Dani Treisman

Loyola University Chicago

STAT 321 – Modeling and Simulation

December 6, 2019

Batting Order Optimization Using Simulation

1. Introduction

In 2019, the use of data science techniques, generally referred to as "analytics", has become as necessity in Major League Baseball (MLB). More specifically, teams are using various statistical methods to optimize different aspects of their team. One aspect that has been focused on by the media is the strategy of defensive shifts. A defensive shift is where the players on the field stand in different areas of the field depending on the hitting statistics of each batter. This strategy has been employed heavily in recent years due to the dramatic increase in available data.

Another common strategy that managers use in MLB games is putting the pitcher in the 8^{th} spot in the batting order. Pitchers are generally the worst hitters on the team and usually occupy the 9^{th} spot in the order. However, managers will occasionally move the pitcher up one spot to stimulate more offense. The goal of this simulation study is to determine whether that strategy, along with other batting order variations, does, in fact, lead to a significant increase in runs.

2. Data

The data used is from player hitting statistics of the 2018 Chicago Cubs on the MLB website. Throughout a given season, a team may employ several different lineups due to strategy, injuries,

or high/lack of performance. To maintain simplicity, the lineup chosen is the most commonly used lineup for the Cubs in 2018 according to baseball-reference.com. The statistics for each batter were calculated as follows: On Base Percentage (OBP) was calculated as 1000 – OBP, and all other statistics were calculated by dividing a player's walks, singles, doubles, triples, and homeruns, each by the number of plate appearances for that player. Each player then received a vector containing the cumulative probabilities of those calculations. **Figure 1** shows the final data table.

- Figure 1 -

Player	OBP	Walk %	Single %	Double %	Triple %	Homerun %
Almora	657	711	929	983	985	1000
Baez	662	709	862	928	942	1000
Bryant	604	727	886	958	966	1000
Rizzo	594	718	904	955	957	1000
Contreras	639	751	911	968	979	1000
Schwarber	579	761	899	932	939	1000
Russel	655	750	936	986	988	1000
Heyward	634	729	920	972	981	1000
Pitcher	870	895	985	990	991	1000

Something to keep in mind is that these numbers do not consider many factors that could influence offensive performance such as weather, home field advantage, and others. They are simply the raw statistics of those players the Cubs 2018 season. When performing this simulation, the average number of runs scored will likely be lower than the actual number of runs scored since the simulation assumes that the lineup stays the same the entire game. Typically, the starting pitcher does not play the entire game and the pitcher's spot in the order is replaced by a pinch hitter.

3. Method

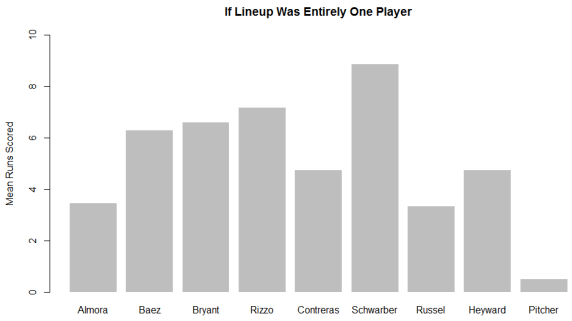
The program simulates a full 27 outs of offense and is contained mostly within one function called *game()* that takes a batting order table in the above format as a parameter. *game()* begins by setting all bases, outs, and runs to 0. The core function of the simulated game relies on a random number generated between 0 and 1000. Before each turn this random number is generated and where it falls in the cumulative probability distribution for the current batter, will determine the outcome of the at-bat. *game()* loops through the batting order until there are 27 outs, and then repeats 1000 times.

This main goal of the simulation study will compare batting orders with the pitcher 8^{th} vs 9^{th}, determine how much worse it is to have the pitcher 1^{st}, and compare all three to a custom order determined by a simplified annealing process. However, as I will explain below, I decided to also evaluate putting the 6^{th} batter (Kyle Schwarber in this case) first, as well as an order determined by decreasing OBP.

