The Report Committee for Siddhartha Thakur
Certifies that this is the approved version of the following report:

Comparison of Prediction Methods for Batter-Pitcher Matchups

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: _____
James Eric Bickel

Co-Supervisor: _____
John Hasenbein

Comparison of Methods for Batter-Pitcher Matchups

by

Siddhartha Thakur, B.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2016

Acknowledgements

This work would not have been possible if not for the guidance and support of Dr. Eric Bickel, my advisor. I would like to thank Dr. John Hasenbein, for his encouragement and help. I would also like to thank Zachary Smith, for his invaluable knowledge and guidance and Andrew Beck, for his help and support.

Abstract

Comparison of Methods for Batter-Pitcher Matchups

Siddhartha Thakur, M.S.E.

The University of Texas at Austin, 2016

Supervisor: James Eric Bickel

Co-Supervisor: John Hasenbein

Baseball is full of confrontations and these confrontations between a batter and the pitcher is what makes the game. If a formula would be able to predict the probability of the outcome correctly, when they meet, wouldn't it instill confidence in the minds of the head coach (or you if you are playing the fantasy) to select someone who would be on the winning end? We would like to know for sure, which of our batters are good, and what out of the small amount of possible outcomes, will be the result when he faces this other good pitcher from the team you face next. It seems the past performance of the batter against this pitcher can be a good indicator, and that is what presumably the methods currently used utilize. But the utility of the Batter vs. Pitcher data in predicting the future outcome is a debate going on for quite a time now. The reason for this debate stems from the fact that the sample size of this data is so small that it becomes hard to comprehend when to prefer information you get from a sample size of thousands of at-bats against all pitchers vs. maybe a few dozen against specific individuals. The report will discuss one of the famous methods, called Log5 [1] that has been utilized so far when it comes to

measuring the outcomes of these confrontations. It also discusses the other methods like logistic regression based on the past data and the new and upcoming Morey-Z. [3]

Table of Contents

(i) List of Tables

(ii) List of Figures

1. Introduction

Baseball lends itself to statistics to a greater extent than any other sport. With discrete plays, it has a small number of possible outcomes and normally players act individually rather than performing in groups. Baseball is also the ultimate skill sport; the biggest, strongest and fastest guy does not always win. This makes statistics an important utility in the measure of their talent level. The journey of statistics in baseball was long. From when the first ever box score (appeared in New York Morning News, 1845) contained only runs and outs, to Sportvision coming through to develop technology that measures speeds and trajectories of pitched baseball (PITCHf/x®), the leaps and bounds have been magnificent. Despite this advancement, there are still some indicators or formulas that are a matter of dispute, which brings us to the topic of this report, Batter vs. Pitcher Matchups.

Baseball is full of confrontations and these confrontations between a batter and the pitcher is what makes the game. If a formula would be able to predict the outcome probability correctly, when they meet, wouldn't it instill confidence in the minds of the head coach (or you if you are playing the fantasy) to select someone who would be on the winning end? We would like to know, which of our batters are good, and what out of the small amount of possible outcomes, will be the result when he faces this other good pitcher from the team you face next. It seems the past performance of the batter against this pitcher can be a good indicator, and that is what presumably the methods currently used utilize. But the Batter vs. Pitcher data utility in predicting the future outcome is a

debate going on for quite a time now. The reason for this debate stems from the fact that the sample size of this data is so small that it becomes hard to comprehend when to prefer information you get from a sample size of thousands of at-bats against all pitchers vs. maybe a few dozen against specific individuals. The next section of the report will discuss one of the famous methods, called Log5 [1] that has been utilized so far when it comes to measuring the outcomes of these confrontations. It also discusses the other methods like logistic regression based on the past data and the new and upcoming Morey-Z.

2. Methods

The idea behind Batter and Pitcher matchups when predicting a particular outcome probability is to use the batter's statistic, the pitcher's statistics and the league average (to gauge where these players stand compared to an average player) for that outcome. The methods discussed here combine these statistics to predict the probabilities of the result of each plate appearance.

2.1. Log5

Bill James, in the 1981 Baseball Abstract (and 1983 Baseball Abstract), published a method to analyze how well one team plays against another. In his words, the Log5 method is a way of gauging the odds when two known forces collide. It can be figured out by following a logarithmic approach, where a team is assigned a Log5 score or a talent weight according to their won-lost percentage. This is the number which when added to 0.5 and divided by the sum, produces that team's won-lost percentage [1]. The win percentage of a team that has a Log5 of 0.333 is:

$$\frac{\mathbf{0.333}}{\mathbf{0.5 + 0.333}} = \mathbf{0.400}.$$

He calculates the probability of a 0.400 team winning against a 0.600 team by getting the Log5 score of both these teams and taking the ratio of their log values [1]. According to this, the probability of a 0.400 team winning this matchup is:

$$\frac{\mathbf{0.333}}{\mathbf{0.333 + 0.750}} = \mathbf{0.308}.$$

The Log5 estimate for the probability of team A defeating team B [1] can also be given by the following formula, without using the logarithmic step:

$$\left(\frac{\boldsymbol{Wins\ A\ \times\ Losses\ B}}{(\boldsymbol{Wins\ A\ \times\ Losses\ B}) + (\boldsymbol{Wins\ B\ \times\ Losses\ A})}\right).$$

When applied to Batter vs. Pitcher matchups the formula (modified by Dallas Adams as credited by Bill James [1]) includes the batter's average (BA), the pitcher's average against (PAA) and the league average (LA) to give us what that batter's average would be against this pitcher. The league average is included in the formula so as to free the formula from the assumption of the league average being fixed at 0.500. The formula to get the batter's average against that specific pitcher (E_m) is given as:

$$\boldsymbol{E_m} = \left(\frac{\frac{\boldsymbol{BA \times PAA}}{\boldsymbol{LA}}}{\frac{\boldsymbol{BA \times PAA}}{\boldsymbol{LA}} + \frac{(\boldsymbol{1-BA}) \times (\boldsymbol{1-PAA})}{(\boldsymbol{1-LA})}}\right).$$

A mathematical proof can be seen in Matt Haechrel's article published in Fall 2014 Baseball Research Journal where he also generalizes the formula to be useful for when there are more than two outcomes. [2] Similarly, the method can also be used to calculate the probability of other outcomes such as whether a batter gets struck out in a confrontation with a pitcher or he reaches base.

