Implementation and validation of a probabilistic open source baseball engine (POSBE): Modeling hitters and pitchers

Rhett Tracy Schaefer

Purdue University

Follow this and additional works at: https://docs.lib.purdue.edu/open_access_theses

Part of the Computer Sciences Commons, Sports Management Commons, and the Statistics and Probability Commons

Recommended Citation

https://docs.lib.purdue.edu/open_access_theses/812

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
PURDUE UNIVERSITY  
GRADUATE SCHOOL  
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By  Rhett Tracy Schaefer

Entitled
Implementation and Validation of a Probabilistic Open Source Baseball Engine (POSBE): Modeling Hitters and Pitchers

For the degree of  Master of Science

Is approved by the final examining committee:

Dr. David Whittinghill
Chair

Dr. Esteban Garcia

Jeffrey Brewer

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University’s “Policy of Integrity in Research” and the use of copyright material.

Approved by Major Professor(s):  Dr. David Whittinghill

Approved by:  Dr. Patrick Connolly  4/20/2016

Head of the Departmental Graduate Program  Date
IMPLEMENTATION AND VALIDATION OF A PROBABILISTIC OPEN SOURCE BASEBALL ENGINE (POSBE): MODELING HITTERS AND PITCHERS

A Thesis
Submitted to the Faculty
of
Purdue University
by
Rhett Tracy Schaefer

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science

May 2016
Purdue University
West Lafayette, Indiana
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iv</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACKNOWLEDGEMENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ABSTRACT</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 1. INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

  1.1 Scope | 3 |
  1.2 Significance | 3 |
  1.3 Definitions | 4 |
  1.4 Assumptions | 4 |
  1.5 Limitations | 4 |
  1.6 Delimitations | 5 |
  1.7 Chapter Summary | 5 |

<table>
<thead>
<tr>
<th>CHAPTER 2. LITERATURE REVIEW</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

  2.1 Sports System Simulations | 6 |
  2.2 Poissonian and Bayesian Prediction Models | 16 |
    2.2.1 Poisson Probability Distribution Models | 19 |
    2.2.2 Bayesian Prediction Models | 21 |
  2.3 Sports Experience Simulations | 23 |
  2.4 Chapter Summary | 27 |

<table>
<thead>
<tr>
<th>CHAPTER 3. METHODOLOGY</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
</tr>
</tbody>
</table>

  3.1 Research Question | 29 |
  3.2 Hypothesis | 29 |
  3.3 Apparatus | 29 |
  3.4 Testing Conditions | 29 |
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>Testing Procedures</td>
<td>30</td>
</tr>
<tr>
<td>3.6</td>
<td>The Probabilistic Algorithm</td>
<td>31</td>
</tr>
<tr>
<td>3.7</td>
<td>Data Sources</td>
<td>33</td>
</tr>
<tr>
<td>3.8</td>
<td>Specific Measures for Success</td>
<td>33</td>
</tr>
<tr>
<td>3.9</td>
<td>Threats to Validity</td>
<td>33</td>
</tr>
<tr>
<td>3.10</td>
<td>Chapter Summary</td>
<td>34</td>
</tr>
<tr>
<td><strong>CHAPTER 4.</strong></td>
<td>RESULTS</td>
<td><strong>35</strong></td>
</tr>
<tr>
<td><strong>CHAPTER 5.</strong></td>
<td>CONCLUSIONS</td>
<td><strong>38</strong></td>
</tr>
<tr>
<td><strong>LIST OF REFERENCES</strong></td>
<td></td>
<td><strong>41</strong></td>
</tr>
<tr>
<td><strong>APPENDICES</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Appendix A</td>
<td></td>
<td><strong>44</strong></td>
</tr>
<tr>
<td>Appendix B</td>
<td></td>
<td><strong>46</strong></td>
</tr>
<tr>
<td>Appendix C</td>
<td></td>
<td><strong>51</strong></td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Hitter’s Card</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Pitcher’s Card</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Dice Probabilities</td>
<td>8</td>
</tr>
<tr>
<td>2.4 Example Poisson Distribution</td>
<td>16</td>
</tr>
<tr>
<td>2.5 Bayes Theorem Infographic</td>
<td>18</td>
</tr>
<tr>
<td>4.1 POSBE Simulated P-Values</td>
<td>35</td>
</tr>
<tr>
<td>4.2 ‘Hitters 1-5’ Doubles Histogram</td>
<td>36</td>
</tr>
<tr>
<td>Appendix Figure</td>
<td></td>
</tr>
<tr>
<td>B.1 Concept-Loading Screen</td>
<td>48</td>
</tr>
<tr>
<td>B.2 Concept-Favorite Team Selection</td>
<td>48</td>
</tr>
<tr>
<td>B.3 Concept-Main Menu</td>
<td>49</td>
</tr>
<tr>
<td>B.4 Concept-Game Mode</td>
<td>49</td>
</tr>
<tr>
<td>B.5 Concept-Settings</td>
<td>50</td>
</tr>
<tr>
<td>B.6 Concept-Game View</td>
<td>50</td>
</tr>
<tr>
<td>C.1 Hitters 1-5 Singles</td>
<td>51</td>
</tr>
<tr>
<td>C.2 Hitters 1-5 Doubles</td>
<td>51</td>
</tr>
<tr>
<td>C.3 Hitters 1-5 Triples</td>
<td>52</td>
</tr>
<tr>
<td>C.4 Hitters 1-5 Homeruns</td>
<td>52</td>
</tr>
<tr>
<td>C.5 Hitters 1-5 Walks</td>
<td>53</td>
</tr>
<tr>
<td>C.6 Hitters 1-5 Hit-by-Pitch</td>
<td>53</td>
</tr>
<tr>
<td>C.7 Hitters 1-5 Strikeouts</td>
<td>54</td>
</tr>
<tr>
<td>C.8 Hitters 6-9 Singles</td>
<td>54</td>
</tr>
<tr>
<td>Appendix Figure</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>C.9 Hitters 6-9 Doubles</td>
<td>55</td>
</tr>
<tr>
<td>C.10 Hitters 6-9 Triples</td>
<td>55</td>
</tr>
<tr>
<td>C.11 Hitters 6-9 Homeruns</td>
<td>56</td>
</tr>
<tr>
<td>C.12 Hitters 6-9 Walks</td>
<td>56</td>
</tr>
<tr>
<td>C.13 Hitters 6-9 Hit-by-Pitch</td>
<td>57</td>
</tr>
<tr>
<td>C.14 Hitters 6-9 Strikeouts</td>
<td>57</td>
</tr>
<tr>
<td>C.15 Starting Pitchers Singles</td>
<td>58</td>
</tr>
<tr>
<td>C.16 Starting Pitchers Doubles</td>
<td>58</td>
</tr>
<tr>
<td>C.17 Starting Pitchers Triples</td>
<td>59</td>
</tr>
<tr>
<td>C.18 Starting Pitchers Homeruns</td>
<td>59</td>
</tr>
<tr>
<td>C.19 Starting Pitchers Walks</td>
<td>60</td>
</tr>
<tr>
<td>C.20 Starting Pitchers Hit-by-Pitch</td>
<td>60</td>
</tr>
<tr>
<td>C.21 Starting Pitchers Strikeouts</td>
<td>61</td>
</tr>
<tr>
<td>C.22 Relief Pitchers Singles</td>
<td>61</td>
</tr>
<tr>
<td>C.23 Relief Pitchers Doubles</td>
<td>62</td>
</tr>
<tr>
<td>C.24 Relief Pitchers Triples</td>
<td>62</td>
</tr>
<tr>
<td>C.25 Relief Pitchers Homeruns</td>
<td>63</td>
</tr>
<tr>
<td>C.26 Relief Pitchers Walks</td>
<td>63</td>
</tr>
<tr>
<td>C.27 Relief Pitchers Hit-by-Pitch</td>
<td>64</td>
</tr>
<tr>
<td>C.28 Relief Pitchers Strikeouts</td>
<td>64</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

Nick Schneider was vital to the success of this project. Once the baseball engine logic was established in the pseudocode, he helped to implement and convert the pseudocode into the correct C++ syntax as well as to provide critical code audits throughout the life of the project. His assistance was crucial to success of the baseball engine. This project could not have been completed without his help.

