CHRISTENDOM COLLEGE

THE MASSEY METHOD & MARCH MADNESS: AN APPLICATION OF LINEAR ALGEBRA TO SPORTS ANALYTICS

A THESIS SUBMITTED TO

DR. DOUGLAS J. DAILEY

IN CANDIDACY FOR THE DEGREE OF

BACHELOR OF ARTS

DEPARTMENT OF MATHEMATICS AND NATURAL SCIENCE

BY

MICHAEL TOWNSEND

FRONT ROYAL, VIRGINIA

APRIL 2025

To my math teachers over the years, who cultivated my passion for the subject.

"If you challenge conventional wisdom, you will find ways to do things much better than they are currently done."

–Michael Lewis, *Moneyball*

Table of Contents

Section 1 Introduction

In a world where computing power dominates the menial tasks of man, there emerges a compelling search for new objects of its statistical and analytical capabilities. Computer analysis is not only useful for carrying out unfathomably large calculations, but it provides the capability of processing large quantities of data. Among many popular sources of entertainment, sports stands out as an ideal object for computer analytics given the critical impact of points, statistics, and averages on the outcomes of particular players, teams, and even managers. The question arises: How can computer models be applied to statistics in sports? The answer has long been present. For years, companies such as ESPN, AWS, and IBM have been perfecting deeply sophisticated computer models that can assimilate data, recognize trends, and provide context known as sports analytics. In his article on quantitative methods applied to sports, Ted Hayduk refers to sports analytics as, "the use of quantitative methods by sports organizations to inform decision-making" [**6**, p. 12]. In this sense, the term sports analytics does not necessarily involve computer analysis, but computing power has now become essential in aggregating, storing, and analyzing massive amounts of data.

With respect to sports analytics, there are several methods of utilizing data to inform decision-making. Hayduk defines three forms of analytics in sports: descriptive, prescriptive, and predictive analytics. Descriptive analytics is a term used to describe the use of historical data in explaining characteristics of a variable or dataset [**6**, p. 12]. In other words, data is analyzed to provide context or insight into what has happened in the past. For example, exploring a team's roster and game data could discover a correlation between a significant stretch of losses in the season and the injury of key players on the team. Information gained from descriptive analytics is beneficial for connecting cause and effect of historical phenomena. Following in turn, prescriptive analytics is defined in relation to its goal: "to recommend the best course of action given a set of known alternatives and a set of constraints related to the problem or research question at hand" [**6**, p. 13]. Namely, prescriptive analytics provides context and insight into a present situation or decision by formulating a best course of action based on the data. In the context of an organization's decision to select a new draft pick, insights into each players stats, shooting percentage, and defensive efficiency are valuable in assessing player compatibility with the team. Poorly informed decisions are often bad decisions. In sports, prescriptive analytics supplies organizations, coaches, players, and even fans with the insights they need to make well-informed decisions. Finally, predictive analytics is a class of analytics that utilizes data to explain potential movements in a particular variable [**6**, p. 13]. Analyzing potential movements in a variable helps predict or forecast future outcomes and effects of the variable. For example, given a consistent shooting percentage in basketball, an increase in the number of three-point shots attempted per game by a team could increase their total points per game and ultimately lead to a projected increase in winning percentage. To stay ahead of the game, organizations need to innovate and develop effective ways of analyzing data and forecasting future events. These analytical categories support a diverse range of practical models.

One application of predictive analytics is bracketology, the tradition of attempting to predict outcomes of games in the NCAA college basketball tournament known as March Madness. Predictive models can be created that analyze team statistics in search for key indicators of team strength, which in turn can assist an accurate selection of winning teams. These models can range from simple winning percentage models to deeply complex multi-factor models that can combine several computer models for increased accuracy. A wide variety of models are available due to the wide variety of factors that can affect the game of basketball. One example of such a model is the Massey Method, a rating model that uses principles of linear algebra to generate team ratings based on teams' relative strengths, particularly the point differential of past matchups [**8**].

The purpose of this thesis is to provide a thorough explanation of the principles of linear algebra behind the Massey Method, detail a few modifications and applications of the model, and build a computer model applying these principles. In an explanation of the

Massey Method, the major topics of discussion are how ratings are calculated and what ratings represent. The influence of external factors on sports games can introduce weighting schemes to the model to improve the accuracy of ratings. Massey ratings can have a number of modifications, including an application to strength of schedule ratings. Modern companies including FiveThirtyEight and the Bowl Championship Series incorporate Massey ratings into their own computer models. This thesis culminates in the creation of a Python computer program that applies the principles of the Massey Method to 2024 NCAA men's basketball in search of the perfect March Madness bracket.

