The Impact of an Automatic Strike Zone

MLB – Spring 2020

Devon Goetz
Department of Mechanical Engineering
dkgoetz@mit.edu

Zack Kopstein
Department of Mechanical Engineering
kopstein@mit.edu

Simran Pabla
Department of Electrical Eng. and Computer Science
simmyp@mit.edu

Claire Wichman
Department of Mechanical Engineering
cwichman@mit.edu

Christina Chase *
MIT Sports Lab
cchase@mit.edu

Anette E. Hosoi †
MIT Sports Lab
peko@mit.edu

May 12, 2020

Abstract

In any baseball game, umpire mistakes happen. Some accept this as a part of the game, while others seek to correct these imperfections; doing so would mean the implementation of an automatic strike zone. This paper aims to quantify the exact impact that implementation would have. It first examined the impact umpires have had on existing games, and found that they are causing real and tangible changes. Offensively advantaged mistakes lead to more walks, just like defensively advantaged mistakes lead to more strikeouts. Next, a predictive model was created in order to step through each game and create a new version, one without any mistakes. These new games saw fewer pitches when the offensive mistakes were removed, and more when the defensive mistakes were removed, showing that the correction of these games could lead to a more fair game.

Keywords baseball · strike zone · metric analysis · predictive algorithm

1 Introduction

In Major League Baseball, any play that transpires is ultimately at the discretion of the umpire. Many calls are obvious and clear, and the umpire plays little to no role in the outcome of the event. But inevitably, this is not always the case. What happens when the umpires make mistakes?

The MLB relies on humans to officiate all of the games, but humans are far from perfect. They carry inherent biases, are subject to external influences, and sometimes just make simple mistakes. We analyzed the difference these mistakes, when made at the plate, can make on the outcome of an inning. As we began to understand the impact these imperfections are having, we created a corrective and predictive algorithm to see how a game might transpire were it to be called perfectly. We then compared our manufactured "correct" games to the ones that had actually occurred, allowing us to draw conclusions about what a perfectly called game might look like, and how it might differ from reality.

The MLB currently possesses the ability to implement an Automatic Strike Zone, eliminating the need for an umpire to be calling balls and strikes. Before using it, however, they want to know how this addition might change the game, or

*Corresponding author
†Corresponding author

how it could be implemented to most closely mimic the good habits of human umpires, while eliminating the costly mistakes they might be making.

In this report, we will walk through our Approach and Methodology, as well as the results we have compiled. It will conclude with a discussion of the potential impact of this report, as well as the drawbacks and areas for improvement.

2 Methodology

This problem naturally lent itself into two parts. First, we quantified the impact umpires are currently having. In order to do this, we compared the metrics from existing innings with mistakes, to existing innings without mistakes. Once that was done, we then wanted to quantify how the game might have been different had no mistakes happened. We corrected mistakes and predicted new game outcomes, and then analyzed the difference between those.

2.1 Comparing Existing Games

We began by simply classifying mistakes. In order to do this, we separated pitches into four categories: correctly called balls, correctly called strikes, balls that were called strikes, and strikes that were called balls. Balls that were called strikes we labeled as defensive mistakes, because it is advantageous for the pitcher to have an extra strike, while we labeled strikes that were called balls as offensive mistakes.

Next, we created a new view in the Google Cloud dataframe for the mistake categorization, and from there were able to sum the number of mistakes made in an inning. Our first analytical approach was to examine the outcome of an inning, and see how those outcomes may vary depending on the number of mistakes made. Our four primary metrics are:

(1) Number of walks

(2) Number of strikeouts

(3) Number of runs scored

(4) Number of pitches thrown (as a way to estimate game time and duration)

We summed these for each inning, and put them into an additional column in our new dataframe view as well. Once we had this information, we were able to compare the average outcomes of innings with no mistakes, and compare them to a gradient of incorrect innings, trying to determine the influence of how many mistakes were made. We also broke it down one step further, and compared flawed at bats to at bats that had been called without mistakes. This allowed us to see a more immediate impact of the mistakes.

2.2 Correcting Mistakes

Knowing the effect umpire calls had on games that had already transpired was valuable in knowing the difference these mistakes were making. The next step we took was to create a model that allowed us to walk through a game and actually correct the mistakes that were made.

2.2.1 Transition Matrix

Creating a transition matrix was a major key in developing the predictive algorithm. Using the count, the number of outs, the difference between home and away score, the runners, and whether the inning is in the top or bottom we were able to define any situation that could happen in the game. Being able to represent the situations as these states allows us to build a transition matrix. This is a giant probability table that takes any single state and predicts what next state is most likely to follow. Figure 1 shows a small section of what our transition matrix has to handle.

There is the initial state at the top, and from there, any number of events can transpire. Some are more likely than others, like here the batter may be most likely to make an out. This moves us to our next state with a reset count and one more out. Or the batter could hit a triple, resetting the count, moving the runners, and scoring a run. And from each of those states, more situations follow. This graphic is a tiny sample of the actual breadth of the transition matrix; it accounts for every single state that could exist, and stores the probabilities for transitioning between any two states in the enormous state space.