Dr. David Whittinghill played a significant role in the completion of this project. He served as the chair of the committee for this thesis and met with the researcher on a regular basis to discuss, assist, and troubleshoot any issues with the project. His support and encouragement was invaluable to the success of the project.

Nathan Hankey played a major role in finding the most appropriate means of analyzing the simulation results. This project depended heavily on sophisticated statistical analysis and his assistance made it possible to explore a much broader range of tools and approaches and thus find the best approach for analyzing our particular data.

This researcher would also like to thank Dr. Esteban Garcia Bravo and Professor Jeffery L Brewer for the excellent support and feedback throughout the entire process.
ABSTRACT

Schaefer, Rhett, T. M.S., Purdue University, May 2016, Implementation and Validation of a Probabilistic Open Source Baseball Engine. Major Professor: Dr. David Whittinghill.

This manuscript details the implementation and validation of an open source probabilistic baseball engine (POSBE) that focuses on the hitter and pitcher model of the simulation. The simulation produced outcomes that parallel those observed in actual professional Major League Baseball games. The observed data were taken from the nineteen games played between the New York Yankees (NYY) and Boston Red Sox (BOS) during the 2015 season. The potential hitter/pitcher outcomes of interest were singles, doubles, triples, homeruns, walks, hit-by-pitch, and strikeouts. The nineteen game series was simulated 1000 times, resulting in a total of 19,000 simulations. The eighteen hitters and twenty-seven pitchers were each divided into four groups based on similar characteristics: Hitters 1-5 in the batting order, Hitters 6-9 in the batting order, Starting Pitchers, and Relief Pitchers. Using the Kolmogorov-Smirnov test, the simulated data were compared against the observed data to obtain appropriate p-values. The calculated p-values were all greater than 0.05 indicating that the POSBE algorithm predicts hitter and pitcher outcomes as they relate to empirical observation.
CHAPTER 1. INTRODUCTION

Developing a realistic software baseball engine can be a daunting task due to the complexity of baseball. Countless rules, obscure scenarios, and myriad statistics all need to be incorporated into the engine to ensure it produces realistic results. There has been little research regarding the best practice and processes of organizing and structuring a software-based, realistic baseball engine. Numerous commercial products are available like MLB The Show (PlayStation 4) and Baseball Mogul (PC), but even if one purchases a game like this there is no way to see the engine’s algorithms or methods since they are generally proprietary technology. It makes sense that the information used to build these commercial products is kept secret, because it would be bad business practice to give up their competitive advantage.

To address this gap, the researcher has developed an open-source baseball engine that will be able to generate realistic results based on historical data inputs. Tests of the simulation indicated that the hitter and pitcher models paralleled empirically observed results from actual professional Major League Baseball (MLB) games.

This manuscript documents the implementation, testing, and validation of said engine as follows. Section 2.1 describes a variety of research on predictive sports system simulations. The studies focus primarily on predicting game outcomes and modeling players. The results and conclusions reached in the studies demonstrated effectiveness in
realistically predicting player and game outcomes. Section 2.2 first describes Poissonian and Bayesian Models and what their relevance is to this thesis. The Subsection 2.2.1 focuses specifically on Poisson Probability Distribution Models and how it has been used to predict soccer and water polo outcomes. Similarly Subsection 2.2.2 discusses Bayesian Prediction Models and the research that has been done with them in relation to predicting the outcomes of soccer games. The results of both models, Poisson Probability Distribution Models and Bayesian Prediction Models, performed extremely well in predicing the outcomes of sporting events. Lastly Section 2.3 describes Sports Experience Simulations. Although this section is an independent inquiry from the previous two sections focusing on systems-based simulations, it provides insight on how to best communicate the results of the probabilistic open source baseball engine. All the studies but two (the others involve basketball and golf) focus on visual baseball simulations. Many of the research topics in this section tested factors like eye movements of hitters, the effects of focus of attention on hitting performance, and sports specific decision-making in terms of reaction time. The results of these studies give an examples of how this baseball engine could be applied in an experience-based approach.

A series of simulations were run with the goal of the simulation producing results that mirror those observed in actual, professional baseball games. This simulation focuses predominantly on hitter and pitcher models as it represents the foundation of any simulated game of baseball. Section 3 details how the simulation was constructed and the statistical considerations that went into the design.

Section 4 discusses the results of the pitcher and hitter models that serve as the core functionality to determine the outcomes of the baseball games. Using the Kolmogorov-
Smirnov test the researcher was able to determine the results of the study to be significant. Thus the researcher concluded that the hitter and pitcher probabilistic models produced realistic results that paralleled those observed in the nineteen games played between the New York Yankees and Boston Red Sox during the 2015 season.

Section 5 discusses alternate explanations, potential alternate implementations, and discussion about the attempts to integrate traditional Bayesian modeling into the baseball engine and, lastly, considers the implications of this research in terms of potential application areas (HBB – Honoring Black Baseball) and recommended future directions for this work.

1.1 Scope

The study was delimited to the implementation and validation of an open source baseball engine focused specifically on modeling hitters and pitchers. The goal of this study was to produce hitter and pitcher statistical outputs that would parallel those observed in the actual professional MLB games between the New York Yankees (NYY) and the Boston Red Sox (BOS). The accuracy of the engine was evaluated using the nineteen games played between the NYY and BOS during the 2015 season.

1.2 Significance

The significance of implementation and validation of the baseball engine is that it will be the first open source baseball engine available to the open source development community. By the virtue of the engine being open and well-documented, it will be outwardly verifiable and extensible. Anything the researcher may not have implemented can be identified, and also any potential future MLB rule changes can be added or subtracted into the engine by the open source development community.
1.3 Definitions

- MLB – Major League Baseball
- NYY – New York Yankees
- BOS – Boston Red Sox
- Position Player – A player on defense who plays as an infielder, outfielder, or catcher.
- Small ball – an offensive baseball strategy that favors situational hitting and baserunning tactics over pure hitting in an effort to make efficient use of scoring opportunities in any given inning.
- Situational hitting – A baseball hitting strategy, usually with runners on base, with the goal being to advance or score the baserunners in the most effective way possible.

1.4 Assumptions

The assumptions for this project include:

- Interchanging different players and their statistics should produce outputs that parallel those observed in actual professional Major League Baseball (MLB) games.
- Using data from seasons not being specifically tested should produce realistic results that would be observed in actual professional MLB games during that season.

1.5 Limitations

The limitations for this project include:

- The researcher focused on professional Major League Baseball (MLB) games.
• The researcher evaluated the player and pitcher data of the 2015 nineteen game 2015 series between the New York Yankees and Boston Red Sox.

• The researcher only incorporated pitching substitutions into the engine—no hitter, baserunner or fielder substitutions.

• The researcher primarily focused on the batter/pitcher interaction on an at-bat basis, thus putting the most focus upon hitter and batter modeling.

1.6 Delimitations

The delimitations for this project include:

• The researcher did not allow position players to enter the game as the pitcher (this applies in long extra-inning games).

• The researcher did not look at the minor league baseball teams that serve under their MLB counterparts.

• The researcher did not program a detailed, multi-team baseball simulation.

• The researcher did not address complex baseball scenarios, in which the outcome is decided by the umpire’s judgement, interpretation, or discretion.

1.7 Chapter Summary

This chapter introduced the scope and significance of the research in this proposal. Additionally this chapter discussed the assumptions, limitations, and delimitations that will or will not be included in the researcher’s. Lastly, several significant words and phrases were defined to assist reader understanding of specific baseball vernacular.
CHAPTER 2. LITERATURE REVIEW

Most people and game companies who have produced baseball engines have chosen not to share the algorithms or methods they used to create their baseball engine due to their proprietary, commercial nature. This can be attributed to the financial incentive of keeping it a secret and maintaining a competitive advantage over their competitors by having the most realistic simulation on the market. As such, research regarding how to build an accurate and realistic baseball simulation is limited. In the research paper, *Building a Baseball Simulation*, the researcher specifically mentions the sensitive nature of the topic. “Other game makers are more secretive about the specifics of their algorithms” (Hastings, 1999, p. 34). The following sections will explore sports system simulations, Bayesian statistical modeling, Poisson distributions, and sports experience simulations that relate to the creation of a realistic open source baseball engine.

