Analyzing coaching decisions in college baseball using a reproducible in-game win probability model

September 15, 2018

Abstract

With the advancement of sports analytics, the notion of win probability has become more mainstream recently. However, the methods behind developing such probabilities are often proprietary in nature, and thus hidden from the general public. Moreover, analysis in sports is primarily focused on professional teams, with some notable exceptions in college basketball. In this paper, we use a mixture of generalized additive models (GAM) and multinomial logistic regression to develop a reproducible win probability model for college baseball, using completely open-source and publicly available data through stats.ncaa.com. Finally, we discuss the application of this model to analyze common decisions that coaches often face throughout the course of a game.

1 Introduction

The ultimate goal in all sports is winning, so it is natural for coaches and analysts to study the affects that certain decisions have on their team's chances to win. In the NFL, for instance, there is recent evidence that head coaches are becoming more aggressive in 4th down and short goal-to-go situations, with Coach Doug Pederson of the Philadelphia Eagles' "Philly Special" play call in the 2018 Super Bowl as a famous example. It is likely that given the current game situation, Coach Pederson felt that the decision to go for it, regardless of outcome, added to the Eagles probability of winning the game.

There has been much work on win probability added (WPA) across all major sports, including the National Football League (NFL) (Yurko et al. (2018); Burke (2010); Lock and Nettleton (2014)), the Nation Hockey League (NHL) (Pettigrew (2015)) and Major League Baseball (MLB) (FanGraphs (2018); Studeman (2004); Albert (2015); BaseballReference (2004)). With the notable exception of the research in NCAA Basketball (Benz (2017)), the majority of win probability research is limited to professional sports, with little work being focused in college baseball. The few rule differences (i.e. metal bats in college baseball, 3-point distance in college basketball) and major contrast in talent level are enough to to distinguish the college game from professional, thus warranting unique research in the area.

Outside of the notable exception Goldner (2017), the application of the recent win probability research emphasizes player evaluation, with little focus on in-game decision making. For example, (Yurko et al. (2018)) present a reproducible method for evaluating offensive football players in the NFL using a well-calibrated in-game win probability model. Moreover, statistical analysis in sports frequently relies on proprietary and costly data sources, ultimately limiting the potential of the field. This paper applies the methods described by Yurko et al. (2018) to baseball create an in-game win probability model for NCAA Division I baseball, and focuses the application of the model to in-game decision making. The ability to conduct this research using (Yurko et al. (2018)) as a primary resource magnifies the significance of reproducible methods, open-source data, and the quality of work presented by the authors.

1.1 Previous work in baseball

Analysis in a given sport begins with breaking down the game into its most basic units and assigning an appropriate value to indicate its success or failure. Baseball can be broken down and analyzed on a pitch-by-pitch basis, given the appropriate data is available. For instance, the MLB uses state-of-the-art tracking technology called Statcast to track pitch-level data that includes the pitch velocity, spin rate and access, among other data points. Fortunately, Statcast data has been made publicly available through `baseballsavant.com` and available for analysis through the `baseballr` package for `R` (Petti (2018)). Pitch-level data is extremely useful for analyzing pitcher and hitter tendencies (Albert (2018)) and predict future performance (@mducondi (2018)), among others. However, the pitch-level is not widely available below the professional level, so it is necessary to look elsewhere for the purposes of this paper.

The next natural step above pitch-level, and where this paper will focus, is at the play-level. MLB play-level data is made available through Retrosheet at `retrosheet.com`, which houses data as early as the 1921 MLB season. Like Statcast, Retrosheet data is also available through the `baseballr` package. One objective of this paper is to produce Retrosheet-type play-by-play data for NCAA Divison I College Baseball, which will be described in Section 2.