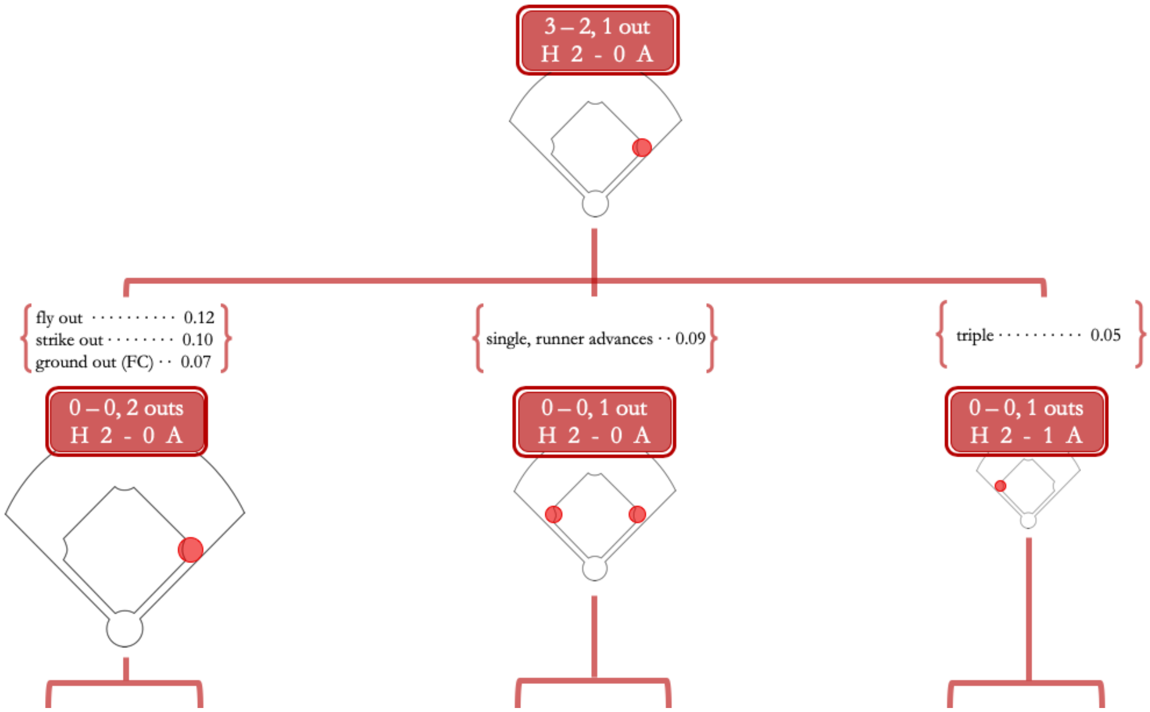

Figure 1: This shows a small portion of the transition matrix. It takes in a state (as defined by count, outs, run differential, top or bottom, and runner configuration) and contains the probabilities of that state moving into any next possible state.

2.2.2 Creating a Predictive Algorithm

When our algorithm identifies a mistake, it creates a new, corrected state. For example, if the count is 0-1 and a pitch was mistakenly called a strike, the count moved to 0-2. We identify the mistake, and correct the count to 1-1. Using this new state, we calculate the most likely next state to occur, and proceed through. Figure 2 shows the first few innings for one team's offense. On that third batter, we identified the second strike as a mistake, and made it a ball instead.

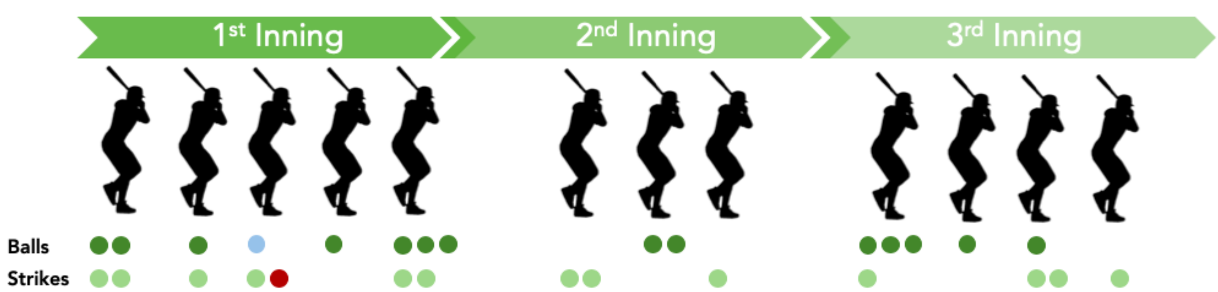

Figure 2: As we step through each pitch of the game, we look for mistakes. Once we find one, we correct it. Here, there is a ball mistakenly called a strike, so the count is corrected from 0-2 to 1-1.

We then use our transition matrix to take the new state and proceed through the at bat. We use the matrix until we reach the end of the at bat, at which point we have a new at bat outcome. From here, three things can happen, as shown in Figure 3. The at bat outcome can remain the same, the batter can get out when they previously got on base, or the batter can get on base when they previously got out.

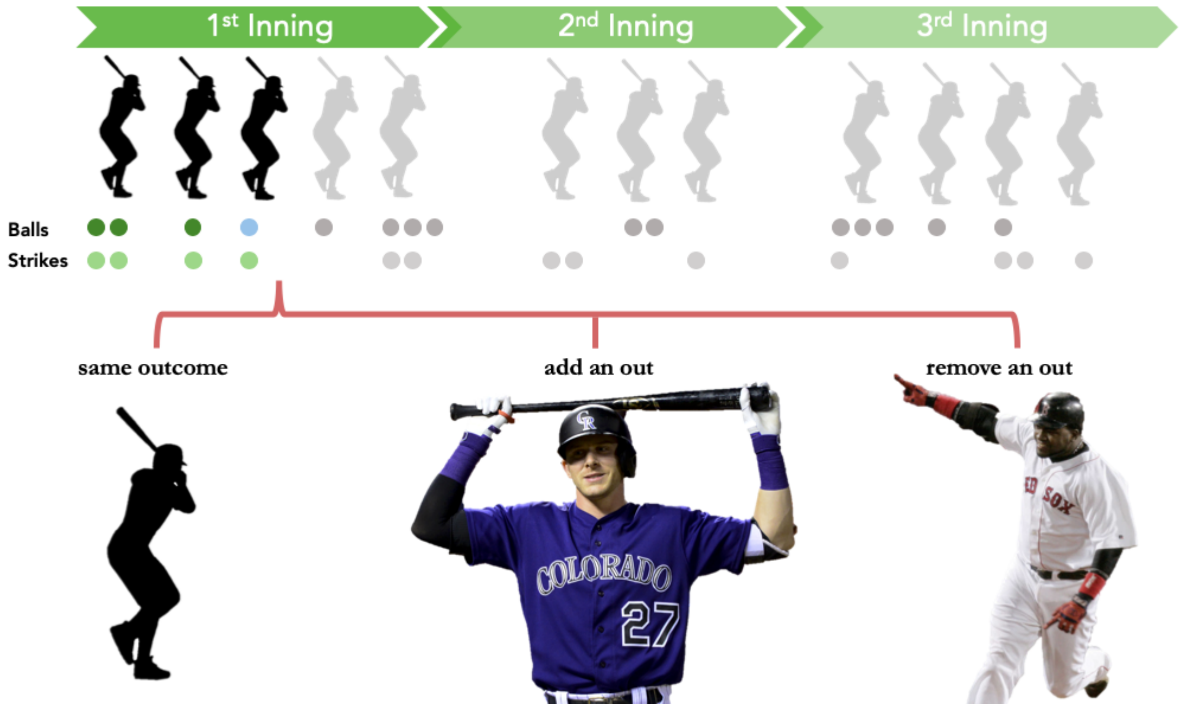

Figure 3: Above we see the existing game state. After we've changed the at bat, there are three possible paths. The at bat outcome can be unchanged, we can add an out, or we can remove an out.

