Group 3

Undergraduate Division

Optimizing Batting Lineup Order Across Minor League Baseball

ABSTRACT

This paper presents a novel simulation-based approach for optimizing batter lineups in minor league baseball. Our simulator goes beyond basic hitting statistics by incorporating key factors such as baserunner advancement, player speed, pitcher fatigue, handedness matchups, and situational outcomes like double plays and sacrifice flies. We propose a dynamic model that can simulate the expected runs scored by different lineups. We demonstrate the simulator's usefulness by comparing real-world lineups, revealing optimal configurations that could significantly impact game outcomes. Our framework offers a valuable tool for coaches and analysts to make informed decisions in the complex and uncertain environment of minor league baseball.

1. BACKGROUND

How do managers decide what the batting lineup should be for any given game? For years, baseball strategy followed an old-fashioned approach to most questions. With the success of the data-driven Oakland Athletics in the early 2000's and the subsequent movement dubbed "Moneyball," professional baseball teams began to integrate more mathematical thinking into their decision-making. Prior to this, traditional minds had won out when it came to lineup construction: speedy batters hit leadoff, bat control guys second, best hitters third, and cleanup hitters fourth. This idea went mostly unchallenged until 2006, when Tom Tango, Michael Littman, and Andrew Dolphin published "The Book: Playing the Percentages in Baseball." Utilizing the new concept of sabermetrics, the authors came up with an analytically-based batting order: have your high on-base hitter first, your best all-around hitter second, don't worry too much about who bats third, and put your best power hitter fourth. Fast-forward to 2024, and many MLB teams have adopted this new approach. However, since "The Book" was published, there has not been much public work challenging the ideas written about or further modernizing lineup construction in baseball. One specific issue that managers may have today is what to do when the "optimal" lineup is not available. While in MLB, teams have consistent rosters and don't have to shuffle their lineups too often, in the minor leagues, chaos reigns. Players are constantly being called up or sent down, with extremely small sample sizes for different lineups and batting orders. So how can managers maintain sound decision-making with so much uncertainty? In this paper, we propose a novel simulation-based approach to assessing the strength of lineups managers actually have at their disposal. We propose a new multifaceted approach for simulating scoring in baseball, accounting for many important factors in the game often overlooked in the name of efficiency.

2. DATA

In order to calculate individual players' batting statistics across the different levels of play, we made use of the "game_info" and "game_event" tables, containing detailed information about the game state as well as time stamped labels for events like hits, ball bounces, and catches. We also made use of the "player_pos" table containing detailed player-tracking information for each play in order to compute features like player handedness and player speeds. Key information we did not have from "game_info" and "game_events" were detailed run and out information. In order to compute more robust detailed game statistics such as double plays, sacrifice flies, and runners scoring or getting tagged out at home plate, we combined "player_pos" with "ball_pos," a table containing detailed ball-tracking information for each play.

3. METHODOLOGY OVERVIEW

In order to determine the success of a certain lineup, we decided to create a game simulator that calculates the number of runs scored by a specific lineup in a game. Previous baseball simulators typically only include basic hitting percentages such as walk percentage, single percentage, double percentage, triple percentage, and home run percentage. They fail to include very important factors such as the advancement of runners on the base path, batting differences based on opposing pitcher, pitcher fatigue, player speed, and directional tendencies. We propose a novel and more accurate simulator taking into account the many nuances of baseball beyond just basic hitting percentages.

4. SIMULATOR COMPONENTS

Every player's basic batting percentages (walks, singles, doubles, triples, home runs) were used as the foundation for our simulator, as those statistics are the most impactful.

4.1 RUNNER ADVANCEMENT

When a player hits a single, typical simulators will assume that every base runner will just advance one more base. Those familiar with baseball know that often that assumption is not the case. A player on first base is sometimes able to advance to third base, and a player on second base might even be able to score on just a single. We examined various factors as to how to predict these more infrequent advancements to identify the best indicators. When looking at second base runner outcomes on a single, we find faster players are most likely to score, while slower players are more likely to stay on third base (Figure 1):