The value of a single play can be determined by how it affects the team's ability to score runs, which in turns affects the outcome of the game. For this reason, play-by-play analysis in baseball begins with *expected run value* (ERV), or the runs a team can expect to score the rest of the inning, given the current base-out situation. For instance, a team can expect to score more runs in the inning when the bases are loaded with zero outs than with the bases empty with two outs. ERV is calculated by finding the average number of runs scored the rest of the inning for all of the 24 base-out states, and is often presented in an *expected runs matrix*, which displays the run expectancies. Figure 1.1 shows an example of the ERV matrix for MLB for the years 2010-2015 (Lichtman (2018)). This example shows that from 2010-2015, an average of 0.481 runs were scored the rest of an inning where an at bat begins with bases empty and no outs (often written at '0 0', in the top, left most entry).

The scoring environment heavily impacts the values in an expected runs matrix. For example, scoring was higher in MLB over 2003-2009 (4.8 runs per game) than it was from 2010-2015 (4.26 runs per game). The reasons for the scoring difference over the two time periods is another area of study, but the values impact the expected runs matrix, giving a value of 0.547 at the '0 0' state (Lichtman (2018)), much higher than the 0.481 in 2010-2015. As for college baseball, in 2017 an average of 5.69 runs were scored per game, significantly changing the run environment, and thus the expected runs matrix (in Section 3).

The expected runs matrix is useful to determine the *run value* of a play, given by

$$\text{run value} = \text{ERV}_{\text{end state}} - \text{ERV}_{\text{start state}} + \text{Runs scored}. \tag{1.1}$$

A positive run value helps a team's potential to score runs, either directly or indirectly, while a play with negative run value has the opposite effect. For example, using the values from Figure 1.1, a play that begins with a runner on first with one out, and, after a double by the batter that does not score a run, ends with runners on second and third, results in a run value of

$$\text{run value} = 1.376 - 0.509 + 0 = 0.867.$$

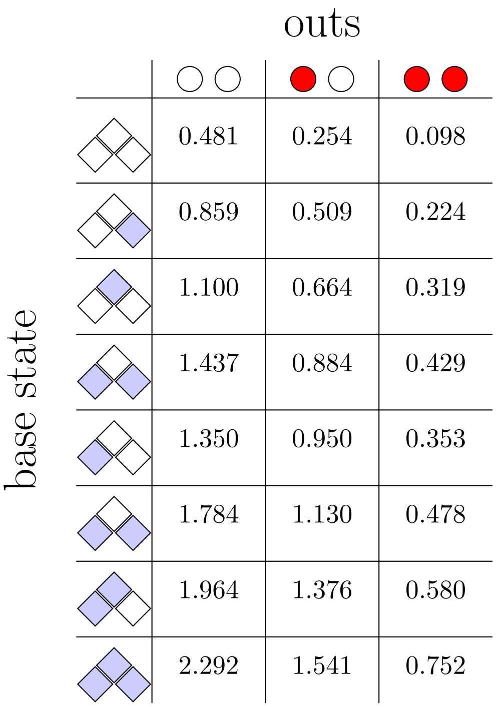
Run value is particularly useful to baseball statisticians because it is used to calculate some of the new sabermetrics, such as weighted-on-base-percentage (wOBA).

Despite its usefulness, ERV ignores crucial factors of a baseball game that also influence a team's ability to win: the inning and the score. For example, a three-run home run matters less to a team who is currently winning by eight runs in the ninth inning than it does to a team currently losing by one run in the seventh, despite having the same run value in two scenarios. This is the motivation behind *win probability added* (WPA), given by

$$WPA = \text{win probability}_{\text{end state}} - \text{win probability}_{\text{start state}}. \tag{1.2}$$

1.2 Structure of the paper

The concept of WPA is no different than in the NFL (Yurko et al. (2018)) or college basketball (Benz (2017)), however calculating a team's win probability at a given state is unique to the sport. This paper will demonstrate the steps to creating a well-calibrated win probability model specific to NCAA Division I College Baseball, and is laid out as follows. Section 2 describes the data and methods for open-source collection. Section 3 builds an in-game game win probability used to evaluate decision making in Section 4. Section 5 will conclude with a brief summary and discussions of future work.



