Hughston Preston

SPM 295 M003

December 8, 2020

Simulating the Probability of an Immaculate Inning

Introduction/Lit Review:

Baseball is a game of frivolities, quirks, and obscure traditions. This is all part of the excitement and lore of the game. Things like inside the park homeruns, balks, bunts for hits, triples, and even successful pickoff attempts are things that the casual fan may not understand or truly grasp the rarity of them. However, the dedicated members of the baseball community recognize the rarity of these occurrences and allow for them to further enrich the enjoyment of the game. In their eyes, it only adds to the time old adage: "You never know what you're going to see at the ballpark today". However, the true king of all frivolities in the game is the aptly named Immaculate Inning.

The Immaculate Inning is the perfectly ideal inning for a pitcher, where he strikes out all three batters faced on three pitches each, thus achieving the minimum total of nine pitches to achieve the lofty goal of striking out all three batters in an inning. However, unlike the Perfect Game or the No Hitter, the Immaculate Inning may be viewed as far more luck dependent and less skill dependent. And yet, only 94 pitchers in professional baseball's recorded history have accomplished the feat (Baseball Almanac). It's perhaps the most pointless, beautiful, and rare accomplishment in the game. In this project, I wanted to see just how rare the Immaculate Inning is by simulating it (McDaniel).

The first aspect I wanted to consider was how to run a simulation. The Monte Carlo Simulation is what we have learned in class, so this is where I began my research. Our class utilized Wayne Winston's University of Houston Coursera video to demonstrate how a Monte Carlo simulation could be utilized to utilized 2014 Baseball Reference at-bat outcome data to simulate how many runs are scored in a game for each MLB team. This exercise was done in Excel and does a very good job of showing how an MC simulation uses the probabilities and

weighted outcomes to build a broader model that can simulate more complex outcomes and strategies. Winston also explains some of the limitations in building this simulation where deeper level probabilities, such as the probability that a runner on first base is able to advance three bases on a double into the gap instead of his standard of two, may be unknown (Winston).

I found another baseball application of the Monte Carlo Simulation in R. Allan Freeze's 1973 Analysis of Baseball Batting Order using the MC Simulation. This was notably a simulation based upon the Sports Illustrated Baseball game of the early 1970s that used rules and special dice to carryout various realistic baseball scenarios (Sports Illustrated Baseball). This simulation was carried out 200,000 times and used specific rule inputs that standardized variables like pitching and defensive abilities while Freeze randomly generated different batting orders. He concluded that batting order only holds a small influence on a team's success amounting to roughly a 3-game swing in a 162-game season between the most and least advantageous lineups that he generated (Freeze).

I looked into various tutorials upon how to build a Monte Carlo simulation in R-Studio. While we had prepared these simulations in excel in class, I was hoping that I could find a more streamlined and organized method of conducting my simulations in R. Homer White's GitHub tutorial was one of the references that I found most insightful. This page demonstrates some basic ways to create randomly generated numbers in R-Studio, and it also gave me some guidance on how to gauge sample sizes with the Law of Large Number's Theorem which states that the more random simulations we include, the closer our simulated values will get to their actual probabilities even when they are weighted (White). I found this interesting because one of my biggest concerns in planning my model was how I would weight the value of an Immaculate

Inning, and this theorem allowed me to see that it might be best to ignore weighing my simulated results.

As a result, this is where I began to separate my simulation methodology away from something that was designed around our class example of the Monte Carlo Simulation which was heavily predicated on weighing results in accordance to value of outcome. The main question that I wanted to address in my research project was how rare Immaculate Innings are, not how valuable they are.

Data and Methodology:

To begin, I wanted to get as granular into the makings of an Immaculate Inning as possible. I originally wanted to incorporate models for each pitcher in baseball and explore how batter plate discipline (i.e the probability of a batter chasing a pitch out of the zone, or the probability of a batter to swing and miss at each pitcher's pitch), but this became excessively complex. I resolved to build a simulation upon three variables that were a how many strikes the pitcher threw, how many balls he threw, and how many balls were put in play against him. These are the three true outcomes of every pitch: it can either be hit, called a ball, or called a strike. Additionally, I found that it was redundant to include plate disciplines that a pitcher imposed upon a batter because in the end they merely reflected some smaller piece of each of these three true outcomes. I also found that in building this simulation that incorporating more than one pitcher's template of (Strike%, Ball%, and Ball-In-Play%) percentages were an auxiliary goal that in the end proved to be far too ambitious. As a result, I simulated only one pitcher's Immaculate Inning and decided to choose Mets closer Edwin Díaz. I chose Díaz because I believe that leverage high relievers that minimize balls in play and strive for highest strike

percentages are far more likely to succeed in this simulation at throwing an immaculate inning than contact minded starting pitchers may be. I recognize that by only including one pitcher in my simulation that I reduce the power of any of my conclusions significantly, however; I felt that building a simulation on a pitch-by-pitch basis would be far more interesting than incorporating more pitchers but starting at a broader at-bat by at-bat basis.