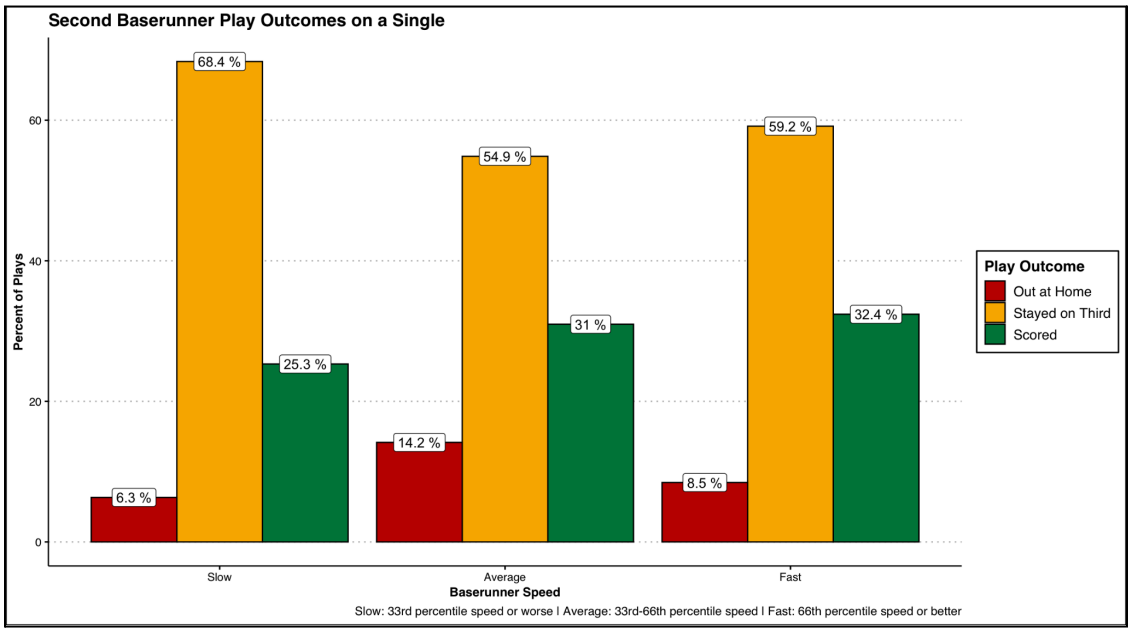

Figure 1: Determining the tendencies of runners on second base after a single depending on speed. *Slow runners, identified by having a 33rd percentile or worse in speed, were much more likely to stay on third base, as represented by the yellow bar. Average runners, identified as being between the 33rd and 66th percentiles of speed, were the most likely to be caught out at home compared to the other groups, as represented by the red bar. Fast runners, identified as being*

above the 66th percentile for speed, were most likely to score from second base on a single, as represented by the green bar.

We then found that advancing from first base to third base on a single is actually more dependent on where the ball was hit, instead of the speed of the base runner. Figure 2 shows a visual example of this trend, with blue points representing hits in which a runner advanced to third base:

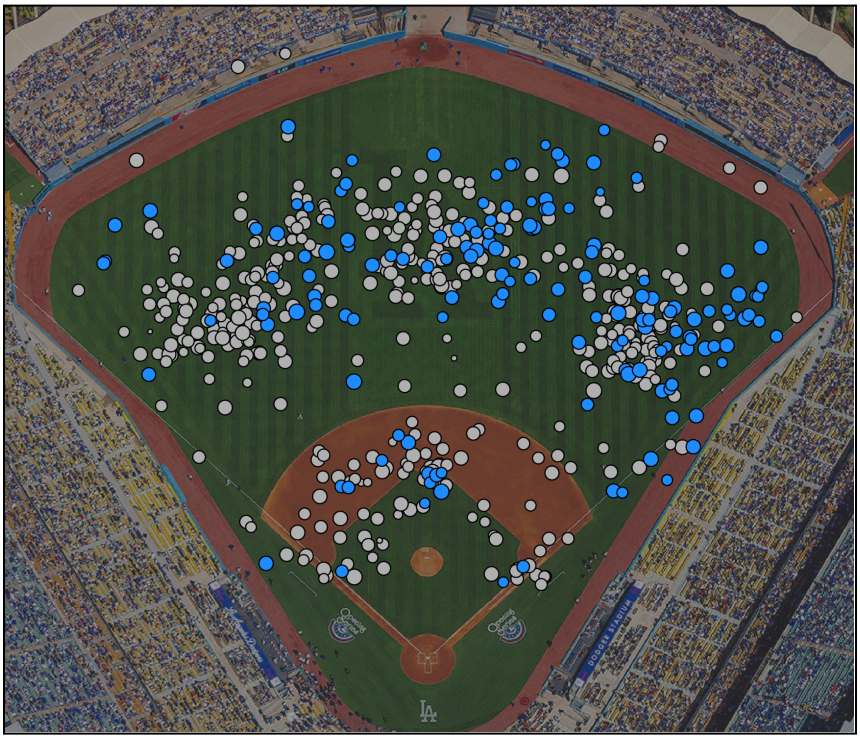

Figure 2: Spray chart of singles when a runner is on first base. *The blue points represent instances where the first base runner advanced to third base on a single. Visually it can be seen that the majority of blue points are located in right field and right-center field, meaning that when a ball is hit there on a single, it is much more likely for a runner to advance to third base from first base.*