outs

base state	○ ○	● ○	● ●
◇◇◇	0.481	0.254	0.098
◇◇◆	0.859	0.509	0.224
◇◆◇	1.100	0.664	0.319
◆◇◆	1.437	0.884	0.429
◆◇◇	1.350	0.950	0.353
◆◇◆	1.784	1.130	0.478
◆◆◇	1.964	1.376	0.580
◆◆◆	2.292	1.541	0.752

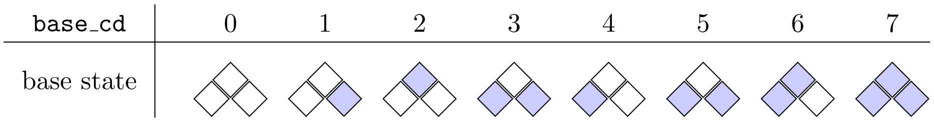
FIGURE 1.1: Run expectancy matrix for MLB 2010-2015. Lichtman (2018)

2 Data

The data used for this paper is play-by-play data for 316 teams playing over 8100 games per year from 2017-2018, resulting in 16532 unique games played over the two-year span. The data was collected from the NCAA statistics webpage: `stats.ncaa.com` using the `tidyverse, XML, stringr` and `RCurl` packages in `R`. The methods were inspired by those used in `R` packages for accessing sports data: `nflscrapr` Yurko et al. (2018), `nhlscrapr` (Thomas and Ventura (2017)) and `ncaahoopr` (Benz (2018)).

Once collected, the play-by-play data was parsed using advanced regular expressions and other data manipulation techniques to extract detailed information about each play, closely resembling the parsed Retrosheet data used for MLB. In addition to the year, date, id number and team identifiers, among other common game traits, Table 2.1 gives a description of the variables found in the parsed play-by-play dataset.

The elements in the parsed dataset are sufficient to conduct the type of research described in Section 1.1. Particularly, the `runs_roi` column describes the runs scored the rest of the inning given the `base_cd_before` and `outs_before` the play occurs: the three elements necessary to compute the expected runs matrix. From there, one can compute the run value of a given play (or coaching decision) using the play's `event_cd`[1]. For reference, the `base_cd_before` entries are given in Figure 2.1.



`base_cd`	0	1	2	3	4	5	6	7
base state	◇◇◇	◇◇◆	◇◆◇	◆◇◆	◆◇◇	◆◇◆	◆◆◇	◆◆◆

FIGURE 2.1: `base_cd` values for base states.

[1] Coincides with Retrosheet event codes.

Variable	Description
`top_inning`	Indicator if top (1) or bottom (0) of the inning
`bat_order`	Where the current batter hits in lineup (1-9)
`base_cd_before`	Base code describing the base state before the play (0-7)
`outs_before`	Outs before the play
`event_cd`	Code that describes the event (0-23).
`hit_type`	Descriptor if the hit is a fly out (FO), ground out (GO), line out (LO), or bunt (B)
`outs_on_play`	Outs on the play
`outs_after`	Outs after the play
`runs_on_play`	Runs on the play
`runs_this_inn`	Runs scored in current inning
`runs_roi`	Runs scored rest of inning, including the play
`away(/home)_score_after`	Away(/Home) score after the play
`sh_fl`	Indicator identifying a sac bunt
`sf_fl`	Indicator identifying a sac fly
`int_bb_fl`	Indicator identifying an intentional walk

Table 2.1: Description of the play-by-play dataset.

3 Win probability model

We now move on to the primary focus of this paper in modeling the win probability (WP) for the hitting team at any state of the game. All win probability models begin with the current score and time as basic indicators. For baseball, the inning to indicate time, but in addition to net score, we will include *expected score difference*, which we will estimate using expected runs.