2.2.2a Unchanged Outcome

After we correct the out, our transition matrix may output a state that has the same effect as the original at bat. It doesn't matter if the fly out turned to a ground out, the result was the same. Our algorithm can just stitch reality back on.

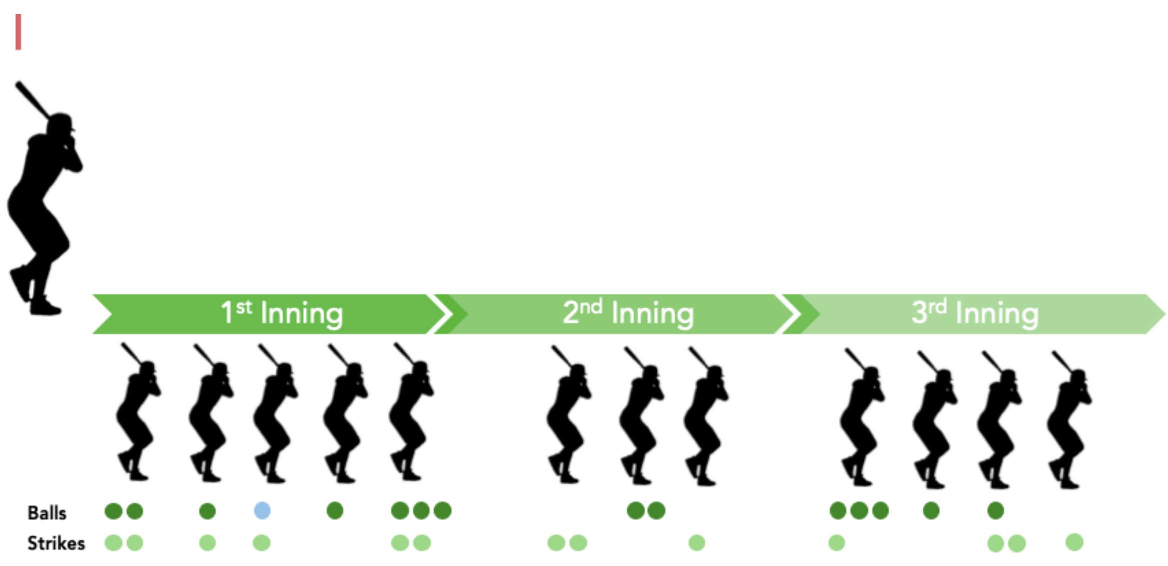

Figure 4: After the at bat ends in the same way, we can just stitch reality back on.

2.2.2b Add an Out

In reality, the batter would gotten on base but in our corrected at bat, the batter gets out. We remove that batter from the bases and stitch reality back on from the new state of the game, which has one more additional out. We have to terminate the inning earlier because we've added an out, so now there are a few at bats from reality that get deleted.

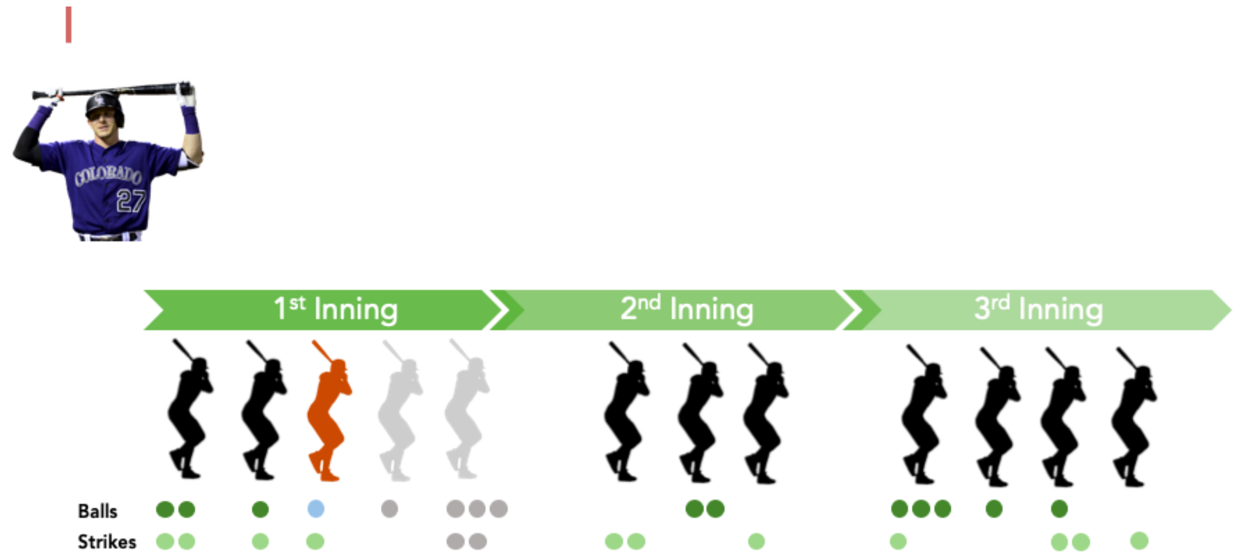

Figure 5: When we add an out, we have to terminate the inning earlier, because we don't have enough outs to allow for all the batters to continue.

When we go to stitch reality back on, however, there is a new runner configuration. In Figure 6, we see that the batter who got on was taken off, and the lead runner moved backwards. We move the new runners with a probability function. We wanted to preserve as much of reality as possible, so this function takes in the real out come of at bats and tells us based on probability and some random perturbation, where the runners will move.