2.1 **Sports System Simulations**

Baseball simulations have existed for almost a century and a half when the simulation, Parlour Baseball, was first created in 1866. Research studies have been conducted using baseball simulations to answer various questions ranging from baseball strategies to human kinetics. In 1973, R. Allan Freeze was one of the first researchers to use a baseball simulation for analysis. Using a Monte Carlo simulation, he analyzed the effect that batting order had on the outcome of baseball games. Traditionally, a team’s best hitters
bat third through fifth in the batting order. The good hitters, who get on base consistently, typically bat first and second in the batting order. While batters six through nine are the team’s least productive hitters (the ninth batter being the worst). After analyzing the data from 220,000 baseball game simulations, Freeze (1973) determined that batting order “only exerts a small influence on the outcome of baseball games…using the best batting order rather than the worst is less than three wins per 162-game season” (p. 728).

In baseball, there is a metric called WAR (Wins Above Replacement). WAR is used to judge a player’s overall performance and contribution to their team. This statistic combines the value of the player in all aspects of the game (hitting, pitching, fielding, and baserunning). Baumer, Jensen, and Matthews (2015) criticize this method because it’s dependence “upon proprietary data, ad hoc methodology, and opaque calculations” (p. 69). Consequently, the researchers proposed and tested their own method, openWAR, which was “based on public data, a methodology with greater rigor and transparency, and a principled standard for the nebulous concept of a ‘replacement’ player” (Baumer, Jensen, & Matthews, 2015, p. 69). After running several simulations using the openWAR method, their results indicated a more comprehensive and accurate measure of player performance that was transferrable to other domains.

A year before the turn of the 21st century, Hastings (1999) studied different approaches to building a baseball simulation. The approaches ranged from simple to complex. The simple game that Hastings described did not factor pitching into the equation and solely focused on modeling the hitter’s batting average and power potential. Each hitter’s plate appearance is determined by a hitter card (shown below in Figure 2.1) and three dice.
The batter rolls three dice simultaneously one of which is red and the others white. The red die determines the column and the sum of the white dice determines the row of the particular hitter’s grid in which is the outcome of the at-bat. The pitcher’s card determines whether the hitter reaches base on a walk or hit batsman or makes an out (Hastings, 1999, p. 33).

To create a hitter’s card for each player and type of hit you would only need to compute each player’s success percentage. Next, find each percentage and round it to an integer of 216th. Lastly, ensure that the total number of symbols for each type of hit, corresponds to the correct number of 216ths (Hastings, 1999, p. 33). The dice probabilities for each row is shown below in Figure 2.3.

<table>
<thead>
<tr>
<th>Row</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
</table>

Figure 2.3 Dice Probabilities
With baseball being a complex game, there are specific scenarios (sacrifices, hit-by-pitches, and walks) when an at-bat is not charged to the hitter. As a result, the hitter card success percentages would be described as ‘hits per plate appearance’ as opposed to ‘hits per at-bat’; however, the primary issue with this simple baseball simulation is that there is no way to realistically incorporate pitcher performance (besides strikeout and walk percentages). The next section of study focuses simulating batting averages and differentiating homerun hitters against a variety of different skilled pitchers.

Another approach used that incorporates pitchers is to take the averages of the hitter and pitcher and then split the difference. Hastings (1999) explains this method, “For example, for the .250 hitter against the pitcher who allows a .200 average to hitters facing him, simulate so that the hitter’s conditional probability of a hit given a charged at-bat is .225” (p. 34). But again this method violates common sense. Hastings (1999) continues:

If, in a league which only hits .250 as a whole, a .280 hitter faces a .280 pitcher, the simulated batting average would be .280 under this plan. It makes better sense that the .280 hitter should hit .280 against the average .250 pitcher and higher than .280 against the relatively poorer .280 pitcher (p.34).

Hastings states he has heard of a popular simulation using the split the difference (between batting averages) approach above but hasn’t been able to confirm it.

Another well-known baseball simulation, Strat-O-Matic attempts to make sense of this issue by taking a different approach. In this simulation both the hitter and pitcher cards have plate appearance results have a card that gives the result of the plate appearance with each card having a fifty percent probability of being used. It’s important to note that these cards would be different than ones shown in Figure 2.1 and Figure 2.2,
as both cards would not have any blank spaces. There are only minor issues with this solution due to the arbitrary selection of 0.5 probability or the fact that the hitter has no influence on the result of the plate appearance half of the time (Hastings, 1999, p. 34).

Next, Hastings (1999) discusses rumors of artificially producing data based on previous games played to steer player statistics towards their actual averages from past seasons (p. 34). As a result, the simulation would make a player who is performing at a higher level than normal (aka having a “career year”) less likely to continue producing those same statistics. For anyone who is an athlete, sports fan, or statistician this is problematic because variation is common in sports and is only an issue if it becomes a reoccurring theme.

The last approach Hastings proposes is cancellation. Hastings (1999) supports this scheme for the reason that it simulates the “underlying physics of how the hitter-pitcher confrontation plays out…” and “exposes a degree of freedom in how pitcher and hitter cards can be created” (p. 34). This would allow for the batter and pitcher to roll their dice at the same time. Additionally, if on the same turn a pitcher rolls an out and the batter rolls a hit, the pitchers result would take precedence over the batter’s result. While only considering walks, hits, and outs, Hastings (1999) developed a cancellation algorithm for the pitchers and batters simultaneous dice rolls:

1. If the pitcher lands on a walk, that is the outcome.
2. If the hitter lands on a hit and the pitcher does not land on a cancel square, the outcome is the indicated hit.
3. If the hitter lands on a hit and the pitcher lands on a cancel square, the outcome is the indicated out on the pitcher card.
4. If the hitter lands on a blank and the pitcher on an out, the outcome is the out. (p. 34)
The cancellation algorithm ignores complex baseball occurrences like wild pitches, the hitter being hit by a pitch, errors, etc. Building other aspects of the baseball simulation would be solved in a similar fashion. The same overall hit method would be used for doubles, triples, and homeruns. Pitcher cards would include wild pitches, hit by pitch, and strikeouts to predict tendencies and batter cards would include errors based on league error frequencies for a realistic simulation.

Based on this cancellation approach Hastings wrote a program using data from the 1997 MLB season to create hitter and pitcher cards. The researcher used the Chicago Cub’s first baseman, Mark Grace, a .315 hitter, along with twelve pitchers from the St. Louis Cardinals. The researcher simulated 50 plate appearances against each pitcher that resulted in significant variation—similar to that of an actual MLB game—while revealing hitting trends that paralleled his actual game performance. For example, Mark Grace was known for hitting doubles but not a lot of homeruns. These results were consistent with the simulated results. After a statistical analysis, Hastings concluded “that the simulated batting averages are not significantly more variable than those observed in the real player’s performance” (1999, p. 37).

Most recently in 2015, Ruiz and Perez-Cruz developed a model for predicting Men’s NCAA (National Collegiate Athletic Association) Division I basketball results. This model is based on previously designed simple soccer models and Poisson factorization while accounting for the structure of Men’s NCAA Division I basketball. The researchers modified the previous research in two ways. This way, Ruiz and Perez-Cruz (2015) were able “to capture both the specific behavior of each NCAA conference and different strategies of teams and conferences” (p. 39). Each team was differentiated
using attack and defense coefficients in order to give specific teams unique characteristics. Ruiz and Perez (2015) were able to conclude that their model “tends to provide results that differ more from the implicit probabilities of the [six online] betting houses…” (p. 39). Additionally, the results from the model outperformed—higher mean profit—the winner of the recent Kaggle 2014 NCAA Tournament competition. Kaggle is an online community, made up of companies and researchers who compete with one another to create the world’s best predictive and analytic models.