I decided to use Edwin Díaz's career totals of balls, strikes, and balls in play to have a larger sample size of data to build the simulation upon than just a single season total of his may have achieved. Díaz has also been a fairly streaky pitcher with lengthy hot and cold streaks, so I believe that his career totals are a far more accurate representation of his holistic abilities.

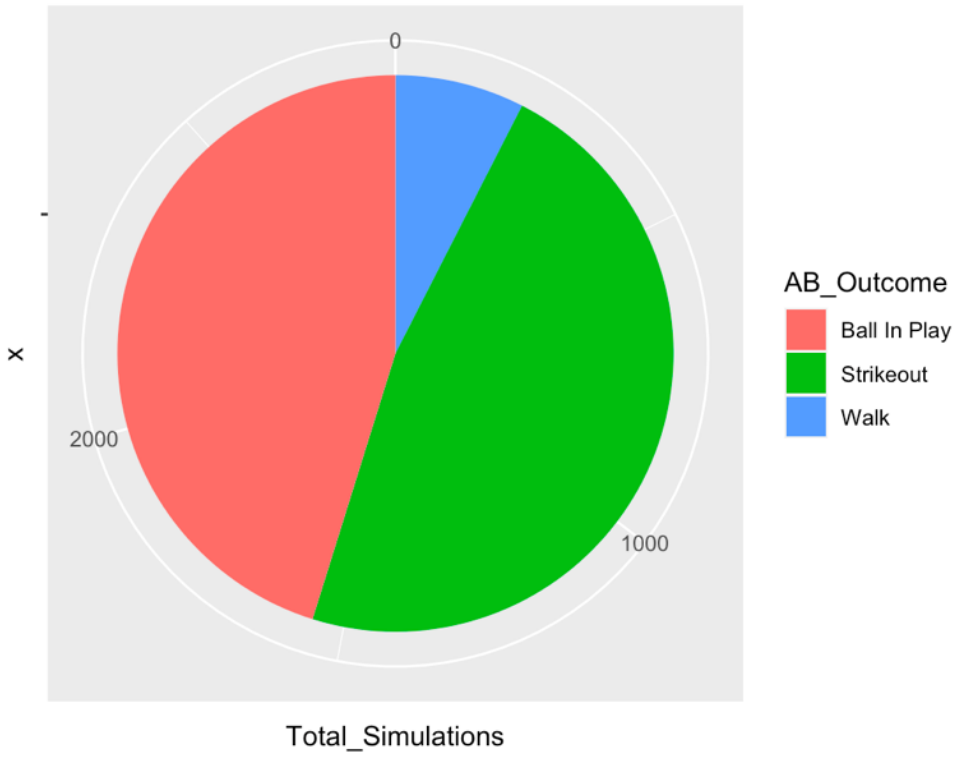
In R, I generated a 10,000 simulated pitch sample off of Diaz's data. This was 10,000 individually random orders and selections of each of the previous three true pitch outcomes. I then began the long journey of translating this 10,000-pitch data frame into actionable play-by-play data that could articulate what the count was of each pitch based upon the previous pitches thrown as well as organize the data into discernable plate appearances and eventually innings. Because this was built upon three true outcomes of a pitch, each at bat was simplified into just three true outcomes: A Walk, a Strikeout, or a Ball in Play. However, at-bats demonstrated more variation in their length (number of pitches thrown) per at-bat.