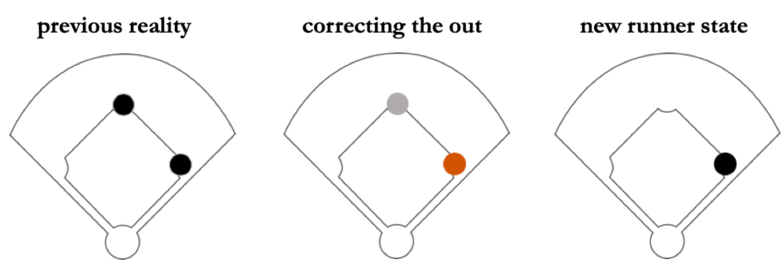

Figure 6: The real inning had runners on first and second, but the runner on first got out in the new predicted at bat. We have to take that runner off the base and restore the pre-at-bat runner configuration.

2.2.2b Remove an Out

This time, we encounter a batter who got out in reality, but in our predicted game, that batter got on base. We add the new baserunner, but this time we also have to add more "fake" batters because otherwise we'll run out of real batters before we reach 3 outs. We continue through our transition matrix until we get back to the original number of outs, and then stitch reality back on, once again moving the runners with our probability function.

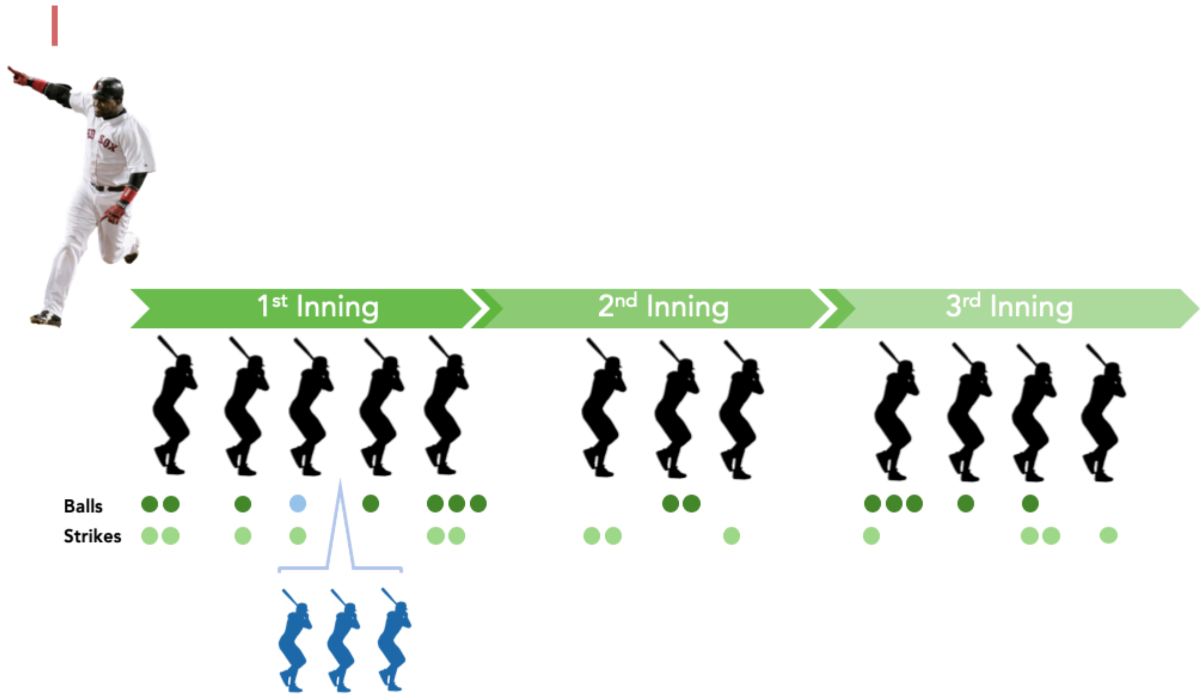

Figure 7: If we were to continue with reality after removing the out, we would reach the end of the inning before we reached 3 outs. Our algorithm adds in fake batters, using the transition matrix to create their outcomes, and then once we've returned to the original out state, we can stitch reality on.

For example, maybe this fourth batter only had a runner on first when he hit a single in reality, so no one scored. But maybe now, based on our introduction of fake batters, the bases are loaded when he hits that single. Again trying to maintain as much of reality as we could, we take the real outcome of the at bat and use our probability function to move the runners.

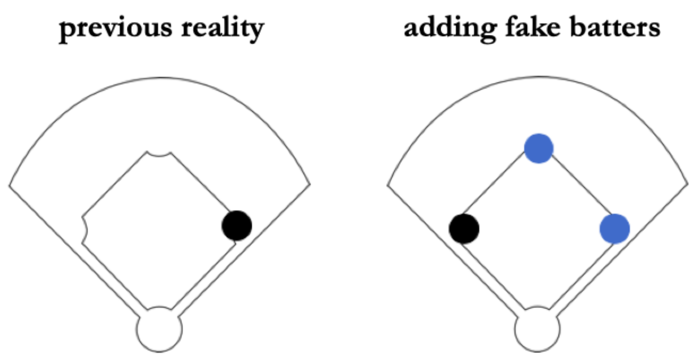

Figure 8: When the fourth batter originally stepped up to the plate, there was a runner on first. Now, because of the changed outcome previously, the bases are loaded. We use our probability function to tell us how these runners will move based on the true outcome of the batter's at bat.

In summation, our predictive model is meant to find and correct a mistake, complete the associated at-bat, and remove or add batters as needed to align these modifications of the game state with the state that allows us to return to reality.

With these new "corrected" games, we tallied the same metrics as before (strikeouts, walks, runs, and pitches) in order to compare the games that actually transpired and the corrected games that we manufactured. This gives us insight into the way an automatic strike zone could affect the games and how baseball would look were there to be no mistakes behind the plate.

3 Results and Discussion

3.1 Comparing Existing Games

As mentioned in the 2 section, and specifically enumerated in the 2.1 subsection, our first portion of the project was to see how mistake free innings compared to those with mistakes.

Before that could happen, it was important to understand how many mistakes are actually happening in games. The columns in Figure 9 represent the number of innings that occurred with the number of mistakes indicated on the x axis. The figure tallies the total number of mistakes of an inning; for example, there were 67,174 innings with no mistakes, and 792 innings with four mistakes. It also splits these mistakes into offensive and defensive mistakes; there were 19.120 innings that had one offensive mistake, and 38,012 innings that had one defensive mistake.