In the Men’s NCAA basketball study, A Mixture-of-Modelers Approach to Forecasting NCAA Tournament Outcomes, a group of researchers worked as a team to predict the outcomes of the 2014 Men’s NCAA basketball tournament. Through their research they were able to support their notion that “simple algorithms outperform more complicated models with numerous features” (Yuan, Liu, Yeh, Kaufman, Reece, Bull, Franks, Wang, Illushin, & Bornn, 2015, p. 14). Additionally, their most successful models did not contain “contaminated data.” A set of data is considered “contaminated” when it contains results from that year’s March Madness tournament. Ultimately, most of the researcher’s models did outperform their baseline prediction. The study concluded with the following four recommendations for developing a similar NCAA March Madness tournament model: “pay careful attention to the issue of feature contamination and feature selection, choose modeling approaches befitting the quality and quantity of available training data, customize algorithm selection based on target loss function, and don’t cast aside simple models…” (Yuan, Liu, Yeh, Kaufman, Reece, Bull, Franks, Wang, Illushin, & Bornn, 2015, p. 14). By following these four recommendations, future predictions models should yield superior results.
Another group of researchers took on the challenge of looking at each potential matchup in the 2014 Men’s NCAA basketball tournament. The name of this study was *Nearest-neighbor Matchup Effects: Accounting for Team Matchups for Predicting March Madness*. The researchers hoped to differ from previous predictive models by basing their model entirely on the overall strength of the team. Instead, Hoegh, Carzolio, Crandell, Hu, Roberts, Song, and Leman (2015), introduced their “nearest-neighbor matchup effects framework, which presents a flexible way to account for team characteristics above and beyond team strength that may influence game outcomes” (p. 29). Although the relative strength of the team remains the dominate force in the model, specific matchup characteristics do play a significant role in the model’s functionality. The study used a simple linear model for predicting matchups; however, this highly flexible framework can support more complex relative strength models and non-linear additive models.

Next, Gupta introduced a new approach to predicting Men’s NCAA basketball tournament brackets. In the study, *A New Approach to Bracket Prediction in the NCAA Men’s Basketball Tournament Based on a Dual-proportion Likelihood*, the researcher began by reviewing relevant previous methods and approaches. After this review, the researcher proposed a novel rating system for teams that incorporated game data from the previous season up to the tournament. The system included a four-predictor probability model that was used to predict the bracket results of each Men’s NCAA basketball tournament from 2009-2014. Additionally, Gupta (2015) performed a Monte Carlo simulation to determine if the model required modifications “from an expected value-based prediction system…in order to have the maximum bracket score within a defined
group” (p. 53). The results of his Monte Carlo simulation indicated that modifying the model’s selections—selecting a high probability upset in the first three rounds—increased the likelihood of a more accurate bracket.

The final Men’s NCAA basketball tournament study, Building an NCAA Men’s Basketball Predictive Model and Quantifying its Success, focused on a prediction model using logistic regressions. The researchers, Lopez and Matthews (2015), attempted “to quantify the degree to which luck played a role in the success of this model by simulating tournament outcomes under different sets of true underlying game probabilities” (p. 5). Like a couple of the other Men’s NCAA basketball tournament studies reviewed above, the researchers stress the importance of informative, uncontaminated data. A simple model using traditional statistical methods with high quality data can match or surpass the accuracy of more complex models. The results of this study showed that the researcher’s model had approximately a 12% chance of being the best competing submission and slightly less than a 50% chance of finishing as one of the top ten scores. Lastly, it is important to note that even with a highly realistic predictive model, it takes an enormous amount of luck to win an NCAA tournament bracket pool.

Similar to the bracket prediction research in Men’s NCAA college basketball, a National Hockey League (NHL) study was performed comparing two different vector machines—relevance vector machine (RVM) and support vector machine (SVM). This study was called, Riding a Probabilistic Support Vector Machine to the Stanley Cup. The purpose of the study was to evaluate NHL team playoff performance expectations and identify the teams who were under-achievers and over-achievers. Demers (2015) found: “Despite the potential of the RVM approach, the SVM algorithm proved to be superior in
the context of hockey playoffs” (p. 205). Due to the competitive nature of the NHL playoffs, chance and other intangible influences played a significant role in determining the outcomes. The results of the study showed that NHL playoffs are phenomenally competitive and playoff upsets will continue to occur.

The final article, *The Implied Volatility of a Sports Game*, discusses Polson’s and Stern’s research regarding predicting the outcome of a sports contest. The study is based on previous research by Stern, who applied the Brownian Motion Model to baseball scores even though it was originally intended for basketball scores. The Brownian Motion Model is based off a simple relationship between the home team’s lead/deficit and a particular point in time during the game. Utilizing those two factors, the model derives the home team’s probability of victory. The purpose of Stern’s (1994) research was to obtain “a simple relationship between the home team’s lead (or deficit) at time $t$ and the probability of victory for the home team” (p. 1128). Although Stern’s process originally appeared too discrete to properly model baseball, the Brownian motion model worked with baseball surprisingly well. Twenty-one years later, Polson teamed up with Stern to further investigate the unpredictability of sports scores. Polson and Stern (2015) extended Stern’s previous model “to calculate the time-varying implied volatility during the game using inputs from real-time, online betting and to identify betting opportunities” (p. 145). A team’s chances of winning continuously changes over the course of the game. At the end of a game when the score is close, every time a team turns the ball over to the other team, that team’s chances of winning drastically changes. Through this study, the researchers illustrated this concept by showing how the market-implied volatility changed throughout Super Bowl XLVII (Baltimore Ravens versus San Francisco 49ers).
2.2 Poissonian and Bayesian Prediction Models

The second section of this literature review will begin by giving a brief introduction explaining Poisson and Bayesian predictions models. Next, a few papers will be presented describing Poisson and Bayesian models and their application in sports system simulations research. Poisson distributions are used to count data and rare events over a specific time. In the context of an individual inning, runs are rare in a baseball game. The Poisson distribution adequately conveys how often runs are scored during innings, but it is not perfect. For instance, examine an example distribution (Figure 2.4) below:

![Example Poisson Distribution](http://stats.seandolinar.com/mlb-poisson-distribution-to-model-runs-scored-per-inning/)

**Figure 2.4 Example Poisson Distribution**

- Red line = average runs for a given situation
- Blue area = actual run frequency
- Gold line = distribution obtained from regression

This graph from 2013 displays the run expectancy data when an inning starts. This graphic indicates that this team, the Pittsburg Pirates, should expect to score less than a
half of a run per inning. The Poisson distribution line (red line) underestimates probability of a shutout (no runs scored) inning or a big-run inning (three or more runs scored). The reason it is not a perfect fit is because baseball is not random. Situations occur where more talented teams will score multiple runs in an inning versus less-talented teams who will not score any runs—resulting in wider run scoring variance.

The Bayesian probability theorem measures the degree of belief that something will happen using conditional probabilities. The variables are defined based on historic data, which then can be applied to similar future events. As more evidence/variable(s) are considered the degree of belief will change. The infographic (Figure 2.5) on the next page shows a basic example of how Bayesian prediction works.
Figure 2.5 Bayes Theorem Infographic
http://www.sports-management-degrees.com/baseball/
2.2.1 Poisson Probability Distribution Models

Prior to the 1980s, several researchers opted for the Negative Binomial model and rejected the Poisson model for modeling professional European soccer scores. The issue with the Negative Binomial model is that it does not account for any variation in a team possessing more talent than another. Consequently, Maher investigates the capabilities of the Poisson model to predict soccer outcomes. The Poisson model established offensive and defensive strength coefficients for each team. With these parameters, Maher’s (1982) model showed only “some small systematic differences, an independent Poisson model gives a reasonably accurate description of football [American soccer] scores” (p. 109). To fix some of these discrepancies, the researcher used a bivariate Poisson model that considerably improved the model fit for soccer score differences.

Eighteen years later, another soccer simulation was created based on a Poisson model to predict the distribution of scores in the 1998 World Cup tournament. This simulation uses a rating for each team and the location of the match. The success of the model’s simulation relied on two assumptions: “The number of goals scored by a team in a soccer match is Poisson distributed” and “It is independent of the number of goals scored by the opposing team” (Dyte & Clarke, 2000, p. 994). In order to improve the effectiveness of the model’s predictions, manual adjustments were made to FIFA’s (International Federation of Association Football) rating data. Ultimately the model performed relatively accurately with a similar number of upsets results (the team expected to win loses) and total number of goals scored in the tournament when compared to the predictions. The researchers suggest that future models could be improved by updating the team ratings as the tournament progresses.
Most of the previous studies on Poisson modeling in sports use an independent Poisson distribution. The independent models are typically used to model the two team’s number of goals; however, this study considers a bivariate Poisson model that allows for a correlation between the two competing team’s scores. Karlis and Ntzoufras (2003) believe this correlation is “a plausible assumption in sports with two opposing teams competing against each other” (p. 381). To demonstrate the bivariate Poisson model, the researchers use data sets from soccer and water-polo. Upon extending the bivariate Poisson model to the diagonal inflated model, the model was not only able to handle correlation but overdispersion as well. Incorporating overdispersion into the model “improves in the precision the estimation of draws and, at the same time, allows for overdispersed, relative to the simple Poisson distribution, marginal distributions” (Karlis & Ntzoufras, 2003, p. 381). The models and methods used in this study can easily be integrated into more complex models.