3.1 Expected runs

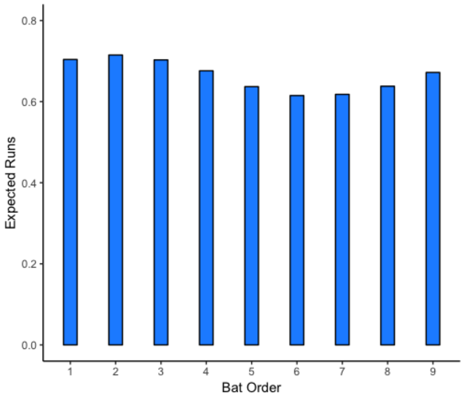
As described in Section 1.1, expected runs is a common method for evaluating a play in baseball, however it is heavily influenced by the run environment in which the game is being played. In college baseball, for example, there were 5.69 runs scored per game in 2017, compared to the approximately 4.63 runs per game in the MLB that same year. As a result, the run expectancy in 2017 college baseball at a given state will be higher than in the MLB. Using the data described in Section 2, we can create the expected runs matrix for college baseball from 2017 and 2018, given in Table 3.1.

	0	1	2
0	0.664	0.351	0.137
1	1.120	0.665	0.288
2	1.415	0.847	0.391
3	1.820	1.139	0.547
4	1.698	1.145	0.462
5	2.104	1.423	0.649
6	2.373	1.599	0.712
7	2.753	1.902	0.951

Table 3.1: Expected runs matrix for college baseball in 2017 and 2018.

One main aspect that is ignored when computing an expected runs matrix is the spot in the lineup where the current batter hits, which tends to coincide with the ability of the hitter (i.e., the third batter tends to be a better hitter than the eighth or ninth batters). Intuitively, a team should expect to score more runs when their better hitters are coming to bat (top of the lineup) than the weaker hitters (seven through nine). This can be seen in Figure 3.1, which displays the average runs scored the rest of the inning in a college baseball game given current position in the batting order, regardless of base-out state.

Figure 3.1: Expected runs in college baseball by batting order and distribution of runs scored in an inning: 2014-2018



As expected, Figure 3.1 shows that a team can expect more runs when at the top of the order, with a noticeable decline in spots five through seven. Unexpectedly, however, there is a slight increase of expected runs at spots eight and nine, with the possible explanation being that the lineup "turns over", or goes back to the top with better hitters, after those spots.

The expected runs presented in Table 3.1, as in most expected runs matrices, are calculated by taking the average runs scored the rest of the inning at the given state: base code and outs. We propose to extend the expected runs matrix by adding "batting order" to the state. For each of the previous 24 states, we compute the average runs scored the rest of the inning, resulting in the familiar 8×3 expected runs matrix for each of the 9 spots in the batting order for a total $8 \times 3 \times 9 = 216$ unique states. Finally, we add net score to the estimated expected runs, creating our next indicator, *estimated score differential* (ESD):

$$\text{ESD} = \begin{cases} \text{ER} + \text{away score} - \text{home score} & \text{if } \texttt{top\_inn} = 1 \\ \text{ER} + \text{home score} - \text{away score} & \text{if } \texttt{top\_inn} = 0, \end{cases} . \tag{3.1}$$

Throughout our work with expected runs, we built several models to predict the runs rest of inning using various inputs, including base code, batting order and outs. However, the predicted expected runs did not outperform the "simple" average expected runs well enough to justify the complexity.

3.2 Model Structure

The application of the paper is coaching decision evaluation, so we will model win probability without taking the teams playing or location into account (i.e., we do not include team strength as an indicator). Ignoring team strength should indicate that each team starts with a 50% win probability, however in baseball the home team has a built-in pre-game advantage by hitting in the bottom of each inning, resulting a home team winning percentage of 59% in 2017 and 2018 college baseball.

