Evaluating Run Scoring Efficiency in MLB Through Markov Chain and similar Modelling Processes

Brett Gustin
Syracuse University
December 2023
Undergraduate Research Thesis

Abstract

Baseball's unique aspect of a 162 game season is part of what makes the sport so special. The durability needed for a Major League Baseball year is hard to compare to any other sport. The length of the regular season leads to believe that the best teams are the ones that can come out on top in the standings after an enduring battle all year. However, once in the playoffs, anything can happen in the sport. In the past nine years, five wild card teams have made it to the World Series with three of them being crowned champions. If it is all about making it to the dance, then the goal is to perform with consistency in the regular season. Broken down in its simplest form, the game of baseball is won when a team scores more runs than the other. Doing so over the long course of a season will set you up as a team for playoff standing. The goal of this research work is to quantify how teams set themselves up for playoff contention in terms of scoring runs efficiently. Through various modelling processes including Markov Chain simulation, understanding how previous teams succeed through consistent run production allows for accurate prediction for teams to come.

I. Literature Review

The way in which baseball players are evaluated has drastically changed with an influx of performance based data. The construction of a lineup for every game in baseball brings uniqueness to the sport with different possibilities each day. As metrics continue to influence the value of players, the use of data to create the optimal lineup has changed the intuition behind batting order slots. The numerical order of the batting lineup has seen differences overtime in what is valued in each position. For instance, the first two hitters in the order were previously defined as players who need to be high speed, purely contact hitters. Through the use of analytics, what is valued in the one and two hole has shifted towards the metrics of OBP(on-base percentage) and SLG(slugging), proving these early hitters indeed are better offensive hitters. Putting certain players ahead in the order contrary to traditional beliefs shows the impact of data driven answers(Kalkman, 2012).

Lineup Order Ideology

Numbers show that since your first two hitters have a higher chance of expected plate appearances, you want high volume hitters in these positions. This ideology is the backbone of a much more complex concept of how to optimize the best batting order. Each position in a lineup has changed their role with plate appearances being a large influence on the order. From there, stats such as Plate Appearance percentage take into account exactly how much can be utilized from each position in the batting order. Figure 1 is helpful with understanding the influence of plate appearances per game broke down at every position in the batting order.

Batting Order Position (BOP)	PA/Game
1	4.83
2	4.72
3	4.61
4	4.49
5	4.39
6	4.26
7	4.14
8	4.02
9	3.90

Figure 1(Sherman,2012)

Given these findings, there is indication that the higher up in the order a player is, the more likely they are to see more plate appearances through the order. As a result, the higher in the

order players have a larger chance to create an expected outcome of their plate appearance. Certain outcomes have a different run expectancy to them, or in other words how much that particular outcome influences runs that can be scored such as a double, homerun, or a strikeout. From this point, run expectancies can be compared at positions, and by player, to show the influence that batting certain hitters in certain positions can make(Sherman,2012).

Markov Chain Introduction

Identifying the importance of plate appearances per game leads to an understanding of run expectancy at each position in the order. Essentially, an optimized lineup can be connected to putting runs on the board from a players run expectancy(*Sports Betting Dime,2022)*. The process of creating an optimized lineup with relation to run expectancy has been introduced through data simulation. The two common approaches to simulate a batting lineup's expected runs is through Combinatorial and Markovian methods. The difference in the methods center around how we simulate run scoring. The Markovian method simulates while transitioning to different conditional states in a baseball game. The Combinatorial approach takes plate appearances and permutates it to simulate the ideal lineup that maximizes runs scored(Saito, 2022). The more frequently used model to determine run expectancy simulation for batting lineups is the Markov Chain method, also known as Monte Carlo simulation. The mathematical modelling of Monte Carlo is evaluated by the movement between states. A process is started in an original state and moves from different states through transitions. The chain process of transferring from certain states can be identified after a certain number of steps. With the probability of the chain being at each state known, a move to another given state depends solely on the state of the chain before(Polaski,2013). By definition, these models create a simulation designed for taking probabilities and finding their sampling distributions. The processes of simulation to different states can be translated to the sport of baseball with metrics to tell the story. The Markov chain states of probabilities can represent the different points of a game based on the base states and outs for the offensive team. To introduce the Markov chain for a baseball game, a utility metric can be collected through half inning simulations. The process of Markov Chain allows for evaluation of the simulation given expected outcomes. As a version of what machine learning can do, lineup optimization can be considered through chain movement. Monte Carlo has the ability to determine which roster or lineup generates the highest probabilities to obtain the most amount of wins. Therefore, these solutions create value and significance for how players can be measured towards team success(Albert 2023, Henderson 2016, McIntyre 2016, Thaker 2011).

Representation of Markov Chain Processes in Baseball

The use of Markov Chain simulation can be broken down in the game of baseball to represent simulated results of expected outcomes. In a half inning, there can be zero, one, or three outs, and eight configurations of runners on first, second, and third. According to Schorsch's markov chain findings, a total of 24 possible states that an inning can be in with each

state connected to others depending on the possible outcome of the hitter at bat(Schorsch, 2015). When a 25th state is added as the end of a half inning, all the possible transitions from states in the chain can be tracked based on possible hitter outcomes. Figure 2 shows all the possible transitions from the given state of (3,1), or runner on third and one out.

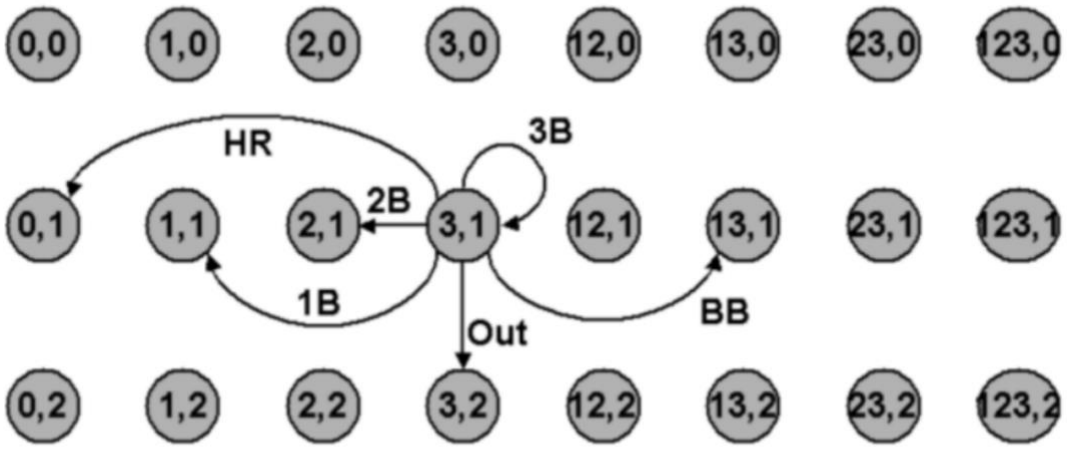


Figure 1: The 24 states with example transitions shown

(Schorsch)

Creating transitions in probabilities by the number of occurrences at different states is explained and gathered through data. The states are identified and can be applied as parameters in the Markov Chain model. These states or possible outcomes on offense can be used for the model until the third out stage is applied for an inning to end. This transition matrix in the run scoring process is a parameter to simulate what is defined as the run-scoring process(Sokol 2003, Thomay 2014). Throughout the process of nine innings, the transition of states can be identified in a block matrix, or the possibilities to travel to different parts of the chain.

Using probabilities to determine base state combinations allows for movement through transitions in the chain. For instance, a movement from (0,0) to (1,0) is a runner on first and still no one out. Therefore, that transition is the probability of a single, hit by pitch, walk, and error(Asaro, 2015). The key to a Markov chain simulation is to determine how many runs one can expect a team to get at any state of the half-inning. During the transitions, a Markov chain has the ability to maximize an outcome based on the simulations of changes in the chain. This part of the simulation is what drives the use of markov methods for baseball lineup optimization. In order to consider the idea of maximizing the potential of an every day starting nine, you can consider the optimal lineup as the one “to produce, on average, the most number of runs per game. Thus, the factor we need to maximize is the number of runs”(Engel, 2012). There is intuition that goes into determining how to build the transitions and how stats in the

lineup are built in as well. The work of Daniel Ursin creates a fundamental understanding of how to intuitively decide on how the matrices are built, which is a crucial part of the model. From there, the source identifies how to run a single inning run distribution with the transition matrices. Then this can be done at a nine inning level to cycle through a lineup of nine batters and take the probabilities of teams scoring runs in the model to expected runs(Ursin, 2014).

