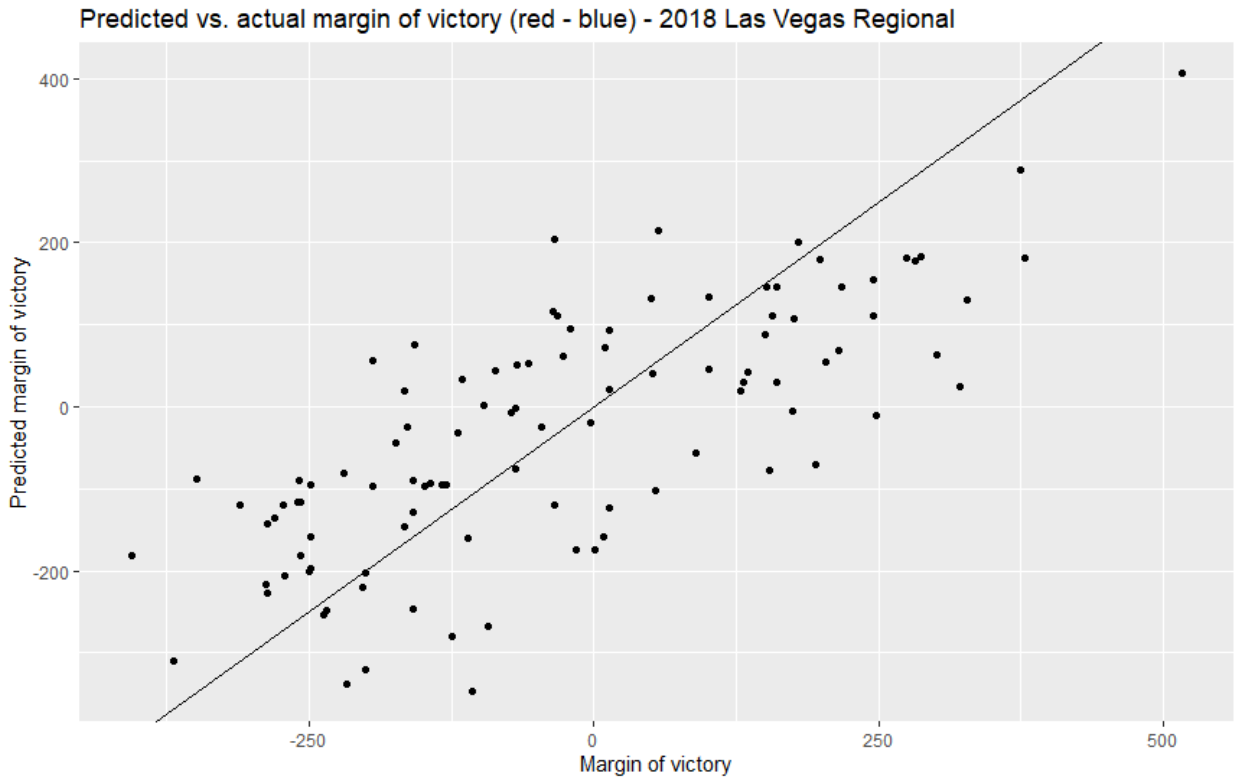


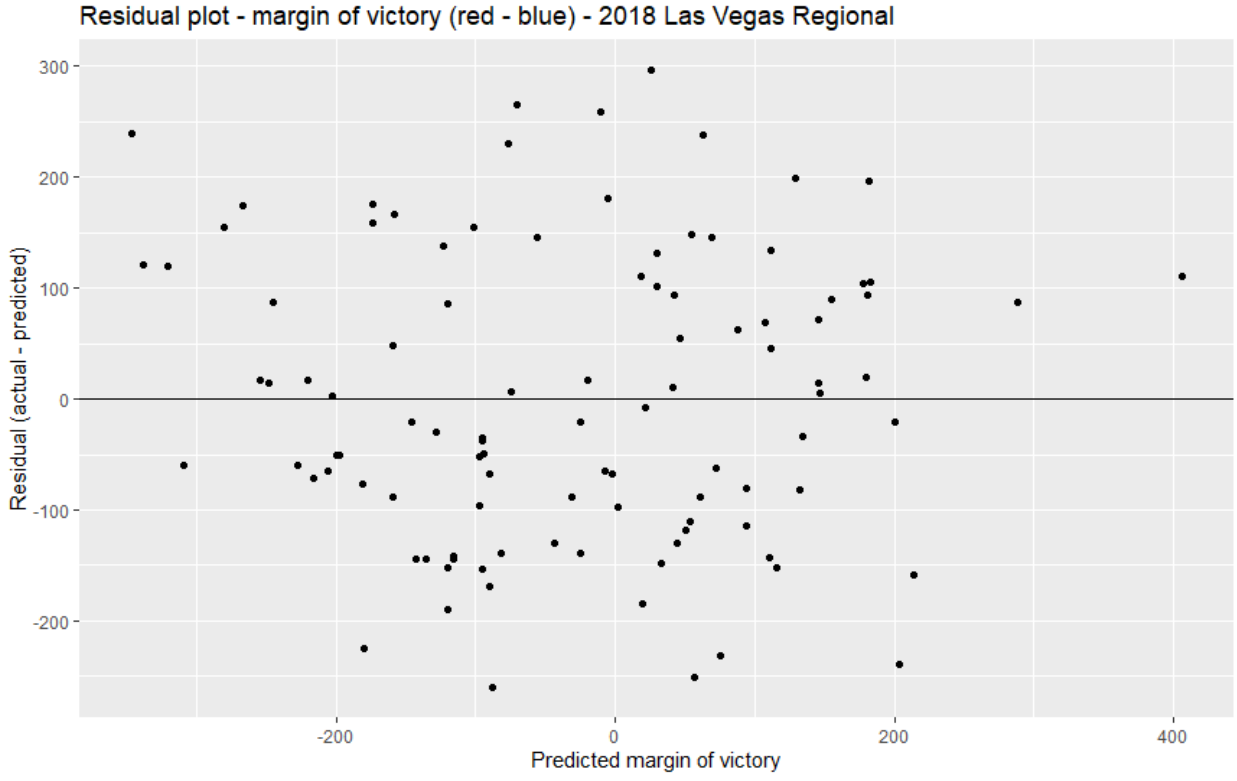