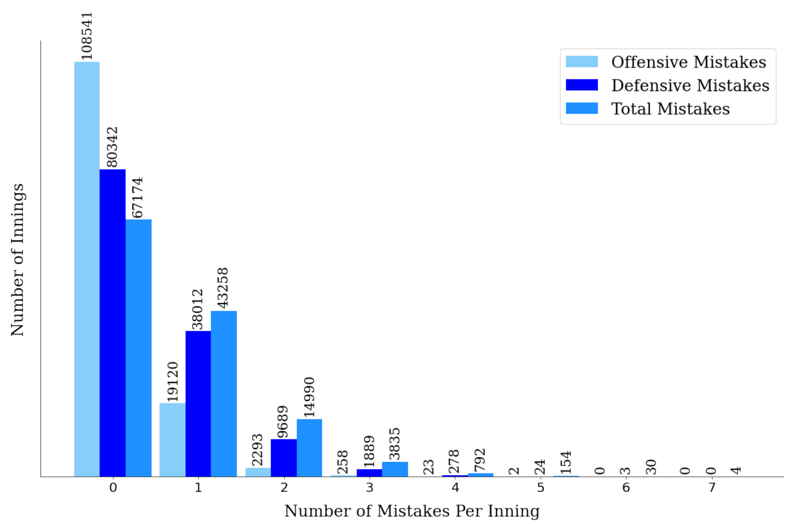

Figure 9: A bar chart that displays the total number of innings that have a certain amount of mistakes. There are three categories: offensive, defensive, and total. There is a clear downward trend, and the majority of the innings have between zero and two mistakes.

Offensive mistakes are the least common, as there are the most innings with zero offensive mistakes, and the offensive mistake bar is well below the others in all other categories. Overall, Figure 9 would indicate that umpires generally do a good job at correctly calling balls and strikes. When they are messing up, it is more often in the favor of the defense, meaning pitches outside of the strike zone are being called strikes.

When we did break out into innings, we found surprising results. The type of mistake did not affect the metric; the total number of mistakes was more influential. Innings with. a lot of offensive mistakes had similar metrics to the innings with defensive mistakes, which does not follow the instinct which tells us offensive mistakes should help the offense produce more runs. Figure 10 is a table that contains the differences between mistake innings and perfect innings across all four metrics. Each inning was grouped by the number of net mistakes they contained (calculated as the number of offensive mistakes minus the number of defensive mistakes). The average number of each metric was taken across that group, and then the average was taken across the innings with no mistakes. The perfect inning averages were subtracted from the mistake innings to find the delta values. This is the difference that is displayed, and any of the statistically significant results are bolded. As a note, the Net Zero column was only innings with a net zero value, not a total zero value. The total zero mistake innings were used as the reference for all cells, as previously mentioned.

From this, we concluded that the number of pitches was driving all of our metrics, including mistakes, rather than the other way around. Innings that had more pitches had more chances for umpires to make mistakes, just like they had more chances for walks and runs and strikeouts.

better for defense ← **Net Mistakes per Inning [Offensive - Defensive]** → better for offense

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5
ΔPitches	23.32	11.71	8.97	6.55	4.11	2.08	4.96	2.96	6.2	9.12	13.9	8.82
ΔRuns	2.62	1.39	0.87	0.63	0.34	0.16	0.43	0.26	0.64	1.16	1.69	-0.38
ΔBB	1.24	0.41	0.39	0.32	0.19	0.09	0.32	0.22	0.51	0.84	1.35	1.74
ΔK	1.12	0.9	0.56	0.44	0.29	0.15	0.16	0.04	0.04	-0.05	-0.03	0.12

Figure 10: The innings were grouped by the number of net mistakes they contained (offensive mistakes - defensive mistakes), and then each metric was averaged across that group. The average across perfect innings was subtracted from that average. We can see the trends all increase regardless of which team had the advantage.

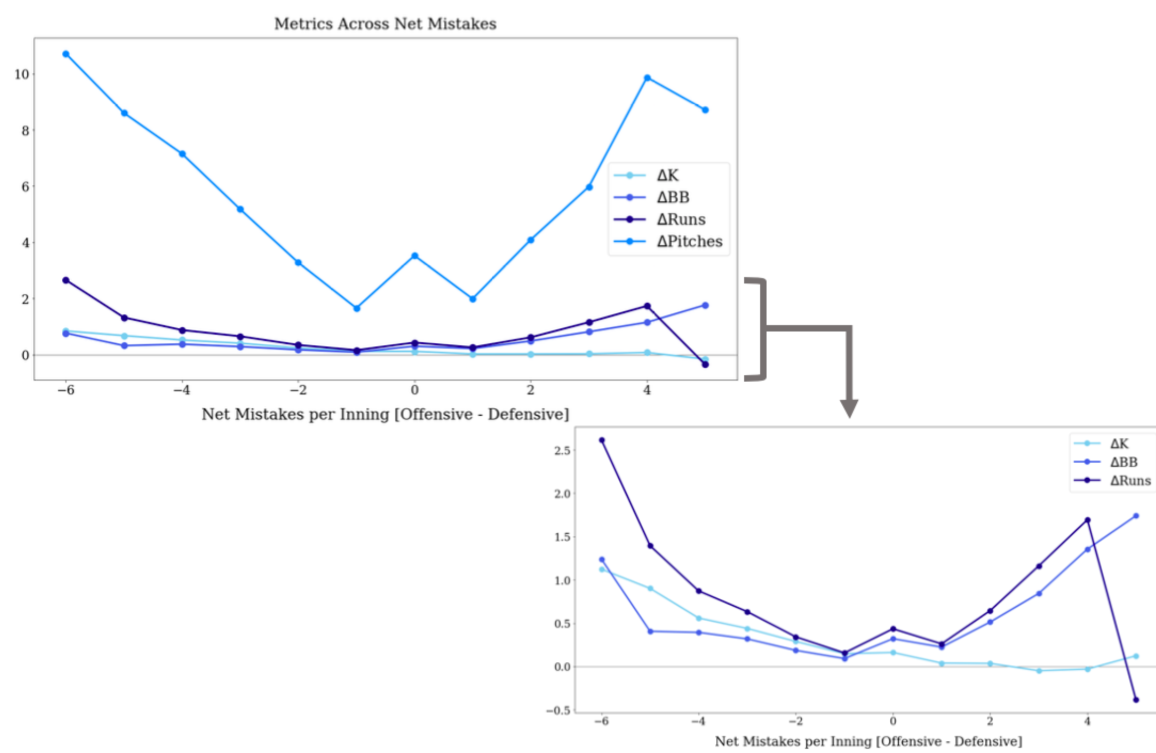

Figure 11: When graphed linearly, the trends from the Table in 10 become clear. No matter what kind of mistake is made, as the number of mistakes increase, the metric differences get more drastic.