Expected Run Outcomes in Markov Chain Simulation

Creating transition matrices in a Markov model and then simulating half innings can result in finding a maximum desired outcome. Introduced with the work of Max Marchi, the desired outcome of runs expected can be used as a statistical measurement to make conclusions with Markov chain applications(Marchi, 2019). While creating the matrices to states in a game, it is important to understand the probabilities associated with moving in the chain in order to create effective results. From previous research findings with Eugene Beyder, the matrices of a Markov Chain can apply to baseball by including data from a previous season that tracks all offensive plays for a batting lineup. This allows for recording of the occupancy of bases and how many outs prior to the play to know what a player will do in a given situation. From there, a 24X25 matrix of transition states allows for every entry into a matrix for a team dependent on their situation and where they are in the chain. Therefore, the data in the matrix can represent the number of times in a simulated game that a transition from a starting state to another chain state occurs. A function can be created to determine the probabilities of changing states, by taking the occurrence of a transition and dividing that by the sum of the row that the entry occupies in the matrix(Beyder, 2015). While moving through the chain, transitions can be based on run outcomes at given base out states. Combining states to transitions allows for the understanding of runs that can be scored from different states.

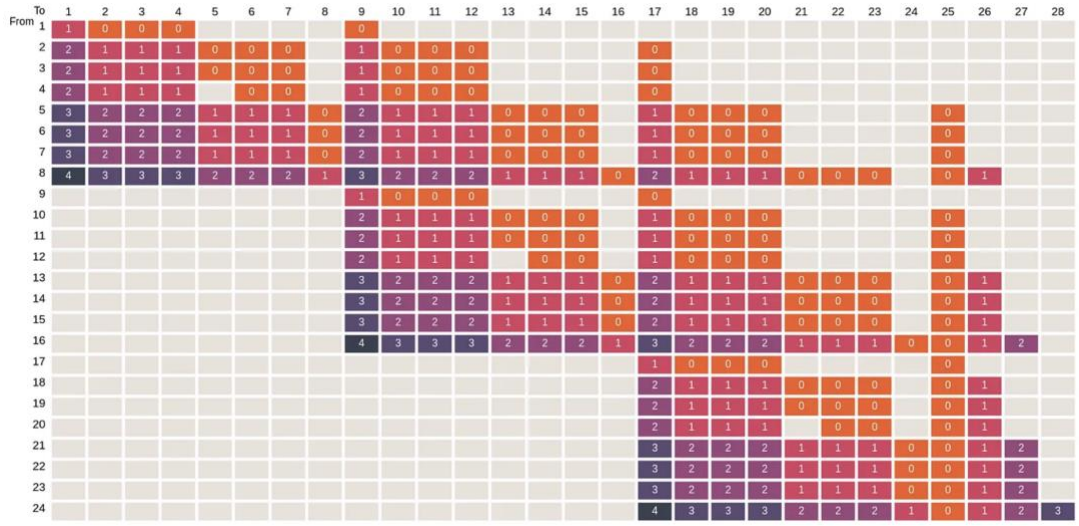


Figure 3(Calestini, 2018)

The third figure represents where a transition for a run scored can be made within the matrix model. For instance, in the eight state of bases loaded to state one of nobody on is an expected

run outcome of 4. This example creates a block matrix to show that all transitions cannot lead to runs scored. The number of expected runs can be collected from specific player data by the sum of transition probabilities times the run vector. As described in other words, an expected outcome is found by multiplying runs gained at a state, with the frequency of being in that state, and the initial run vector state(Harvard University, n.d.). The run vector is the run each transition creates and is represented by Figure 3. The matrix then can be computed with 24 values for each possible state in a game if a batter comes to the plate. The expected run outcome at each state given player data since 1921 is produced in Calestini's work.

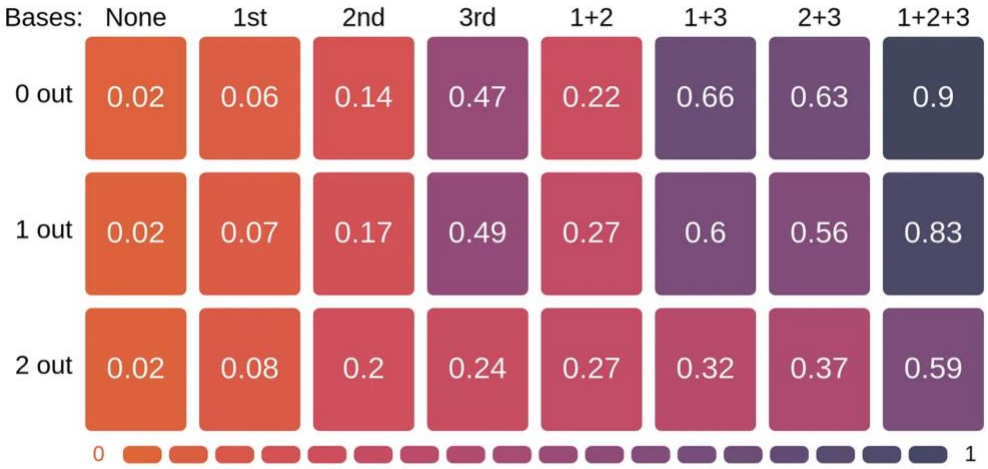


Figure 4(Calestini)

While revealing the aggregated expected run totals given the transition matrix values, the Markov Chain can be used as a comparison tool for evaluation. From the batter perspective, how the runners and outs influence your chance of scoring an RBI is obtained in a metric for expected runs in the transition states. From this point, the work of Bruce Bukiet finds that the batting order that produces the highest expected number of runs in a game is also the order that produces the greatest expected number of wins(Bukiet n.d., Calestini 2018).

The term stochastic matrix is used as a way to summarize the large matrix transition to different probability states. Certain transitions to particular states are introduced as block matrix to understand how a model would know certain out states to end a half inning while in the simulation. From this point, an example on expected number of runs through the use of the model is simulated and related to true runs scored. The work by *Statshacker* calculates expected runs per half inning as an evaluation tool metric. An "actual" run total was created by the average number of runs scored divided by the average number of innings pitched. The findings indicate the two sets of results being strongly correlated. This correlation concludes that in this project that some stable season-over-season average effect is missing. A further discussion indicates that an explanation for this is non-batting events such as the impact that base stealing has(Statshacker, 2018).

Year	Actual	Markov	Difference (%)
2017	0.526	0.509(1.043)	3.2
2016	0.506	0.491(1.008)	3.0
2015	0.485	0.472(0.993)	2.7
2014	0.464	0.450(0.973)	3.0
2013	0.478	0.463(0.981)	3.1
2012	0.493	0.480(1.003)	2.6
2011	0.493	0.480(1.003)	2.6
2010	0.498	0.483(1.008)	3.0

Figure 5(Statshacker)

These results provide comparison between actual expected runs scored and what a Markov chain process can conclude to. For the example research from *MaplePrimes*, a simulation is run and then order permutations follow that given different lineup outcomes in permutations from 1 to 9. Then, runs scored is created with these lineups as a tool to rank them to determine the optimal lineup outcome. This is seen through a ranking of a given starting lineup and their metrics including OBP, SLG, OPS, and AVG. Different lineups can be produced that result in varying calculations for expected runs scored. Deciphering between the best and worst permutations can create value for certain hitting metrics over others(MaplePrimes, 2013). Markov chain simulation can have different approaches to create a maximized outcome. A popular approach comes from evaluating expected runs to create statistical value to the process to create conclusions. From half inning observations, expected run outcomes can conclude that a batting team scores incrementally and each subsequent score is generally less likely to occur. It was also found that the most common half inning is one without the batting team scoring(Spencer, 2017). Another approach can be through modelling packages as introduced by Spedicato's work in modelling Markov chain through R programming. The package markov chain allows for the handling of simulations and their transition matrices for an expected outcome through R(Spedicato, n.d.).