When the ball was hit to left field on a single (Figure 2), a runner on first base advanced to third only 17.9% of the time, compared to right field's 39.5% and center field's 26.92%. If the ball is not hit into any of those locations, there is only a 23.2% change of advancing. So to determine

where the ball was hit on these singles, we calculated each player's hitting tendencies and accounted for them in our simulation.

The same process was taken for looking at runners on 1st base during doubles, where the most predictive indicator was speed again. This time the differences were even more staggering (Figure 3):

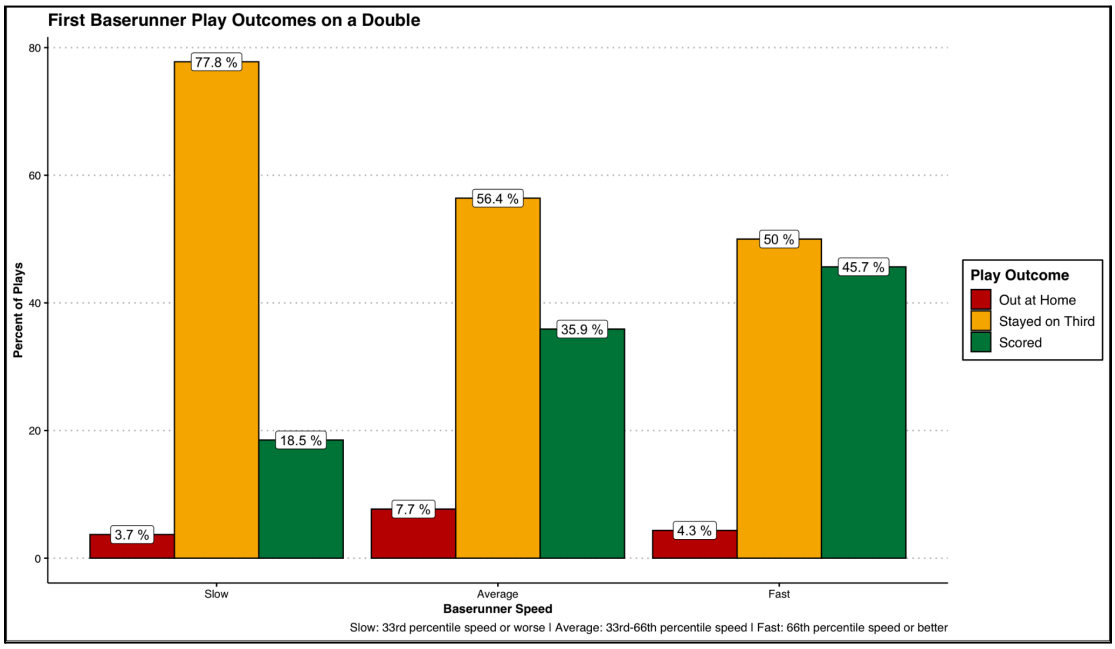

Figure 3: Determining the tendencies of runners on first base after a double depending on speed. *Slow runners, identified by having a 33rd percentile or worse in speed, were much more likely to stay on third base, as represented by the yellow bar. Average runners, identified as being between the 33rd and 66th percentiles of speed, were again the most likely to be caught out at home compared to the other groups, as represented by the red bar. Fast runners, identified as being above the 66th percentile for speed, were most likely to score from second base on a single, as represented by the green bar.*

4.2 OUT TYPE

Typically, baseball simulators treat all outs as equal, but we know that fly outs can sometimes lead to players scoring while ground outs can sometimes lead to double plays. To determine what

the outcome was for every out accrued, we first had to get each player's likelihood for striking out, grounding out, and flying out. When a player strikes out, there is no difference in base runners, as the out count just increases by one. We examined fly outs in the context of sacrifice flies, where the batter hits a fly out with a runner on third to potentially send him home. When a player flies out, we found the location of the fly ball to be more predictive than speed, where a center field fly out has a probability of sending the man on third home of 81.8% compared to a left field or right field fly out probability of 90%. When a player grounds out, there was no indicative factor of speed or direction, but it was estimated that a ground out turns out into a double play 42% of the time when there is a runner on 1st base.