Figure 11 shows this parabolic tendency of the metrics. Regardless of how the mistakes were helping or hurting a team, everything was going up. The walks, strikeouts, and runs look linear when on the same scale as the pitches, but when broken out it is clear they act parabolic as well.

The next step taken was to break this analysis into at bats so as to see the impact of these mistakes. By looking on a more granular scale, we were able to eliminate other confounding variables and uncover the effect of the mistakes.

Figure 12 shows the trends for all four metrics across net mistakes per at bat. For walks, we see a distinct increase when there are more mistakes that benefit the offense, and we see the opposite trend in strikeouts. Strikeouts increase as there are more defensive mistakes. This confirms that umpires are, in fact, influencing the outcome of at bats in a real and tangible way.

better for defense ← **Net Mistakes per At Bat [Offensive - Defensive]** → better for offense

	-3	-2	-1	0	1	2	3
ΔPitches	**1.22**	**1.4**	**0.91**	**1.75**	**1.14**	**1.62**	**1.83**
ΔRuns	**-0.11**	**-0.09**	**-0.04**	**-0.05**	**-0.02**	**-0.04**	0.03
ΔBB	**-0.07**	**-0.02**	**0.01**	**0.12**	**0.14**	**0.37**	**0.77**
ΔK	**0.8**	**0.42**	**0.16**	**0.12**	**-0.01**	**-0.08**	**-0.2**

Figure 12: At bats were grouped by the number of net mistakes they contained, and then these mistaken at bats were compared to at bats with no mistakes. Net mistakes are (offensive - defensive), the delta values are calculated by subtracting perfect at bat averages from mistake at bat averages, and the bolded values are statistically significant.

In Figure 13 the line graphs for each metric is displayed. the pitches follow no real pattern; this confirms the hypothesis that the analysis itself is biased against this metric, because any pitch logged as a mistake was not a pitch hit into play, forcing the pitch count upwards. But it also means that the pitches are no longer driving the other metrics, so conclusions the influence of mistakes can be more readily drawn. The runs have a very slight increase as we increase the offensive advantage, but this is more situational and difficult to influence in a single at bat. Runs also display very little change, with an extremely slight upward trend as net mistakes increase and we lean towards an offensive advantage. The last data point is more dramatically negative, but is not statistically significant. The most prominent trend lines are, again, the strikeouts and walks.

Figure 13: This figure graphs the trends from the Table in 12 linearly. While pitches have no pattern, and runs have a very small one, walks and strikeouts have strong trends. Both, however, give a greater advantage than disadvantage. Fore example, there is a more drastic increase in walks when there are offensive mistakes than there is a decrease in walks when there are defensive mistakes.

Another interesting conclusion we can draw here is that the mistakes in one team's favor have a greater impact than the mistakes against a team. When there are more offensive mistakes, or net positive, there are a great deal more walks, but a smaller decrease in strikeouts. There certainly are fewer strikeouts, but the difference is of a much smaller magnitude than the difference in walks. Similarly, when there are more defensive mistakes, there is a larger increase in strikeouts than there is a decrease in walks.

In order to increase granularity even further, the at bats were classified by their combination of offensive and defensive mistakes, instead of by their net mistakes. Figure 14 shows the tables for each metric broken out by these mistake combinations.

The difference in pitches was still an overall positive one, with values getting larger as the total number of mistakes increases, an expected outcome again due to the nature of the analysis.

At (1, 1) on the Strikeout Table, there is a positive delta value, perhaps suggesting that defensive mistakes play more strongly in influencing the outcome of a strikeout. As expected, the trends get stronger as we move towards more uneven at bats, with predominantly offensive or defensive mistakes.

Looking at walks, significantly more occur when the offense has the advantage, noting that when the defense has the advantage, there aren't that many fewer walks. The greens are much darker than the reds. As an example, where there is an increase by 0.77 walks per at bat with 3 offensive mistakes and 0 defensive, there is only a 0.07 decrease in walks when that advantage is flipped.

The number of runs scored is the most surprising trend. No matter the kind of mistake, runs seem to decrease when compared to perfect at bats. However, defensive mistakes do represent a larger decrease in runs, perhaps suggesting some slight impact on at bat outcome.

From the in depth analysis of existing innings and at bats, it was concluded that umpires are tangibly influencing the outcome of the game. The umpires have a direct influence on the outcome of an at bat, so there is valid grounds for considering the implementation of the automatic strike zone.