Section 2 The Massey Method

Introducing the Massey Method involves a thorough exploration of certain concepts from linear algebra. Kenneth Massey, in his undergraduate thesis at Bluefield College [**8**], introduced a technique for rating sports teams based on previous game statistics and point differential between teams. In this section, the following explanation of the Massey Method closely follows an outline of the method provided by Tim Chartier et al. in their book *Mathematics and Sports* [**2**, pp. 55-70]. In a series of matrix manipulations, Massey turned a linear system of game data into a matrix equation that yields team ratings based on cumulative point differential. Massey's own creative genius was in developing a "Massey Matrix" that can turn a matrix system with no solutions into solvable form by centering ratings around zero. Once ratings are calculated, they can be utilized in forecasting future matchup outcomes between teams. The process begins with an accurate and effective way of representing teams' previous matchup data: a system of linear equations.

All predictive models use previous data to forecast future results; Massey represented point differentials of games as equations and a whole schedule or season as a system of linear equations. For example, take the equation $r_1 - r_2 = 6$. Each r_i represents the particular rating for team i. The constant 6 represents the point differential resulting from a game between team 1 and team 2. Taken as a whole, the equation reads, *in a particular matchup between teams* 1 *and* 2*, team* 1 *beat team* 2 *by* 6 *points.* This equations represents one game, but there are usually many more games and teams playing each other in a season. Introducing a few more games into the system, suppose we have: $r_1 - r_2 = 6$, $r_2 - r_3 = 3$, $r_4 - r_3 = 2$, $r_4 - r_1 = 7$, $r_3 - r_1 = 1$. This system of equations forms a foundation for comparing various teams, especially in future matchups. We can convert this system of equations into a matrix equation of the form $M\vec{r} = \vec{d}$, where matrix M stores matchup data, vector $\vec{r}$ holds teams' ratings, and vector $\vec{d}$ contains point differentials [**2**, p. 61]. See System 2.1.

Each column of matrix M is associated with a particular team, and each row with a particular matchup. This is a simple example, but the method remains open to a vast

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 2 \\ 7 \\ 1 \end{bmatrix} \tag{2.1}$$

amount of games and teams.

The question remains: what exactly does a rating in this method signify? It seems clear from the previous representation that ratings are variables intrinsically related to each other by point differential. So naturally, solving the system of equations for these variables will yield a rating. The problem is that the system is unsolvable because transitivity does not hold [**2**, p. 60]. For example, $r_1 - r_2 = 6$ and $r_2 - r_3 = 3$. We would expect $r_1 - r_3 = 9$ to be true if transitivity holds. In other words, team 1 should beat team 3 by 9 points. But the new matchup $r_3 - r_1 = 1$ is introduced that violates transitivity and throws off the system. This unpredictability in sports leads us to conclude that an exact solution is impossible. But, obtaining an approximation is not out of the question. An explanation requires the introduction of the linear algebraic concept of least squares approximation.

The least squares method provides a way of obtaining an approximated solution to the linear system. Put simply, least squares regression finds a "line of best fit" among scattered or variant data, minimizing error by taking squares of the "residuals" of each equation [**2**, p. 62]. Since System 2.1 has no solutions, Chartier et al. state, "we will use the method of least squares to find the vector $\vec{r}$ such that the length of the vector $\vec{d} - M\vec{r}$, which is the residual error, is minimized" [**2**, p. 62]. To do so, consider the system $M^T M\vec{r} = M^T\vec{d}$.

$$\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \\ -4 \\ 9 \end{bmatrix} \tag{2.2}$$

In this new matrix system, the rating vector $\vec{r}$ remains the same and the vector $M^T\vec{d}$ becomes the sum of point differentials for the corresponding team. The matrix $M^T M$ compacts the game data into a 4×4 matrix where columns represent specific teams [**2**, p. 62]. Analyzing the matrix multiplication behind $M^T M$, the main diagonal is populated when a row is multiplied by its corresponding identical column. The result is a total count of games played by the corresponding team, since game data in corresponding rows and columns will align and multiply to produce 1 each time the particular team plays. These 1s are added up for each matchup to populate the main diagonal, resulting in total number of games played by the specific team. In all other positions of the matrix $M^T M$, non-zero components in row-column combos only align when the corresponding teams play each other. Since the matchup results in a winner and a loser, the product of aligning non-zero components must be -1 for each matchup. Thus, a -1 goes into position (i, j) provided teams i and j play each other. If they play a rematch, an additional -1 is added. However, the rank of an $n \times n$ matrix of this form is at most $n - 1$ due to the symmetry of matchups because $\vec{b} = (1, 1, 1, ..., 1)$ is always a null vector of the system [**2**, p. 62]. Note the following theorem.

Theorem 2.1 *The normal equations, $M^T M\vec{r} = M^T \vec{d}$, of a linear regression always have a solution. This solution is unique if and only if M has full column rank, or equivalently, M has no null vector.*

Since M has a null vector, System 2.2 does not have a unique solution. Our search for unique team ratings leads us to a key concept of matrix modification.