In addition to the above format, we can also write this formula more compactly in terms of the odds ratios, if B_o = BA/(1-BA), P_o = PAA/(1-PAA) and L_o = LA/(1-LA), the odds ratio of matchup ($E^*_m = E_m/(1-E_m)$) is:

$$\boldsymbol{E^*_m} = \frac{\boldsymbol{B_o \times P_o}}{\boldsymbol{L_o}}.$$

The odds ratio version of Log5 can further be written in this form:

$$\boldsymbol{ln(E_m^*) = ln(B_o) + ln(P_o) - ln(L_o).}$$

2.2. Logistic Regression

Logistic regression is one of the regression models where the dependent variable is categorical. Developed by David Cox in 1958, the model is used to estimate the probability of a binary response based on one or more predictor variables. The estimation is made using a logistic function, which is a cumulative logistic distribution. The binary logistic regression finds the best fitting model between the dichotomous (containing data coded as 1 for success and 0 for failure) dependent variable, with a set of independent variables $(X_1, X_2 \ldots X_k)$ generating the coefficients of a formula to predict a *logit* transformation of the probability of success. The *logit* transformation is defined as the logged odds of the probability of success (p):

$$\boldsymbol{logit(p) = ln\left(\frac{p}{1-p}\right) = b_o + b_1X_1 + b_1X_2 + \cdots + b_kX_k.}$$

In the prediction of the outcome of the confrontation between a batter and a pitcher, this model can be used to estimate the probabilities using the baseball statistics for the players and the league. In fact if we look at the odds ratio version of the Log5 formula, we can say that it is a generalized logistic regression model with $\boldsymbol{b_o = -(ln\ L_o)}$, $\boldsymbol{b_1 = 1,\ X_1 = ln\ (B_o)}$, $\boldsymbol{b_2 = 1,\ X_2 = ln\ (P_o)}$, where "p" gives the batting average of the resultant matchup.

Now, as the conventional baseball wisdom suggests, when a pitcher and a hitter both have the same handedness, the pitcher typically has the advantage in these

confrontations and hence handedness should have some say in predicting the outcome of the matchup. We can say the same thing for other statistics like ground ball and fly ball ratios. A logistic regression model could help us in considering any other significant statistics that would affect these outcome probabilities. The process section of this report discusses how we run a logistic regression model on the past three years of data and develop equations that would predict the probability for, let's say, a strikeout when a particular batter meets another pitcher.

A logistic regression model can also be built for multiclass dependent variables, where the dependent variable is not binary and can take more than two possible discrete outcomes. These are known as multinomial logistic regression models, and can be used to calculate the probabilities for a fixed number of outcomes in the sample space of a plate appearance.

2.3. Morey-Z

Morey and Cohen (2015) proposed a new formula [3] to estimate the outcomes of low probability events (in particular HR%) resulting from a specific batter/pitcher matchup, which is known as the Morey-Z formula. According to them, the need for this change was because Log5 formula has a bias in estimation that becomes more apparent as the underlying league averages depart from 0.500. They go on to construct Monte Carlo simulations, to explore "Hit" probability and the "Home run" probability in order to determine if Log5 predictions are skewed toward a 0.500 taking a wide variety of plausible matchup probabilities. [3]

The Morey-Z formula is given as (using the same notations as the Log5 formula)[3]:

$$\boldsymbol{E_m} = \left(\frac{\frac{\boldsymbol{BA-LA}}{\sqrt{\boldsymbol{LA(1-LA)}}} + \frac{\boldsymbol{PAA-LA}}{\sqrt{\boldsymbol{LA(1-LA)}}}}{\sqrt{\boldsymbol{2}}} \times \sqrt{\boldsymbol{LA(1-LA)}} \right) + \boldsymbol{LA}.$$

The authors proposed this formula that characteristically estimates outcomes from the same inputs as the Log5 procedure, but according to the results given in the paper the formula yields estimates that demonstrate greater accuracy in estimating outcomes of low probability events in outlier matchups.

3. Dataset

The dataset required to get the baseball statistics for this project were gathered from the free Retrosheet files [4]. These Retrosheet's data files contain play-by-play records for all home games at each of the 30 home stadiums for every contest (regular as well as post-season) since 1921. With this at hand, just about any split and any statistical enquiry are possible. Each row in the events file is an "event" that has taken place during a ball game. Most of the time this is the conclusion of a plate appearance but it also includes stolen bases, wild pitches, balks, etc. Anytime there is an out or a change in base state, it deserves its own row. [4]

Retrosheet uses .EVA and .EVN format for its files and a special parser, CWEVENT [5] from the Chadwick software tools was used to convert them into .csv format. These .csv formats from year 2000 to 2015 were then taken and uploaded as MySQL database, a glimpse of what can be seen below.

Search: seq_events =

seq_events	GAME_ID	YEAR_ID	AWAY_TEAM_ID	INN_CT	BAT_HOME_ID	OUTS_CT	BALLS_CT	STRIKES_CT	PITCH_SEQ_TX
1	NYA200010100	2000	SEA	1	0	0	3	1	BBBCB
2	NYA200010100	2000	SEA	1	0	0	3	2	CBFBBS
3	NYA200010100	2000	SEA	1	0	2	3	2	BBBCFB
4	NYA200010100	2000	SEA	1	0	2	3	1	BBBCX
5	NYA200010100	2000	SEA	1	1	0	3	2	BCBBCFX
6	NYA200010100	2000	SEA	1	1	1	0	2	CFS
7	NYA200010100	2000	SEA	1	1	2	1	1	BCX
8	NYA200010100	2000	SEA	2	0	0	0	1	CX
9	NYA200010100	2000	SEA	2	0	1	3	2	SBBBFB

Figure 1: Screenshot of the Retrosheet play-by-play data in MySQL

Each row begins with a twelve character ID called the GAME_ID, which identifies the date, location, and the number of the game. For example, in the above

screenshot, the first column reads NYA200010100, which means the home team is New York Yankees with the single game being played on 10th October 2000. There are other columns that can be seen in the above figure like, AWAY_TEAM_ID (identifies the away team), PITCH_SEQ_TX (identifies the pitches thrown till the plate appearance ends, details of which are shown in Appendix A1). The rest major column names that are used to write SQL commands are EVENT_CD (describes the event type that occurred in codes ranging from 2-23, the description can be seen in Appendix A2), BAT_ID (describes who is the batter on base and is the RetrosheetID of that player, Alex Rodriguez is rodra001), PIT_ID (RetrosheetID for the pitcher who is pitching), BAT_HAND_CD and PIT_HAND_CD (gives the handedness of batter and the pitcher) etc. [4] A SQL command snippet can be seen below to create a table "Stat567_Bat" containing the Batters' various baseball statistics for the season 2005-2007. This is further exported as a .csv file.

```
#----------------------Create tables for Batters' statistics for the season 2005,
2006, and 2007-------------------------------------------------------------------

CREATE TABLE Stat567_Bat
AS (SELECT bat_id
, COUNT(distinct game_id) as G
,SUM(IF(AB_FL = 'T',1,0))+SUM(IF(event_cd= 14,1,0))+SUM(IF(event_cd= 15,1,0))
+SUM(IF(event_cd= 16,1,0))+SUM(EVENT_TX like '%SF%')+SUM(EVENT_TX like '%SH%') as
PA

,(SUM(IF(event_cd= 20,1,0)))/(SUM(IF(AB_FL = 'T',1,0))+SUM(IF(event_cd= 14,1,0))
+SUM(IF(event_cd= 15,1,0))+SUM(IF(event_cd= 16,1,0))+SUM(EVENT_TX like '%SF%')
+SUM(EVENT_TX like '%SH%')) as 1BPerc
```

Figure 2: SQL Command snippet to create tables containing batter's statistics

In order to build a regression model predicting the outcome of a batter/pitcher matchup, the head to head dataset for years 2008 and 2009 was used from the

chadwickbureau retrosplits repository [6] which contained files from 1974 to 2015. These files contained outcomes of all the matchups that occurred during that year in both regular and postseason games, if any. As we can see below, the files mention the year, the batter's ID, pitcher's ID, the number of plate appearances between them and result of these appearances (if it was a hit, what kind of hit, if not a hit, was it a walk or a hit by pitch, and all such information.)