Alternative Lineup Optimization Findings

While evaluating hitter value through discoveries of a Markov Chain, there are other methods and approaches to maximizing outcomes of a starting lineup in baseball. The work of Parker Chernhoff suggests that maximizing variance inflation factors through regression modelling can determine what metrics have statistically significant correlation with runs scored(Chernhoff, n.d.). Finally, an evaluation of lineup optimization can be related to daily fantasy sports and average fantasy point scored. Some of these methods include creating optimization models based on the lineup of a team and their respected projected points. Parameters such as salary and position create the maximum potential of a lineup given average

and projected points in fantasy. With a certain salary to create the best fantasy lineup to project for a week, a tool modeling a player's projection with fantasy points creates lineup optimization(Github 2021, Zamora 2018).

II. Methodology

A. Data Summary

In order to implement a simulation process for MLB run efficiency, proper data collection of batting outcome events is needed. Breaking down what Markov Chain simulation possesses allows for an understanding of what data needs to be gathered. By turning the game into transition states, every moment in a baseball game is at a state and has the possibility to move from that state. Moving through transition states is defined through a probability of how frequent that outcome or move to another part of a game can occur. The probabilities necessary to define moving through states in a baseball game can be obtained through batting outcome events at the team level. Data is collected from over seven years of MLB team offensive statistics, excluding the pandemic season. It is important to note that during this time there was an additional playoff team added to each league. This data was collected through offensive team statistics on baseballreference.com. Since the analysis for the Markov Chain is to scale how efficient playoff teams are at scoring, the data is split this way. Outcome events for the simulation are collected for playoff teams and non-playoff teams with offensive categories including walk, HBP, single, double, triple, homerun, and out. These are all of the possible events that a state can transition to in the chain. During collection of the data the variables are separated as shown below.

Fig 1

Postseason Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
BB&HBP	62	628	57	506	581	672	752
1B	62	881	58	779	830	915	1022
2B	62	286	29	213	270	302	355
3B	62	24	8.3	6	19	29	54
HR	62	216	36	127	198	236	307
Outs	62	4109	45	3984	4075	4140	4208
Total	62	6143	94	5915	6081	6204	6338

Missed Postseason Summary Statistics

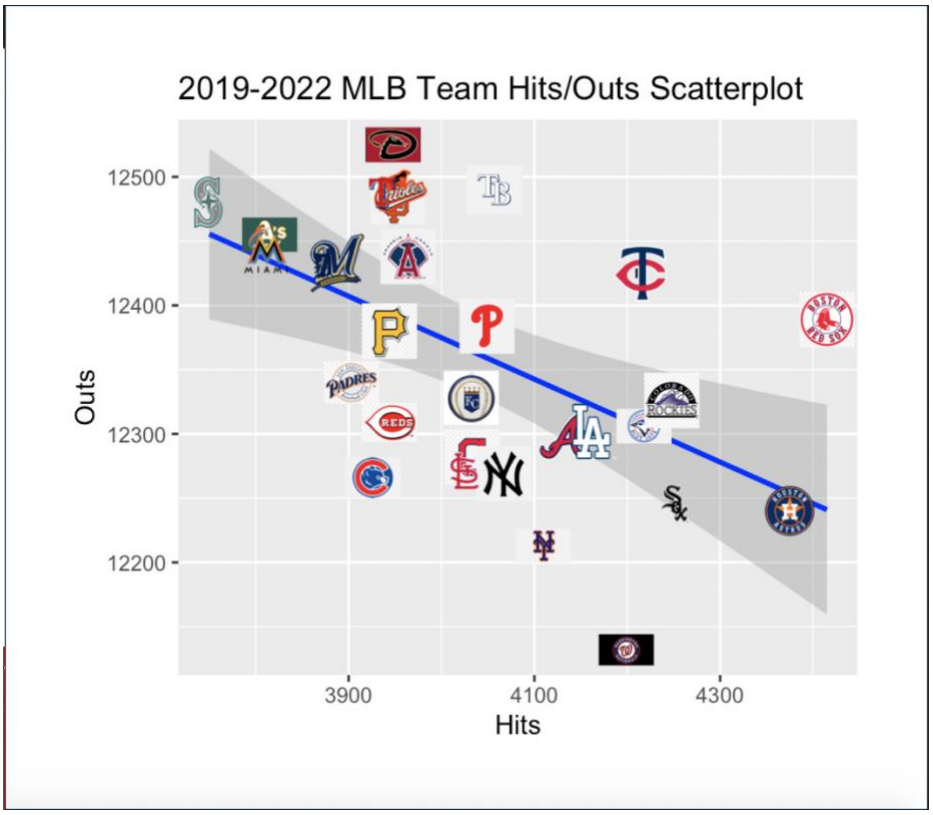
Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
BB&HBP	118	557	63	427	507	606	690
1B	118	873	63	746	824	922	1031
2B	118	267	26	219	249	284	352
3B	118	26	9.1	5	20	32	56
HR	118	185	33	110	161	214	262
Outs	118	4129	50	3967	4101	4164	4247
Total	118	6037	101	5799	5974	6102	6409

Within the preparation for running models to predict run efficiency, there are two ways in which the batting outcome data is used. For linear regression analysis, the categories of events remain separated other than walk and hit by pitch being one variable. For Markov Chain analysis, a walk, hit by pitch, or single are grouped together as they create the same result in the simulation(a runner advancing to first).

B. **Preparing Regression & Simulation Modelling**

After collecting data based on team statistics and separated by playoff appearances, it is important to approach the question of run efficiency in multiple capacities. The purpose is to create a focus on different ways to evaluate a consistent scoring team in MLB, and taking multiple approaches helps define that term. The first way to conceptualize run efficiency is through linear regression of the batting outcomes collected as independent variables to a dependent variable of runs scored per team. While split between postseason and non-postseason teams, modelling through regression analysis will help gain insight into the most influential variables that separate successful teams for the run scoring process. These results create a glimpse into what is defined as most significant towards creating runs across an entire year. By separating regression models on the postseason and non-postseason data, comparison of these variables justify what matters towards offensive production.

Fig 2



By compiling team batting outcomes, not only can playoff and non-playoff teams be compared but individual team success over the given amount of seasons. Figure 2 demonstrates how well teams over the past few seasons have consistently gathered hits while limiting how often they get out. This is the first way to evaluate how consistent a team is offensively, in the simplest terms. How good is a team at successfully getting hits that lead to run creation. This figure

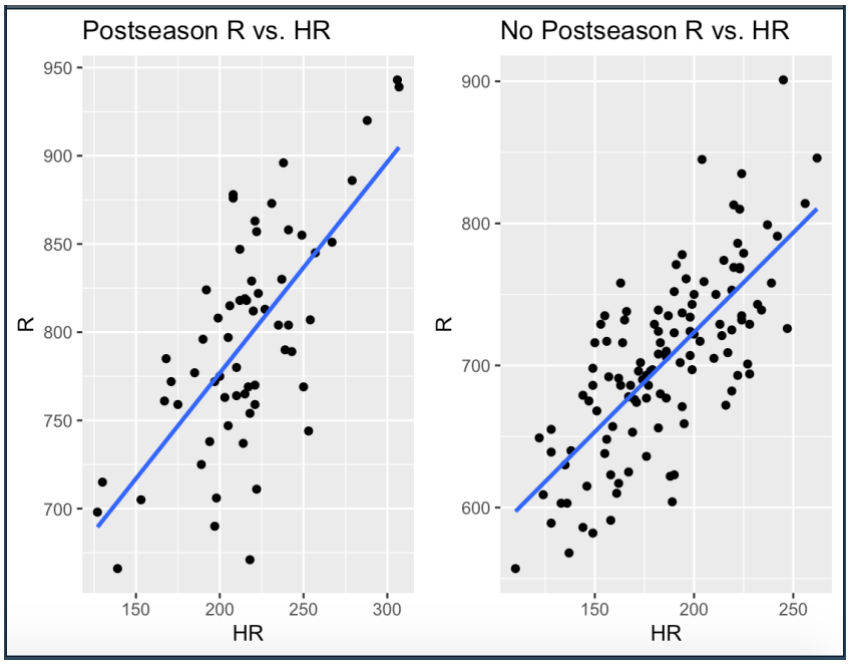
demonstrates what each teams general approach is to winning. For instance in these years in which the Houston Astros have dominated in MLB, the graph depicts just how efficient their offense has been. Teams that have been the most “hitting efficient” aim to be close to the bottom right of the graph, with the most hits and least amount of outs. Teams such as the Astros, Red Sox, and surprisingly the Nationals stick out as teams that in the past three years have been hitting efficiently. The graph also depicts teams that rely strongly on pitching for their success. For instance, the Milwaukee Brewers offensive consistency is below average with a large amount of outs given their low number of cumulative hits. Despite this, we know that they have been a winning regular season team, indicating a lot of their success stems from their pitching rotation. Finally, two teams that stand out as teams that have hit consistently over these seasons are the White Sox and the Colorado Rockies. Both these teams however have struggled to win over these seasons, showing that their problems are mainly with pitching as their hitting has been consistent over time.