The final article in this subsection proposes an approach to analyze and predict the results the FIFA World Cup in 2014. This study, Prediction of Major International Soccer Tournaments Based on Team-specific Regularized Poisson regression: An Application to the FIFA World Cup 2014, “is based on a regularized Poisson regression model that includes various potentially influential covariates describing the national teams’ success in previous FIFA World Cups” (Groll, Schauburger, & Tutz, 2015, p. 97). The model’s framework incorporates team-specific variables further improving the simulation’s accuracy. To test the model, the researchers repeatedly simulate the FIFA World Cup 2014 and recorded each team’s probability of winning. The simulation results
show Germany and Brazil to be the two favorites to win the 2014 World Cup, with Germany favored over Brazil.

2.2.2 Bayesian Prediction Models

This study, *Predicting Football Results Using Bayesian Nets and Other Machine Learning Techniques*, evaluates the performance a Bayesian network models versus other machine learning techniques to predict the game outcomes of the professional European soccer team, Tottenham Hotspur, from 1995-1997. “Bayesian networks (BNs) provide a means for representing, displaying, and making available in a useable form the knowledge of experts in a given field” (Joseph, Fenton, & Neil, 2006, p. 544). The goal of the study was to determine the accuracy of the Bayesian network compared to alternative machine learning models. The other models evaluated were MC4, Naïve Bayesian Learner, Data Driven Bayesian, and K-nearest neighbor learner. The results of the study show the Bayesian network model to be superior to the other learning machines in the domain of predictive accuracy. The results are especially surprising because this study’s assumptions positioned the Bayesian network model at a disadvantage. The applications of the Bayesian network model are plentiful with its ability to accurately predict the outcome of sporting events while only using minimal data.

Researchers from Seoul National University in South Korea developed a soccer prediction framework, called FRES (Football Result Expert System), utilizing Bayesian inference, rule-based reasoning, and an in-game time-series approach. By using the Bayesian network component and rule-based reasoner, games could be accurately predicted, even when data between the opposing teams is scarce. Additionally the researchers, Min, Kim, Choe, Eom, and McKay’s (2008) framework was “able to
consider many factors, such as current scores, morale, fatigue, skills, etc. when it predicts the results of sports matches: most previous work considered only one factor, usually the score” (p. 551). The last feature they incorporated into their framework was an in-game time-series approach. This approach resulted in more realistic and accurate predictions because it was able to account for a team’s psychological momentum. The Oxford Dictionary of Sports Science & Medicine (2006) defines psychological momentum as:

"The positive or negative change in cognition, affect, physiology, and behavior caused by an event or series of events that affects either the perceptions of the competitors or, perhaps, the quality of performance and the outcome of the competition. Positive momentum is associated with periods of competition, such as a winning streak, in which everything seems to ‘go right’ for the competitors. In contrast, negative momentum is associated with periods, such as a losing streak, when everything seems to ‘go wrong’.

The ability to integrate an intangible concept, like psychological momentum, into a framework is not only novel to this field of study but yields compelling results as well. Upon running the simulation, FRES was able to give plausible and reliable predictions for the 2002 FIFA World Cup.

Also in 2008 two researchers, Baio and Blangiardo, proposed a soccer prediction outcome framework using a Bayesian hierarchical model. The predictive strength of this framework was evaluated based on data from the 1991-1992 Italian Serie A championship. To mitigate the overshrinkage issue associated with the Bayesian hierarchical model, Baio and Blangiardo (2008) used “a more complex mixture model that results in a better fit to the observed data” (p. 253). The mixture model performance was tested using 2007-2008 Italian Serie A championship. Ultimately, the results of the test indicated that the framework produced reasonable results.
In the final study of this section, *pi-football: A Bayesian Network Model for Forecasting Association Football Match Outcomes*, the researchers attempted to use a Bayesian network model to predict soccer outcomes using significant subjective variables that were not included in the historical data. This model, *pi-football*, was used to produce outcomes from the 2010-2011 English Premier League (EPL) matches. The framework first ran the objective prediction—supported by historical data—and was later modified according to the results of the subjective variables. The revision was done by evaluating time-dependent data using degrees of uncertainty. The researchers’, Constantinou, Fenton, and Neil (2012), prediction results showed that “using an appropriate measure of forecast accuracy, the subjective information improved the model such that posterior forecasts were on par with the bookmakers’ performance” and “…the model generates profit under maximum, mean, and common bookmakers’ odds, even allowing for the bookmakers’ built-in profit margin” (p. 322). Thus, the *pi-football* framework not only produces accurate match outcomes, but is precise enough to beat the bookmakers’ odds.

2.3 Sports Experience Simulations

One of the first sports experience simulations was performed by the researcher Wayne Burroughs in 1984. In his work, published in the *International Journal of Sports Psychology*, he aimed to “evaluate the effectiveness of visual simulation training film approaches to enhancing the visual pitch recognition and pitch location skills of collegiate baseball batters” (Burroughs, 1984, p. 118). This study, *Visual Simulation Training of Baseball Batters*, used an apparatus called, Visual Interruption System (designed for bullpen use), that recorded pitch location scores in real time. The results of Burroughs (1984) study indicated “significant gains in [baseball batters] location scores
and non-significant changes in pitch recognition scores (probably due to initially high pretest levels)” (p. 117).

In another baseball study, *Behavior of College Baseball Players in a Virtual Batting Task*, a batting simulation was used to determine the information used to hit a ball. To conduct this research the “measures of spatial and temporal swing accuracy were used to test whether” experienced college baseball players “(a) use speed to estimate pitch height, (b) initiate a constant swing duration at a fixed time to contact, (c) are influenced by the history of previous pitches and pitch count, and (d) use rotation direction” (Gray, 2002, p. 1131). The results of the study showed that changing the speed of the pitch led to errors in the height of the batter’s swing, because batters use an estimate of pitch speed to predict the pitch height. Additionally, Gray’s (2002) findings indicated that batters control their swing based on what pitches have been thrown, pitch count knowledge, and the rotation of the ball after it leaves the pitcher’s hand (p. 1148). Ultimately this study showed that effective hitting results from a combination of perception and action.

In the same year researchers Kato and Fukuda (2002) analyzed “visual search strategies of baseball batters during the viewing period of the pitcher’s motion” (p. 380). In their study, *Visual Search Strategies of Baseball Batters: Eye Movements During the Preparatory Phase of Batting*, 18 participants (9 experts and 9 novices) measured and analyzed eye movements while watching a videotape of a pitcher throwing 10 pitches from the perspective of a right-handed batter. The results showed significant differences between the experts and novices. “Novices moved their eyes faster than experts, and the distribution area of viewing points was also wider than that of the experts” (Kate & Fukuda, 2002, p. 380). Additionally, the experts looked at the pitcher’s arm longer than
the novices during the final two pitching phases. Based off their findings, the researchers determined that experts focus their vision (set a visual pivot point) on the pitcher’s elbow while using their peripheral vision to judge the pitcher’s motion and ball trajectory.

Next, a group of researchers from Canada observed the performance effects of internal and external attention instructions on both highly skilled and low skilled golfers. Each group was made up of ten golfers, who were given a 9-iron that would be used to hit the golf ball to an orange pylon located at different distances (10, 15, 20, or 25 m) away.

Under internal focus of attention instructions, the participants were told to concentrate on the form of the golf swing and to adjust the force of their swing depending on the distance of the shot. For the external focus of attention conditions, the participants were told to concentrate on hitting the ball as close to the target pylon as possible. (Perkins-Ceccato, Passmore, & Lee, 2003, p. 593).