The final consideration when building the win probability model for baseball is that in any given inning, the state of the game for the hitting team is different in the top of the inning than the bottom. For instance, in the top of the first inning, both the hitting (away) and fielding (home) teams have 27 offensive outs remaining in the game (assuming a 9-inning game). In the bottom of the first, however, the hitting (home) team still has 27 outs remaining, while the fielding (away) team has 24 (after 3 outs in the top of the inning). As a result, we will build two separate models, one for each of the half-innings.

3.3 Generalized additive model

To estimate WP, we use a generalized additive model (GAM), similar to that proposed for the win probability model proposed by Yurko et al. (2018). Contrary to standard linear regression, GAMs allow the relationship between the explanatory and response variables to vary according to smooth, non-linear functions.

We use a generalized additive model with a logit link function and ESD and inning (and their interaction) to create two WP models for the hitting team, home and away, both of the form:

$$\log\left(\frac{p\,(\text{Win})}{p\,(\text{Loss})}\right) = s(\text{ESD}) + s(\text{inning}) + s(\text{inning}, \text{ESD}). \tag{3.2}$$

By taking the inverse of the logit:

$$p(\text{Win}) = \frac{1}{1 + e^{-\alpha}} = \frac{e^{\alpha}}{1 + e^{\alpha}}, \tag{3.3}$$

where α is the right-hand-side of (3.2), we arrive at the WP for the hitting team at the given state.

3.4 Win probability in the bottom of the ninth inning

The final step in building the win probability model for baseball is to consider the unique state of the game in the bottom of the ninth (or final) inning. The rules of baseball dictate that if the home team is winning at the end of the top of the ninth inning, then the game ends without the bottom half being played. This fact alone distinguishes the structure of the data in the bottom of the ninth inning from the rest in that there will be no instances where the home team is winning.

A second aspect to consider is that the home team has three unique outcomes at the end of the bottom of the ninth: win, lose or “force extra innings”, where an additional full inning will be played if the game is tied. Consider the example where the game is tied in the bottom of the ninth inning and the home team has a runner on third with one out. In that scenario, the home team has three outcomes with associated probabilities:

$$\begin{aligned} \text{Win now (or walk-off):}\ & p(\text{Walk-off}) \\ \text{Remain tied and force extras:}\ & p(\text{Extras}) \\ \text{Lose:}\ & p(\text{Loss}) = 1 - p(\text{Walk-off}) - p(\text{Extras}). \end{aligned} \tag{3.4}$$

Equation 3.4 is the motivation behind using multinomial logistic regression to estimate the probabilities for both walk-off and forcing extras relative to the “Loss” outcome. Similar to Equation 3.2, we use ESD as an indicator but drop inning as it is specifically for the ninth, to give the form:

$$\begin{aligned} \log\left(\frac{p(\text{Walk-off})}{p(\text{Loss})}\right) &= s(\text{ESD}) \\ \log\left(\frac{p(\text{Extras})}{p(\text{Loss})}\right) &= s(\text{ESD}). \end{aligned} \tag{3.5}$$

Taking the inverse of the logit (3.3) gives the two probabilities, $p(\text{Walk-off})$ and $p(\text{Extras})$. Finally, we use $p(\text{Walk-off})$ and $p(\text{Extras})$ as indicators to estimate the home team's probability to win the game given the current state in the ninth inning:

$$\log\left(\frac{p(\text{Win}_9)}{p(\text{Loss})}\right) = s(p(\text{Walk-off})) + s(p(\text{Extras})). \tag{3.6}$$

We note that we use the ninth-inning model for extra innings as the current state is exactly the same as the ninth.

3.5 Win probability calibration

We use a 75/25 train/test split on events that were not marked "No event" (i.e., substitutions, ejections, rain delays, etc.) from the parsed play-by-play data from 2017 and 2018. Though we did not think ties were possible in baseball, we found and removed 36 games where the game ended with the scored tied. The total dataset resulted in 1,047,869 rows training data and 349,291 rows of test data.