A	B	C	D	E	F	G	H	
YEAR	PHASE	RESP_BAT_ID	RESP_PIT_ID	B_PA	B_AB	B_H	B_TB	B_2B
2008	D	andeb003	bradc001	1	1	0	0	
2008	D	andeb003	howej003	2	2	0	0	
2008	D	andeb003	kazms001	3	2	0	0	
2008	D	andeg001	beckj002	3	2	0	0	
2008	D	andeg001	delcm001	1	1	0	0	
2008	D	andeg001	lesti001	7	7	2	2	

Figure 3: Retrosplits headtohead-2008 csv file

For the regression model, this data was modified in RStudio such that rather than showing the total plate appearances for a particular matchup, it has a separate row for each plate appearance with the outcomes distributed over these rows. In essence, the resultant datafile then gives us a binary dependent variable in terms of any outcome for each plate appearance that we can then match with the batter and pitcher statistics coming from the Retrosheet data using their RetrosheetID.

A	B	C	D	E	F	G	H	I
YEAR	PHASE	RESP_BAT_ID	RESP_PIT_ID	B_PA	B_AB	B_H	B_TB	B_2B
2008	D	andeb003	bradc001	1	1	0	0	
2008	D	andeb003	howej003	1	1	0	0	
2008	D	andeb003	howej003	1	1	0	0	
2008	D	andeb003	kazms001	1	1	0	0	
2008	D	andeg001	beckj002	1	1	0	0	
2008	D	andeg001	beckj002	1	0	0	0	
2008	D	andeg001	delcm001	1	1	0	0	

Figure 4: Head to head data 2008 after modifications to the plate appearance

The statistics from the Retrosheet dataset for the players as well as the head to head splits, were also verified based on their statistics at baseball-reference.com. [7] These data files are a useful source to look back and see how well these formulas perform.

4. Process

The first step during the process was to look at confrontations between a particular batter and pitcher over the last 15 years (2000-2015) with the help of the Retrosheet data. Python and one of its module xlwings interacted with the MySQL database through Excel and VBA macros. We then used the equations (Log5 and Morey-Z) to predict the outcome probability of the matchup and compared them to the actual frequency. For these purposes, each of the plate appearances (PA) were considered to be resulting into seven fixed outcomes- Single (1B), Double (2B), Triple (3B), Home Run (HR), Walk (BB), Hit-by-Pitch (HBP), or Out (O). The Out contains all those plate appearances, which resulted in either a Generic Out, a Strike out, a Sacrifice Fly, a Sacrifice Hit, Error or Fielder's Choice. For example, here we will see how Alex Rodriguez (RetrosheetID- rodra001) fared against Justin Verlander (RetrosheetID-verlj001) during 2000-2013. Their individual statistics were taken and the outcome percentages were calculated by dividing them with their respective plate appearance. The table below shows Alex Rodriguez's statistic:

	Alex Rodriguez						
PA	1B%	2B%	3B%	HR%	BB%	HBP%	O%
9433	0.1462	0.0420	0.0021	0.0584	0.1222	0.0170	0.6121

Table 1: Batting Statistics for Alex Rodriguez during 2000-2015

We also find the same statistic for Justin Verlander using the Retrosheet dataset.

	Justin Verlander						
BF	1B%	2B%	3B%	HR%	BB%	HBP%	O%
9160	0.1475	0.0425	0.0041	0.0218	0.0734	0.0080	0.7027

Table 2: Pitching Statistics for Justin Verlander during 2005-2015

After putting in these values along with the league values for MLB during 2005-2015, we get the probabilities for this matchup. Using the number of matchups between them that actually happened we find the expected amount of each of these outcomes. In this case the Alex Rodriguez has met Justin Verlander 45 times.

Statistics	Actual	Log5 Expected*	Morey-Z Expected
1B	5	6.36	6.46
2B	1	1.75	1.82
3B	0	0.08	0.11
HR	5	2.17	2.05
BB	6	4.87	4.66
HBP	1	0.68	0.62
GO	27	29.09	29.27

Table 3: Comparisons between Actual and Expected Runs Scored (* -- Log5 probabilities do not add up to one and hence they are normalized)

We can see that even though the expected numbers (other than the homerun numbers, maybe mainly because it is Alex Rodriguez) are in the same line as the actuals, the number of matchups between them (45) does not seem a very significant number of plate appearances to make any judgment. To increase these numbers of matchups and to judge if these two methods actually give an indication of the outcome of the confrontation, we decided to compare the cluster matchups. [Though if for a minute we

stop to look at these numbers, and say they come from significant matchups, Log5 seems to do a better job, MSE (Log5)=2.08, MSE (Morey-Z) = 2.58]

4.1. Handedness Matchups

The most basic way to compare clusters is to compare the handedness matchups, for example, Left Hand batter vs. Left Hand Pitcher, Left Hand batter vs. Right Hand Pitcher and so on. In the example below, we look into one of these types of matchups- Right-Handed Batters (RHB) vs. the Left-Handed Pitchers (LHP). The Retrosheet data was used to fill in statistics for all right-handed batters (against all pitchers) and all left handed pitchers (against all batters) in MLB from 2000-2015. The actual value is how these right-handed batters fared against these left-handed pitchers during the same years.

Thus for Right-Handed Batters:

	Right-Handed Batters						
PA	1B%	2B%	3B%	HR%	BB%	HBP%	O%
1737868	0.1548	0.0467	0.0040	0.0271	0.0769	0.0100	0.6806

Table 4: Batting Statistics for Right Handed Batters in MLB (2000-2015)

For the Left-Handed Pitchers:

	Right-Handed Batters						
BF	1B%	2B%	3B%	HR%	BB%	HBP%	O%
817458	0.1545	0.0470	0.0044	0.0264	0.0868	0.0085	0.6724

Table 5: Pitching Statistics for Left Handed Pitchers in MLB (2000-2015)

The total number of such matchups is 584577, which should be significant enough for us to compare the expected value with the actual values.

