Weather, Travel, and Baseball: Markov Chain Simulation of Major League Baseball Games

Brian Cropp[1]

[1]*9325 Slayter Union, Denison University, Granville OH 43203 Email: cropp_b1@denison.edu*

Abstract

There is a long history of Major League Baseball game prediction with the sport's history dating back over 100 years. Modern tools allow for advanced methods of data analysis to be applied to historical Major League Baseball data. This study focuses on proving that outside factors including weather, travel, and time combined with classic Major League Baseball statistical performance contribute to the outcome of a game. Using Markov Chain simulation, game outcome results were predicted. The results of this study improve accuracy with the addition of outside factors. A minimal amount of improvement in accuracy is reflected.

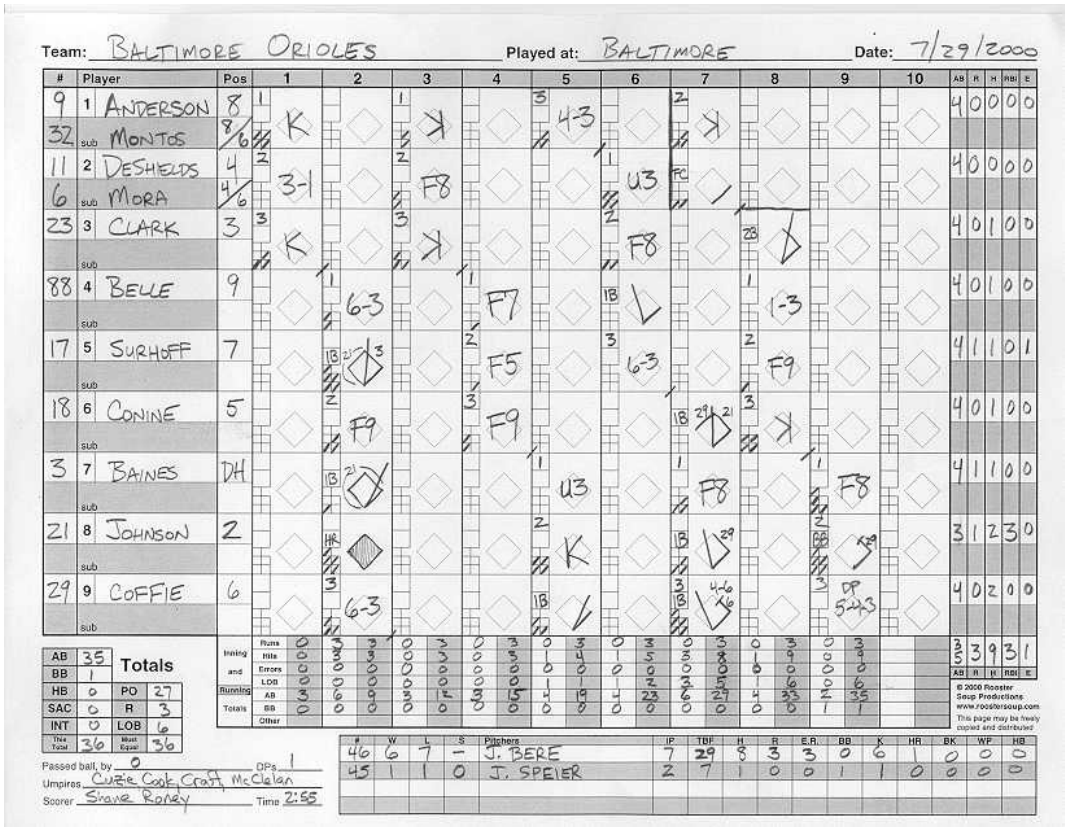
Introduction

Long before the Hollywood hit *Moneyball* popularized statistical analysis in Major League Baseball, Bill James coined the term 'sabermetrics' in 1980 to define "the search for objective knowledge about baseball"[1]. The history of the statistics in baseball truly dates back to as early as 1925 when F.C. Lane published his book Batting, where he developed theories and ideas on how mathematics can actually

[1] Phil Birnbaum. "A Guide to Sabermetric Research." SABR, Society for American Baseball Research.

tell us something about the game of baseball (Birnbaum). Access to data dates back so far because of the simplicity required to score a game. Figure 1 depicts a baseball scorecard, a manual pencil and paper method of keeping track of statistics and performance throughout a baseball game. The extended history of the sport, as well as the long-practiced collection of game and player statistics offers an interesting dataset for exploration. Not only can trends in the game be observed, but examination of changes and improvements to the sport can be hypothesized about, measured, and tested.

Figure 1. Baltimore Orioles vs Cleveland Indians July 29th, 2000



Team: BALTIMORE ORIOLES Played at: BALTIMORE Date: 7/29/2000

#		Player	Pos	1	2	3	4	5	6	7	8	9	10	AB	R	H	RBI	E
9	1	ANDERSON	8	1 K		1 K		3 4-3		2 K				4	0	0	0	0
32	sub	MONTOS	8/6															
11	2	DESHIELDS	4	2 3-1		2 F8			1 U3	FC				4	0	0	0	0
6	sub	MORA	4/6															
23	3	CLARK	3	3 K		3 K			2 F8		2B			4	0	1	0	0
	sub																	
88	4	BELLE	9		1 6-3		1 F7		1B		1 1-3			4	0	1	0	0
	sub																	
17	5	SURHOFF	7		1B 21 3		2 F5		3 6-3		2 F9			4	1	1	0	1
	sub																	
18	6	CONINE	5		2 F9		3 F9			1B 29 21	3 K			4	0	1	0	0
	sub																	
3	7	BAINES	DH		1B 21			1 U3		1 F8		1 F8		4	1	1	0	0
	sub																	
21	8	JOHNSON	2		HR			2 K		1B 29		2 BB 29		3	1	2	3	0
	sub																	
29	9	COFFIE	6		3 6-3			1B		3 1B 4-6 6		3 DP 5-4-3		4	0	2	0	0
	sub																	
														35	3	9	3	1
														AB	R	H	RBI	E

Inning and Running Totals		1	2		3		4		5		6		7		8		9		10	
	Runs	0	3	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3		
	Hits	0	3	3	0	3	0	3	1	4	1	5	3	8	1	9	0	9		
	Errors	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	LOB	0	0	0	0	0	0	0	1	1	1	2	3	5	1	6	0	6		
	AB	3	6	9	3	12	3	15	4	19	4	23	6	27	4	33	2	35		
	BB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1		
	Other																			

Totals

AB	35		
BB	1		
HB	0	PO	27
SAC	0	R	3
INT	0	LOB	6
This Total	36	Must Equal	36

Passed ball, by 0 DPs 1

Umpires Cuzie Cook, Craft, McClelan

Scorer Shane Roney Time 2:55

#	W	L	S	Pitchers	IP	TBF	H	R	E.R.	BB	K	HR	BK	WP	HB
46	6	7	–	J. BERE	7	29	8	3	3	0	6	1	0	0	0
45	1	1	0	J. SPEIER	2	7	1	0	0	1	1	0	0	0	0

© 2000 Rooster Soup Productions www.roostersoup.com This page may be freely copied and distributed

SOURCE: http://www.baseballscorecard.com/gallery.htm[2]

[2] Baseball Scorecard's website offers an extended history of scoring methods used historically. A full collection of historical Major League Baseball scorecards can be found on their site - http://www.baseballscorecard.com/gallery.htm

Because of the data available, Major League Baseball has always been a popular avenue of exploration for sports analytics. The game has been played with virtually the same rules for over 100 years. In its current form, there are 30 different MLB teams that play a total of 162 games a season. This means for any given starting player, hundreds of at-bats are recorded, and hundreds of pitches are thrown.