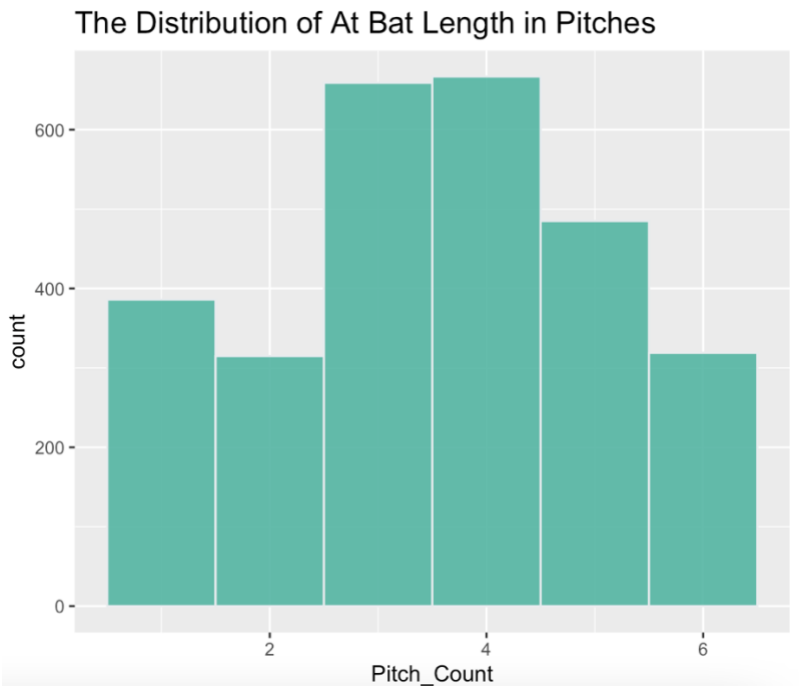
It's important to note that foul balls were not considered in my simulation. All foul balls were cumulatively grouped with ordinary strikes because I could not find any reliable data that discerned foul balls away from strike totals.

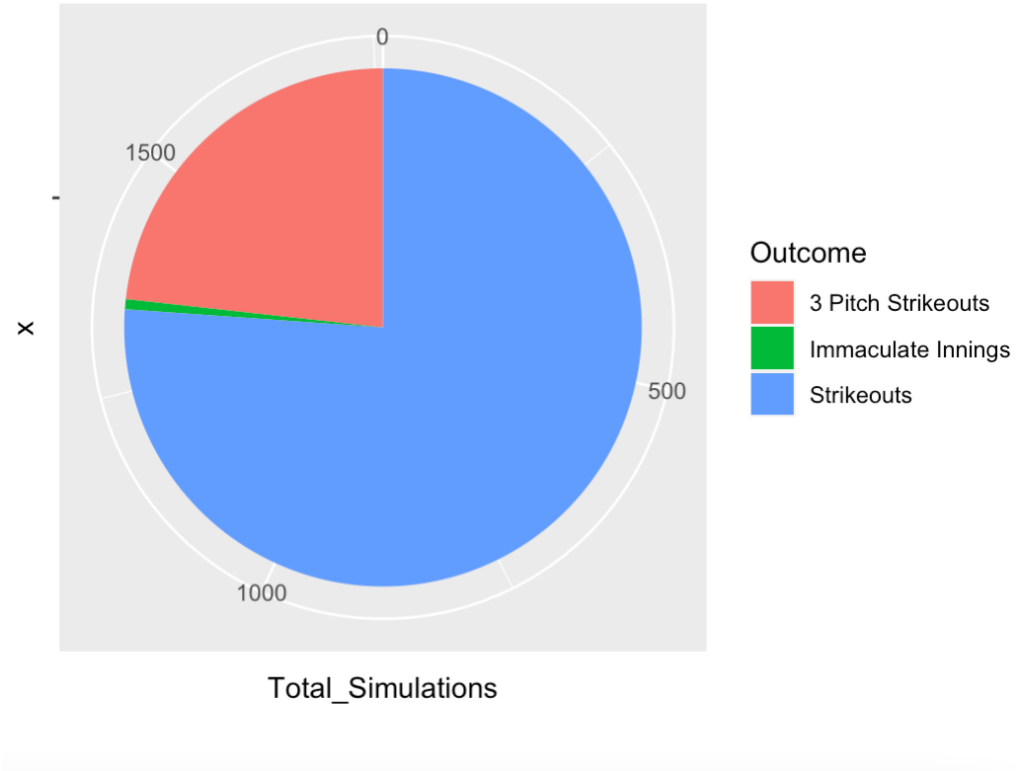
After this larger data frame was developed, it was looped through in its original order and three pitch strikeouts and eventually immaculate innings were scored with binomial variables that were finally totaled to calculate their probabilities of each. It is important to note that the

random order that the 10,000 pitches were developed in was maintained throughout this process, and I made sure to calculate immaculate innings in a manner that preserved the integrity with which the 10,000 pitches were sequenced. The summative results from the simulation were then illustrated in ggplot2, and these graphs and tables are listed below.