Evaluating these metrics by team gives indication as to what clubs have stability in their offense as it relates to hitting efficiency. The goal is to take this insight to explain hitting consistency’s impact on creating runs to define run efficiency. This is done through multiple forms of modeling, the first being linear regression. Regression analysis allows an understanding of what hitting outcome variables are most influential in consistently scoring runs. The two different regression models focus on impact of various hit events on runs scored for data split by postseason appearance, and then with a dummy variable of Postseason. This variable indicates a 0 or 1 for if the team made the postseason that season or not, as shown in the formula

$$R \sim Walk/HitByPitch + FirstB + SecondB + ThirdB + HR + Out + Postseason$$

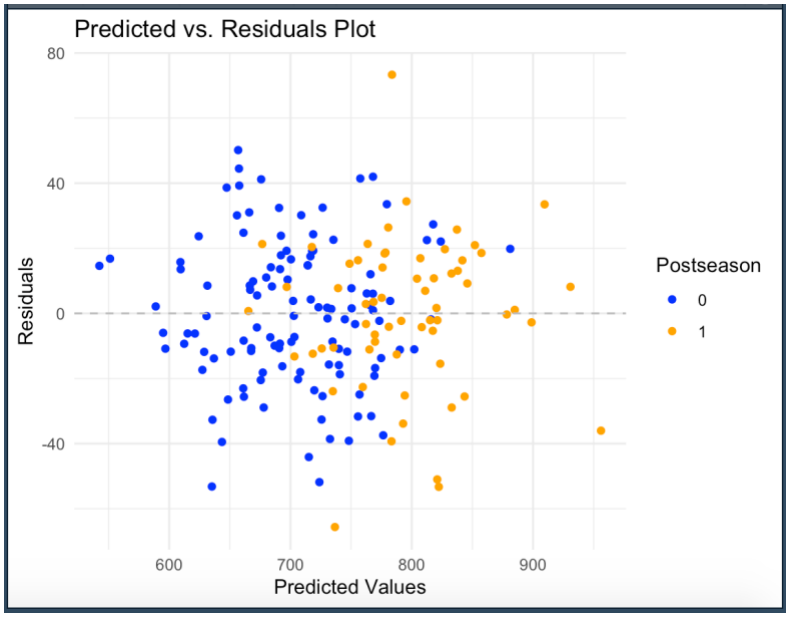
Through comparative analysis of the regression models that include and exclude the variable of Postseason, the most influential batting outcome variables on long-term run production can be identified. Using the model interpretation, predicted results can be formed to expand on these variables and their impact on consistently scoring runs. The below scatterplots demonstrate the influence of a significant batting outcome variable, HR, on runs scored for MLB teams depending on a postseason appearance.

Fig 3



As seen in the results to follow, one of the most significant predictors in evaluating stability in a team's run scoring process is with the HR. These scatterplots indicate a linear relationship for home runs on runs scored for both playoff and non playoff teams. The postseason teams have higher home run and runs scored totals but both types of teams have a linear relationship between the variables. The dependent variable of runs scored is linearly associated with the home run, meaning more home runs for a team tends to lead to more scoring efficiency. To go beyond the influence of just one independent variable, the regression models are used to create prediction outcomes on runs scored. Based on the model accuracy, a runs scored metric is predicted for teams based on postseason appearance. A fitted vs residual plot demonstrates the accuracy in the predicted runs scored model as seen in Figure 4.

Fig 4



This plot displays the residuals or the difference between the actual runs scored values and the predicted runs scored along with the predicted values. This visual determines that there is consistency in the variance of the residuals in relation to the predicted values. Therefore, the model and runs predicted model have homoscedasticity with the independent variables. Also, the plot has the dummy variable of if the team made the postseason by the color of each point in the scatterplot. The predicted values of runs scored tend to be greater for the postseason teams while the residual variance remains constant. These regression models help create an overview on run efficiency by defining scoring influence through batting outcomes. Using these linear models help to then create predictions on how to quantify consistent run production.

C. **Implementing Markov Chain Analysis**

An overview of run efficiency in MLB can be provided with the regression analysis, however a more complex approach can lead to further findings to define success of the run scoring process. Given the divide between teams based on the playoffs, two Markov chain models are used to represent the run scoring process through transitional states. Determining the different states in which successful teams score helps provide a more in depth explanation to run efficiency. The batting outcome events are turned into probabilities based off the expected amount of times that event is likely to occur. Assigning these probabilities to the motion of states allows for simulation as a process through the game. For example, the probability of transitioning from a 1,0 state(runner on first and nobody out) to 12,0(runner on first and second and nobody out) is the probability of a walk, hit by pitch, or a single. A 1,0 state to 3,0 is the probability of a triple occurring and a run scoring. All movement throughout the game can be defined in these states with third out absorption states that end the time step process of

simulation. Breaking down the game into these states allows for modelling to depict where the most influential moments are for teams that consistently score well. In order to do this, there are general assumptions made in the Markov chain simulation model to create reliable results that decrease certain factors into the probabilities. This includes the following:

- No stolen base attempts through states
- No fielder's choice or errors(fielding independent)
- Runners advance by force

Using these assumptions each base, out state has a probability that fits into a matrix form that reads as rows that are the current state to columns that are a possible state the game can transition to. This matrix form covers all the possible states that the game can go to and from and how likely these events occur. The figure below represents the run analysis model in matrix form.

Fig 5

	0,0	1,0	2,0	3,0	12,0	13,0	23,0	123, 0	0,1	1,1	2,1	3,1	12, 1	13, 1	23, 1
0, 0	0.0351148	0.24552794	0.04658081	0.0039279	0	0	0	0	0.66884854	0	0	0	0	0	0
1,0	0.0351148	0	0	0.0039279	0.24552794	0	0.04658081	0	0	0.66884854	0	0	0	0	0
2,0	0.0351148	0	0	0.0039279	0.24552794	0	0.04658081	0	0	0	0.66884854	0	0	0	0
3,0	0.0351148	0	0	0.0039279	0	0.24552794	0.04658081	0	0	0	0	0.66884854	0	0	0
12,0	0.0351148	0	0	0.0039279	0	0	0.04658081	0.24552794	0	0	0	0	0.66884854	0	0
13,0	0.0351148	0	0	0.0039279	0	0	0.04658081	0.24552794	0	0	0	0	0	0.66884854	0
23,0	0.0351148	0	0	0.0039279	0	0	0.04658081	0.24552794	0	0	0	0	0	0	0.66884854
123,0	0.0351148	0	0	0.0039279	0	0	0.04658081	0.24552794	0	0	0	0	0	0	0
0,1	0	0	0	0	0	0	0	0	0.0351148	0.24552794	0.04658081	0.0039279	0	0	0
1,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0.24552794	0	0.04658081
2,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0.24552794	0	0.04658081
3,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0	0.24552794	0.04658081
12,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0	0	0.04658081
13,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0	0	0.04658081
23,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0	0	0.04658081
123,1	0	0	0	0	0	0	0	0	0.0351148	0	0	0.0039279	0	0	0.04658081
0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
123,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0, 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
123,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The figure represents the transition states up to 23,1 or runners on second and third and one runner out. A lot of the original states to transition states in this glimpse of the matrix form have a probability of zero, because the game cannot go backwards in transition states that have a smaller number of outs. The rest of the probabilities begin the diagonal matrix that forms the Markov chain for simulation processes. The models are once again split by if the team made the playoffs in order to determine how much more often successful teams create opportunity in key situations. This modelling process is used to determine statistically where these key situations are relative to states of the game. In order for the model to interpret movement through the states as it relates to run production, a metric is used as a vector at each state of expected runs. This metric known as RE24, is a measured value that evaluates the expected number of runs at each state of the matrix form. This is combined to the transition states to interpret the production of runs at each state. By having an expected run total at each state,

implementing the transition probabilities helps calculate a new found expected runs metric to compare for teams.