In looking closely at the Figure 1, the collection of events in the game happens on what is called a play-by-play basis. In Baseball, offensive events are separated by at-bats. On offense, each team sends a player to face the opposing team's pitcher. The pitcher from the team on defense will pitch the ball, hoping to achieve an out. The offensive player at-bat has the chance to swing and hit the ball into fair play. A 'play' in baseball is defined as an action that occurs resulting in a new baserunner, a movement of a baserunner, a run, or an out[3]. The abundant amount of data that exists over Major League Baseball's history has been collected at this play-by-play basis.

With the advancement of statistical modelling tools and an increased access to publicly available data, there is now a new opportunity to combine historical Major League Baseball play-by-play data with other outside factors. The data presented in Figure 1 does not capture certain variables that might impact the performance of both teams of offense. In an interest to advance the examination of

[3] A full breakdown of the rules of Major League Baseball can be found on their website - https://img.mlbstatic.com/mlb-images/image/upload/mlb/ttajxpedgemiy4sx0cer.pdf

historical baseball game predictions, this study will focus on variables pertaining to weather, and travel as factors not typically modeled in other projects.

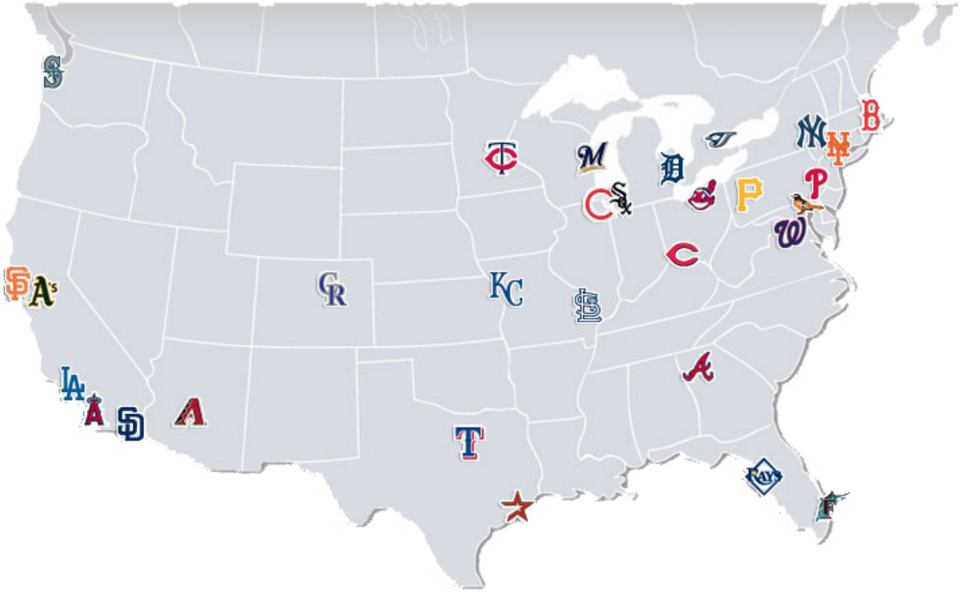
Weather variables such as temperature, vary from day-to-day and from stadium-to-stadium. Climate also varies from franchise to franchise. For example, the Seattle Mariners, located in the Pacific Northwest experience a radically different spring, summer and fall[4] for home game and team training compared to teams located in Florida or Texas. In 2015, the average temperature at Safeco Field in Seattle[5] was 67.9 degrees (Fahrenheit) compared to Globe Life Park in Arlington, TX[6] where the season average temperature was 84.4 degrees (Fahrenheit)[7]. In addition to temperature, stadium location can affect the wind speed measured on the field. High wind speeds can impact the trajectory of the baseball in flight. Tropicana Field[8] in Tampa, FL had an average of 0 wind (miles per hour) measured in the 2015 season. This can be compared to Oracle Park[9] where wind averages speeds of 11.8 mph.

Travel comes into question regarding the long distance's teams travel to play one another. Teams fly all over the US, changing time zones while staying on the road for long periods of a time[10]. League scheduling attempts to optimize team travel schedules in order to minimize travel time and miles flown. Figure 2 maps the

[4] The Major League Baseball season historically extends from late March to the end of September/Early October. Postseason is traditionally held through October. A full look at the 2019 schedule can be seen can be see here - https://www.baseball-reference.com/leagues/MLB-schedule.shtml
[5] Home of the Seattle Mariners franchise.
[6] Home of the Texas Rangers franchise.
[7] See Appendix A for full Table I. 2015 Average Season Temperature and Wind Speed
[8] Home of the Tampa Bay Rays franchise.
[9] Home of the San Francisco Giants franchise.
[10] The MLB provides an interactive map showing individual teams schedules, and the total number of miles they will travel per year - http://mlb.mlb.com/schedule/interactive-team-map.jsp

30 Major League Baseball franchise locations on a map of the United States. While optimization of schedule is attempted in order to equate travel, the teams located in the Midwest or Central East Coast travel much fewer miles than other coastal teams. In the 2019 season, the Seattle Mariners[11] and the Los Angeles Angels both travel over 44 thousand miles. Compare the Mariners and Angels to the Chicago Cubs who will only travel a total of 24,271 miles in the 2019 season[12].

Figure 2. Location of Major League Baseball Franchises



SOURCE: http://www.wordsmithingpantagruel.com/2011/09/[13]

[11] The Seattle Mariners played a series of games in the beginning of the 2019 season in Japan as part of potential Major League Baseball worldwide expansion. This distance impacts miles traveled greatly - https://www.mlb.com/mariners/tickets/japan-opening-series

[12] See Appendix B for Table 2. 2019 Total Miles Travelled by Teams

[13] Visual map of Major League Baseball Franchises referenced from blog - http://www.wordsmithingpantagruel.com/2011/09/

It is clear that factors pertaining to weather and travel differ from team to team. This study will focus on combining historical play-by-play Major League Baseball data combined with travel and weather data to improve prediction of game outcome. This study aims to predict Major League Baseball game outcome predictions by using these outside variables. Based on specifics of the game, this study will focus on only the 2015 season for initial analysis. Based on this goal, the testable hypothesis this study has centered its examination on is the following:

> *Outside factors including weather, and travel combined with classic Major League Baseball statistical performance contribute to the outcome of a game.*

Dependent Variable Selection for Measuring Game Outcome

In developing a model to predict the outcomes of MLB games, choosing the dependent variable for which the model to predict is an important. Outcome could be measured in different ways. The game outcome can be classified by a ‘W’ or ‘L’. Otherwise, the total number of runs could be forecasted.