4.3 STOLEN BASES

Stolen bases were also incorporated into our simulator. Typically a runner who frequently steals bases are placed very early on in a batting order, so we knew its incorporation would be vital. Every player's stolen base percentage from 1st to 2nd base, as well as from 2nd to 3rd base were calculated and subsequently added to the simulator.

4.4 INNING LENGTH

The first is the idea of a "long inning," which is the idea that within a single inning, the more batters the pitcher faces, the more advantage the batters have due to pitcher fatigue and mental tear. However, after extensive work on the topic, we determined that the data did not show significant enough results to include it in the simulator.

4.5 INNING COUNT

The second is the idea that depending on the inning, batters will either do better or worse. This was based on the thought that over time, pitchers in a game get more fatigued and do worse the longer they stay in, meaning batters would probably do best in the later middle innings, and then get worse when the relief pitchers come in during the last innings. The results are pictured in Figure 4:

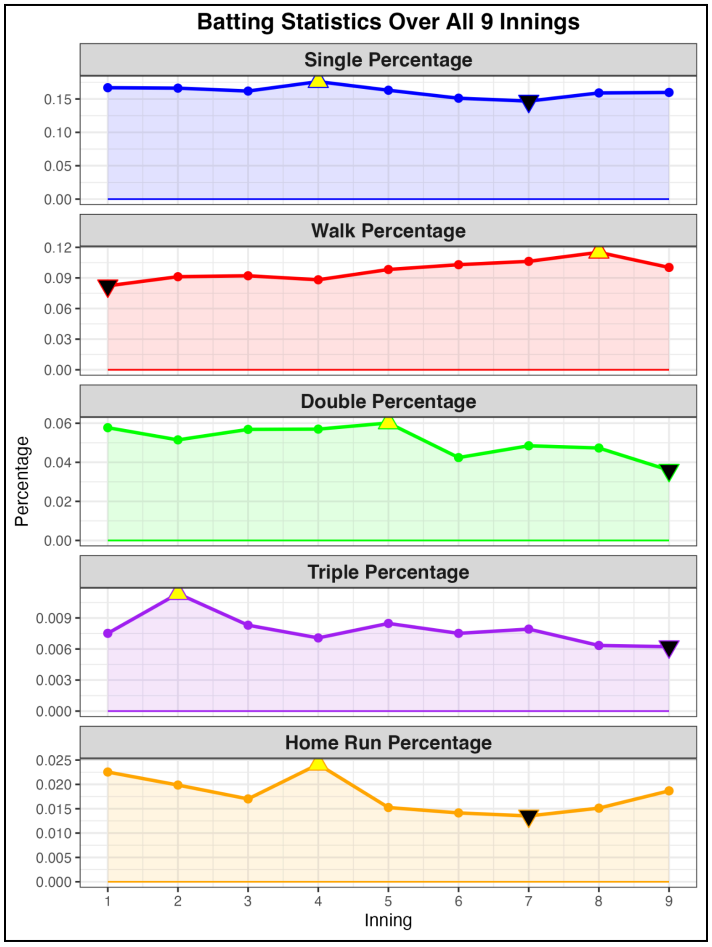

Figure 4: Determining which innings were most advantageous for batters depending on batting statistic. *Downward black arrows show the innings in which the batters did the worst in that batting statistic, while upward yellow arrows indicate the best inning for batters in that batting statistic. The last 3 innings seemed to be where the batters did the worst compared to other innings, whereas the early-middle innings were most advantageous for batters.*

These results aligned with our hypotheses, which the batters typically did worse during the first and last innings (represented by the black downward arrow), and they did best in the middle innings (represented by the yellow upward arrow). Due to these differences, we took each player's typical batting statistics and modified them relative to the average statistic for that specific inning, making the simulator more dynamic over innings.