RE24 = RE End State – RE Beginning State + Runs Scored

This formula depicts the average expected runs at each state as a metric to incorporate on the process of moving from states in the simulation. By matrix multiplication within the simulation, the probabilities of batting events occurring determines the new expected run total starting at certain given states. The models are presented by playoff teams and non-playoff teams, indicating where there is difference in the expected runs scored. More specifically, which exact game states do consistently successful playoff teams score in that put them over other teams. Determining where in the game these outcomes occur help approach defining the term of an efficient run scoring team.

III. Results & Predictive Analysis

Generating proper analysis through multiple modelling techniques was completed prior to Markov Chain simulation. Testing the process of simulating runs scored between postseason and non-postseason teams across multiple models helps gain an understanding of the run scoring process prior to Markov Chain. The first way this is implemented is in the linear regression results.

<u>Postseason Team Results</u>

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  205.23896  296.29026   0.693   0.4914
WalkHitPitch   0.29045    0.06293   4.616 2.39e-05 ***
FirstB         0.37596    0.06687   5.622 6.51e-07 ***
SecondB        0.79194    0.11743   6.744 9.90e-09 ***
ThirdB         1.07233    0.41934   2.557   0.0133 *
HR             1.31204    0.10007  13.111  < 2e-16 ***
Out           -0.11159    0.06867  -1.625   0.1099
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.04 on 55 degrees of freedom
Multiple R-squared:  0.8668,    Adjusted R-squared:  0.8522
F-statistic: 59.64 on 6 and 55 DF,  p-value: < 2.2e-16
```

Non-Postseason Results

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  227.56245  197.98198   1.149   0.2529
WalkHitPitch   0.28289    0.04063   6.963 2.47e-10 ***
FirstB         0.33238    0.03930   8.457 1.22e-13 ***
SecondB        0.83173    0.08681   9.581 3.32e-16 ***
ThirdB         0.50456    0.24197   2.085   0.0393 *
HR             1.29236    0.07429  17.396  < 2e-16 ***
Out           -0.10836    0.04460  -2.430   0.0167 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.49 on 111 degrees of freedom
Multiple R-squared:  0.8793,    Adjusted R-squared:  0.8728
F-statistic: 134.8 on 6 and 111 DF,  p-value: < 2.2e-16
```

Upon gathering the regression results, there are baseline quantitative takeaways to implement into an efficient run production offense. To begin, looking at both Adjusted R-squared values, it can be confirmed and validated that these modelling processes are accurate and a solid beginning representation of the run scoring process. This is because the adjusted R-squared values for both models is above .85, meaning over 85% of the variance in the data is represented in the dependent variables on the independent variable. In other words, what it takes to predict scoring runs is highly represented with the batting outcome variables included in the process. This makes sense as the simulation process for Markov Chain includes all of the batting outcome variables needed to produce and score runs. Implementing those variables into regression modelling allows for accurate representation of what impacts run scoring over time for teams that made and missed the playoffs. When split by postseason appearance or not, the most statistically significant part of the run scoring process for both types of teams was with the home run batting outcome. Both sets of teams had statistical significance of the highest level between 0 and 0.001 for walk or hit by pitch, single, double, and home run. These are all the most statistically significant batting outcome variables that are correlated to higher consistency in runs scored. The variable HR for both regression models is the most influential variable because of the coefficient values. For postseason teams, a one unit change in the amount of home runs attained results in an 1.31 increase on runs scored. Since this coefficient is higher for postseason teams, the regression modelling emphasizes that playoff teams in the past decade have focused on home runs more than non-playoff teams, and have benefited more off of this. Another result of the regressions is that the out outcome was slightly statistically significant for the non-postseason teams while not significant at all for postseason teams. The coefficient is negative for both, representing that a one unit increase in outs negatively impacts runs scored. However, the significance for the non-postseason teams

indicates that consistently getting out at a higher clip does matter if the offensive production is not up to par with some of the rest of the teams in the league.