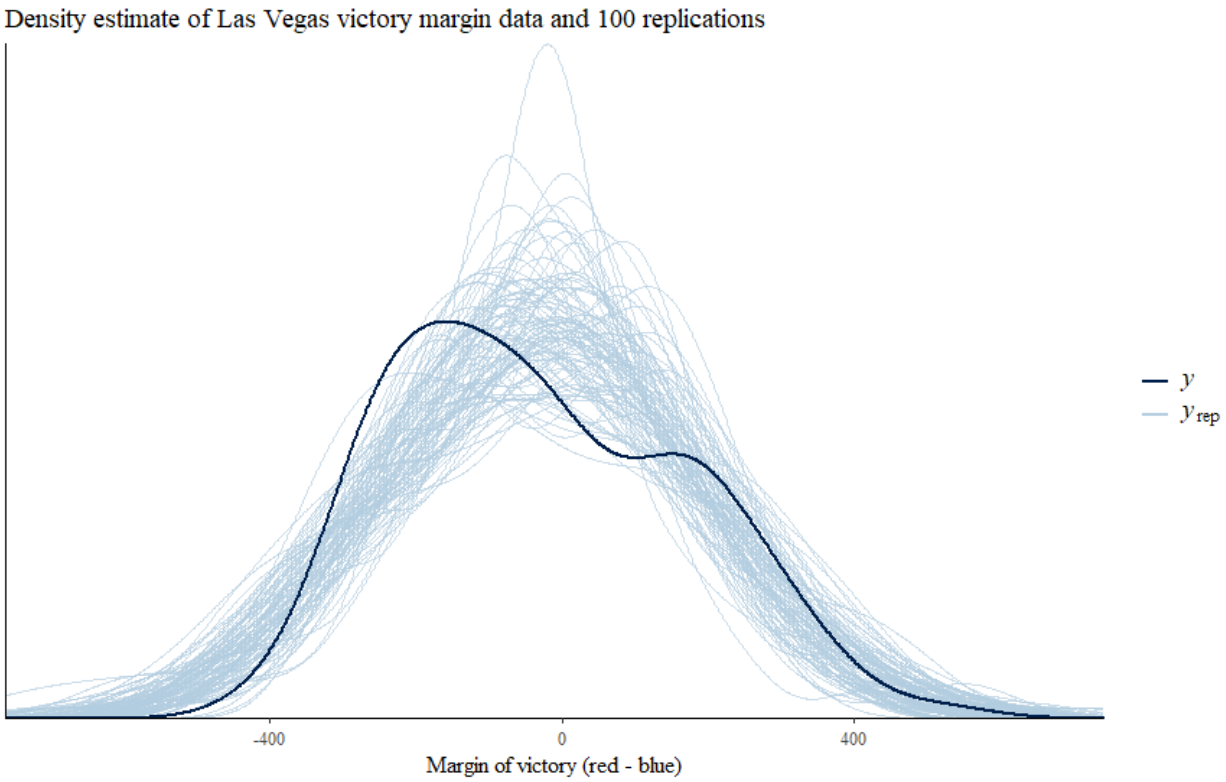


The accuracy statistic, as calculated by R, is 71.28%.