4.6 HANDEDNESS

The third is the idea that batters perform differently depending on the handedness matchups. It is typically known that batters hit better when they face pitchers of the opposite handedness, but we needed to test by how much that is true and how much that is false conventional wisdom. The results proved that this idea is statistically correct, as represented by Figure 5:

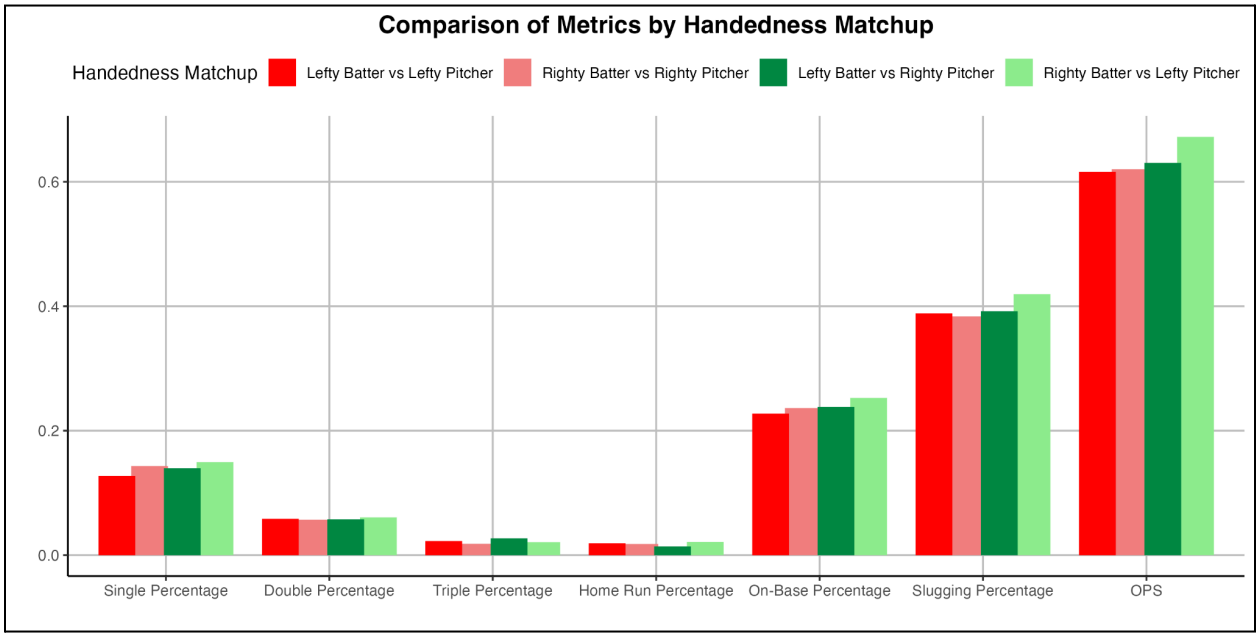

Figure 5: Comparing batter performance depending on batter and pitcher handedness matchups. *Green bars, both light and dark, indicate matchups in which the batters and pitchers have opposite handedness. Red bars, both light and dark, indicate matchups in which the batters and pitchers have the same handedness. Opposing handedness matchups showed much better success for batters.*

Batters facing opposite hand pitchers tend to perform better. Due to this, we added an option in our simulator to add the pitcher's handedness and subsequently altered the batters' statistics accordingly.

5. RESULTS

We decided to run the game simulator on a sample lineup of the 9 players with the most at bats in the data. We made two separate simulations, one for exclusively single game simulations and one for multiple games. For our single game simulator, we are able to see the play by play of the simulated game (Figure 6):

	First	Second	Third	Outs	Outcome	Runs_Scored	Hitter	Innings
1	0	0	0	0		0	0	1
2	1	0	0	0	Single	0	1	1
3	1	0	0	1	Out	0	2	1
4	1	1	0	1	Walk	0	3	1
5	1	0	1	1	Single	1	4	1
6	1	0	1	1	Single	1	5	1
7	1	0	1	2	Out	0	6	1
8	1	0	1	2	Single	1	7	1
9	1	0	1	3	Out	0	8	1
10	0	0	0	0		0	8	2
11	0	0	1	0	Triple	0	9	2
12	1	0	1	0	Single	1	1	2
13	1	0	1	1	Out	0	2	2
14	1	0	1	2	Out	0	3	2
15	1	0	1	3	Out	0	4	2
16	0	0	0	0		0	4	3
17	0	1	0	0	Double	0	5	3
18	0	1	0	0	Double	1	6	3
19	0	1	0	1	Out	0	7	3
20	1	0	0	1	Single	1	8	3
21	1	0	0	2	Out	0	9	3
22	1	1	0	2	Walk	0	1	3
23	1	1	0	3	Out	0	2	3
24	0	0	0	0		0	2	4
25	1	0	0	0	Walk	0	3	4
26	0	1	1	0	Double	0	4	4
27	0	1	1	1	Out	0	5	4
28	0	1	1	2	Out	0	6	4
29	0	1	1	3	Out	0	7	4
30	0	0	0	0		0	7	5
31	0	0	0	1	Out	0	8	5
32	1	0	0	1	Walk	0	9	5
33	1	1	0	1	Single	0	1	5
34	1	1	0	2	Out	0	2	5
35	1	1	1	2	Walk	0	3	5
36	1	1	1	3	Out	0	4	5