Results and Discussion:




The Distribution of At Bat Length in Pitches
600
400
200
0
count
2
4
6
Pitch_Count


0
1500
500
1000
x
Total_Simulations
Outcome
3 Pitch Strikeouts
Immaculate Innings
Strikeouts

	Total number	Proportion
Pitches	10,000	--
Plate Appearances	2,831	--
Strikeouts	1,339	47.298%
Walks	213	7.524%
Balls in Play	1,279	45.178%
3-Pitch Strikeouts	409	14.447%
Immaculate Innings	11	1.166%

As you can see above, we found that roughly 1.166% of Díaz's innings were Immaculate. We can also see that Díaz's results did not fair too far from his career averages. A quick jump to his FanGraphs page and you can find that he has a career BB% of 8.8% which is just slightly higher the 7.524% we projected, and Díaz holds an elite K% 39.5% which is a couple of paces below our simulations projected rate of 47.298% (FanGraphs: Edwin Díaz). Now, I do believe that our inability to incorporate foul balls into our simulation played a big role in the differences between these two rates. When an opponent fouls off 2 strike pitches it allows them to last deeper into counts which will slightly boost their walk rates. However, we holistically claimed that all fouls are ordinary strikes which is not the case. As a result, our strike rate for Díaz is likely inflated higher than it truly is, and that is likely why our strikeout rate is a bit higher than his actual rate. This foul-ball effect can also be shown in our plate-appearance length histogram. We witnessed no at-bats last longer than 6 pitches, and that is because in a world without foul balls 6 pitches is mathematically the longest a plate appearance can last. In reality I believe that

the plate appearance length distribution likely shifts further right than ours does and potentially skews farther right than ours does. This because of the foul ball effect, but also because taking the first pitch or taking until you see a strike is a somewhat common hitter's approach in baseball and as a result a hitter swing rate and ball in play rate are likely not as standard in each count as our simulation assumes. This means that there are likely less 1 and 2 pitch at bats than what we found.

Conclusion:

Because of some of these issues and because I believe leverage relievers with high strikeout rates like Díaz are well optimized for achieving an Immaculate Inning, I think that our Immaculate Inning rate and Three Pitch Strikeout rate are probably somewhat inflated. With that being said, I did not expect the simulation to be quite as accurate as it was in predicting Díaz's K% and BB%. This simulation is definitely something to be improved upon in the future, but it lays a strong foundation for exploring this topic further.

In the future, it would likely be best to run this simulation on all MLB pitchers or at least more than just one. There is certainly room to explore the correlation between pitcher quality or archetype and their immaculate inning rates. This could grant baseball fans some insight onto whether Immaculate Innings are just gimmicks or whether they truly do represent pitcher skill.

References:

Works Cited

Baseball Almanac, Inc. “Immaculate Innings.” *Baseball Almanac*, www.baseball-almanac.com/feats/feats17.shtml.

“Edwin Díaz.” *Edwin Díaz - Stats - Pitching | FanGraphs Baseball*, FanGraphs, www.fangraphs.com/players/edwin-diaz/14710/stats?position=P.

Freeze, R. Allan. “An Analysis of Baseball Batting Order by Monte Carlo Simulation.” *Operations Research*, vol. 22, no. 4, 1974, pp. 728–735. *JSTOR*, www.jstor.org/stable/169949. Accessed 9 Dec. 2020.

Gilfillan, Josh. “Cumulative Sum in R with Reset after Reaching Threshold.” *GitHubGist*, GitHub, 2016, gist.github.com/jgilfillan/23336d0f5bcfffe6a71d0bdd634d023e.

Kenton, Will. “Monte Carlo Simulation.” *Investopedia*, Investopedia, 16 Sept. 2020, www.investopedia.com/terms/m/montecarlosimulation.asp.

McDaniel, Rachael, and Frank Jackson. “Pitchers as Efficiency Experts.” *The Hardball Times*, FanGraphs, 23 Oct. 2015, tht.fangraphs.com/pitchers-as-efficiency-experts/.

“R If Else Elseif Statement.” *Learn By Example*, Learn By Example, 20 Apr. 2020, www.learnbyexample.org/r-if-else-elseif-statement/.

“Sports Illustrated Baseball.” *BoardGameGeek*, boardgamegeek.com/boardgame/4642/sports-illustrated-baseball.

White, Homer. “Beginning Computer Science with R.” *6.2 Monte Carlo Simulation*, 9 Apr. 2020, homerhanumat.github.io/r-notes/monte-carlo-simulation.html.

Winston, Wayne. “3.4 Baseball Monte Carlo Simulation.” *Coursera*, The University of Houston, www.coursera.org/lecture/mathematics-sport/3-4-baseball-monte-carlo-simulation-gn7kW.