Results – Las Vegas Data

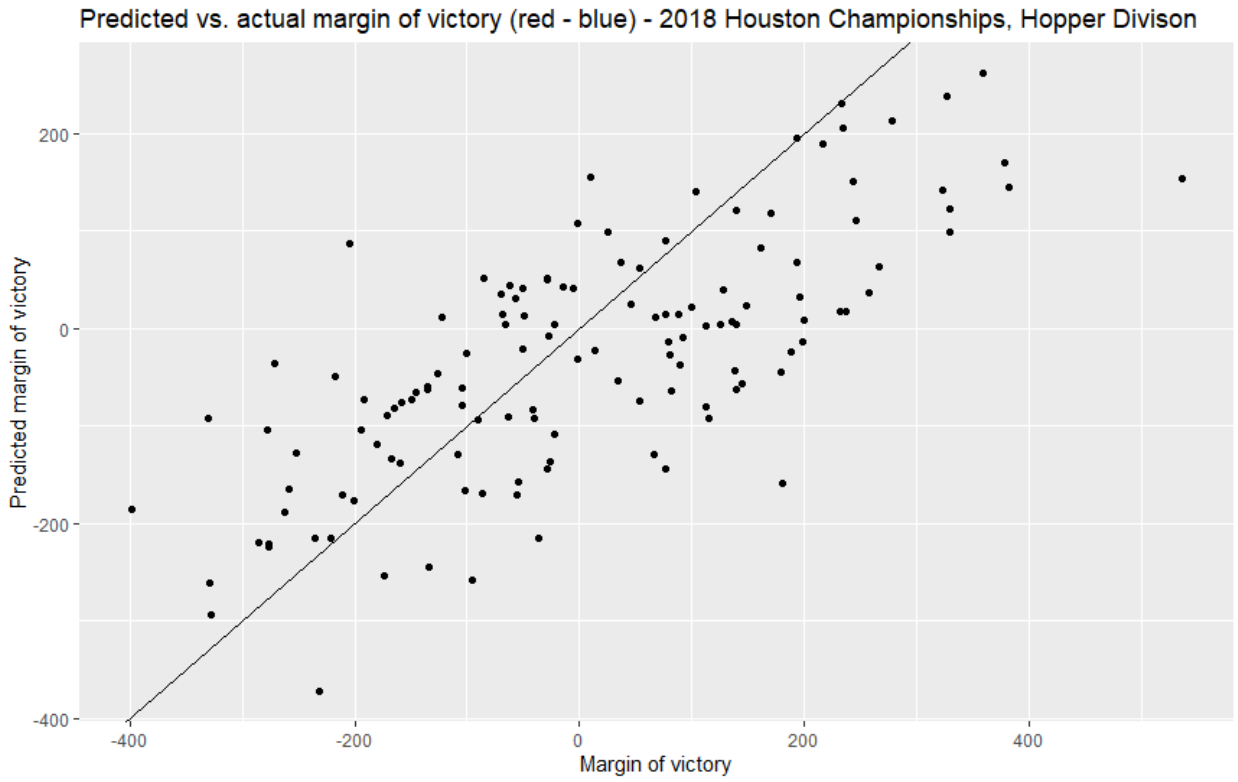


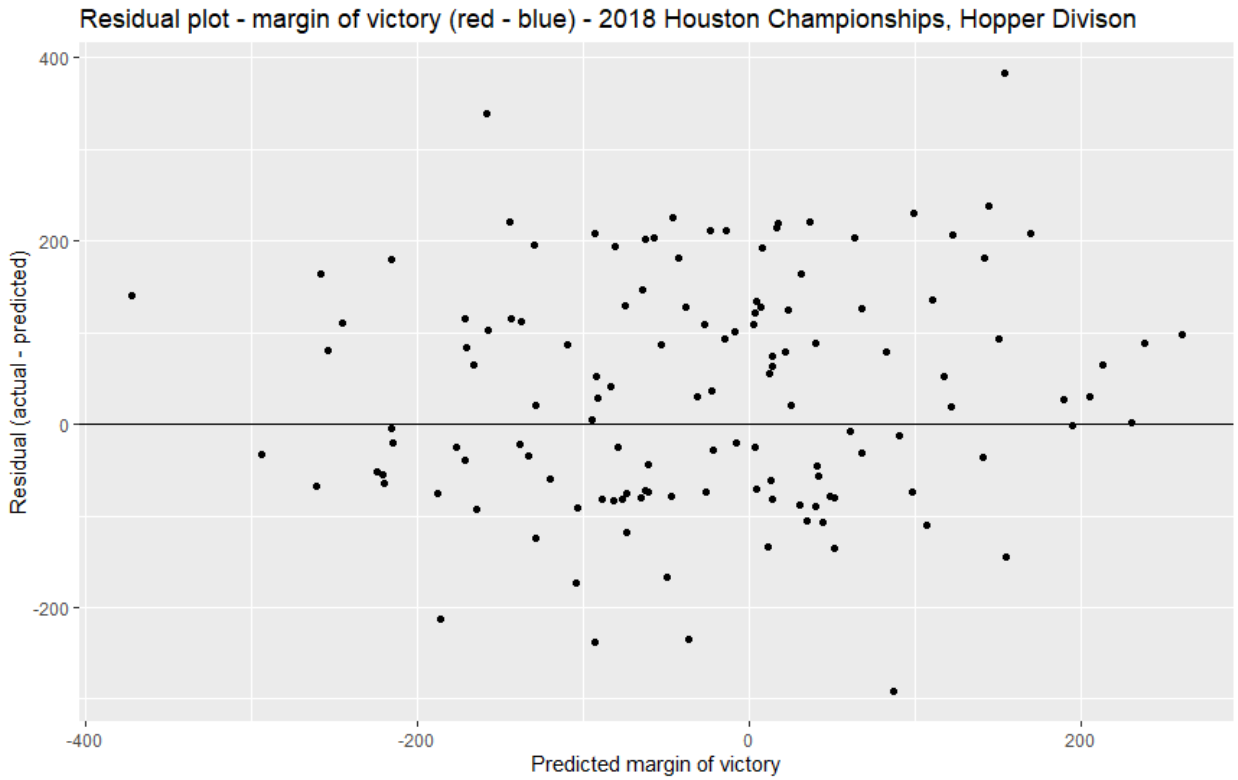


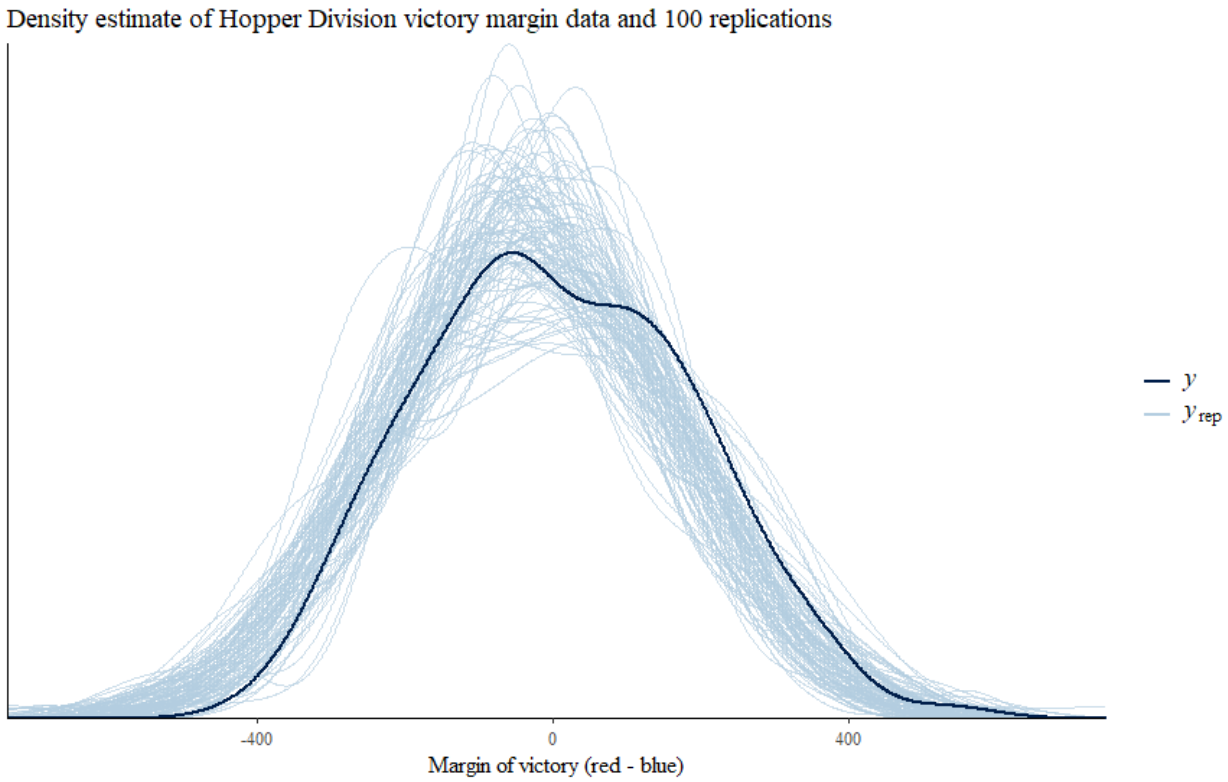


The accuracy statistic is 78.64%.

Results – Hopper Division data







The accuracy statistic is 73.08%.

Discussion

The residual plots of the data all exhibit random scatter, and the posterior predictive checks all show for the most part that the generated distributions are similar enough to the original distributions of victory margins.

Why did I separate my data by competition instead of pooling the data and then randomly selecting observations from there? I primarily wanted to account for competition-specific trends in player tendencies. As the competition season progresses, alliance strategies and playstyles usually shift; the so-called metagame also heavily depends on which robots are present at each competition and their capabilities. As an example, if Team A is in an alliance with Team B, it may be relegated to a "support" role (defense against opposing robots, achieving objectives worth fewer points) if Team B has a demonstrated ability to handle the heavy lifting of scoring points on its own. If Team A is in an alliance with a weaker Team C, it may be placed in a "captain" role where its alliance partners must ensure that the path is clear for Team A to score as many objectives as possible. If Team A is based in California and Team B (or C) is based in Massachusetts, it is likely that the two teams will never meet to begin with. As such, further work on this project would likely involve finding some way to standardize data from each dataset to a predetermined distribution.

Additionally, there may be value in disaggregating the predictors—instead of counting how many objectives an alliance scored, counting how many objectives each individual team scored would

allow for greater fine-tuning. Unfortunately, for this approach to work, it would be necessary to simulate enough data (again, based off predetermined information) to meet the needs of the model, there would not be enough existing data (see above sections) to ensure a reliable fit. Prior information on the relative abilities of each team would have to be incorporated into the regression, but one would need to be careful to ensure that bias in the priors is minimized wherever possible.

This marks the end of the report proper: to see my code and data, please refer to the Github repository link.