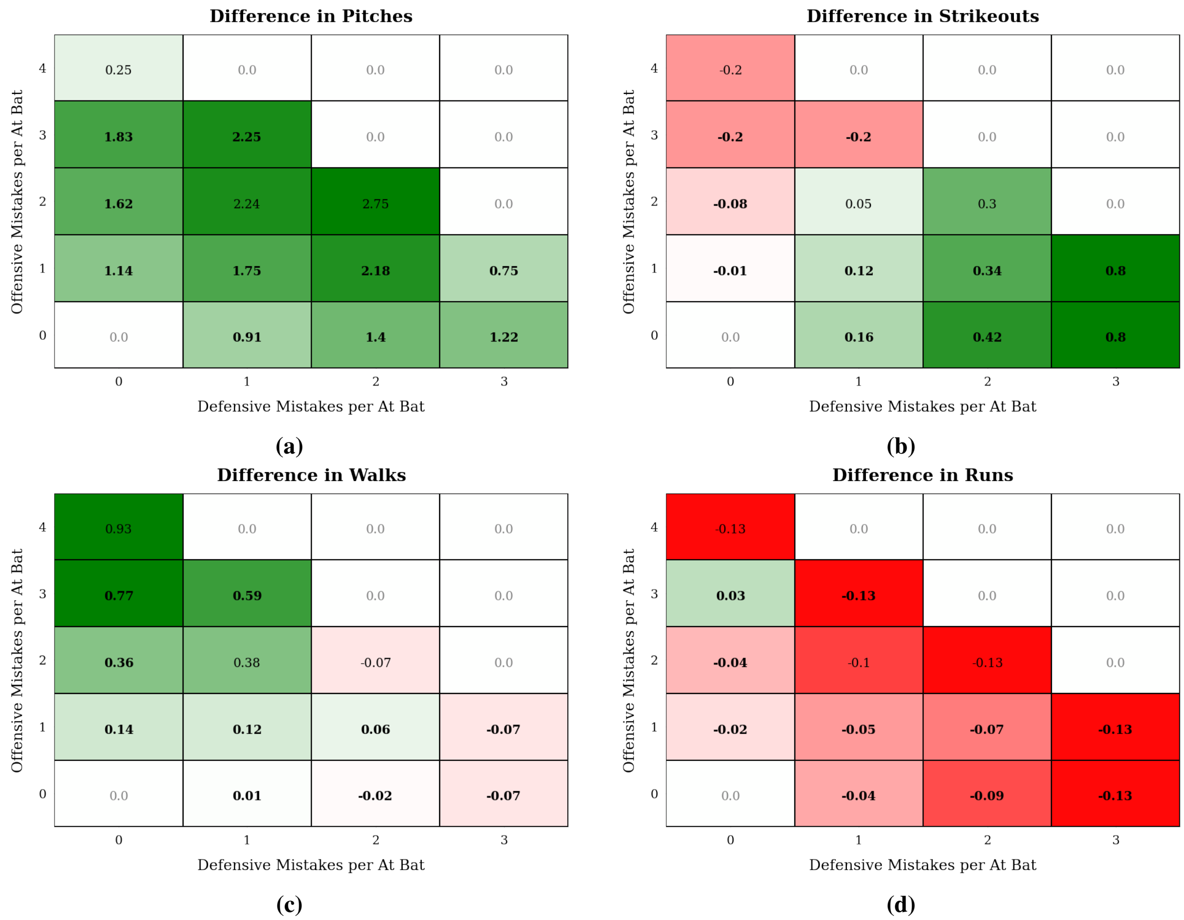

Figure 14: Four tables, each a breakout of one metric and how that fluctuates as the number of offensive and defensive mistakes change. (a) The number of pitches increases regardless of mistakes, a byproduct of the fact that in order for a mistake to be tallied, a pitch that is not hit has to occur, automatically increasing the pitch count. (b) The number of strikeouts decreases as more offensively advantageous mistakes occur and increase as the number of defensively advantageous mistakes occur. (c) The opposite trend is seen in walks; they increase with more offensive mistakes and decrease with more defensive mistakes. (d) As perhaps the most puzzling trend, the number of runs scored decreases no matter the amount of mistakes that occur. For all four, bolded numbers are statistically significant.

3.2 Corrected Inning Comparisons

All metrics in this section are defined as original innings minus the corrected innings. The existing analysis took the innings with mistakes minus the innings without mistakes, and this section is intended to mirror that process. This time, though, the innings without mistakes are just a new version, not different ones that actually happened.

With this new delta definition, the same analysis was performed, and the results for each metric, broken out by their combination of offensive and defensive mistakes, is in Figure 15.

For the difference in pitches, when the defensive mistakes are corrected, more pitches have to be thrown, whereas when the offensive mistakes are corrected, fewer pitches have to be thrown. This makes sense, because by eliminating the defensive advantage, batters are staying at bat longer and the situation generally worsens for the defense.

For strikeouts and walks, there is more of both in the original innings, regardless of mistake type. Walks have less of a pattern, seeming to decrease as the total number of mistakes increase, while for strikeouts we do see a more drastic decrease when we correct defensive mistakes, again eliminating that pitcher advantage.

There are a few hypotheses as to why we might not be seeing the anticipated trends. First is that we under-count strikeouts and walks when we move out of our transition matrix. If a batter gets out, with 2 strikes, we don't know if they flew out, grounded out, struck out, or fouled off a bunt. When we are inside our runner change function, we can