A study performed on behalf of the American Statistical Association (ASA) used regression prediction methods to generate what they called ‘Estimated Run Production’[14]. With some success, their measure of outcome was compared between two teams to tell the winner of the game. Regression prediction was used in their study to predict runs with a degree of variance. The predicted number of

[14] Jay M. Bennett and John A. Flueck. “An Evaluation of Major League Baseball Offensive Performance Models.”

runs by each team was compared between Team A and Team B to pick the winner. The advantage to this method captured randomness in game outcome. With regression predictions and the standard errors of coefficients, simulation of game outcomes was able to estimate the outcome of games well with a degree of randomness.

A similar method of predicting game outcome was used in a study by researchers at Stanford. The study in question utilized Markov Chain simulation in order to predict a 1 or 0 zero binary variable to denote a win vs loss[15]. In analyzing results of their work, the game outcome as a product of comparing the teams predicted numbers of runs was more successful than run prediction. The method actually predicted the total number of runs, and determined an outcome 1 or 0 based on total runs.

This study follows a similar method to the game outcome variable selection as the mentioned research from ASA and Stanford. Observation of these studies indicate best practices in dependent variable selection. Both studies focus their predictive model building to estimate runs. Analysis of a model's predictive power can be measured in terms of accuracy to total number of runs, and also in terms of wins vs losses. This study will aim in predicting the final score (regression analysis) in order to game outcome (classification).

[15] Nico Cserepy and Robbie Ostrow. "Predicting the Final Score of Major League Baseball Games."

Independent Variable Selection for Inputs

For this project to stand out in the realm of sabermetric research, the input variables into the model is an opportunity for a unique dive in potential outcome forecasting. With the extended availability of MLB data, many projects have used a variety of input variables to develop prediction models. This study stands out from the rest in combining typical player and team statistics with other factors that affect play. The independent variables that set this study apart from others encapsulate travel and weather data.

Individual and team statistics have been used to generate predictions on MLB game outcomes in many studies prior to this one with high success. In observing previous work done in game outcome prediction however, studies that utilize other outside factors not measured by the typical individual and team statistics are most insightful. A thesis paper written at Minnesota State University looked at the losing streaks that teams would go on throughout the season[16]. Losing streaks helped to predict an amount of randomness in game outcome. Probabilities based on historical statistics can fail to capture streaks of possible failures (losses) or successes (wins). This study improved prediction by using predicting streaks in game outcomes. The study produced an edge in prediction performance based on establishing an amount of randomness. Streaks are a proponent of a teams performance based on their probabilities they perform at.

[16] Jordan Robertson Tait, "Building a Predictive Model for Baseball Games" (2014)

A study published at the University of South Florida-Sarasota looked at the biases created by weather and home field advantage in the National Football League[17]. The project, although not identical sports, used possible methods in consideration for this study that basically creates an input variable for players/teams based on a combination of stats, environment (weather and home field advantage) and randomness. The result is similar to how the popular blog FiveThirtyEight generates their MLB predictions. FiveThirtyEight uses what is coined an 'Elos' values in their MLB predictor methods[18]. This variable is a consideration of player/team stats along with outside factors that contribute to the game. Both studies show that it is possible in a model to combine different types of statistics to create some sort of profile of a player or team in certain situations.

Markov Chain Simulation

With developed independent variables in which to track and observe Major League Baseball games, the next step must be to select viable methods of prediction in which to best forecast the outcomes of games. Statistical programming in R[19] will allow for different methods going into our prediction model to be tested and examined to check for the best accuracy. With the help of previous

[17] Richard Borghesi. "The Home Team Weather Advantage and Biases in the NFL Betting Market."
[18] Jay Boice. "How Our MLB Predictions Work." fivethirtyeight.com/methodology/how-our-mlb-predictions- work/
[19] R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing

research into MLB win prediction, we can narrow down these methods down to one distinct approach; Markov Chain simulation.

A popular method in MLB game prediction is Markov Chain simulation. Major League Baseball historical play-by-play data offers the opportunity to simulate games at the at-bat level. With probabilities of potential certain events occurring, simulation of points being scored and can generate game outcomes.

The Markov Chain method of simulation is helpful in performing these in-depth predictions. Baseball is a perfect avenue for Markov Chain simulation because of the rules of the game. The process of Markov Chain simulation considers the potential 'states' that any given circumstance may result in[20]. Any given at-bat in an MLB game has a certain set of potential outcomes. In simple terms, a player at-bat with a runner on first base has a specific set of possible resulting outcomes for which could happen. The player could hit a single to right field, or a single to center field, and or the player might strikeout. All these possible outcomes have a certain probability of actually occurring. Therefore, certain of these possible outcomes are more likely to occur than others. In the process of Markov Chain simulation, a resulting outcome to this player's at-bat is generated. In this way of simulation, an entire game can be simulated a number of times, in a number of different ways.

[20] Mark D. Pankin "Baseball As A Markov Chain"

Markov Chain simulation has been a popular method of prediction in the world of MLB statistics because of the potential 'states' quality of the model's forecasting. The Markov Chain model generates the probabilities for such potential states based upon historical data[21]. With an extensive amount of historical data in the independent variables going into this model, the Markov Chain presents a great option for simulation in this study.

The method of Markov Chain simulation has been successfully used in studies of MLB win prediction in a varied fashion. Using player statistics from the first half of the season, researchers in the 1991 season accurately predicted the divisional winners of both the two MLB leagues[22]. Their work relied on the first 81 games as a training set for simulation (Barry). Using this training set, predictions of future games were predicted[22]. Similarly, the method of simulation was used in a study conducted at Stanford in which the results from the Markov Chain method were compared to Las Vegas betting lines[21]. The study resulted in very accurate predictions when it came to game outcomes and was slightly more accurate than the Las Vegas odds[22].

The potential success for accuracy using the Markov Chain method supports the use of the unique independent variables that this study will focuses on. The probabilities of possible outcomes in certain scenarios will be generated based on historical data. By using independent variables such as travel time, weather, time of

[21] Nico Cserepy and Robbie Ostrow. “Predicting the Final Score of Major League Baseball Games.”

[22] Daniel Barry, and J. A. Hartigan. “Choice Models for Predicting Divisional Winners in Major League Baseball.”.

game, temperature, and fan attendance, the Markov Chain could potentially build even more accurate probabilities.

Methods

To execute game predictions, this study followed a three-step process:

1. Data collection and aggregation
2. Setup of Markov states
3. Simulation of games

Data Collection and Aggregation

The initial step in the process of executing this study was to collect data pertaining to Major League Baseball, weather data specific to game time and location, and travel data unique to the visiting team. The history of Major League Baseball goes hand in hand with analytics and statistical analysis. The sport of baseball has been played professionally in America with virtually the same rules for over 100 years[23]. This fact has resulted in a full and accurate aggregation of Major League Baseball statistics dating back to the 1920's.

Major League Baseball Data Collection

The collection of Major League Baseball statistics for this study required play-by-play data for analysis and simulation. A baseball game is broken

[23] A full list of rule changes over the history of Major League Baseball - http://www.baseball-almanac.com/rulechng.shtml

down into 9 innings. Each inning, both the home and away team trade off playing offense and defense. The game starts with the away team on offense. Each team has three outs per inning to score points (in baseball known as runs). Play-by-play data is broken down further in order to collect individual at-bats.