Full Team Regression

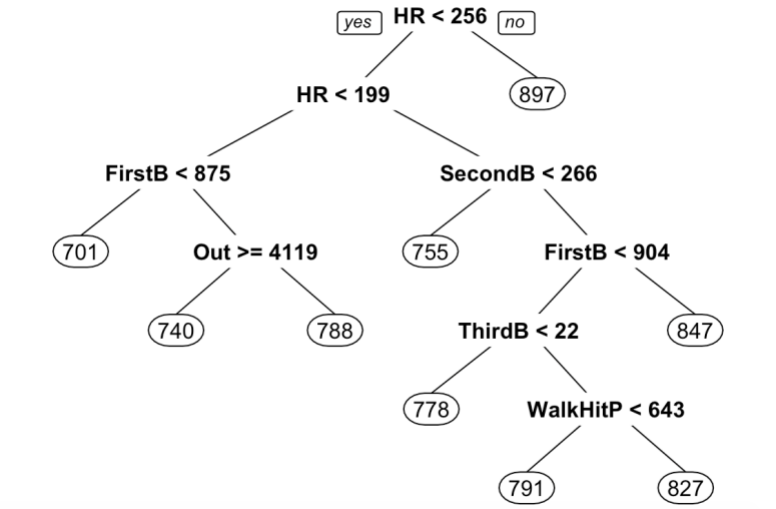
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  205.37388  160.34021   1.281  0.20197
WalkHitPitch   0.28980    0.03276   8.846 1.06e-15 ***
FirstB         0.34552    0.03341  10.341  < 2e-16 ***
SecondB        0.82682    0.06751  12.248  < 2e-16 ***
ThirdB         0.66583    0.20730   3.212  0.00157 **
HR             1.28969    0.05845  22.065  < 2e-16 ***
Out           -0.10729    0.03626  -2.959  0.00352 **
Postseason    14.05926    4.43237   3.172  0.00179 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.76 on 172 degrees of freedom
Multiple R-squared:  0.9155,    Adjusted R-squared:  0.912
F-statistic: 266.1 on 7 and 172 DF,  p-value: < 2.2e-16
```

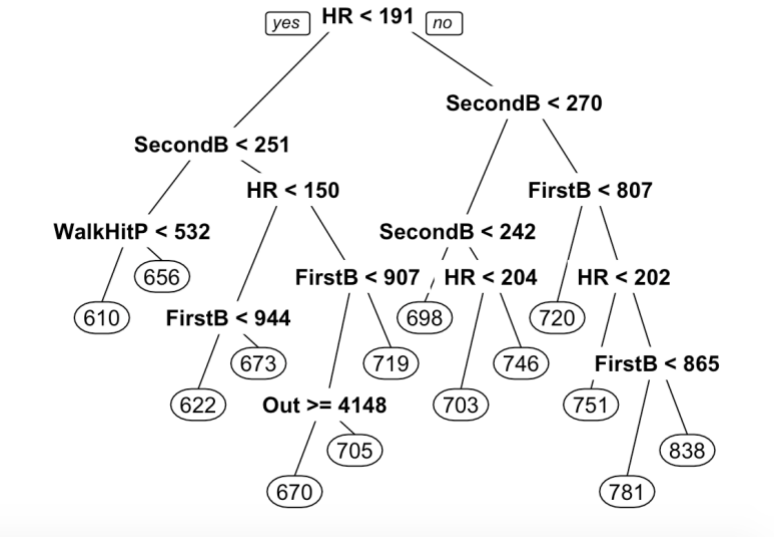
One more linear regression is ran this time with the combination of the postseason and non-postseason teams and the addition of the Postseason variable. This independent variable is considered a dummy variable taking, a in a value if the team is a postseason team. When running this regression the first difference is that the R-squared value is now above .9, indicating a very good prediction of the run scoring process. Secondly, all variables including triples now are very statistically significant, with the Postseason variable also being significant to the second degree(a p-value between .001 and .01). The coefficient of the Postseason variable indicates that a unit increase of making the postseason leads to a 14 unit increase in runs scored. This means that the difference between making the postseason or not in the last decade can come down to around fourteen runs. When this is put into perspective given the length of the season, the difference between being in and out in terms of the postseason as far as efficient run production can seem so slim.

The background provided by the regression results gives an in depth look into how the batting outcome variables are influencing the run scoring process that we are trying to analyze. The difference between making and missing the postseason can be small, and teams that find success in producing runs often do so with help by the long ball(homerun). Taking these results a step further into decision tree modelling helps pinpoint how to predict consistent scoring going further numerically. Once again the data is broke down at the batting outcome level by postseason appearance or not. Decision tree models are created to work off each other and see the difference in run scoring production for teams that are successful or not.

Postseason Decision Tree Results

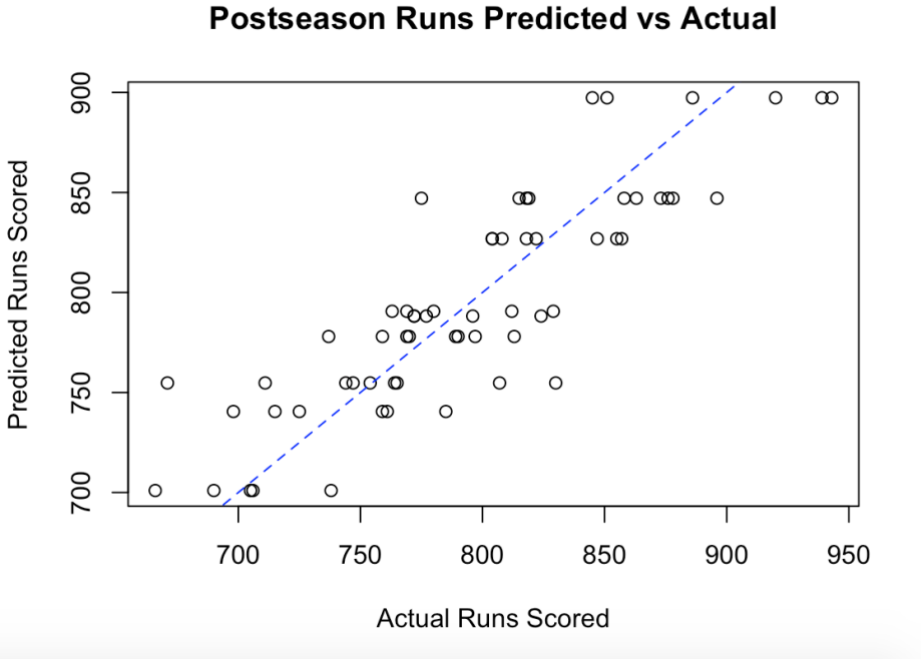


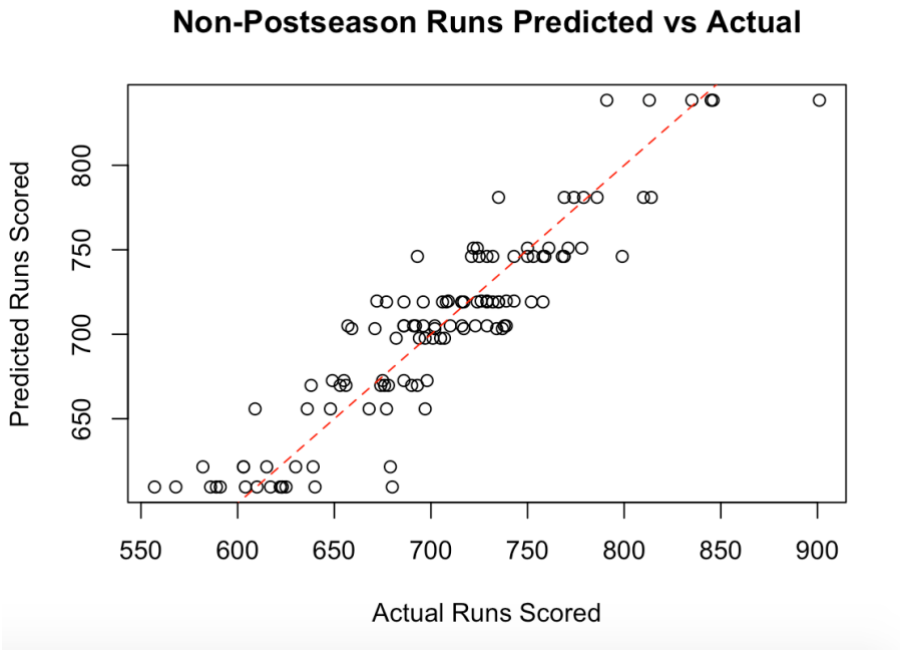
Non-Postseason Decision Tree Results



These decision tree results help quantify where the variance between postseason and non-postseason teams occurs in the form of splits. The breakoffs are a meaning to where producing for that variable matters on the evolution of scoring runs. For instance, the most significant predictor for scoring efficiently over time is with the home run. For both types of teams, in order to be successful the home run has to be a consistent piece of the offense, but in order to make the postseason it has to be done at a much higher rate. This is concluded by the first split of each decision tree, where having more or less than 256 home runs for playoff teams or 191

home runs for non-playoff teams is the biggest indicator of sufficient run production. Stated another way, a playoff team that is expected to hit over 256 home runs is likely to have 897 total runs scored on the season, the highest simulated run production. From there, the most important indication of run scoring for playoff teams is the amount of singles hit. While for non-playoff teams, the production of doubles matters the most following the home run. These results indicate that playoff teams that are scoring more efficiently are doing so with an emphasis on home runs and the hope that singles are also produced. The combination of singles leading to home runs is what greatly impacts elite run scoring. However, non-playoff teams see an emphasis towards getting into scoring situations with extra base hits as an important second indication. Following these two for both sets of teams are walks or hit by pitch. It is important to note that singles are valued more for the impact of run scoring in these models because often times a single can lead to a run scored from a player on base when a walk or hit by pitch might not. However in Markov Chain simulation from transitional states, the advancement into the base out state holds equal value for both of these outcomes. For both decision trees one of the strongest results gathered is that implementing an emphasis on the home run will substantially increase a team's offensive production. Over the stretch of a long season, having consistency in a team's ability to create home run outcomes strongly correlates to efficient scoring. Based on the decision tree models and the division of teams by postseason appearance, predictions can be made to quantify future run production. This indicates where teams need to be from a run production standpoint in order to compete for the playoffs going forward.





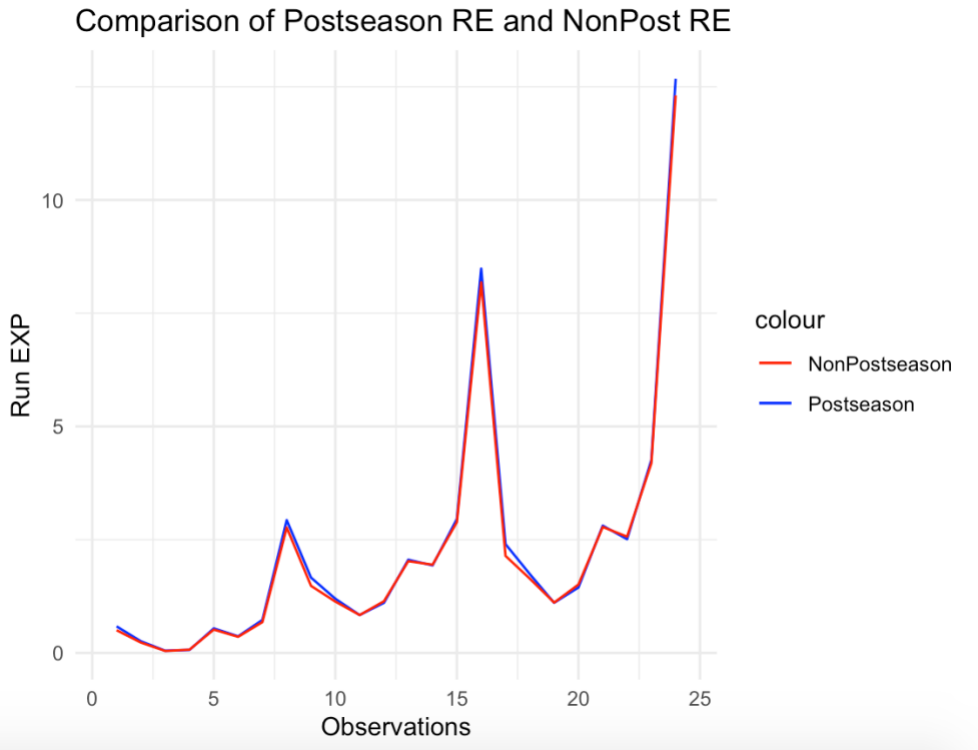
Reviewing both scatter plots, it is clear that the Non-Postseason predictions were a stronger indicator of actual runs scored due to the tight fit in the data. For a large portion of the data, the predictions allow an indication as to where teams will score in the future based off their current run production. The shift in predicted runs scored is compared to show that playoff teams are predicted to score at a higher rate than the non-playoff teams, but there is more variance in how efficient the predicted playoff teams can be. The spread in predicted values for the playoff teams show there are times when teams will both under perform and over perform previous run production. This event happened more for the postseason teams, but both have most of the predictions close to what was previously scored.

Both forms of modelling created insight into how both teams can statistically determine consistent scoring and what separates teams from having success making the playoffs. These results help shape an understanding of the batting outcome correlations to runs scored prior to simulation. Markov chain simulation is run through the transition states given the batting outcome probabilities. Matrix multiplication helps produce a simulated runs scored at a given starting state based on the movement in the chain and the possible transition and absorption states. The transition states correlate to a probability and expected runs scored, while the absorption states have expected run totals of zero since they are three out states. After simulating through the matrix from the start of different transition states, total runs expected can be accumulated to determine the most important states towards run scoring. Below is a summary of the findings given certain transition states and the expected runs for both playoff and non-playoff teams.