The result of the study, *Effects of Focus of Attention Depend on Golfers’ Skill*, showed that the highly skilled golfers performed better while focusing their attention on the external attention instructions. On the other hand, the low skilled golfers performed better while focusing their attention on the internal instructions. These findings showed a relationship between skill and focus of attention instructions when it comes to the variability in performance.

After Rob Gray’s *Behavior of College Baseball Players in a Virtual Batting Task* research in 2002 researchers, Gray, Castaneda, and Brooke took a closer look at a batter’s attention in baseball. The purpose of this study was to determine on what baseball players should focus their attention while batting. Using high-skilled and low-skilled baseball players, they tested four dual-task conditions. The first set of variables dealt with internal skill and external skill execution (hand movement and bat movement). The second set of
variables examined were irrelevant environmental and external environmental (auditory tones and the ball leaving the bat). Their findings showed that highly skilled players had better batting performance in the environmental/external condition and performed worse in the skill/internal condition. On the other hand, the less skilled players performed significantly better in the internal and external skill conditions than the two environmental conditions. Once Castaneda, Brooke, and Gray (2007) analyzed their results they concluded:

…the optimal focus of attention for highly skilled batters is one that does not disrupt proceduralized knowledge and permits attention to the perceptual effect of the action… [optimal focus of attention] for less skilled batters is one that allows attention to the step-by-step execution of the swing (p. 60).

In the following year, two researchers, Nakamoto and Mori, studied the Go/NoGo reaction time (RT) and its relevance to sport-specific decision-making. The studies participants were male college students (20 basketball players, 24 baseball players, and 13 non-athletes), who were tasked with reacting to a Simple RT task and a Go/NoGo RT task that involved baseball related stimulus. As one might expect, in both tasks the baseball and basketball players displayed quicker reactions than non-athletes. But there was significant variation in reaction times based on level of experience for baseball players but not for basketball players. Nakamoto and Mori (2008) stated that “These results suggested that Go/NoGo RT could be used as an index of expertise for sport-specific decision making, if stimulus-relation in Go/NoGo RT task has a natural relation for a particular domain” (p. 163).

Similar to a couple of the other baseball studies discussed in this section, this research paper divides its participants into two groups—skilled and non-skilled baseball
batters. The aim of the study was to observe the differences between the two groups in the visual search strategies during the pre-pitch phase or pitchers “windup”. Both groups watched a pitcher throw ten pitches and were tasked with pressing a button (attached to a bat) when the participant thought they should swing the bat to hit the ball. While performing this task the eye movements, accuracy, and swing timing were measured. “The Expert group shifted their point of observation from the proximal part of the body such as the head, chest, or trunk of the pitcher to the pitching arm and the release point before the pitcher released a ball, while the gaze point of the Non-expert group visually focused on the head and the face” (Takeuchi & Inomata, 2009, p. 971). Consequently, the skilled batters swung significantly earlier (timing) and were significantly more accurate than the non-skilled batters. As a result, the skilled batters performed better than the non-skilled batters by using visual search strategies to look for specific cues (the pitcher’s arm) that improved accuracy and quickened their decision making.

Lastly the researchers, Muhammad Rusdi Syamsuddin and Yong-Moo Kwon, from the Korea Institute of Technology studied how to simulate batting practice in the virtual world using actual baseball data. By utilizing several physics formulas the researchers created pitch-ball and hit ball trajectory generators in a virtual world (Syamsuddin & Kwon, 2011). Additionally, this paper discusses how real-world data can be used to simulate the trajectory of a baseball in virtual reality.

2.4 Chapter Summary

This chapter summarized three areas of interest in relation to the implementation and validation of an open source baseball engine. The first section of this chapter discussed sports system simulations and focuses on previous research performed in the domain of
forecasting sporting event outcomes. Secondly, the Poisson and Bayesian predictions model sections addressed the best approaches to accurately develop a baseball engine. This section was divided into three subsections: a brief introduction to Poisson and Bayesian modeling, Poisson probability distribution models, and Bayesian prediction models. Finally, the last section of the literature review focused on an independent line of inquiry to provide insight as to how best communicate the results of the baseball engine.

Consequently, there has been no prior research on an open source baseball engine. The goal of implementing and validating a new open source baseball engine, the aim of this study, is to produce realistic a baseball engine using hitter/pitchers models to predict the outcomes that parallel actual MLB games.
CHAPTER 3. METHODOLOGY

3.1 Research Question
To what extent does this project’s open source probabilistic baseball engine (POSBE) produce outputs using a hitter and pitcher model that parallel those observed in actual professional Major League Baseball games?

3.2 Hypothesis
The open source probabilistic baseball engine (POSBE) will produce outputs using a hitter and pitcher model that parallel those observed in actual professional Major League Baseball games.

3.3 Apparatus
This study used a C++ compiler, called QT 5.5.1, to simulate 19,000 (nineteen games simulated 1000 times) baseball games between the New York Yankees and the Boston Red Sox based off observed statistics from the 2015 season.

3.4 Testing Conditions
The baseball engine simulations were run in silico (performed on computer or via computer simulation). The simulations were run on the researcher’s PC desktop computer. The computer contains an AMD 8-core 4.0 GHz CPU (Central Processing Unit), 16 GB of memory, and a 500 GB Samsung solid state hard drive.
3.5 Testing Procedures

First the researcher wrote pseudocode to help identify all possible outcomes of each situation or scenario that could occur during a baseball game. Second, before programming the simulation, the researcher mapped out the hitter/pitcher interaction pseudocode and each of the possible outcomes. The purpose of writing the pseudocode was to avoid rework by organizing and structuring the code appropriately. Next, the pseudocode was converted into the correct syntax for C++, using the QT 5.5.1 compiler. After implementing and debugging the code, the researcher simulated the nineteen games between the New York Yankees and Boston Red Sox played during the 2015 season 1000 times—resulting in a total of 19,000 baseball game simulations. The sample size of 1000 was chosen because, as per the central limits theorem, it gives enough data to create a reliable distribution; more than 1000 simulations would not change the distribution.

Once the 19,000 simulations were completed, the actual data from the 2015 New York Yankees (NYY) and Boston Red Sox games (BOS) were used as the observed distribution that was then compared to the simulated distributions. The statistical analysis used to test the simulated data was the Kolmogorov-Smirnov test. This test groups data based on similar characteristics. The hitters and pitchers were both divided into two groups: Hitters 1-5 in the batting order, Hitters 6-9 in the batting order, Starting Pitchers, and Relief Pitchers. The program used to run the Kolmogorov-Smirnov test was R, a free statistical computing and graphics software package. After running the statistical analysis on the seven metrics for each of the four groups, the researcher was able to determine the results and conclusions described in Chapter 4 and Chapter 5 of this manuscript.
3.6 The Probabilistic Algorithm

Every player in Major League Baseball (MLB) has a unique skillset and play style. Consequently no two players are identical, resulting in each player having different probabilities for each possible outcome to an at-bat. First, the researcher generated an algorithm that produced the hitter probabilities for each outcome, which were then normalized based on the team’s performance against that specific opposing team. The algorithm below shows how the researcher derived the probability of a New York Yankees hitter successfully achieving a specific outcome versus a Boston Red Sox pitcher:

1. Algorithm:

\[
\frac{L_h + \frac{O_{TVO}}{A_{TVO}}}{2} = \frac{H_iO}{H_iA} = \frac{L_h}{L_p} = \frac{L_h + L_p}{2}
\]

\[
L_h = \text{likelihood of hitter achieving a specific outcome}
\]
\[
L_p = \text{likelihood of pitcher allowing a specific outcome}
\]
\[
O_{TVO} = \text{team total of a specific outcome versus opponent}
\]
\[
A_{TVO} = \text{total team attempts versus the opponent}
\]
\[
H_iO = \text{hitter total of a specific outcome versus opponent}
\]
\[
H_iA = \text{total hitter attempts versus the opponent}
\]
\[
P_jO = \text{pitcher total of a specific outcome allowed versus opponent}
\]
\[
P_jA = \text{total pitcher attempts versus the opponent}
\]

\[
L_p \text{ was not normalized against the opposing team’s hitters because the pitcher is not dependent on the batter. The hitter is forced to react to the pitch whether it be the type, quality, speed, or location of the pitch. The pitcher makes these decisions independently of the hitter.}
\]
After finding $L_h$, the researcher solved for $L_p$. Lastly the average of $L_h$ and $L_p$ was taken to give the final probability for the hitter/pitcher interaction. This result indicated the probability that a hitter achieves a specified outcome versus the opposing pitcher. The generalized algorithm shown above was applied to eighteen hitters and twenty-seven pitchers on the New York Yankees and Boston Red Sox. The algorithm calculated the probabilities of the seven metrics under evaluation in this study, which the researcher designates as follows:

$ABO_{single} = \text{the at-bat outcome of a hitter getting a hit and earning one base}$

$ABO_{double} = \text{the at-bat outcome of a hitter getting a hit and earning two bases}$

$ABO_{triple} = \text{the at-bat outcome of a hitter getting a hit and earning three bases}$

$ABO_{homerun} = \text{the at-bat outcome of a hitter getting a hit and earning one run for himself and a run for each additional runner on base}$

$ABO_{walk} = \text{the at-bat outcome of a hitter receiving four pitches that the umpires calls balls and earning one base}$

$ABO_{hit-by-pitch} = \text{the at-bat outcome of a hitter getting hit by a pitch from a pitcher and earning one base}$

$ABO_{strikeout} = \text{the at-bat outcome of a hitter accumulating three strikes during an at-bat and resulting in an out}$
3.7 **Data Sources**

The hitting and pitching statistics from the nineteen games played between the New York Yankees and Boston Red Sox during the 2015 season were retrieved from an online MLB statistical repository called, https://www.baseball-reference.com.

3.8 **Specific Measures for Success**

To validate the engine’s ability to accurately predict outcomes, the researcher evaluated each player’s individual statistics during the nineteen games. For hitters the researcher evaluated singles, doubles, triples, homeruns, walks, hit-by-pitch and strikeouts. Pitchers were measured based on singles-allowed, doubles-allowed, triples-allowed, homeruns-allowed, walks-allowed, hit-by-pitch, and strikeouts.

3.9 **Threats to Validity**

The four primary threats to validity were: bad statistical analysis, biased random number generation, faulty code, and not accounting for an important edge case of a MLB rule. Due to the nature of the study focusing on a realistic and accurate representation of an actual MLB baseball game, one of the most serious threats to validity was bad statistical analysis. To mitigate this threat, the researcher worked closely with the Purdue statistics department to determine the most effective statistical test to run on the data and to audit results as they were complied. Secondly, the core functionality of the engine relied on random number generation to determine the result of the at-bat (single, double, triple, home run, walk, hit-by-pitch, or strikeout). Therefore biased random number generation was a serious threat that was mitigated by testing the random number generator to verify it produced a discrete/flat distribution. The results of this test indicated that 99.7% of the time it generated a truly random number. Thirdly, faulty code was another threat to
consider. In order to prevent this problem, the researcher programmed with a collaborator that allowed for both programmers to audit one another’s work and to help ensure as few bugs as possible were present. Lastly, with baseball being a complex game with numerous rules and countless situational variables, missing an important baseball rule edge case was another threat to monitor and mitigate. The researcher has played and followed baseball since he was three years old and used his lifetime of expertise to ensure no important MLB rule edges case were left out of the simulation. Additionally the researcher’s primary collaborator has also followed baseball for a long time and was able to help account for those rare baseball scenarios that researcher might have missed.

3.10 Chapter Summary

This chapter focused the research question, hypothesis, and apparatus that the researcher used for this study. Additionally, the testing conditions, testing procedures, the probabilistic algorithm, and data sources were discussed and served as the framework for testing the simulation. Lastly, the researcher outlined the specific measures for success and the threats to validity for implementing and validating a new baseball simulation algorithm.
CHAPTER 4. RESULTS

Eighteen hitters and twenty-seven pitchers on the New York Yankees (NYY) and the Boston Red Sox (BOS) were modeled based on their performance during the nineteen head-to-head matchups played between the teams in 2015. An algorithm was created to produce hitter and pitcher results that parallel those observed in the actual nineteen games played in 2015. To test whether the simulated results were realistic and representative of the nineteen games, the researcher used the Kolmogorov-Smirnov test to evaluate the algorithm for the hitter and pitcher interaction.

After extensive statistical consultation, it was determined that the researcher’s Kolmogorov-Smirnov test would use a 95% confidence interval. As long as the simulated p-values were greater than 0.05, it was determined that the simulation’s algorithm would be realistically predicting each hitter/pitcher outcome for each of the four groups. The simulated values are displayed in the table below:

<table>
<thead>
<tr>
<th>Group</th>
<th>POSBE Simulated P-Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group 1</strong></td>
<td>Hitters 1-5</td>
</tr>
<tr>
<td></td>
<td>Singles</td>
</tr>
<tr>
<td></td>
<td>0.9417</td>
</tr>
<tr>
<td><strong>Group 2</strong></td>
<td>Hitters 6-9</td>
</tr>
<tr>
<td></td>
<td>Singles</td>
</tr>
<tr>
<td></td>
<td>0.9685</td>
</tr>
<tr>
<td><strong>Group 3</strong></td>
<td>Starting Pitchers</td>
</tr>
<tr>
<td></td>
<td>Singles</td>
</tr>
<tr>
<td></td>
<td>0.4652</td>
</tr>
<tr>
<td><strong>Group 4</strong></td>
<td>Relief Pitchers</td>
</tr>
<tr>
<td></td>
<td>Singles</td>
</tr>
<tr>
<td></td>
<td>0.0775</td>
</tr>
</tbody>
</table>

Table 4.1 POSBE Simulated P-Values

The benefit of splitting the data into multiple groups and then checking the distribution is that it gives a better representation of how the simulation is performing
Looking at the results on a player-by-player basis is problematic because a portion of the players might be returning reasonable results, but maybe some players could be slightly off. Thus by using groups the Kolmogorov-Smirnov test shows how the simulation is performing for each group rather than knowing that for some players it’s not working as accurately. All twenty-eight histograms can found be in Appendix C. Below is the histogram from the ‘Hitters 1-5’ group that displays the simulation accuracy for doubles.

![Histogram of obs.doubles](image)

Figure 4.2 ‘Hitters 1-5’ Doubles Histogram

The higher p-values indicate that observed distribution and simulated distribution are more closely related than the lower p-values. The lowest p-values belong to the ‘Relief Pitchers’ group, which makes intuitive sense. With the scope of the project delimited to nineteen games and the fact that relief pitchers typically only pitch one or
two innings per game, these factors result in a smaller sample size for this group. If the scope of this project were expanded to multiple, full 162 game seasons, the researcher expects that the ‘Relief Pitchers’ low p-values for singles, walks, and hit-by-pitch would be significantly higher as that group has more opportunities to pitch more innings. This expectation is supported by the ‘Starting Pitchers’ group’s data. The ‘Starting Pitchers’ group had greater sample sizes because starters typically pitch at least 6 innings per game—resulting in more opportunities to face hitters.
CHAPTER 5. CONCLUSIONS

The goal of this study was to produce hitter and pitcher outputs that parallel those observed in actual professional Major League Baseball games. To do this the researcher evaluated the nineteen games played between the New York Yankees (NYY) and Boston Red Sox (BOS) during the 2015 season. The results discussed in Chapter 4 indicate that the simulation successfully produced hitter and pitcher outputs that parallel those observed in the nineteen games between the NYY and BOS in 2015. POSBE is the only open source baseball engine available to the open source development community. The self-documented C++ code along with some supplementary documentation is posted on https://posbe.sourceforge.net/. The open source nature of the code will help ensure that it is outwardly verifiable and extensible. Additionally, the algorithm can be applied to any data set/any two teams and produce realistic hitter/pitcher outcomes.