The leading source of play-by-play Major League Baseball data is housed by an organization called Retrosheet[24]. Founded in 1989, Retrosheet aggregates play-by-play accounts of professional baseball games dating back to the sport's origin. Access to data hosted by Retrosheet is free and open to the public for use. While the data collected and host by Retrosheet is advanced, the method of accessing specific game accounts is complicated. Retrosheet does not provide data to users in the currently popularized application programming interface (API) method or through web-scrapping. Instead, the Retrosheet organization aggregates game data in their own file format and requires the use of their own software tools to extract the data requested.

In first collecting the play-by-play data for use in this study, understanding of Retrosheet's unique file formats and software was necessary. The Retrosheet organization aggregates all game account data in what they call

[24] Retrosheet - https://www.retrosheet.org/

‘event files’[25]. These event files are compressed records of Major League Baseball games. Extraction of data from these event files relies on software tools developed by the Retrosheet organization. Retrosheet provides four different software tools for four different applications data extraction. In recent years, data collection in Major League Baseball has developed to encapsulate ultra-specific variables pertaining to play-by-play action[26]. For the purposes of this study, only the BEVENT.EXE and the BGAME.EXE software was required for data collection.

The use of the BGAME.EXE software tool extracts information from the event file summarizing information that stays “constant for each game”[27]. Using the BGAME.EXE application, such variables such as date, time and location can be accessed. For this study, data pertaining to the date, time, location, weather, temperature, wind, and precipitation were all available for collection.

The complication the arose due to the use of Retrosheet as a data source arose with the application of these two software tools. Both the BGAME.EXE and BGAME.EXE files could only be used on Windows based computers.

[25] A sufficient description of Retrosheet event files can be found at this address - https://www.retrosheet.org/eventfile.htm

[26] Retrosheet provides play-by-play data down to the hit location of batted balls. A full breakdown of the 90+ potential hit locations and defnintions can be found here - https://www.retrosheet.org/location.htm

[27] Retrosheet’s account of the BGAME.EXE software tools use - https://www.retrosheet.org/datause.txt

These two applications used the Windows Command Line to call the pacific application, access the Retrosheet data, and then return the sufficient information. In order to capture the data that was needed in the range of years tested in this study, the Microsoft Windows Command Line functions were written in for each of file needed. The basic call of these applications was the following:

```
bevent -y 2015 2015MIN.EVA > 2015MIN_bevent.txt

bgame -y 2016 2016OAK.EVA > 2016OAK_bgame.txt
```

The process in accessing the data files from Retrosheet should follow the following steps:

1. Navigate to the Retrosheet organizations website where they host the event files - https://www.retrosheet.org/game.htm
2. Once on this webpage, the Regular Season Event Files header lists the events files by season. Retrosheet has you download by full seasons at a time. For this study, the event files for the years 2015 through 2018 were downloaded.
3. The download from Retrosheet is saved as a .zip file. The next set is to uncompressed this downloaded file.
4. Included in each full year of file is one event file for each team based on that specific year. Also included in the uncompressed file is a roster file (these files follow the extension .ROS). The BEVENT.EXE and BGAME.EXE files use these roster files to extract data. The files follow the following format in defining their team, and year:

   2017TEX.EVN
   *YEAR*TEAM*.EVN

5. Next, the sufficient software for data extraction must be downloaded. Navigate to https://www.retrosheet.org/tools.htm and download the BEVENT.EXE and BGAME.EXE. In order for these application to be used, place the .exe software files in the same folder as the .EVN file.
6. The final step in the data collection process from Retrosheet is to use the BEVENT.EXE and BGAME.EXE to extract the intended data for use. To do this, use the Microsoft Command Line and issue commands specific for the needed BEVENT and BGAME data.

The BEVENT.EXE processes event files and returns the play-by-play data specific to a team and year. The following variables are returned as a result of BEVENT.EXE process and will be used in the Markov Chain simulation[28]:

Table 3. Vars Returned by BEVENT.EXE used in Markov Simulation

game id*
visiting team*
inning*
batting team*
outs*
balls*
strikes*
pitch sequence
vis score*
home score*
res batter*
res batter hand*
res pitcher*
res pitcher hand*
first runner*
second runner*
third runner*

event text*
leadoff flag*
pinchhit flag*
defensive position*
lineup position*
event type*
flag*
ab flag*
hit value*
SH flag*
SF flag*
outs on play*
RBI on play*
wild pitch flag*
passed ball flag*
num errors*

batter dest* (5 if scores and unearned, 6 if team unearned)
runner on 1st dest* (5 if scores and unearned, 6 if team unearned)
runner on 2nd dest* (5 if scores and unearned, 6 if team unearned)
runner on 3rd dest* (5 if socres and uneanred, 6 if team unearned)

Next, using the BGAME.EXE application, data specific to games is collected. For the purposes of the simulation process, the following variables were needed to be collected[29]:

[28] A full list of returned variables as a result of the BEVENT.EXE function can be observed in Appendix C

[29] A full list of returned variables as a result of the BGAME.EXE function can be observed in Appendix D

Table 4. Vars Returned by BGAME.EXE used in Markov Simulation

game.id
date
day.of.week
start.time
day.night.flag
visiting.team
home.team
game.site
temperature
wind.direction
wind.speed
field.condition
precipitation
sky
time.of.game
number.of.innings
visitor.final.score
home.final.score
visitor.batter.1
visitor.batter.2
visitor.batter.3
visitor.batter.4
visitor.batter.5
visitor.batter.6
visitor.batter.7
visitor.batter.8
visitor.batter.9
home.batter.1
home.batter.2
home.batter.3
home.batter.4
home.batter.5
home.batter.6
home.batter.7
home.batter.8
home.batter.9

When downloaded, the BGAME and BEVENT data is separated by year and by team. With the process of extracting the BGAME and BEVENT data done for each team for the range of years, the next step is to aggregate each type of data by the year. Every team's BGAME file should be merged with one another so that all of the game-by-game data is aggregated for an entire season. The same should be done for the BEVENT data[30]. With full .csv files of play-by-play data and game-by-game data aggregated by year, the Markov Simulation Process can begin.

Weather Data

The data pertaining to the weather at each game was sourced from the returned BGAME game files from Retrosheet.com. The game files contained

[30] Using the *data-merging.ipynb* file can assist in this process

variables measuring the temperature, the wind speed, the wind direction, the precipitation, and the field condition. According to the BGAME file documentation, these variables are recorded at the beginning of each game. This means that this data is not exact for each play-by-play scenario, but influences factors contributing to performance.

Travel Data

Each teams travel statistics for modeling perposes was also sourced from Retrosheet.com. Travel data considers the amount of travel that a team is faced with while on the road. Teams travel from coast to coast, and away series' are often played in three game stretches[31]. Based on the BGAME game files, the home.team variable determined the team hosting the event. Then using the Retrosheet ballpark directory[32]. The Retrosheet ballpark codes provided locations of stadiums in which games were being played. Using this information, manual collection on longitude and latitude location information was collected using GPS Coordinates website[33]. This data was entered into the Latitude/Longitude Distance Calculator provided on the National Hurricane Center website[34]. The distance variable traveled prior to a game was applied to each play-by-play observation.