Figure 6: Example Game Simulation. *The "First", "Second", and "Third" columns represent the bases, with a binary of whether a runner is on that base. "Runs_Scored" is not a cumulative measure, as it is the number of runs scored on that particular at-bat. On line 5, you can see that the runner on second base scored off of a single, while the runner on 1st base advanced to third base. This outcome was caused by the methodology described in Subsection 4.1.*

These single game simulations were only used for looking at an example play by play of a game. We created a multi-game simulator to run 1000 simulations of each game as shown in Figure 6. The simulator then only returns the average number of runs scored, rather than tables for every game.

5.1 GAME SIMULATOR USER INTERFACE

One of the major decisions we made during this project was the implementation of a user interface that allows a user to run the simulator for themselves. Each user can choose the level of play, the pitcher's handedness, the batter from each position, and finally the batting order (Figure 7). Our product then returns the average number of runs that lineup produces. The user can then change the order to their liking, attempting to get a higher average runs scored value, showing

that the lineup has been optimized.

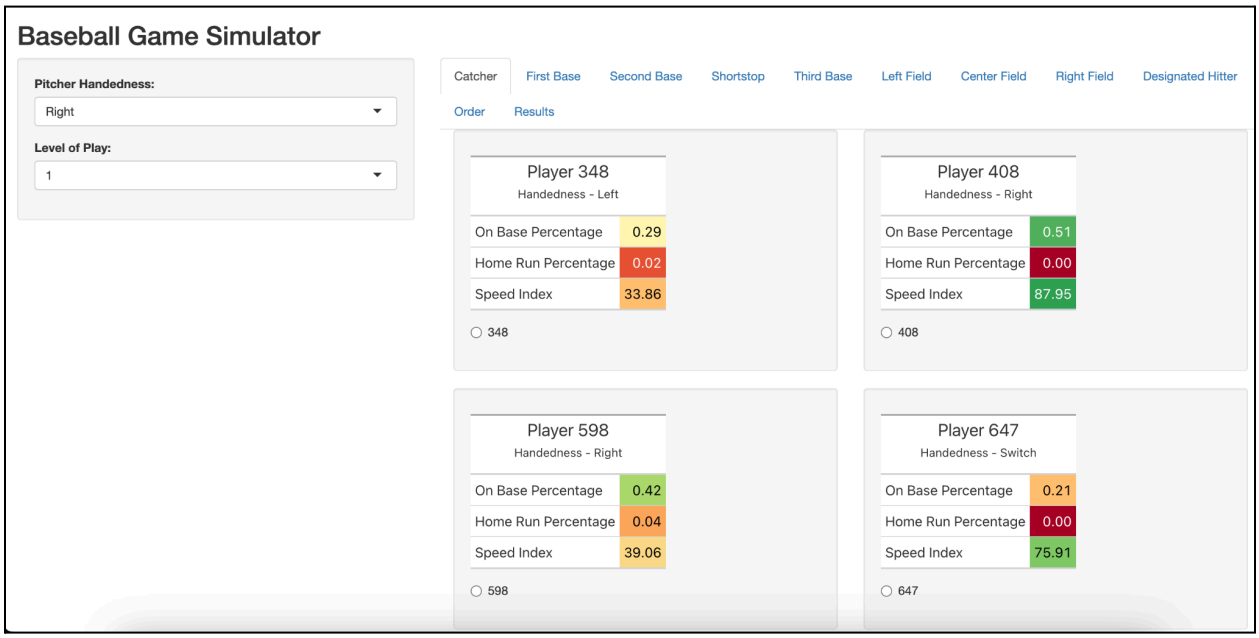

Figure 7: Selecting Lineups in Game Simulator. *The user can choose the level of play, opposing pitcher handedness, and a starting player for each position. This figure shows the first four batters available to select for the Catcher position, including each player's on base percentage, home run percentage, and speed index. These were then colored from red to green based on how they compare to the rest of the league.*