Statistics	Actual	Log5 Expected*	Morey-Z Expected
1B	90985	90473.22	90426.55
2B	29056	27622.25	27478.91
3B	2469	2134.73	2306.22
HR	16350	15612.92	15628.83
BB	52521	46448.75	47318.71
HBP	3947	5487.42	5448.24
GO	390909	396797.71	395969.54

Table 6: Comparisons between Actual and Expected Runs Scored (* -- Log5 probabilities do not add up to one and hence they are normalized)

We see that considering the significant plate appearance between the two matchups, the expected numbers are almost in line with the actual apart from Home Runs and Walks, which are somewhat far off. It should also be noted that the batting and the pitching statistics of the clusters are close to each other, and are almost near the league average of these statistics.

4.2. Plate Discipline Matchups

Nate Silver in his article on baseball-prospectus introduced the plate discipline quotient (PDQ), which is the geometric mean of a player's walk rate and his strikeout rate and the formula is given as: [8]

$$PDQ = \sqrt{BB\% \times K\%}\,.$$

This metric can be used to group batters and pitchers in terms of a 'Finesse' player, a 'Neutral' player or a 'Power' player. The finesse players are the players with a PDQ of 0.10 or less, the power players are the ones with PDQ of 0.14 or more and the neutrals are the rest. In baseball terms, typically a Power hitter will tend to go more for

homeruns and finesse hitters would manage higher averages (because of tendency of hitting more ground shots). We can see the matchup between a Power hitter and Neutral pitcher and calculate the expected outcomes with the Log5 and Morey-Z formulas.

	Power Batters						
PA	1B%	2B%	3B%	HR%	BB%	HBP%	O%
461803	0.1330	0.0477	0.0042	0.0382	0.1180	0.0099	0.6489

Table 7: Batting Statistics for Power Batters in MLB (2000-2015)

	Neutral Pitchers						
BF	1B%	2B%	3B%	HR%	BB%	HBP%	O%
1132228	0.1553	0.0468	0.0049	0.0263	0.0807	0.0088	0.6770

Table 8: Pitching Statistics for Neutral Pitchers in MLB (2000-2015)

Again, the total number of such matchups between Power hitters and Neutral Pitchers in MLB are 251330, which is large enough to be significant for us to compare the actual outcomes with the expected ones form Log5 and Morey-Z formulas.

Statistics	Actual	Log5 Expected*	Morey-Z Expected
1B	33794	33575.69	35115.08
2B	12025	12018.80	11923.42
3B	1069	1059.00	1103.74
HR	9546	9505.13	8697.01
BB	28798	28567.14	26580.32
HBP	2403	2426.89	2380.12
GO	163695	164177.35	165530.31

Table 9: Comparisons between Actual and Expected Runs Scored (* -- Log5 probabilities do not add up to one and hence they are normalized)

We can see that the values lie closer to the actual values in case of Log5 than the expected value from Morey-Z. The mean square error for Morey-Z is really high (mainly

because of the error in Singles and Walks, around 1537790) compared to the Mean squared error for Log5 (47999.6). Apart from these matchup comparisons, in the second part of process a regression model was built to predict the K% (strikeout %) based on the past three years of data, and was compared to Log5 and Morey-Z probabilities.

4.3. Regression Equation for K%

A binomial logistic regression model was built taking the batting and pitching statistics as independent explanatory variables and the strikeout column from the head to head dataset, which is the binary dependent variable. Only those statistics that turned out to be significant were kept as the explanatory variables. For this regression model, the head to head records were taken for the years 2008 and 2009 and were modified such that each row is one plate appearance with the binary variable telling me if there was strikeout or not. The statistics such as K%, BB%, OBP% etc. were calculated for both batters and pitchers using 2005, 2006 and 2007 as the base years for the 2008 matchups while 2006, 2007, 2008 were used as base years for the 2009 matchups. These were then divided randomly, with 70% of this entire dataset forming the training dataset and 30% as the test dataset. The training dataset were the instances the model was trained on, and this model then generates the prediction of matchups in the test dataset. The logistic regression model's code is attached in the appendix B1 and B2 and so are the significant variables as well as the coefficient of these variables. The script in R was written to also give the Log5 and Morey-Z predictions of these matchups, which after rounding up to 2 decimal places were exported as a csv file. Comparisons were made between the formulas and the

model predictions by dividing the matchups into certain predetermined buckets based on the range of K%.

Lower	Bucket	Upper
0	0-0.04	0.04
0.04	0.04-0.08	0.08
0.08	0.08-0.12	0.12
0.12	0.12-0.16	0.16
0.16	0.16-0.20	0.20
0.20	0.20-0.24	0.24
0.24	0.24-0.28	0.28
0.28	0.28-0.32	0.32
0.32	0.32-0.43	0.43

Table 10: Bucket formed for Grouping the predictions

Next, the predictions from logistic regression that were rounded up to two decimal places were grouped within these buckets and a comparison was made between the actual frequency and the average prediction. For example, let's take the bucket 0.04-0.08; the actual frequency is the number of actual strikeouts between matchups that were predicted to be in the bucket of 0.04-0.08. The average prediction is the average of all the predictions made regarding the K% that lie between 0.04-0.08. Similarly, the actual frequency and the average prediction were calculated for each bucket and as well as for the other two formulas. For the model/formula to work perfectly the graph drawn between actual frequency and the average prediction should be a straight line with a slope of 1 (i.e. they should be equal). The graph was plotted and the actual line and the preferred line were shown together as shown in the next section.

5. Results

The regression model to predict the K% of the matchups gives us Batter's K% odds ratio (bKOdds), Pitcher's K% odds ratio (pKOdds), Batter's Handedness advantage (bHandAdv1), and both batter's and pitcher's deviation from the league average (bKZScore and pKZScore). The bKOdds, pKOdds were calculated from the batter's K% and Pitcher's K% in the same way as the odds ratios. The bHandAdv1 was a binary variable which indicates if the batter was at some kind of advantage given both their and the pitcher's handedness. The deviation from the league average was calculated in terms of Z-score, where:

$$bKZScore = \frac{bKperc - \mu(league\ K\%)}{\sigma(league\ K\%\ std\ dev)}.$$

The reason to include this was to keep in consideration how far the batters and pitchers are from a league average player. As we can see from the regression model result below in figure 5, all these variables are significant.

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	0.35600251	0.5502532	0.64698	0.51764525	
bKOdds	-6.71176047	2.19704523	-3.0549	0.00225133	**
pKOdds	-3.96631659	1.6606126	-2.38847	0.01691888	*
bHandAdv1	0.0779418	0.02244063	3.47324	0.00051421	***
bKZscore	0.78094726	0.14774883	5.28564	1.2527E-07	***
pKZScore	0.51641211	0.12424539	4.15639	3.2332E-05	***

Figure 5: Significant variables and the coefficients for the regression model

Using this regression model, the Log5 formula and the Morey-Z formula we found the strikeout probability for a matchup in the test set, separately. Now using the

method specified (making the bucket and plotting the graphs as mentioned on page 26) we compared the graphs for each formula.