[31] 2019 MLB Regular Season Schedule: http://mlb.mlb.com/mlb/schedule/index.jsp?tcid=mm_mlb_schedule#date=05/06/2019
[32] Retrosheet ballpark codes: https://www.retrosheet.org/parkcode.txt
[33] GPS Coordinates: https://www.gps-coordinates.net/
[34] Latitude/Longitude Distance Calculator: https://www.nhc.noaa.gov/gccalc.shtml

Setup of Markov States

The data in it's current form achieves a relational database between the 'game id' variable from the BGAME data and the BEVENT data. With data collected and aggregated in a relational database, the next step in this study will be to create the Markov states for which the model will simulate at-bats. This process will start by creating probabilities for each player based on their historical statistics. Each Markov state in the decision process will simulate a possible outcome to an at-bat. Using the outside factors of travel time, travel distance and weather to generate scalers for the simulation of runs.

The way that Markov Chain simulation works is in calculating probabilities of moving from a 'state' to another 'state'[35]. In baseball, we can define states based upon the current situation a player is facing at-bat. For example, one out with a runner on second would define a potential state. Based on historical performance, the player at the plate has a certain probability in the potential states for which could come next.

For this process to work, the initial step in the Markov process was to define how states would be defined, and then calculate each before and after state for each play in the play-by-play data. For this study, states were defined by the number of base runners on base, their position, and the number of outs. Therefore a runner on first, with 2 outs would be encoded in state form as '2 100'. 2 denotes

[35] Setosa.io provides a visual interpretation of the change in Markov states probabilities that helps understanding - http://setosa.io/ev/markov-chains/

the number of outs in the inning, 1 denotes that there is a runner on first, and the next two 0's denotes that there is no one on second base or third base. The first step in this process was to create this state variable for each play prior to the action of said play[36].

Figure 3. Markov Chain Application to Major League Baseball Data

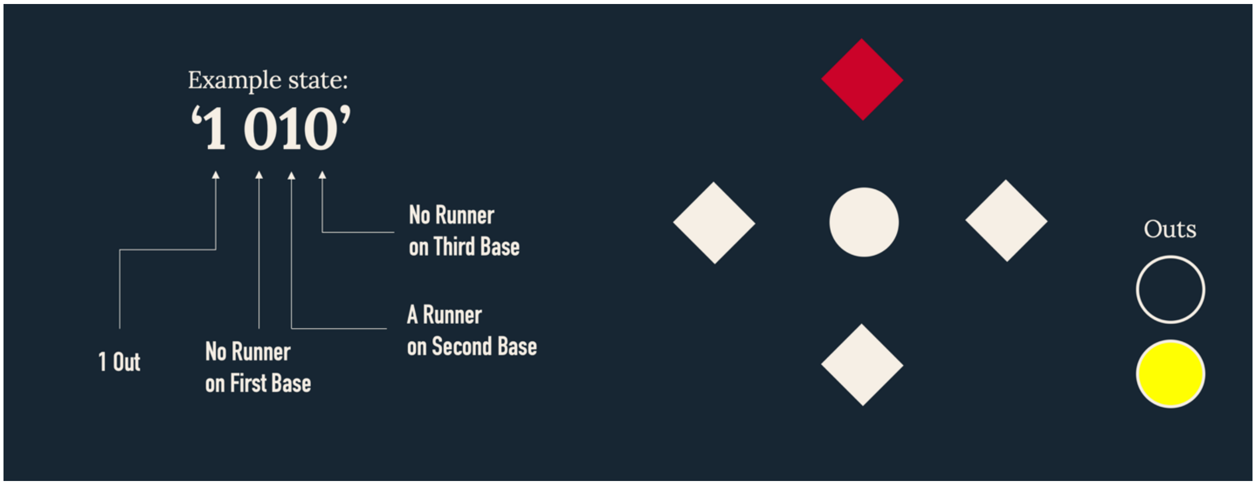


Figure 3 provides a graphical example of this process. An example scenario in which a runner is on second base with one out would be denoted in the Markov state at '1 010'.

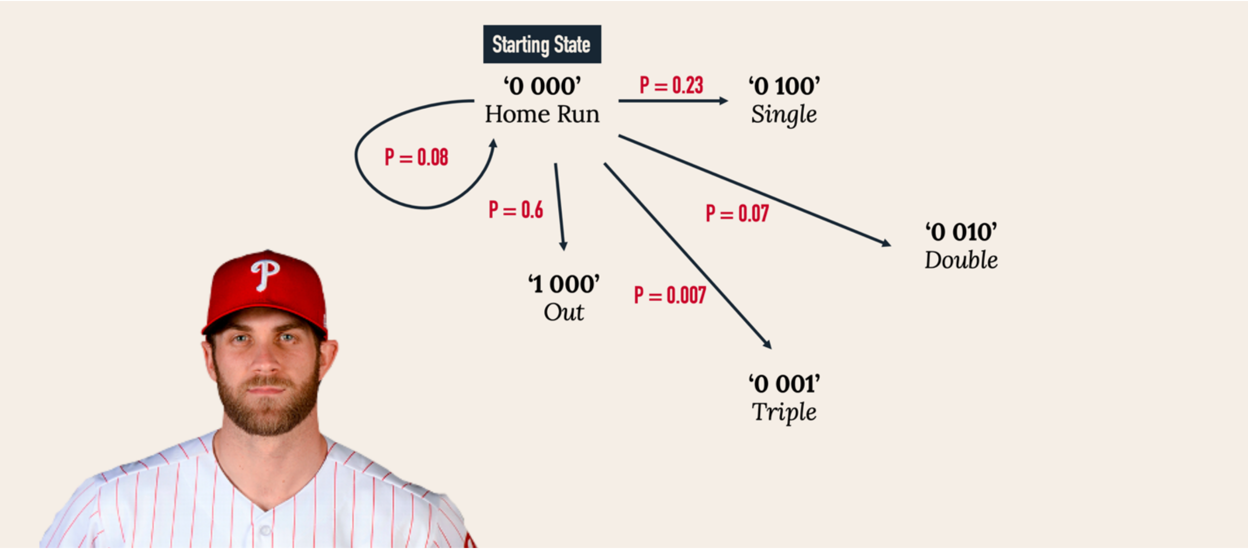
Now figure that the player at-bat gets a hit, and the runner on first advances to second while the hitter goes to first. This action would result in a state of (still) two outs, with runners on first and second base. The resulting state from the action of the previous play would be denoted as '2 110'. For each play again, the after state must also be calculated in addition to the before state.

[36] The states_full_df function in the mlb-markov-chains.R file processes a BEVENT file to create the before and after states.

Once each play in a season is defined by a potential state, probabilities can be calculated. When a given player is at-bat with a state of '2 110', historical data provides a probability for the next potential after state. Maybe the player has a large history of striking out, so it's likely that the state will go to '3 000'. Maybe the player at-bat has a high probability of hitting a single, so the state moves to '2 111'.

For any give player based on historical data, there will be 25 total number of possible states depending on where they are in the inning[37]. The above example is the method for which a possible state will be encoded for programing purposes. The practice of Markov chains relies on decision matrices of probabilities in order simulate a direct relation between two states. It is in this way that the 'On Base', 'Outs' will be updated. Based on the innings and score the game will automatically end after 9 innings when a team is ahead.