The calibration technique mimics that by Yurko et al. (2018) where the estimated win probabilities for each state $\widehat{p_i}$ are binned in 5% increments, with the observed proportion of the state p_i found in each bin. Figures 3.2 and 3.3 show the calibration plots by inning for the away and home teams, respectively.

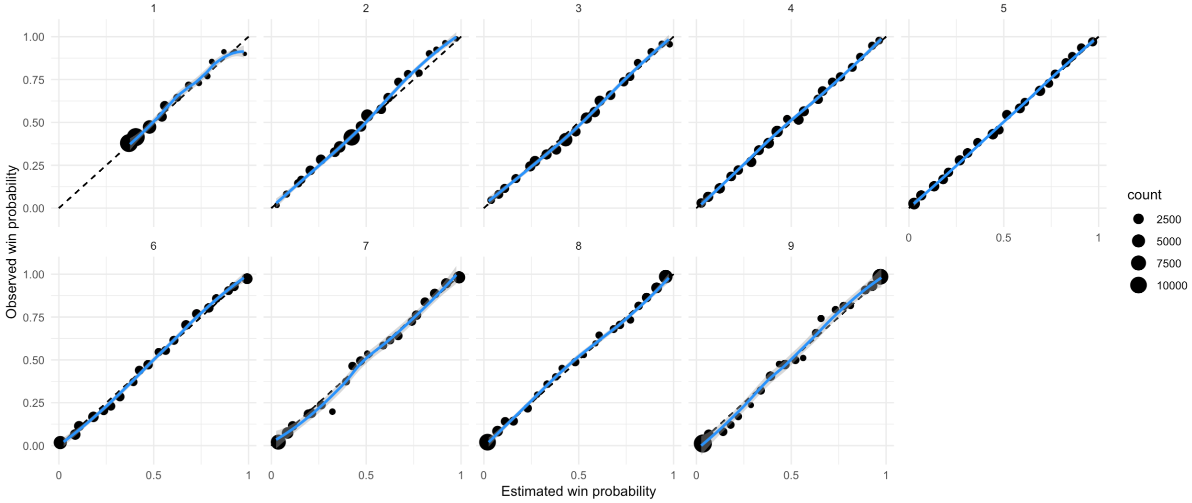


FIGURE 3.2: Away team win probability model calibration by inning

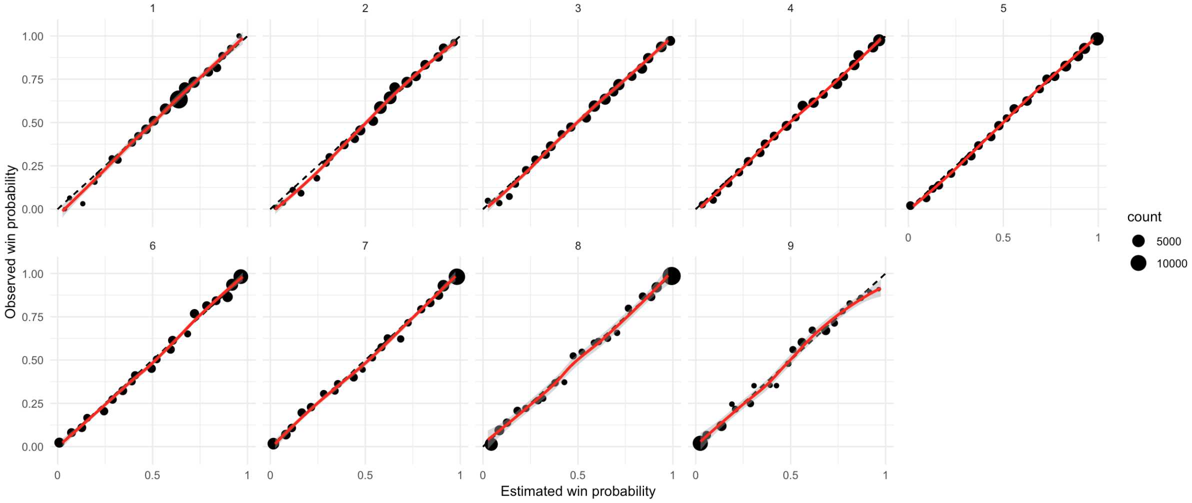


FIGURE 3.3: Home team win probability model calibration by inning

The observed proportion of wins closely matches the estimated proportion of wins within each bin in each of the 18 half-innings, indicating that both models are well-calibrated across all innings and all possible win probabilities. One note to make is that there seems to be no predicted probabilities below $p = 0.4$ in the top of the first inning. We can explain this by noting that the hitting (away) team will never be losing in the top of the first inning, because the home team has yet to hit, and such, cannot score more runs than the away team. Similarly, in the bottom of the 9th inning, there are few predicted win probabilities near $p \approx 1$. Again, we can explain this by noting that the hitting (home) team cannot enter the bottom of the ninth with a lead (since the game will have ended with the home team already have won). The few predicted probabilities above $p = 0.5$ occur when the home team is tied with several runners on base, with few outs, giving a high ESD.

4 Win probability added of coaching decisions