Using Logistic Regression:

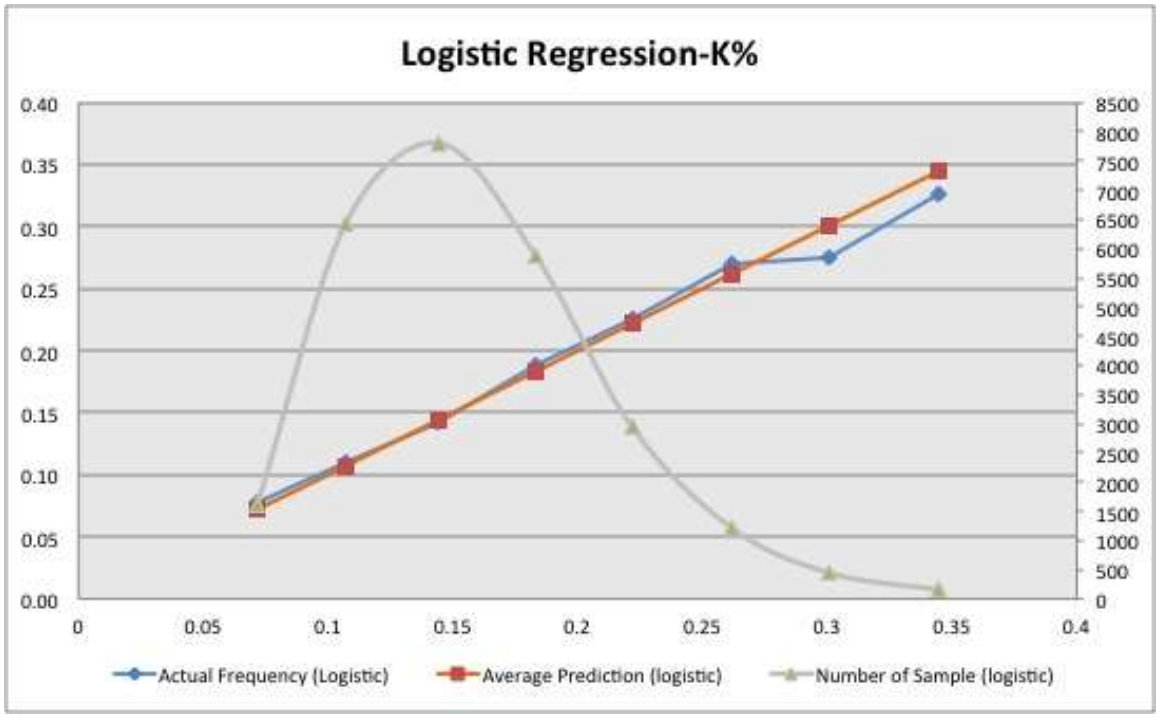

Figure 6: Logistic Regression predictions for K%

In figure 6, the gray line is the number of plate appearances in the bucket where the average prediction and the actual frequency is being compared. We can thus see that logistic regression works for almost all matchups. It makes sense as the logistic regression model looks for explanatory variables to get information regarding the dependent variable and is using last three years of data. It gets a little away from the actual frequency line in the outlier cases where the probability of outcome for the matchup being K% is unusually high. This might be because one of the players in

question is far off from what an average MLB player should be. The fact that such samples are small can also be one of the reasons.

An additional regression analysis was performed while not taking the pitcher's deviation from the league average into consideration. Thus in that case the significant variables are bKOdds, pKOdds and bHandAdv1. The regression model is given below along with the graph in figure 8.

Coefficients:

	Estimate	Std Error	Z value	Pr(>|z|)	
(Intercept)	-3.3175203	0.0505666	-65.60694	2.22E-16	***
bKOdds	4.86943455	0.16283896	29.90337	2.22E-16	***
pKOdds	2.90352874	0.14523866	19.99143	2.22E-16	***
bHandAdv1	0.0831959	0.02242204	3.71045	0.00020689	***

Figure 7: Coefficients of 2nd Logistic Regression on K%

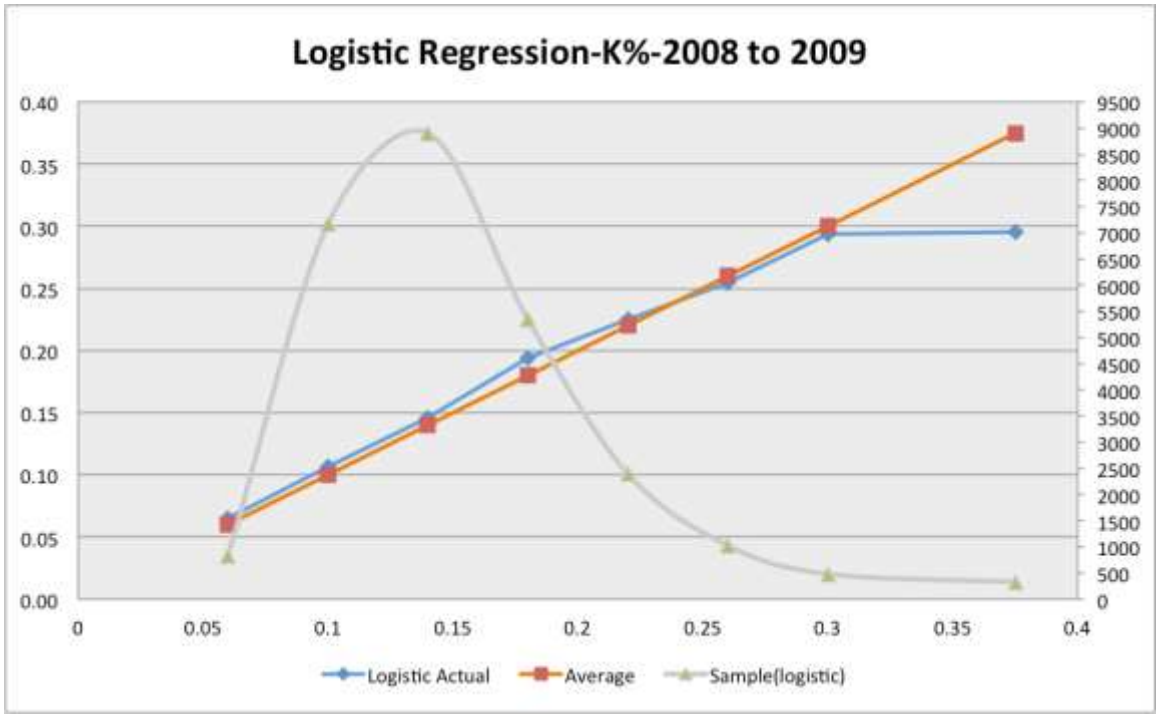

Figure 8: 2nd Logistic Regression for K%

The above graph (figure 8) looks more or less similar to the earlier one, just that it gets a little closer to the actual frequency line on the outliers but a slight away in the

places where it matters. Hence as correctness over a high sample of matchups is desired, we go with the first regression.