Figure 4. Bryce Harper 2015 State Probabilites



[37] Outs cannot be subtracted, so transferring from '2 000' to '1 000' is not possible. Because this is impossible to occur in a game, when probabilities are calculated, such state to state movement has a probability of 0.

Figure 4. depicts example probabilities for Bryce Harper in the 2015 season in a certain state. The starting state is ‘0 000’, meaning that there are zero runners on, and zero outs. In order to generate these probabilities, the model looked at the past performance of Bryce Harper to see how he performed in this exact scenario. Based on historical data pertaining to this situation, Harper has a 60% chance of getting out, a 0.7% chance of hitting a triple, a 7% chance of hitting a double, a 23% chance of hitting a single, and an 8% chance of homering. In the calculating these same probabilities for every player in a lineup, an entire game can be simulated.

Final Simulation + Weather + Travel

In order to run the entire simulation of a whole season, the final simulation relied on the observation a from the BGAME files for each season. Each observation in the data set provides a single game with the starting lineups. The lineups for players hitting one through nine are inputted into the ‘Lineup’ function[38]. This function. This function creates probabilities for each member in the offensive lineup based on historical data. These probabilities are then used in the Markov Chain process in order to simulated games through 9 innings.

Each game is simulated 100 different times. Similar to the study referenced from Stanford[39], the Markov Chain simulation predicts the total number of runs scored by each team. Using this information, a denoted value for 1 win or 0 loss is

[38] Refer to code https://github.com/briancropp/mlb-game-predictions/blob/master/markov.R

[39] Nico Cserepy and Robbie Ostrow. “Predicting the Final Score of Major League Baseball Games.”

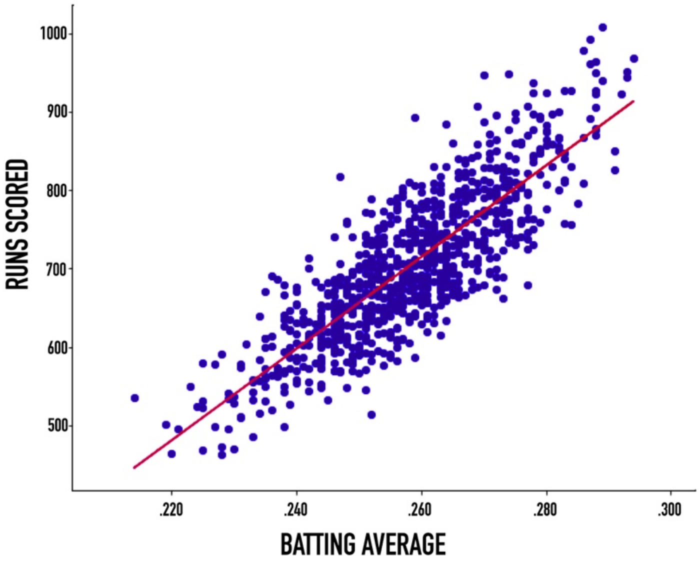
determined for a team. This value for win or loss is determined by which team has the most wins out of the 100 simulations.

In order to incorporate weather and travel statistics into the overall model, historical data was used to produce scaling factors for runs based on weather scenarios. For example, the MLB in 2015 averaged 21% less runs in the direct rain than in the sun. These scalers were imputed into the simulated run for each game.

Results

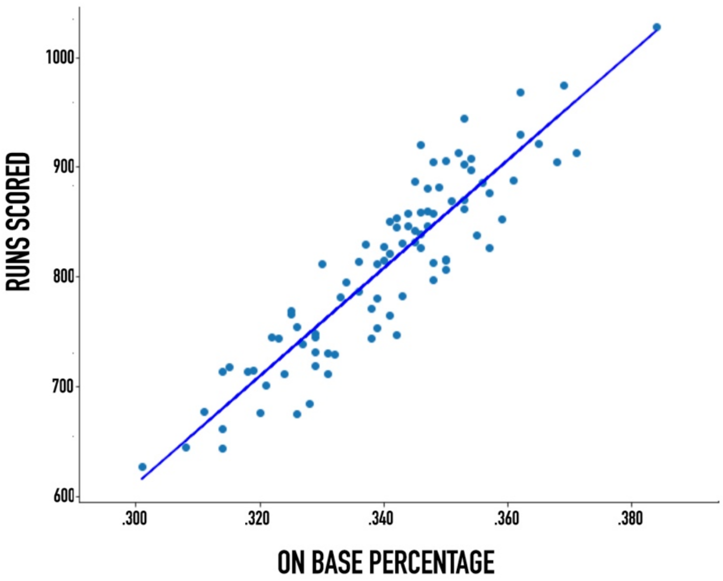
The testable hypothesis that this study intended to prove was *outside factors including weather, travel, and time combined with classic Major League Baseball statistical performance contribute to the outcome of a game.* Previous projects have focused specifically in predicting game outcomes using historical performance in order to predict game outcome or else using weather, travel, and time to predict overall team performance. This study will set itself apart from others by combining these two types of input variables in a Markov decision process.

Figure 5. Team Batting Average vs Runs Scored



From early exploratory analysis, interesting patterns cropped up in run outcomes. Initial analysis can show certain patterns that lead to a winning outcome of a game. In looking for these patterns, the variable 'Runs' provides a measure of score production. Looking at the relationship between Batting Average and Runs shows and obvious relationship. Figure 5. provides a graphical depiction of this relationship. Batting Average is the statistic in baseball that is the number of hits that a player produces divided by their number of at-bats. Figure 5. shows a clear linear relationship between these two statistics. From this exploratory analysis, we can tell that a higher Batting Average contributes to more Runs, and therefore a better chance of winning the game.

Figure 6. Team On Base Percentage vs Runs Scored

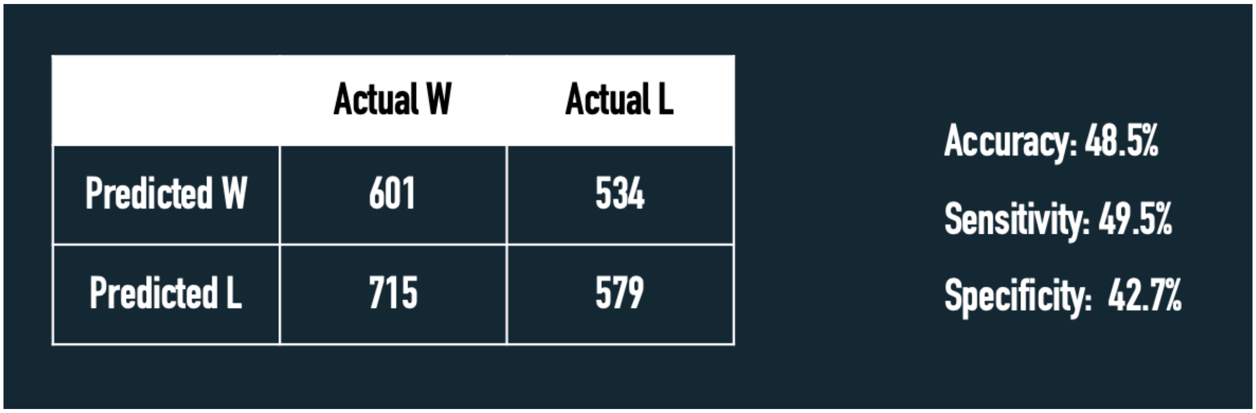


Continued exploratory analysis of Runs, looked to the relationship between Runs scored and On Base Percentage. On Base Percentage is a statistic measured in baseball that tells the percentage a player is to get on base. The calculation of the statistic is generated by the following (Hits + Walks + Hit by Pitch) / (At Bats + Walks + Hit by Pitch + Sacrifice Flies). Again, we see a linear relationship between On Base Percentage and Runs, similar to Figure 5, Figure 6 presents a similar positive linear relationship as Batting Average does. This tells that a higher On Base Percentage contributes to more Runs and therefore more wins.