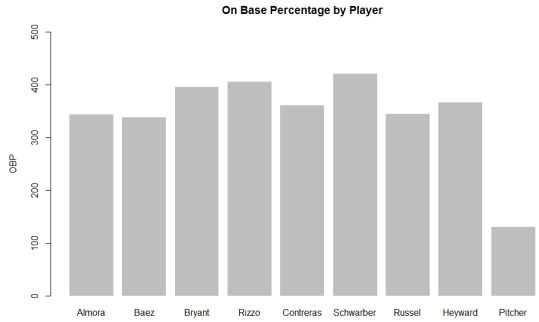
4. Results

I decided to begin by exploring lineups composed of just one batter to evaluate who which player might contribute the most runs produced. I created 9 different lineups, each comprised of 9 copies of themselves. For example, the first lineup is 9 copies of Almora. **Figure 2** shows the results of this first simulation. Schwarber scored the most runs per game at around 8.4, and the Pitcher scored the least at less than 1 run per game. Schwarber scored about two runs per game higher than the next closest player so I decided to utilize this information in the final comparisons and do a simulation with Schwarber batting first. After comparing OBP for each player (**Figure 3**) I hypothesized that runs scored might be related to OBP, so I decided to add a batting order to the comparison that is in order of OBP descending but with Schwarber batting 3^{rd}, the spot typically reserved for the best offensive player.

- Figure 2 -



- Figure 3 -



After running all six simulations, I did not find any significant differences in runs scored between the different batting orders. **Figure 4** shows the average runs scored per game for each batting order. The mean runs per game for each batting order configuration are all very close. Putting the 6th batter (Schwarber) first seems to be lower than range of the other lineups. **Figure 5** shows the pairwise t-test p-values.

- Figure 4 -

Lineup Config.	Runs/Game
Original	4.635
Pitcher 8th	4.638
Pitcher 1st	4.516
Batter 6 → 1st	4.318
Optimized	4.429
Decreasing OBP	4.555

- Figure 5 -

** Significant at *alpha* = 0.05
* Significant at *alpha* = 0.1

	Original	Pitcher 8th	Pitcher 1st	6 → 1st	Optimized	Decr. OBP
Original	-	0.984	0.405	0.025**	0.148	0.583
Pitcher	0.984	-	0.392	0.023**	0.141	0.568
Pitcher 1st	0.405	0.392	-	0.148	0.529	0.783
Batter 6 → 1st	0.025**	0.023**	0.148	-	0.416	0.090*
Optimized	0.148	0.141	0.529	0.415	-	0.372
Decr. OBP	0.583	0.568	0.783	0.090*	0.372	-

Putting the 6th batter first results in a significantly lower (at *alpha* = 0.05) mean runs per game than the original order and the order with the pitcher 8th. The order with decreasing OBP results in a significantly higher (at *alpha* = 0.1) mean runs per game. While I am slightly surprised to have some significant results since , earlier runs of these simulations did not produce significant results and the difference in means between the batting orders was different every time. However, none of my results showed a lineup that was significantly better than the original, in terms of runs scored. This is consistent with real world results since there has not been any significant breakthroughs on lineup optimization or changes in the standard format managers use for their lineups. The most extensive research on optimizing batting orders that I found is in a book titled *The Book* by Tango, Lichtman, and Dolphin (2007). However, their research mostly aligns with the way batting orders have been structured for decades.

5. Conclusion

Sabermetrics has revolutionized almost every aspect of baseball; from fielding shifts to pitching changes, to scouting, and everything in between. In baseball, runs are at a premium and teams are always trying new techniques to try and maximize run production. Moving the pitcher to the 8th spot is a method that some managers swear by and has been debated greatly among sports commentators and fans. I presume however, the debate will continue for much longer unless a significant breakthrough is made in statistical research.