A coach makes a number of decisions throughout a game, such as when to bunt, steal, pinch-hit, etc., in hopes to give their team the best chance to win. Some decisions, like a pitching change or a pinch runner, cannot be analyzed using this model as it only accounts for game state, which will remain unchanged after a substitution (giving WPA= 0). It is conceivable to estimate the WPA of a substitution using a more complex model that includes player and team strengths, however we will focus on two decisions that change the current game state: sacrifice bunts and intentional walks.

With the well-calibrated WP model from Section 4, we evaluate the value of certain decisions using the the concept of *win probability added* (WPA), given by

$$\text{WPA} = \text{WP}_{\text{after}} - \text{WP}_{\text{before}}. \tag{4.1}$$

With the intention of attempting to value the decision to intentionally walk an opposing batter or to sacrifice bunt, we are able to estimate WPA values for both scenarios using the data available. With the WPA estimates, we can pinpoint a general situation in which it is likely to be beneficial to the team.

The outcome of intentional walks (IBB) is known to a coach before they make the decision: the batter will be awarded first base. The 2602 intentional walks issued in 2017 and 2018 had an average WPA of -3% for the pitching team, and there were only 48 instances (1.8%) where the WPA was positive. In all of the scenarios, the game was tied in the bottom of the ninth inning with either a runner on second or first and third, most of which facing a batter third or fourth in the lineup. Finally, the average WPA for those 34 instances was 3%.