Markov Results

The Markov Chain process was tested in the 2015 MLB season with the Weather and Travel Variables and without. The following results compare the two distinctions:

Figure 7. 2015 Markov Process Without Weather and Time Results



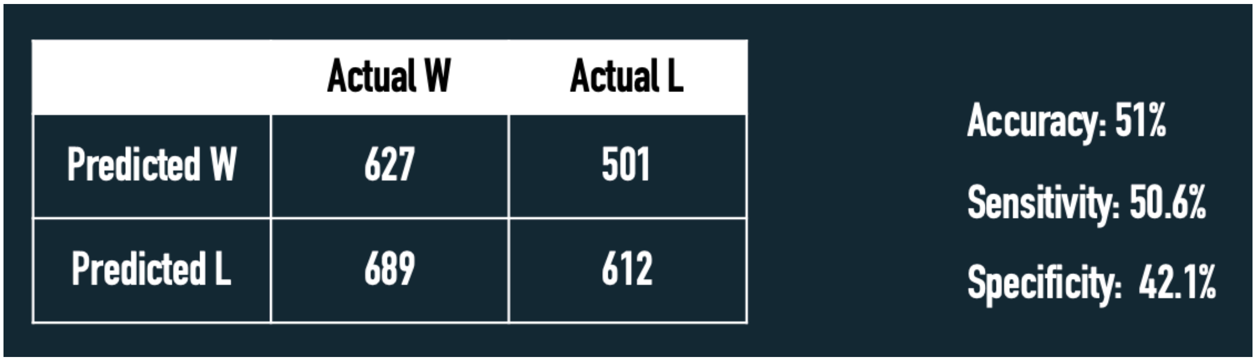
	Actual W	Actual L
Predicted W	601	534
Predicted L	715	579

Accuracy: 48.5%

Sensitivity: 49.5%

Specificity: 42.7%

Figure 8. 2015 Markov Process With Weather and Time Results



	Actual W	Actual L
Predicted W	627	501
Predicted L	689	612

Accuracy: 51%

Sensitivity: 50.6%

Specificity: 42.1%

Discussion

Understanding Results

Initial understanding of results tells that the model simulating runs for using Markov Chain simulation improves with the addition of weather and travel data in the new model. These results tell the testable hypothesis this study aimed to prove is actually correct as accuracy in predictions improved in this study.

It is important to take note that predictions did not improve a great deal from model to model. Figure 8 shows a slight increase overall in Accuracy, and

Sensitivity over the model without weather and Travel in Figure 7. This change is very minimal however.

Ethical Considerations

This study works with human data, which brings into question an IRB certification. The data sourced from Retrosheet Because the human data used is sourced from Major League Baseball, subjects are considered public figures. This fact excludes the possibility of potentially revealing personal private information of individuals. Therefore, the use of this data is exempt from IRB certification.

The player data used in this study is publicly available. The statistics pertaining to performance in Major League Baseball games and weather variables were sourced from game logs hosted on Retrosheet.com. This website has been the data source for a majority of Major League Baseball analytics studies. Retrosheet.com has data for every game dating back to the origin of stat collection in Major League Baseball. The site's proven track record and quality datasets validate the accuracy of the observations. Outside factors including travel time, distance are determined for each game sourced from publicly open APIs. Therefore, both the data pertaining to outside factors and the classic Major League Baseball statistics do not raise ethical concerns.

The intended use for this prediction model is solely for academic purposes. This model is not designed for gambling or any function outside of academia. This exempts the use of results of this study for any commercial purposes.

Results from this study should be considered in relation to the range of years observed. Certain elements of Major League Baseball have slightly changed in the 100+ year history of the game. It should therefore be taken into account that any prediction model that successfully predicts game outcome for current or previous seasons, is not guaranteed to work in following seasons.

Future Considerations

This study ended in positive results. Yet, these positive results only open the door to more questions that need answering. Addition of the weather and travel data on the overall model helped model predictions. Further research should focus on two avenues of new considerations.

First, simulation and predictions should be looked at a team by team basis. In creating this model, the impact of travel and weather data was on a Major League Baseball average basis. Based on data pertaining to the entire league, the effect of for example rain scaled the total runs for a specific team. In the future, simulation of these variables could be broken down to a team level.

Next, more seasons should be compared. The data resulting from the 2015 season has positive results. But what about the 2016 season, or the 2017 one. In order to fully takeaway findings from this study, more modeling and testing across more seasons should be considered.

Conclusion

This study was successful improving Major League Baseball game outcome predictions using added weather and travel data. However, these results only create more further questions about future sabermetric research in the future. This impact of this study tells that weather and travel factor into the game of baseball. As the future of sabermetrics moves forward, prediction models should consider these variables in further research in how they factor into the game.

References

Barry, Daniel, and J. A. Hartigan. “Choice Models for Predicting Divisional Winners in Major League Baseball.” Journal of the American Statistical Association, vol. 88, no. 423, 1993, pp. 766–774. JSTOR, www.jstor.org/stable/2290761.

Bennett, Jay M., and John A. Flueck. “An Evaluation of Major League Baseball Offensive Performance Models.” The American Statistician, vol. 37, no. 1, 1983, pp. 76–82. JSTOR, www.jstor.org/stable/2685850.

Birnbaum, Phil. “A Guide to Sabermetric Research.” SABR, Society for American Baseball Research, sabr.org/sabermetrics.

Boice, Jay. “How Our MLB Predictions Work.” FiveThirtyEight, FiveThirtyEight, 28 Mar. 2018, fivethirtyeight.com/methodology/how-our-mlb-predictions- work/.

Borghesi, Richard. “The Home Team Weather Advantage and Biases in the NFL Betting Market.” Journal of Economics and Business, vol. 59, no. 4, 2007, pp. 340–354., doi:10.1016/j.jeconbus.2006.09.001.

Cserepy, Nico and Robbie Ostrow. “Predicting the Final Score of Major League Baseball Games.” (2015).

Ghahramani, Z (2001) An Introduction to Hidden Markov Models and Bayesian Networks. International Journal of Pattern Recognition and Artificial Intelligence. 15 (1): 9-42

Jensen, Shane et al. "Hierarchical Bayesian Modeling Of Hitting Performance In Baseball". Www-Stat.Wharton.Upenn.Edu, 2019, http://www.stat.wharton.upenn.edu/~stjensen/papers/shanejensen.traj09.pdf. Accessed 19 Feb 2019.