Similarly, we checked the predictions with the Log5 formula:

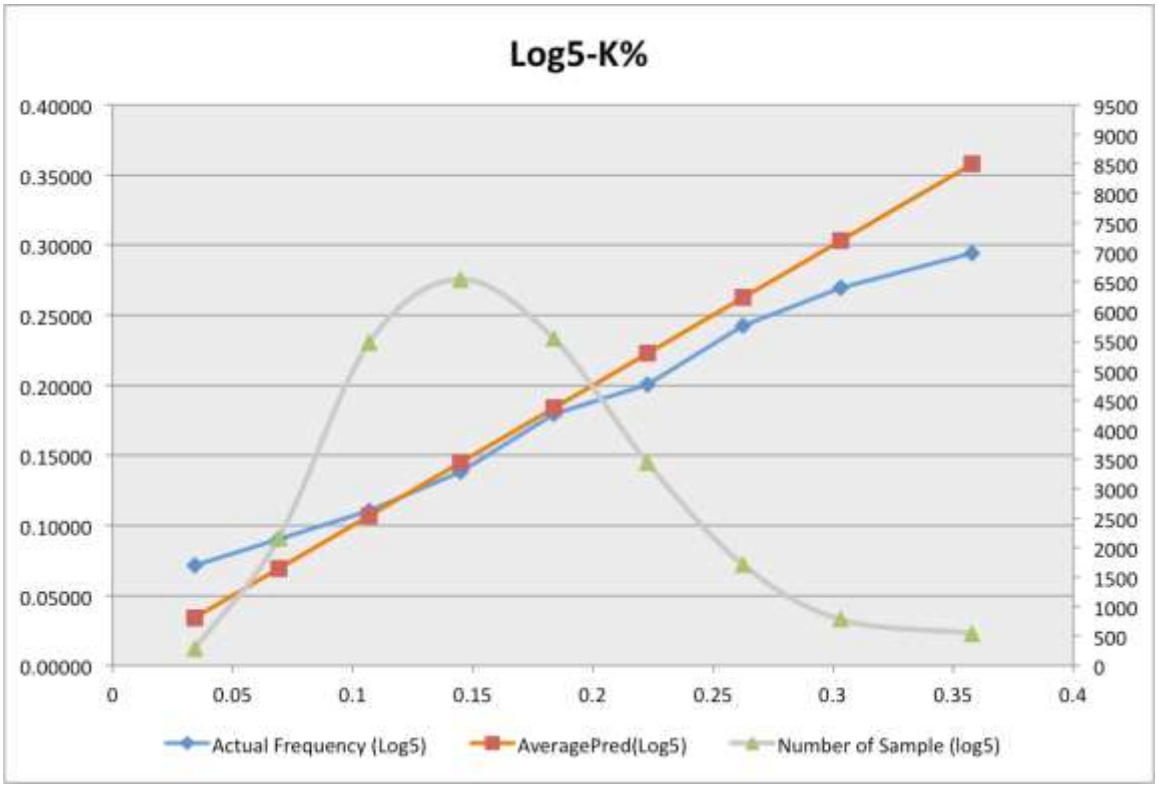

Figure 9: Log5 predictions for K%

Yet again we can see in figure 9 that Log5 is a good estimator where it matters the most; where the major sample of matchups is concentrated (near the average players and not outliers). It does tend to underestimate and overestimate when we move towards the outliers, but numbers of such matchups is smaller. Hence, even though logistic regression seems to be a better predictor of the outcome percentages of strikeout, Log5 is an easier formula to use straight with the statistics, without any kind of training or data modeling needed. The one downside to Log5 is that if we are using it in conditions where we need

to keep the outcomes of a sample space fixed, we will have to normalize it, as the outcome probabilities from the Log5 formula do not necessarily add up to one.

Going one step further, and constructing a similar graph for probabilities predicted through the Morey-Z formula, we get the following figure (figure 10):

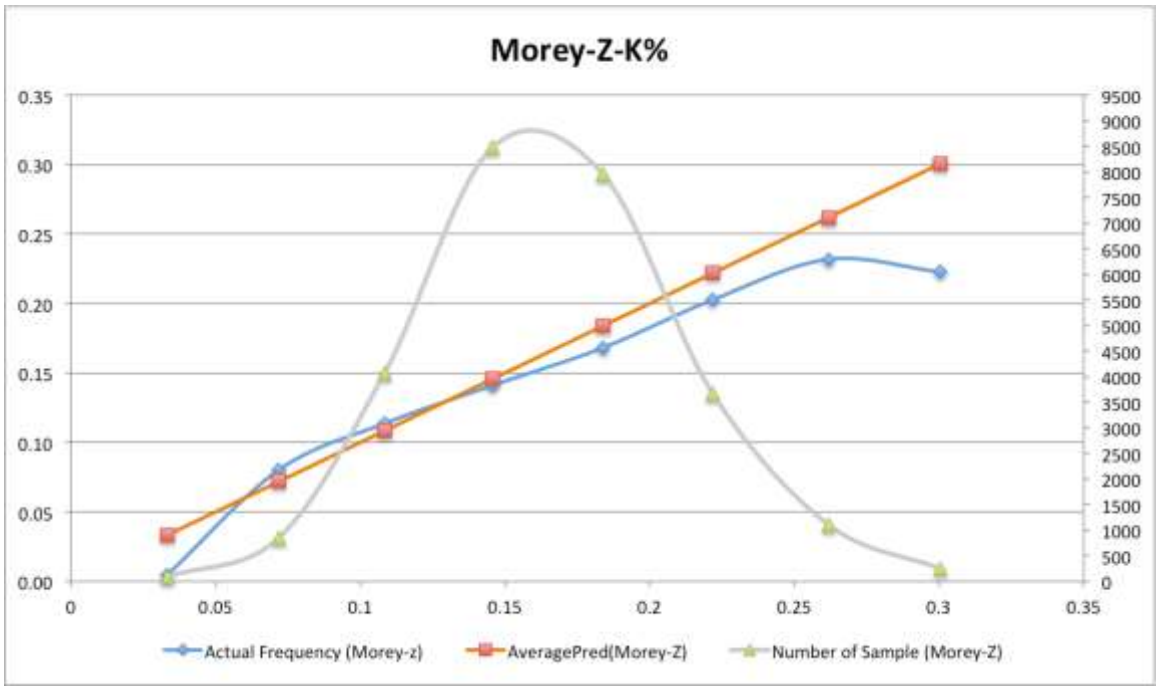

Figure 10: Morey-Z predictions for K%

The Morey-Z results look less reassuring than both Log5, though only slightly. This is because in the zone where the numbers of samples for matchups are high, the Morey-Z average prediction is less than the actual frequency, which was not the case for Log5 and logistic regression. This might also stem from the fact that the formula is made such that if a batter faces an average pitcher (the MLB average player), their matchup outcome statistics would be lower than the batter's.

5. Conclusion and Future Work

To conclude, if one has to use one of these formulas/methods to come up with the matchup outcome predictions, one should choose between logistic regression and Log5. Between both the above-mentioned formulas, Log5 seems to have the upper hand because of how easy it is to get an approximate idea of these outcome probabilities without even having to collect data, modify and build a model.