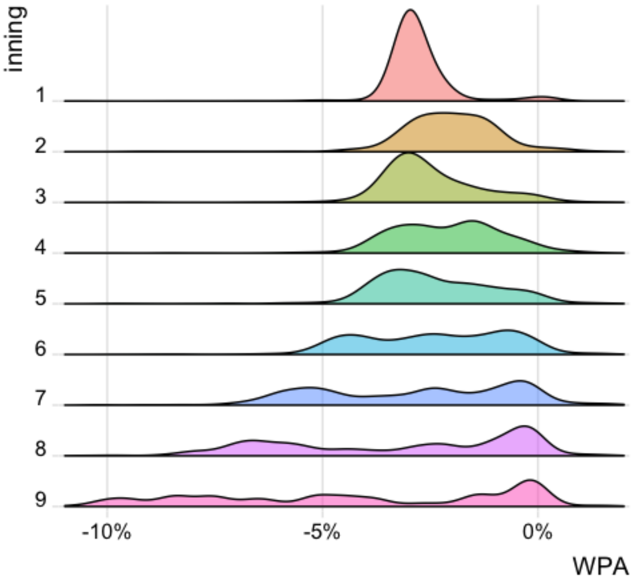


FIGURE 4.1: WPA distribution for SAC bunts for each inning in 2018

Conversely, when a coach calls for a sacrifice bunt, they are doing so with the intention of the batter being thrown out and moving a base-runner forward, which may be beneficial to coaches in certain situations where the game is on the line. Figure 5.1 shows the distribution of WPA for sac bunts for each inning in 2018. We notice that the WPA from sac bunts becomes less concentrated as the game progresses, and actually widens towards becoming more negative as the game goes into the later innings, and that some bunts in the 9th inning can cost a team up to 10% win probability.

The most extreme example of losing WPA through a successful sacrifice bunt we found was in the bottom of the 9th inning of a game between Loyola Marymount and Pepperdine (game_id= 4585160). The situation was in the bottom of the ninth inning, with no outs, runners on first and third, losing by two runs and the second batter at bat. According to the win probability model from Section 4, the home team has a WP= 51% in that given situation. The batter for Pepperdine sacrifice bunted, moving the runner from first to second, resulting in a new situation with WP= 25%, resulting in a WPA of −26%. On the next play, however, the third batter for Pepperdine hit a three-run walk-off home run to win the game.

5 Summary and conclusions

In this paper we showed how to build and validate an expected runs and win probability model using completely free and available data. We can use these models to analyze situations and make suggestions to coaches given the findings. We will continue working on this project by developing an R package, similar to those mentioned in Section 2, to allow for more data acquisition and quantitative analysis of college baseball. Finally, we will explore various machine learning techniques to produce more models using multiple years' worth of data.

References

Albert, J. (2015). Calculation of win probabilities, part ii. https://baseballwithr.wordpress.com/2015/01/27/calculation-of-win-probabilities-part-ii/.

Albert, J. (2018). Using statcast to measure hitters. https://baseballwithr.wordpress.com/2018/01/02/using-statcast-to-measure-hitters/.

BaseballReference (2004). Win expectancy (we) and run expectancy (re) stats. https://www.baseball-reference.com/about/wpa.shtml.

Benz, L. (2017). Ncaa basketball win probability model. https://sports.sites.yale.edu/ncaa-basketball-win-probability-model.

Benz, L. (2018). ncaahoopr: An r package for working with ncaa basketball play-by-play data. https://github.com/lbenz730/ncaahoopR.

Burke, B. (2010). Win probability added (wpa) explained. http://archive.advancedfootballanalytics.com/2010/01/win-probability-added-wpa-explained.html.

FanGraphs (2018). Wpa. https://www.fangraphs.com/library/misc/wpa/.

Goldner, K. (2017). Situational success: Evaluating decision-making in football. *Handbook of Statistical Methods and Analyses in Sports,*.

Lichtman, M. (2018). Re. http://www.tangotiger.net/re24.html.

Lock, D. and Nettleton, D. (2014). Using random forests to estimate win probability before each play of an nfl game. *Journal of Quantitative Analysis in Sports*, 10(2).

@mducondi (2018). Using statcast data to predict future results. https://www.fangraphs.com/community/using-statcast-data-to-predict-future-results/.

Petti, B. (2018). baseballr. https://billpetti.github.io/2018-02-19-build-statcast-database-rstats/.

Pettigrew, S. (2015). Assessing the offensive productivity of nhl players using in-game win probabilities. *MIT Sloan Sports Analytics Conference*.

Studeman, D. (2004). The one about win probability. https://www.fangraphs.com/tht/the-one-about-win-probability/.

Thomas, A. and Ventura, S. L. (2017). nhlscrapr: Compiling the nhl real time scoring system database for easy use in r. `https://CRAN.R-project.org/package=nhlscrapr,r`.

Yurko, R., Ventura, and Horowitz, M. (2018). nflwar: A reproducible method for offensive player evaluation in football. *arXiv:1802.00998*.