Pankin, Mark D. "Baseball As A Markov Chain". Pankin.Com, 2019, http://www.pankin.com/markov/intro.htm. Accessed 19 Feb 2019.

Tait, Jordan Robertson, "Building a Predictive Model for Baseball Games" (2014). All Theses, Dissertations, and Other Capstone Projects. Paper 382.

Yang, Tae Young, and Tim Swartz. “A Two-Stage Bayesian Model for Predicting Winners in Major League Baseball.” Journal of Data Science, vol. 2, 2004, www.jds-online.com/file_download/39/JDS-142.pdf.

Appendix

A. Table I. Average Season Temperature and Wind Speed 2015

HOME TEAM	AVERAGE SEASON TEMPERATURE (FAHRENHEIT)	AVERAGE WIND SPEED (MPH)
ANA	75.2592593	6.30864198
ARI	81.6419753	3.4691358
ATL	79.2098765	7.90123457
BAL	75.962963	4
BOS	69.6419753	10.9382716
CHA	70.2222222	11.0617284
CHN	68.2592593	10.1728395
CIN	74.691358	7.60493827
CLE	69.825	10.2625
COL	72.0740741	6.50617284
DET	69.5555556	9.7037037
HOU	74.2962963	1.24691358
KCA	77.691358	8.5308642
LAN	74.2962963	6.04938272
MIA	75.8024691	0.7654321
MIL	73.5432099	4.50617284
MIN	73.5061728	10.4938272
NYA	73.8518519	9.43209877
NYN	74.4320988	11.5432099
OAK	65.1851852	11.1358025
PHI	75.3703704	9.60493827
PIT	73.9012346	7.96296296
SDN	72.5802469	8.54320988
SEA	67.9259259	2.40740741
SFN	64.7777778	11.7901235
SLN	80.2716049	7.12345679
TBA	72	0
TEX	84.8518519	9.2962963
TOR	68.7160494	6.18518519
WAS	77.7901235	5.75308642

B. Table 2. 2019 Total Miles Travelled by Teams

TEAM	MILES TRAVELLED
Mariners	47704
Angels	44945
Athletics	42119
Rangers	41128
Dodgers	40294
Giants	39341
Astros	38553
Padres	37363
Rays	36916
Red Sox	36896
D-backs	35312
Yankees	35252
Marlins	35226
Rockies	33287
Blue Jays	32895
Orioles	32322
Braves	29236
Royals	29077
Twins	28948
Phillies	28351
Mets	26832
White Sox	26538
Cardinals	26451
Pirates	26134
Brewers	25620
Tigers	25450
Indians	25176
Reds	25108
Nationals	24664
Cubs	24271

C. Table 3. Returned Variables on Play-By-Play basis of BEVENT.EXE

- game id*
- visiting team*
- inning*
- batting team*
- outs*
- balls*
- strikes*
- pitch sequence
- vis score*
- home score*
- batter
- batter hand
- res batter*
- res batter hand*
- pitcher
- pitcher hand
- res pitcher*
- res pitcher hand*
- catcher
- first base
- second base
- third base
- shortstop
- left field
- center field
- right field
- first runner*
- second runner*
- third runner*
- event text*
- leadoff flag*
- pinchhit flag*
- defensive position*
- lineup position*
- event type*
- batter event flag*
- ab flag*
- hit value*
- SH flag*
- SF flag*
- outs on play*
- double play flag
- triple play flag
- RBI on play*
- wild pitch flag*
- passed ball flag*
- fielded by
- batted ball type
- bunt flag
- foul flag
- hit location
- num errors*
- 1st error player
- 1st error type
- 2nd error player
- 2nd error type
- 3rd error player
- 3rd error type
- batter dest* (5 if scores and unearned, 6 if team unearned)
- runner on 1st dest* (5 if scores and unearned, 6 if team unearned)
- runner on 2nd dest* (5 if scores and unearned, 6 if team unearned)
- runner on 3rd dest* (5 if socres and uneanred, 6 if team unearned)
- play on batter
- play on runner on 1st
- play on runner on 2nd
- play on runner on 3rd
- SB for runner on 1st flag
- SB for runner on 2nd flag
- SB for runner on 3rd flag
- CS for runner on 1st flag
- CS for runner on 2nd flag
- CS for runner on 3rd flag
- PO for runner on 1st flag
- PO for runner on 2nd flag
- PO for runner on 3rd flag
- Responsible pitcher for runner on 1st
- Responsible pitcher for runner on 2nd
- Responsible pitcher for runner on 3rd
- New Game Flag
- End Game Flag
- Pinch-runner on 1st
- Pinch-runner on 2nd
- Pinch-runner on 3rd
- Runner removed for pinch-runner on 1st
- Runner removed for pinch-runner on 2nd
- Runner removed for pinch-runner on 3rd
- Batter removed for pinch-hitter
- Position of batter removed for pinch-hitter
- Fielder with First Putout (0 if none)
- Fielder with Second Putout (0 if none)
- Fielder with Third Putout (0 if none)
- Fielder with First Assist (0 if none)
- Fielder with Second Assist (0 if none)
- Fielder with Third Assist (0 if none)
- Fielder with Fourth Assist (0 if none)
- Fielder with Fifth Assist (0 if none)
- event num

D. Table 5. Returned Variables on Play-By-Play basis of BGAME.EXE

game.id
date
game.number
day.of.week
start.time
DH.used.flag
day.night.flag
visiting.team
home.team
game.site
vis.starting.pitcher
home.starting.pitcher
home.plate.umpire
first.base.umpire
second.base.umpire
third.base.umpire
left.field.umpire
right.field.umpire
attendance
PS.scorer
translator
inputter
input.time
edit.time
how.scored
pitches.entered
temperature
wind.direction
wind.speed
field.condition
precipitation
sky
time.of.game
number.of.innings
visitor.final.score
home.final.score
visitor.hits
home.hits
visitor.errors
home.errors
visitor.left.on.base
home.left.on.base
winning.pitcher
losing.pitcher
save.for
GW.RBI
visitor.batter.1
visitor.position.1
visitor.batter.2
visitor.position.2
visitor.batter.3
visitor.position.3
visitor.batter.4
visitor.position.4
visitor.batter.5
visitor.position.5
visitor.batter.6
visitor.position.6
visitor.batter.7
visitor.position.7
visitor.batter.8
visitor.position.8
visitor.batter.9
visitor.position.9
home.batter.1
home.position.1
home.batter.2
home.position.2
home.batter.3
home.position.3
home.batter.4
home.position.4
home.batter.5
home.position.5
home.batter.6
home.position.6
home.batter.7
home.position.7
home.batter.8
home.position.8
home.batter.9
home.position.9
visitor.finishing.pitcher
home.finishing.pitcher
name.of.official.scorer

Github

https://github.com/briancropp/mlb-game-predictions

Tools

Python 3.7.2 – Programming language for all web scrapping, building of Markov states, and Markov simulation

Pandas 0.24.1 – Python package for building data frames and aggregating data.

SQL:2016 – Domain specific language used to do the heavy lifting of relational database storage and access.