Regarding future research, one could be to use some weighted combination of these methods. In the example below, the probability of strikeout% was predicted for a matchup with the help of each method, the final probability was just the average of all three. The resultant graph between the actual frequency and the average predicted probability appears in figure 11:

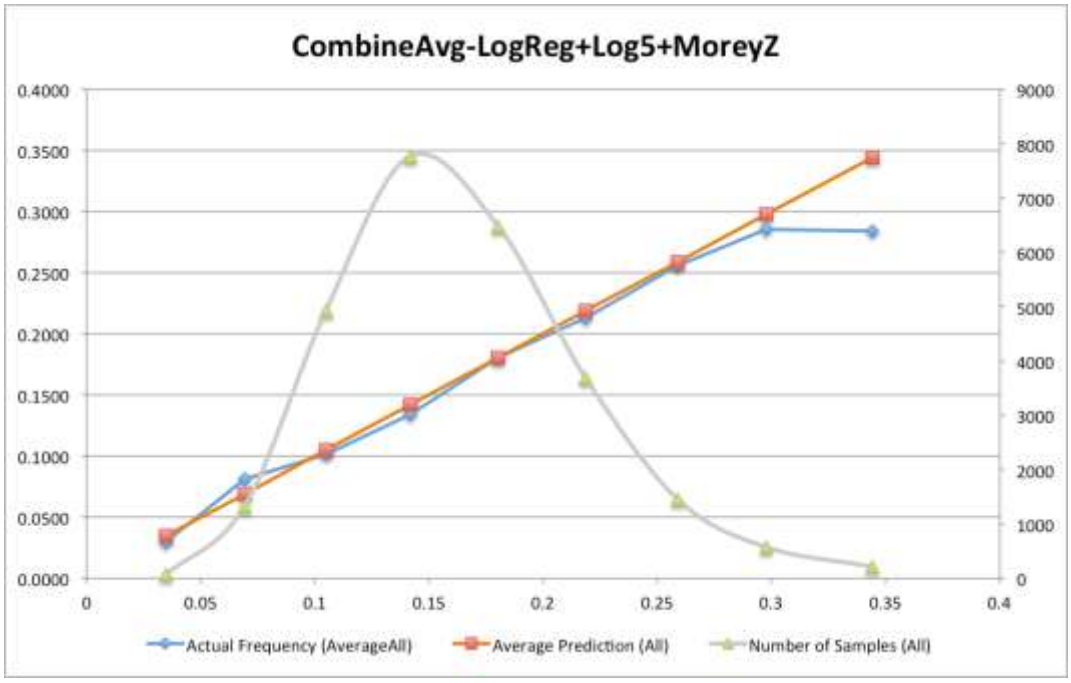

Figure 11: Combined Predictions for the K%

This is comparatively smoother over a large range of samples, and hence can be used as an outcome probability prediction measure. This is a simple average of all three but there can be weighted averages that would give more stress on Log5 and logistic regression and less stress on Morey-Z, and so on. Additionally, the logistic regression can also be solved while keeping outcome as a multinomial dependent variable where the outcome (rather than being binary-> strike out or not) would contain all the outcomes possible in a plate appearance. Work can be done to minimize the work it takes in building this model by using automated scripts that would take data from the past three years and generate the regression models at the end of every year, so that these equations could be used as easily as Log5.

Appendices

APPENDIX A1:

PITCH_SEQ_TX (describes the event that occurred pitch by pitch)

B	ball	O	foul tip on bunt
C	called strike	P	pitchout
F	foul	Q	swinging on pitchout
H	hit batter	R	foul ball on pitchout
I	intentional ball	S	swinging strike
K	strike (unknown type)	T	foul tip
L	foul bunt	U	unknown or missed pitch
M	missed bunt attempt	V	called ball because pitcher went to his mouth
N	no pitch (on balks and interference calls)	X	ball put into play by batter
		Y	ball put into play on pitchout

APPENDIX A2:

EVENT_CD (describes the event type that occurred in codes ranging from 2-23

2	Generic out	13	Foul error
3	Strikeout	14	Walk
4	Stolen base	15	Intentional walk
5	Defensive indifference	16	Hit by pitch
6	Caught stealing	17	Interference
7	Pickoff error	18	Error
8	Pickoff	19	Fielder's choice
9	Wild pitch	20	Single
10	Passed ball	21	Double
11	Balk	22	Triple
12	Other advance	23	Home run

APPENDIX B1:

Code for running a logistic regression model

```
#---------------Loading Data for Strike Out-----------------
library(ggplot2)
library(pROC)
library(caret)
data2<-read.csv("DataForModelFor Model.csv")
train<-data2
#--------Removing the #N/A-----------------
NAs<-train=="#N/A"
train[NAs]<-NA
trainfinal<-train[complete.cases(train),]
names(trainfinal)
str(trainfinal)
table(trainfinal$X2B)
#----Converting the data type of the variables--------------
indx <- sapply(trainfinal, is.factor)
trainfinal[indx] <- lapply(trainfinal[indx], function(x)
as.numeric(as.character(x)))
trainfinal$bHand<- as.factor(as.character(trainfinal$bHand))
trainfinal$pHand<- as.factor(as.character(trainfinal$pHand))
trainfinal$bHandAdv<-
as.factor(as.character(trainfinal$bHandAdv))
indx <- sapply(trainfinal, is.integer)
trainfinal[indx] <- lapply(trainfinal[indx], function(x)
as.factor(as.character(x)))
trainfinal$YEAR<- as.integer(as.character(trainfinal$YEAR))
#trainfinal$GbPit<- as.factor(trainfinal$GbPit)


#-----Splitting into Training and Test dataset--------------
set.seed(10)
trainfinal <- trainfinal[sample(nrow(trainfinal)),]
sam<-sample.split(trainfinal$YEAR,SplitRatio=0.70)
t.trainBball<-subset(trainfinal,sam==TRUE)
t.testBball<-subset(trainfinal,sam==FALSE)
n <- names(t.trainBball)


#--------------Model for Strikeout%-------------------------
```

```
ctrl <- trainControl(method = "repeatedcv", number = 30,
savePredictions = TRUE)
mylogit.K1 <- train(K ~ (bKOdds) +(pKOdds) +bHandAdv
+bKZscore + pKZScore , data =
t.trainBball[,c(1,3:26,28,29,55:57)], method="glm", family =
"binomial", trControl = ctrl, tuneLength = 15)
summary(mylogit.K1)
pred1 <- predict(mylogit.K1, newdata =
t.testBball[,c(1,11:26,28,29,55:57)], type = "prob")
#pred2 <- ifelse(fitted.results > 0.50,1,0)
results1<-data.frame(prob=pred1$`1` , Log5Pred=
Log5SO(t.testBball$bKperc, t.testBball$pKperc),
MorZPred=MoreyZSO(t.testBball$bKperc, t.testBball$pKperc),
actual=t.testBball$K, BatKPerc= t.testBball$bKperc, PitKPerc
= t.testBball$pKperc)


#----------------Import Results--------------------
results.true<-results1
results.true$prob<-round(results.true$prob, digits = 2)
results.true$Log5Pred<-round(results.true$Log5Pred, digits =
2)
results.true$MorZPred<-round(results.true$MorZPred, digits =
2)

write.csv(results.true,"results_ver1_Final.csv")
```