Markov Chain Expected Run Totals for Postseason vs Non-Postseason Teams

Scenario	Postseason RunsExp	Non-Postseason RunsExp
2 Out	0.690611833	0.644815709
Scoring Position 2 Out	0.780335243	0.738333695
1 out	2.528669438	2.45348435
Scoring position 1 Out	2.89531775	2.835999967
0 out	3.61855775	3.53045625
Scoring position 0 Out	4.134626167	4.077317333

This analysis demonstrates the ability for playoff teams to consistently have higher expected run totals for the state transitions that matter. Low out situations with runners in scoring position are the highest expected run totals, with playoff teams scoring at just a .06 higher margin. In total, postseason teams consistently make more out of the positions with runners in scoring position than non-playoff teams. However, the standard deviation of the postseason expected runs is slightly higher at 1.43 vs 1.42, meaning there is more variance in the expected runs for these teams.



The line chart runs through each transition state from the 2 out states to 0 out states and compares the expected runs between postseason and non-postseason teams. In most of the transition states, the expected runs created given the batting outcome possibilities are not highly influenced by a change in postseason appearance. Even if the probabilities are different regarding teams that make and miss the postseason and how likely they are for certain “better”

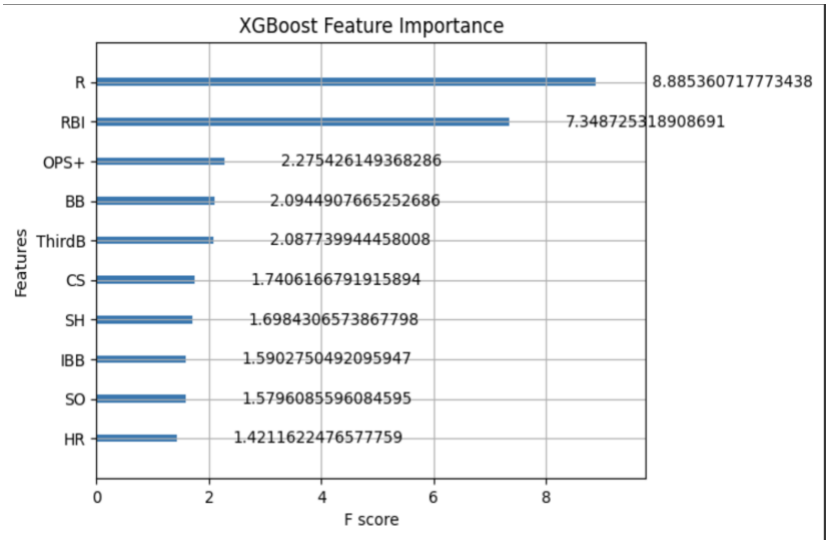
events to occur, the Markov chain simulation has non-postseason teams closely compared. As seen in the peak transition states which are runners in scoring position states(such as bases loaded) and less outs, the difference between expected runs of the two types of teams is the greatest. These are the state transitions in which efficient teams are making the most out of their potential scoring opportunities.

Predicting Postseason Team Makeup Through XGBoost

After defining and quantifying how playoff teams scoring efficiency differs from those who do not make it deep into October, the next step is to use machine learning applications to predict the build or makeup of a winning team. The teams are split once again by their ability to make the postseason in their seasonal year, and includes offensive statistics on top of the base stats that make up movement in the run scoring matrix.

The decision tree model calculates the importance of a specific split of the independent variables associated with a team's offensive success and their ability to make the postseason. Using information gain from the difference in entropy at varying levels of the independent variable statistics, the tree models the most influential variables on the run scoring process. The data of both teams is split into training and testing sets, for the model to learn about the differences that separate postseason teams from their opponents. From there, random forest models use numerous decision trees to build upon the accuracy of the predictions. XGBoosting takes random forest models and has each predictor forest build on the errors from the previous model. This form of lazy learning increases the effectiveness of our final model while randomly sampling the variables that consist of the offensive performance for each different forest.

Using XGBoost, some of the most important parameters to tune included the n estimators, or amount of random forest samples, the learning rate of each model on the previous errors, and the maximum depth of the trees in each forest. Runs scored and RBIs were kept in as separate variables on the binary dependent variable of postseason appearance in order to account for a level of luck between plays that result in a run but are not a run batted in.



The above figure indicates the feature importance between forests in the XGBoost model. In other words with each dependent variable evenly weighed as predictors in each decision tree that makes up the random forest models, the feature importance finds the most influential factors in postseason appearances. Runs and RBIs being at the top intuitively makes sense, with their difference as the significance of luck involved in the run scoring process as runs that scored that were not "batted in". In further research, a threshold could be created to put this into one variable to eliminate correlation as if the team experiences "luck" from a certain amount of runs created that were not from a following at bat, i.e an error, wild pitch, or a double pitch. The next highest influence on identifying playoff teams in the boosted model include OPS+, a weighted version to measure Slugging and On Base % of a team, and their ability to get on base via walks.

With the tuned boosted model, the team statistics hold significance on predicting the makeup of a postseason team. Comparing the model between postseason and non-postseason teams helps distinguish the difference in components of a successful team to predict for future teams. The following figure shows the results from the classification report of the boosted model.

XGBoost Postseason Prediction Model

Classification Report	Precision	Recall	F1-Score
0	0.75	0.82	0.78
1	0.67	0.57	0.62

With the classification target as a binary value of postseason appearance or not, the model can interpret a team's offensive identity and predict their ability to make the postseason or not given the performances of similar teams historically. The overall accuracy of the model is 72%, with the above report diving into these numbers using precision, recall, and F1-score, which weighs the precision and recall to account for uneven totals of identified cases, i.e False Negatives vs False Positives. These metrics determine that the model is much more efficient identifying non postseason teams, and struggles the most with false negatives where the model predicts a team will not make the postseason and it actually does. The most probable cause of this misinterpretation is due to close wild card-like teams that have similar stats to teams that just reached the threshold of making the playoffs. These teams that performed similarly above average but close to the borderline are what cause the most trouble for predictive purposes. Overall, this machine learning application allows for further analysis on the specific splits in offensive variables that help lead teams to making the postseason. The XGBoost model going forward could continue to develop off of more historical data to help determine if the build of a team by it's offensive capabilities have enough for the postseason. However, the farther back the team level data goes historically, the more necessary a weighted metric to compare offensive stats from different generations is.