Initially Bayes’ Theorem was to be used for the algorithms that predict the outcomes of each at-bat. It was to be implemented by using the following as variables: total number of hits divided by plate appearances and a specific hit outcome (for instance, homerun) divided the total number of hits. Although the math worked perfectly, this solution left out alternative batting outcomes like walks or hit-by-pitch. Not accounting for these outcomes prevented the simulation from producing realistic results. Upon further consideration, it was discovered that in order to use Bayes Theorem for the baseball
engine’s algorithm, it would have needed to be written on a pitch-by-pitch basis. Furthermore, Bayesian modeling applies to two possible outcomes in this study -- the batter either hits the ball or does not -- whereas when dealing with pitcher/hitter interaction there are more than two possible outcomes. Because the number of possible outcomes exceeded two possibilities, using Bayes Theorem was not plausible with an at-bat based structured algorithm.

Future researchers should consider implementing the Bayesian pitch-by-pitch algorithm. Had this been a doctoral dissertation, the timeline would have allowed the pursuit of the much larger scope of work this approach would entail. To implement a pitch-by-pitch method, the following would need to be considered: an in-depth analysis of each hitter’s skill dealing with each particular pitch, pitch location, specific hitter/pitcher matchups, weather conditions, situational hitting, etc. Comparing the results of this study and a separate Bayesian approach would be an interesting future research topic.

One final note: at the conception of this project, the ultimate goal was to create a realistic baseball game that highlighted the lives and records of players in the Negro League when American baseball was racially segregated. This project was called—Honoring Black Baseball (HBB). With its purpose being to create an entertaining managerial-style tablet-based baseball game that educated users on the Negro League player’s lives on and off the field. Though the work completed for this project was not included as part of this manuscript, some of the project concept art is included in Appendix B to provide context for this manuscript’s initial inspiration. It is hoped the
HBB project will continue in the future. Those interested in carrying this work forward should please contact the author.
LIST OF REFERENCES
LIST OF REFERENCES


APPENDICES
Appendix A Pseudocode

Pitch Function

IF the ball is in the strike zone & batter does NOT swing
THEN = (+1 strike) or (+1 out) or (+1 out & if outs = 3 then teams switch offense and defense)

IF the ball is in the strike zone & batter does swing and misses
THEN = (+1 strike) or (+1 out) or (+1 out & if outs = 3 then teams switch offense and defense)

IF the ball is NOT in the strike zone & batter does swing and misses
THEN = (+1 strike) or (+1 out) or (+1 out & if outs = 3 then teams switch offense and defense)

IF the ball is NOT in the strike zone & batter does NOT swing
THEN = (+1 ball) or (walk) or (walk & +1 run (if bases loaded))

Hit Function *(make sure to check for baserunners each time the ball is hit)*

IF the ball IS or is NOT in the strike zone and batter hits the ball
AND the ball is foul
THEN = (+1 strike) or (new pitch (if strikes already = 2)) or (+1 out (if ball is caught in the air))
AND the ball is fair & the hitter flies out or grounds out & the fielder does NOT make an ERROR
THEN = (+1 out) or (+1 out & if outs = 3 then teams switch offense and defense)
AND the ball is fair & the fielder DOES make an ERROR
AND no baserunners

IF the error is made by an infielder
THEN = baserunner on 1B

IF the error is made by an outfielder & the ball stays in front of him
THEN = baserunner on 1B

IF the error is made by an outfielder & the ball travels behind him
THEN = baserunner on 2B or 3B or Home (depending on distance away from him)

IF the error is made by an infielder or outfielder & outs = 2
THEN = baserunner on 2B & other baserunners advance 2 total bases

AND the ball is hit outside the fielders range
THEN = hitter reaches base safely & becomes a baserunner

IF the ball stays in the infield &/or in front of the outfielders
THEN = baserunner on 1B
IF the ball goes behind the outfielders
THEN = baserunner on 2B
IF the ball goes behind the outfielders & the outfield is playing in
THEN = baserunner on 3B
IF the ball goes (a specific distance) away from the outfielders & baserunner is “fast”
THEN = baserunner on 3B
IF the ball goes over the fence/wall
THEN = homerun

Baserunner Function
If the hitter reaches base safely
AND hitter hits a single
THEN = hitter is on 1B
IF no baserunners
THEN = no runs and no baserunner movement
IF baserunner on 1B
THEN = no runs and baserunner on 1B moves to 2B
IF baserunner on 2B
THEN = +1 run and no baserunner movement
IF baserunner on 3B
THEN = +1 run and no baserunner movement
IF baserunners on 1B & 2B
THEN = +1 run and baserunner moves from 1B to 2B
IF baserunners on 1B & 3B
THEN = +1 run and baserunner moves from 1B to 2B
IF baserunners on 1B & 2B & 3B
THEN = +2 runs and baserunner moves from 1B to 3B
AND hitter hits a double
THEN = hitter is on 2B
IF no baserunners
    THEN = no runs and no baserunner movement
IF baserunner on 1B
    THEN = no runs and baserunner on 1B moves to 3B
IF baserunner on 2B
    THEN = +1 run and no baserunner movement
IF baserunner on 3B
    THEN = +1 run and no baserunner movement
IF baserunners on 1B & 2B
    THEN = +1 run and baserunner moves from 1B to 3B
IF baserunners on 1B & 3B
    THEN = +1 run and baserunner moves from 1B to 3B
IF baserunners on 1B & 2B & 3B
    THEN = +2 runs and baserunner moves from 1B to 3B
AND hitter hits a triple
    THEN = hitter is on 3B
IF no baserunners
    THEN = no runs and no baserunner movement
IF baserunner on 1B
    THEN = +1 run and no baserunner movement
IF baserunner on 2B
    THEN = +1 run and no baserunner movement
IF baserunner on 3B
    THEN = +1 run and no baserunner movement
IF baserunners on 1B & 2B
    THEN = +2 run and no baserunner movement
IF baserunners on 1B & 3B
    THEN = +2 run and no baserunner movement
IF baserunners on 1B & 2B & 3B
    THEN = +3 runs and no baserunner movement
AND hitter hits a homerun
THEN = hitter doesn’t appear as a baserunner
IF no baserunners
    THEN = +1 run and no baserunner movement
IF baserunner on 1B
    THEN = +2 run and no baserunner movement
IF baserunner on 2B
    THEN = +2 run and no baserunner movement
IF baserunner on 3B
    THEN = +2 run and no baserunner movement
IF baserunners on 1B & 2B
    THEN = +3 run and no baserunner movement
IF baserunners on 1B & 3B
    THEN = +3 run and no baserunner movement
IF baserunners on 1B & 2B & 3B
    THEN = +4 runs and no baserunner movement
Appendix B Honoring Black Baseball (HBB) Concept Art

Figure B.1 Concept-Loading Screen

Figure B.2 Concept-Favorite Team Selection
Figure B.3 Concept-Main Menu

Figure B.4 Concept-Game Mode
Honoring Black Baseball

Settings

Favorite Team: Baltimore Black Sox
Difficulty: Normal
Weather: Overcast

Figure B.5 Concept-Settings

Figure B.6 Concept-Game View
Appendix C POSBE Histograms

Figure C.1 Hitters 1-5 Singles

Figure C.2 Hitters 1-5 Doubles
Figure C.3 Hitters 1-5 Triples

Figure C.4 Hitters 1-5 Homeruns
Figure C.5 Hitters 1-5 Walks

Figure C.6 Hitters 1-5 Hit-by-Pitch
Figure C.7 Hitters 1-5 Strikeouts

Figure C.8 Hitters 6-9 Singles
Figure C.9 Hitters 6-9 Doubles

Figure C.10 Hitters 6-9 Triples
Figure C.11 Hitters 6-9 Homeruns

Figure C.12 Hitters 6-9 Walks
Figure C.13 Hitters 6-9 Hit-by-Pitch

Figure C.14 Hitters 6-9 Strikeouts
Figure C.15 Starting Pitchers Singles

Figure C.16 Starting Pitchers Doubles
Figure C.17 Starting Pitchers Triples

Figure C.18 Starting Pitchers Homeruns
Figure C.19 Starting Pitchers Walks

Figure C.20 Starting Pitchers Hit-by-Pitch
Figure C.21 Starting Pitchers Strikeouts

Figure C.22 Relief Pitchers Singles
Figure C.23 Relief Pitchers Doubles

Figure C.24 Relief Pitchers Triples
Figure C.25 Relief Pitchers Homeruns

Figure C.26 Relief Pitchers Walks
Figure C.27 Relief Pitchers Hit-by-Pitch

Figure C.28 Relief Pitchers Strikeouts