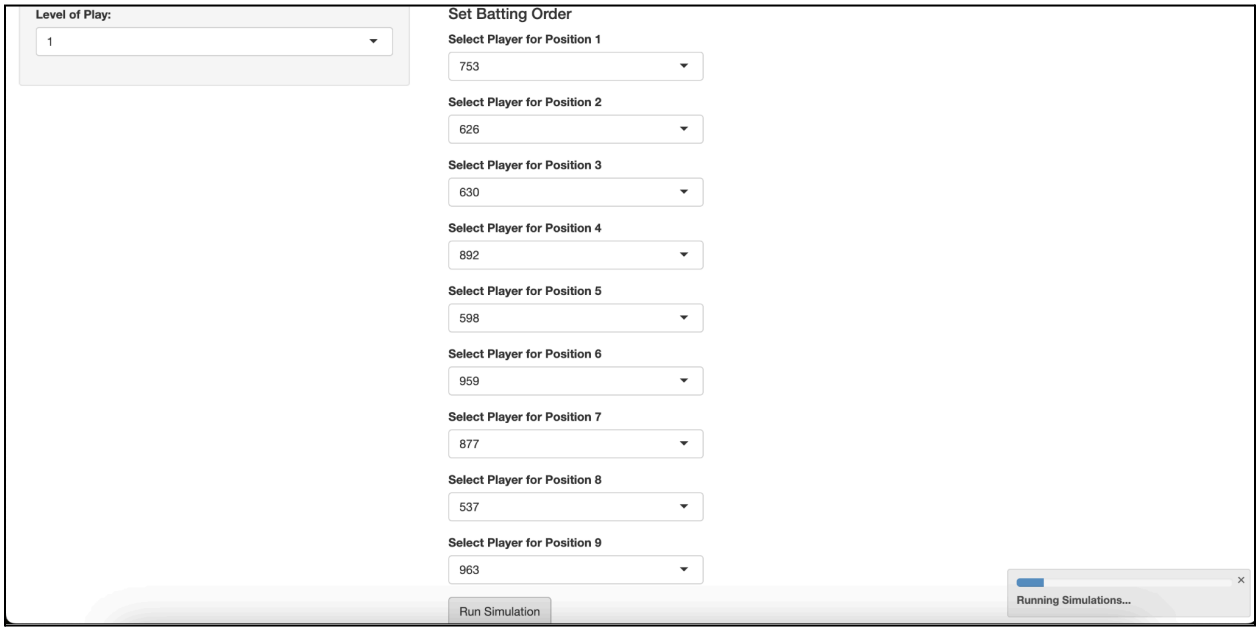

Figure 8: Selecting Batting Order in Game Simulator. *Once the user has chosen a valid lineup, they can set their desired batting order and run the simulation to find the average number of runs scored.*

When applying our game simulator to real-world lineups to achieve optimized solutions, one hurdle we faced was the inconsistency of lineups across the minor leagues. The most commonly-used nine-man lineups batted just two times across both seasons of data. However, when examining 1A, we found that a particular five-man grouping at the top of the lineup batted together for 12 different games, in three distinct orders. One order (Lineup A) was used in 7 games, another order (Lineup B) was used in 4 games, and the last order (Lineup C) was used in 1 game. But which of these orders is the "best"? Coaches would have a hard time knowing which one they should use in any given game due to the miniscule sample size. Using our simulator, we can compare these three lineups and see which one is the most optimal lineup (Figure 8). We found that Lineup A produced on average 6.96 runs per game, Lineup B produced 6.77, and Lineup C produced 7.2. Lineup C may have been discarded by the coaching staff due to one bad performance, however, our analysis shows that this was the best combination of the given batters. If the manager had used Lineup C in all 12 games, the team would be expected to score 3.4 more runs in total. This may not seem particularly significant, but any small difference in runs can be the difference between a win and a loss.

6. DISCUSSION

In this analysis, we have produced a novel framework for evaluating lineup combinations in minor league baseball. Our multifaceted game simulator takes into account the detailed aspects of baseball, such as double plays, baserunner speed, stolen bases, and more. Our work can be

used across the different levels of play provided to us to produce optimized lineups for coaching staff and fans alike to utilize.

7. LIMITATIONS

The primary limitation of our work is the imputation of outs and runs. Due to detailed out and run data being removed for anonymization purposes, our process when calculating certain statistics (such as whether runners were ruled safe or out at home plate or an accurate out count on a play-by-play basis) represents a best guess of the game state and play outcome. In a non-anonymous setting, these findings would be consistently accurate and trivially correct.