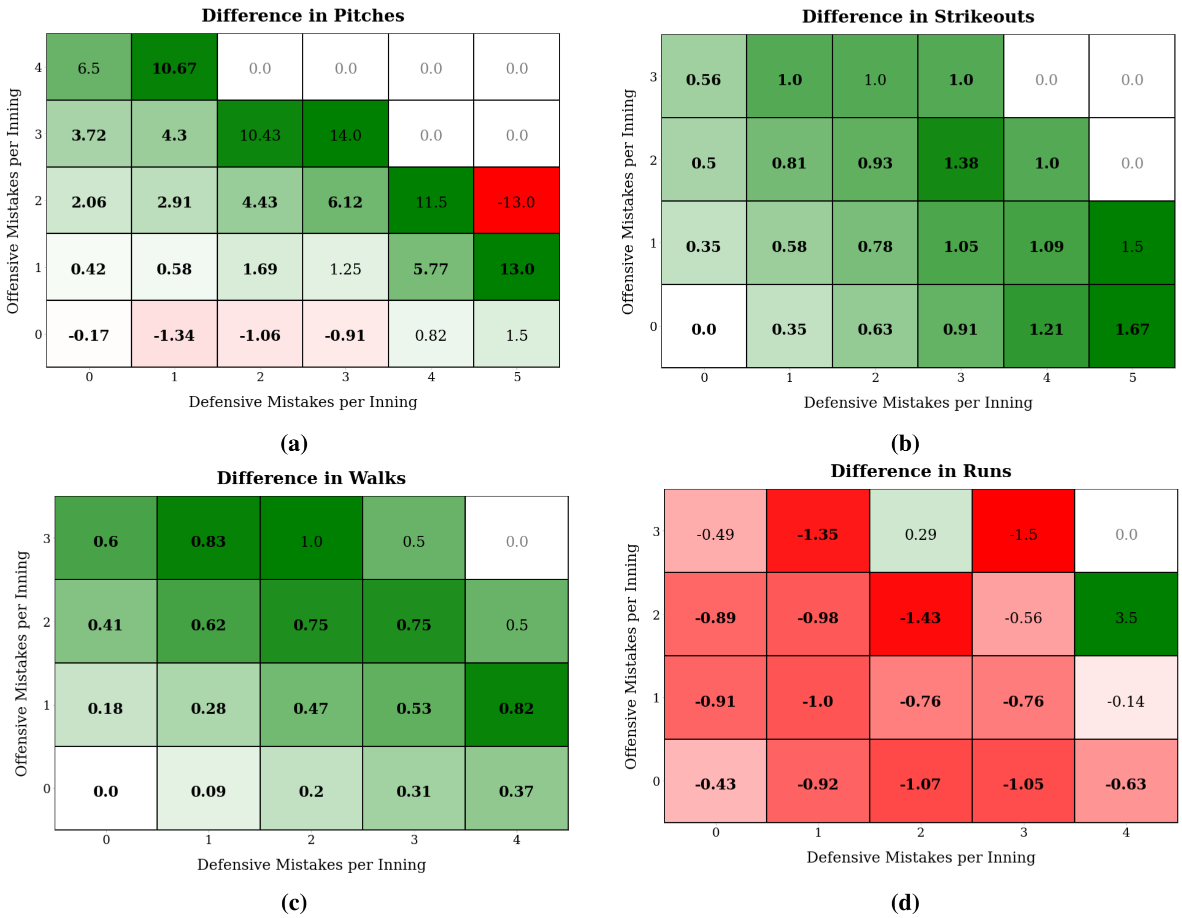

Figure 15: (a) When defensive mistakes are corrected, more pitches are thrown, and the opposite is true when offensive mistakes are corrected. This follows intuition, as the elimination of a defensive advantage should lead to more pitches having to be thrown. (b) Strikeouts increase regardless of type of mistake, as do (c) walks and this is likely due to the undercounting of these metrics that comes from the nature of our transition matrix. (d) The runs lack a strong pattern; in general, runs increase in the corrected innings.

tabulate strikeouts and walks, and we can count them if they are the direct result of the corrected pitch, but as of now, our transition matrix doesn't include specific outcomes, so those are not currently being tallied.

For runs, there is not too much of a pattern. The main takeaway here is that no matter what kind of mistake we are changing, there is an overall increase in runs being scored when our innings have no mistakes. This is a big of a confusing finding; intuitively, removing offensive advantages should lead to fewer runs being scored.

4 Conclusions and Future Work

4.1 Conclusions

Throughout the course of this analysis, it has been determined that umpires are, in fact, making a real impact on the game. Their offensive and defensive mistakes help either team, and can impact a team's ability to produce or prevent offense.

Based on our existing inning analysis, the implementation of the automatic strike zone would make the game more fair, as offensive and defensive mistakes tend towards favorable, and debatably undeserved, outcomes. Right now, from our predictive model, we'd anticipate a decrease in strikeouts and walks if this automatic strike zone were to be implemented, which might lead to more balls in play, which is a more exciting game. It is up to the MLB to decide which their ultimate goal is, but an increase in both fairness and excitement seems like a promising future.

4.2 Future Work

Predictive Model Improvements

Though our model was able to provide some interesting and valuable insights, there were many assumptions made and places where its current ability is limited.

As we insert fake batters and move runners, we generate new states that weren't in the original game. A small subset of these states has never shown up in the dataset, so our transition matrix isn't equipped with any information on the best next state for it. In these cases, we perturb the state slightly to help nudge the game forward. We do so by increasing the strike count until the state can be found in the transition matrix. This introduces a slight bias, because in the small subset of cases where we get stuck, we rather arbitrarily punish the batter. The best way to resolve this issue is to add more data and develop an even more thorough transition matrix, so there are no blank states.

Additionally, changes to the game cause new runner configurations and associated changes in run differential. While we account for the changes in runner configurations in our states, we don't update run differential in our states before passing them through the transition matrix. Thus, our current model assumes that run differential doesn't impact the most likely next state. This assumption is made to simplify the problem, but we would advise updating run differential as a possible next step for increasing the accuracy of this model.

One large gap is the previously stated reason for undercounting the strikeouts and walks. The transition matrix produces an out, but it doesn't know if that out was a groundout or a strikeout. In order to correct this, the transition matrix would have to take into account the at bat outcome, making it incredibly massive. We do not feel that this is a reasonable approach, but rather think there could be some probabilistic approach taken, where if the transition matrix produces an out, a separate function is run to determine the probabilities of which type of out it could be. A similar function would have to be developed to differentiate between walks, singles, and hit by pitches.

Finally, we believe to make this model more robust, we need to introduce some randomness. Currently, the function to move our runners takes in a random number and then proceeds according to the probabilities of certain runner movements, but our transition matrix just moves into the next most likely state. In order to remove some of the biases and repeated results, a random perturbation of that next state, where it can move into a state based on the probability of that state occurring, would be necessary.

Next Steps

One of the main things that remains to be done is to decide if this automated strike zone is changing the game in a positive or negative way. This numerical analysis is just one piece of the puzzle, there are players, coaches, managers, fans, and many more people and factors to take into account before any final decision is made. Additionally, even if the MLB does decide an automated strike zone would be the right choice, how should it behave? Should it be the same size and shape as a textbook strike zone? Should it stay consistent throughout the game? Should it be modified to most closely match how the game is being called currently? Further analysis of this kind may be needed for each of these scenarios in order to see how potential shape changes of the zone might influence the outcome of a game.

5 Acknowledgments

The authors would like to thank Peko Hosoi for her mentorship and support with every step of this project, as well as the rest of the 2.980 Staff for all of their help and support. They owe a huge thanks to Travis Buck and Reed MacPhail, their sponsors and mentors from the MLB for their helpful input and guidance, as well as the abundance of data they provided. The authors would also like to thank Devavrat Shah, Anish Agarwal, and Dennis Shen for their insights and guidance on causal inference and the predictive algorithm. They would like to thank Ramzi BenSaid for his assistance with their data sets and their Google interactions. Finally, they would like to thank the other students in the 2.980 class for their feedback and comments throughout the semester. None of this project would have been possible without all of these people and countless others. Thank you!