APPENDIX B2:

Coefficients of significant variables in the logistic regression

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	0.35600251	0.5502532	0.64698	0.51764525	
bKOdds	-6.71176047	2.19704523	-3.0549	0.00225133	**
pKOdds	-3.96631659	1.6606126	-2.38847	0.01691888	*
bHandAdv1	0.0779418	0.02244063	3.47324	0.00051421	***
bKZscore	0.78094726	0.14774883	5.28564	1.2527E-07	***
pKZScore	0.51641211	0.12424539	4.15639	3.2332E-05	***

APPENDIX C:

Model for modifying the head to head seasons data.

```
#--------------------Read data and make workdata out of it---------------------
data1<-read.csv("headtohead-2 seasons.csv")

#--------------Get The qualified Batters (PA>502)----------------------
workDatasort<-data1[order(data1$RESP_BAT_ID),]
result<-NULL #made a new dataframe which will contain the output data frame
count=1
PA_tot=0
a=1
for(x in 1:(nrow(workDatasort)-1)){
  print(x)
if(as.character(workDatasort$RESP_BAT_ID[x+1])==as.character(workDatasort$RESP_BA
T_ID[x])){
    count=count+1
  }
  else{
    for(y in a:(count)){
      PA_tot=PA_tot+workDatasort$PA[y]
    }
    row<-c(as.character(workDatasort$RESP_BAT_ID[x]),PA_tot)
#-----------------Making what will consist of the row in the new dataframe
    result<-rbind(result,row) #Binding
    a=count+1
    count=count+1
    PA_tot=0
  }
}
for(z in a:count){
  PA_tot=PA_tot+workDatasort$B_PA[z]
}
row<-c(as.character(workDatasort$RESP_BAT_ID[x+1]),PA_tot) #Making what will
consist of the row in the new dataframe
result<-rbind(result,row)
qual<-data.frame(result[which(as.numeric(result[,2])>1004)])

#-----------------Get qualified Pitchers(BF>502)---------------------------
workDatasort_pit<-data1[order(data1$RESP_PIT_ID),]
result_pit<-NULL
count=1
PA_tot=0
b=1
for(x in 1:(nrow(workDatasort_pit)-1)){
  print(x)
```

```
if(as.character(workDatasort_pit$RESP_PIT_ID[x+1])==as.character(workDatasort_pit
$RESP_PIT_ID[x])){
    count=count+1
  }
  else{
    for(y in b:(count)){
      PA_tot=PA_tot+workDatasort_pit$PA[y]
    }
    row<-c(as.character(workDatasort_pit$RESP_PIT_ID[x]),PA_tot)
    result_pit<-rbind(result_pit,row)
    b=count+1
    count=count+1
    PA_tot=0
  }
}
for(z in b:count){
  PA_tot=PA_tot+workDatasort_pit$B_PA[z]
}
row<-c(as.character(workDatasort_pit$RESP_PIT_ID[x+1]),PA_tot) #Making what will
consist of the row in the new dataframe
result_pit<-rbind(result_pit,row)
qual_pit<-data.frame(result_pit[which(as.numeric(result_pit[,2])>1004)])

#-----------Make FinalData for one plate appearance in one row----------------
finaldata08<-NULL
for (i in 1:nrow(data1)){
  print(i)
  if((data1$RESP_BAT_ID[i] %in%
qual$result.which.as.numeric.result...2.....1004..)&((data1$RESP_PIT_ID[i] %in%
qual_pit$result_pit.which.as.numeric.result_pit...2.....1004.. ))){
    print("y")
    for (j in 1:data1$PA[i]){
      row<-data1[i,]
      finaldata08<-rbind(finaldata08,row)
    }
  }
}

#--------Make anything other than the first Plate appearence zero---------------
finaldataTotal<- finaldataTotal[finaldataTotal$PA!=0,]
finaldata08<- finaldataTotal
p=1
while (p < (nrow(finaldata08))){
```

```
  print(p)
  if((finaldata08$RESP_BAT_ID[p]==finaldata08$RESP_BAT_ID[p+1]) &
(finaldata08$RESP_PIT_ID[p]==finaldata08$RESP_PIT_ID[p+1])){
    PA=finaldata08$PA[p]
    for(u in 5:12){
      v=p+1
      for(g in v:(p+PA-1)){
        finaldata08[g,u]=0
      }
    }
    p=p+PA
  }
  else{
    p=p+1
  }
}
copyfile<-finaldata08
#-----Spread the data in outcomes on the 1st line of PA to the rest----
i=1
while (i < (nrow(finaldata08))){
  if((finaldata08$RESP_BAT_ID[i]==finaldata08$RESP_BAT_ID[i+1]) &
(finaldata08$RESP_PIT_ID[i]==finaldata08$RESP_PIT_ID[i+1])){
    c=0
    y=0
    PA=finaldata08$PA[i]
    for(j in 5:12){
      #print (j)
      a=finaldata08[i,j]
      if(a!=0){
        y=y+1
        #print (c+i)
        o=c+i
        r=o+a-1
        for(z in o:r){
          finaldata08[z,j]=1
          print (z)
        }
        if(y>=2){
          finaldata08[i,j]=0
        }
        c=c+a
      }
    }
```

```
    i=i+PA
  }
  else{
    i=i+1
  }
}

#-----------------Writing the csv-----------------------------
write.csv(finaldata08,"DataForModel.csv")
```

References

[1] James, Bill. *The Bill James Baseball Abstract, 1983*. New York: Ballantine, 1983.

[2] Haechrel, M. (2014, September 22). Matchup Probabilities in Major League Baseball. *The Baseball Research Journal*. 43(2), 118-123.

[3] Morey, L. C., & Cohen, M. A. (2015). Bias in the Log5 estimation of outcome of batter/pitcher matchups, and an alternative. *JSA Journal of Sports Analytics, 1*(1), 65-76.

[4] Retrosheet Event Files. Play-by-play event files (n.d.). Retrieved April 11, 2016, from *http://www.retrosheet.org/game.htm*

[5] Cwevent: Expanded event descriptor¶. (n.d.). Retrieved April 11, 2016, from *http://chadwick.sourceforge.net/doc/cwevent.html*

[6] Chadwickbureau/retrosplits. (n.d.). Retrieved April 11, 2016, from *https://github.com/chadwickbureau/retrosplits/tree/master/splits*

[7] Baseball Reference. (n.d.). Retrieved April 11, 2016, from *http://www.baseball-reference.com/*

[8] Baseball Prospectus | Lies, Damned Lies: Batter vs. Pitcher Matchups. Retrieved April 12, 2016, from *http://www.baseballprospectus.com/article.php?articleid=1986*