ACKNOWLEDGEMENTS

We would like to thank SMT for hosting this challenge, providing us the opportunity to work on such an incredible project using immense tracking data. We believe that tracking data is becoming more important in the Sports Analytics industry each year, so this project is very advantageous.

We would also like to thank everyone who has helped us during this project, particularly Meredith Wills, for their expertise and guidance. Additionally, we would like to thank the judges who have taken the time to examine all of the submissions.

Finally we would like to thank Rice University and their Sport Management and Sport Analytics programs for encouraging us to pursue these types of data projects, allowing us to strengthen our skills in the sports analytics field.

Sincerely,

Lucca Ferraz and Jonah Lubin

REFERENCES

Cwiklinski, P. (n.d.). *How sabermetrics influence baseball batting order strategy*. Sports Betting Dime. https://www.sportsbettingdime.com/guides/strategy/batting-order-sabermetrics/

Driveline Baseball. (2020, February 26). *SEQNZR: Simulation Modeling for lineup optimization*. https://www.drivelinebaseball.com/2020/02/seqnzr-simulation-lineup-optimization/

Macleod, N. (2024, June 30). *Building a baseball offensive simulator in R:*. Medium. https://medium.com/@nicholas.macleod9/building-a-baseball-offensive-simulator-to-optimise-hitting-lineups-in-r-611b13fb368f

Marx, D. D., Marx Scheuerell, A. C., & Tesar, M. L. (2024, April 25). *Is a Major League Hitter Hot or Cold?*. Society for American Baseball Research. https://sabr.org/journal/article/is-a-major-league-hitter-hot-or-cold/

Rees, L. P., Rakes, T. R., & Deane, J. K. (2015). Using analytics to challenge conventional baseball wisdom. *Journal of Service Science (JSS)*, *8*(1), 11–20. https://doi.org/10.19030/jss.v8i1.9493

Sherman, N. (2012, October 12). *Optimizing Order, part 1: How to build the ideal lineup*. Bluebird Banter. https://www.bluebirdbanter.com/2012/10/12/3490578/lineup-optimization-part-1-of-2

Silver, B. (2022, April 8). *How using analytics helps optimize the Philadelphia Phillies' lineup*. Sports Illustrated Inside The Phillies. https://www.si.com/mlb/phillies/opinions/analytics-optimize-philadelphia-phillies-lineup-opening-day-joe-girardi-bryce-harper-gossip

APPENDIX

Simulator Guideline:

1. Choose the pitcher handedness and level of play, as seen on the left hand side of the screen.
2. For each position tab, choose a player you want inserted in the lineup and then scroll down and press “Save Selection”.
3. Make sure that the batter chosen in the designated hitter slot is not a duplicate of someone already chosen from the 8 position tabs.
4. Click on the order tab and set the order as you wish.
5. Scroll down on the order tab after setting the order and press “Run Simulation”
6. Wait patiently while the progress bar is showing.
7. Once the progress bar disappears, click on the results tab, which will show the average runs per game.

Sample Size Details:

Each player was only included in their primary position, which was deemed to be the position they played the most games at. However, every player with at least 5 games of hitting appears in the designated hitting spot.

To be included in the simulation data, the players had to have played at least 5 games at their primary position. They also had to have had 10 or more at bats in any given level to be included in that respected level's data.

We believe that this is a strong enough representation of a player's skill to be included in the simulation. Increasing the sample size to 30 for example would have gotten rid of the vast majority of the data, only leaving a handful of batters in the entire data set, so we could not have the sample size be too large. This lower sample size does lead to some inflated numbers in the data set, but if given a larger set of data, our simulation could theoretically increase the sample size to ensure more precise results.

Calculating Player Handedness:

In order to figure out whether batters were right-handed, left-handed, or switch hitters, we first used the game_events table to find the timestamp of the pitch. We then took the average batter position at the timestamp grouped by individual batters to see where batters normally stood. If their average x position (in coordinates) was greater than 1(center of home plate has a coordinate of 0), we labeled them as left-handed batters. Players with average x position less than -1 at the time of the pitch were labeled as right-handed batters. Players with average x positions in between -1 and 1 were labeled as switch hitters.

For determining the handedness of pitchers, we utilized a very similar approach. Once we had the timestamps of the pitches, we took the average x position for each pitcher. Pitchers with

average x positions greater than 0 were labeled as left-handed, and pitchers with average x positions less than 0 were labeled as right-handed.