Conclusion

Through various forms of modelling and simulation, the research question regarding run scoring production was statistically analyzed to create meaningful and valid results. Regression modelling demonstrated that the difference in certain batting outcome's influence on runs scored could be miniscule in relation to the length of the season. From there, a complex look into where quantitative results split demonstrated how many runs can be produced and predicted for teams with similar batting outcome totals. All of these results help to build simulation modelling regarding Markov Chain matrix movement. These conclusions helped specify where postseason teams are more efficient at producing runs when it matters the most. This was often in high stake situations with less outs so more opportunities to consistently put up runs. Overall, these results help create meaning behind persistent run scoring in MLB. Understanding how the run scoring process operates at its highest efficiency helps lead to predictive analysis for future teams to model their performance on.

Works Cited

Albert, J., & Hu, J. (2020, July 30). *Probability and Bayesian Modeling*. Retrieved April 15, 2023, from https://bayesball.github.io/BOOK/probability-a-measurement-of-uncertainty.html

Asaro , V. J. (2015). *Markov League Baseball: Baseball Analysis using Markov chains*. Cal Poly. Retrieved April 16, 2023, from https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1063&context=statsp

Beyder, E. (2015, June). *Simulation model using standardized lineup to evaluate player offensive player* . Retrieved April 15, 2023, from https://core.ac.uk/download/pdf/48501665.pdf

Bukiet, B. (n.d.). *A Markov Chain Approach to Baseball*. Retrieved April 14, 2023, from https://web.njit.edu/~bukiet/Papers/ball.pdf

Calestini, L. (2018, September 10). *The Elegance of Markov Chains in Baseball*. Medium. Retrieved April 15, 2023, from https://medium.com/sports-analytics/the-elegance-of-markov-chains-in-baseball-f0e8e02e7ac4

Cella, S. (n.d.). *Markov Chain Baseball*. Retrieved from https://www.lmtsd.org/cms/lib/PA01000427/Centricity/Domain/172/Markov%20Chain%20Baseball.pdf

Chernoff, P. (n.d.). *Sabermetrics - Statistical Modeling of Run Creation and Prevention in Baseball*. Florida International University Digital Commons. Retrieved April 15, 2023, from https://digitalcommons.fiu.edu/cgi/viewcontent.cgi?article=4852&context=etd

Engel, E. (2012, December 17). *Application of Monte Carlo Markov Chains to Batting Order Optimization*. Retrieved from https://www.math.wustl.edu/~feres/Math350Fall2012/Projects/mathproj03.pdf

GitHub. (2021). *DFS-with-R/coach: Lineup optimization for Daily Fantasy Sports*. DFS-With-R. Retrieved April 15, 2023, from https://github.com/dfs-with-r/coach

Harvard University. (n.d.). *Valuing Situations and Actions in Baseball*. Lecture. Retrieved April 14, 2023, from https://canvas.harvard.edu/files/11808242/download?download_frd=1

Henderson, K. (2016, May). *Baseball portfolio optimization* . Retrieved April 15, 2023, from https://scholarworks.uark.edu/cgi/viewcontent.cgi?article=1037&context=ineguht

How sabermetrics influence baseball batting order strategy. Sports Betting Dime. (2022, April 12). Retrieved April 14, 2023, from https://www.sportsbettingdime.com/guides/strategy/batting-order-sabermetrics/

Kalkman, S. (2012, October 9). *Optimizing your lineup by the book*. Beyond the Box Score. Retrieved April 15, 2023, from https://www.beyondtheboxscore.com/2009/3/17/795946/optimizing-your-lineup-by

MapleGridComputing. (2013, May 27). *Money ball with Maple: How to optimize the 2013 Blue Jays lineup so they can start winning*. MaplePrimes. Retrieved April 15, 2023, from https://www.mapleprimes.com/posts/147767-Money-Ball--With-Maple-How-To-Optimize

Marchi, M., Albert, J., & Baumer, B. (2019). 9- Simulations. In *Analyzing Baseball Data with R*. essay, CRC Press, Taylor & Francis Group.

McIntyre, J. (2016, March 8). *Linear optimization* . RPubs. Retrieved April 15, 2023, from https://rpubs.com/Koba/linear-opt-baseball

Polaski, T. (2013, February 12). *A Markov Chain Model for Run Production in Baseball* . Retrieved April 15, 2023, from http://faculty.winthrop.edu/polaskit/Spring13/Baseball.pdf

Saito, R. (2022). *Combinatorial and Markovian baseball lineup optimization*. Retrieved April 15, 2023, from https://designday.jhu.edu/wp-content/uploads/formidable/6/Baseball_Lineup_Optimization_Presentation.pdf

Schorsch, E. (2015, May). *Baseball Lineup Optimization* . Research Gate. Retrieved April 15, 2023, from https://www.researchgate.net/profile/Emanuel-Schorsch/publication/328529782_Baseball_Lineup_Optimization/links/5bd2a90b4585150b2b877139/Baseball-Lineup-Optimization.pdf

Sherman, N. (2012, October 12). *Optimizing Order: How to build the ideal lineup*. Bluebird Banter. Retrieved April 15, 2023, from https://www.bluebirdbanter.com/2012/10/12/3490578/lineup-optimization-part-1-of-2

Sokol, J. (2003). *A Robust Heuristic for Batting Order Optimization Under Uncertainty* . Georgia Institute of Technology. Retrieved April 15, 2023, from https://www2.isye.gatech.edu/~jsokol/boouu.pdf

Spedicato, G. (n.d.). *The Markovchain Package: A package for Wasily Handling SDscrete Markov Chains in R*. Cran. Retrieved April 15, 2023, from https://cran.r-project.org/web//packages//markovchain/vignettes/an_introduction_to_markovchain_package.pdf

Spencer, S. (2017, December 3). Run Expectancy Distributions of MLB Game States. Retrieved April 15, 2023, from https://ssp3nc3r.github.io/post/2017-12-03-run-expectancy-distributions/

Thaker, R. (2011, November 30). *An analysis of lineup optimization in baseball*. ininet.org. Retrieved April 15, 2023, from https://ininet.org/an-analysis-of-lineup-optimization-in-baseball.html

The Markov Chain Model of Baseball. Statshacker. (2018, July 26). Retrieved April 15, 2023, from http://statshacker.com/blog/2018/05/07/the-markov-chain-model-of-baseball/

Thomay, C. (2014). *Markov Chain Theory with Applications to Baseball*. Retrieved April 15, 2023, from https://openworks.wooster.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=6727&context=independentstudy

Ursin , D. (2014, December). *A Markov Model for Baseball with Applications* . Retrieved April 15, 2023, from https://dc.uwm.edu/cgi/viewcontent.cgi?article=1969&context=etd

Zamora, R. (2018, September 7). *Coach package for lineup optimization*. DFS with R . Retrieved April 15, 2023, from https://www.dfswithr.com/blog/coach-package-for-lineup-optimization/