Massey introduces a series of matrix manipulations that ensures a unique solution to System 2.2. His signature solution is the Massey Matrix [**8**, pp. 33-34]. This matrix is formed by replacing a row of matrix $M^T M$ with all 1s and its corresponding row entry in $M^T \vec{d}$ with a 0. This centers all the ratings around zero, meaning $r_1 + r_2 + r_3 + r_4 = 0$.

$$\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \\ -4 \\ 0 \end{bmatrix} \tag{2.3}$$

At first glance, this alteration seems to delete matchup data from the system. However, the symmetry of data ensures that no matchups are lost and ratings are only affected proportionally. In System 2.2, matrix $M^T M$ has rank 3. The substituted row of all 1s in System 2.3 cannot be a linear combination of the other rows of the matrix since they add up to 0. Therefore, matrix $M^T M$ of System 2.3 has rank 4, and thus, a unique solution of team ratings as outputs. For this example, team rating estimates are $r_1 = -0.5$, $r_2 = -2.25$, $r_3 = -1$, and $r_4 = 3.75$. In other words, team 4 would be expected to beat team 1 by $3.75 - (-0.5) = 4.25$ points.

To recap, we will trace the representation of a rating throughout this process. In System 2.1, a particular rating is a number whose only restriction is its numerical difference from other ratings, namely point differential. In reality, the numbers chosen are arbitrary since it is only the distance between them that matters. Since the system violates transitivity, it yields no solution. System 2.2 is created to solve the transitivity problem [**2**, p. 62]. It essentially aggregates a team's matchups to produce one equation that can approximate a solution or "line of best fit." In System 2.2, a particular rating becomes a summation of individual matchup ratings, creating an approximation instead of an exact value or prediction. System 2.3 is created to solve the unique solution problem [**2**, p. 62]. Ratings still represent approximations dependent on total point differentials, but the only difference is that they are now centered around zero. Difference in numerical length between ratings remains proportional, but now we obtain a unique solution.

On observing the method's base principles, it is clear that the Massey Method provides certain advantages in its ratings. Many other statistical models for sports are simply based on team record and win/loss data. While these models may produce accurate estimations, they cannot directly account for point differential, which is inherent to the Massey Method. Due to its reliance on point differential to output ratings, the Massey Method inherently accounts for ties in its model and no external adjustments are necessary [**7**, p. 137]. A game resulting in a tie can be represented by a zero point differential in the Massey Method. Honing in on the original Massey matrix system, when a tie game is added to the data, cumulative point differential remains the same and an additional game is added to the team's total count and matchup count. Essentially, the addition of a tie game does not affect the overall ratings of either team. On the topic of handling ties, Langville and Meyer claim, "the above Massey method naturally incorporates ties. For a tied event between teams i and j, the m_{ij} and m_{ji} entries increment by one, yet the information in the point differential vector $\mathbf{p}$ remains unchanged" [**7**, p. 137]. As a result of this, ignoring ties or adjusting the model to incorporate them is unnecessary, giving the Massey Method a natural versatility.

In its current form, however, the Massey Method still has limitations. Blowout wins may disproportionately inflate the ratings of teams based on their significant point differential. These situations can have a wide variety of causes. The question of what "significant point differential" means is unclear. In other words, what is considered a blowout victory? In his Great Courses lecture series, Tim Chartier introduces the idea of capping off blowouts to prevent inflation of team ratings [**1**, p. 264]. For example, say we capped off any blowout wins at 25 points. This means that any win by more than 25 points would be counted in our rating system as a 25 point win. The challenge is that blowout wins may or may not be an indication of relative strength between teams. The winning team may truly be 25+ points better than the other, but there can be many external factors that cause the exaggerated scoreline. A term known as "garbage time" refers to the last few minutes of a basketball game when a team holds a substantial lead and makes substitutions to their less experienced players, often decreasing their intensity of play. In this situation, the scoreline often does not reflect the teams' relative strengths. Overall, a well-defined cap for blowout victories can provide immense advantages to the accuracy of rating models.

A countless number of factors remain. Some teams may have easier schedules. Do home court advantage and time-in-season have an effect on game outcomes? These are all useful components in perfecting the accuracy of a model in order to more effectively estimate and predict outcomes. One benefit of the Massey Method is that it remains easily customizable for including these various external factors in how the model weights particular games.

Section 3 Weighting the Model

Due to its unique game data representation, there are a variety of weighting schemes that can be applied to the Massey Method. A weighting scheme is a systematic and well-defined method of determining how much a game should affect a team's overall rating based on certain internal or external factors. The unweighted Massey Method assumes all games are of equal importance in determining a rating. However, what if a team has players sustaining injuries? Does a team have a better chance of winning at home? Are late season games more indicative of a team's true strength? Weighting schemes can range from simple to deeply complex, taking into account hot streaks, team rosters, or injury reports. But, just because a weighting system is complex does not necessarily mean it is advantageous or more accurate. It is true that external factors affect games, but it is difficult to predict and quantify these effects. So, a finely-tuned and precisely weighted model can have immense advantage in rating accuracy, but an imprecise or faulty weighting system often leads to inaccurate results. Given this need for precision, an evaluation of possible weighting factors is crucial.

We begin with a concept known as temporalized weighting. This section primarily references an outline of weighting schemes for the Massey Method by Amy Langville and Carl Meyer in their book *Who's #1?: The Science of Rating and Ranking* [**7**, pp. 147-150]. A temporalized weighting system weights games differently based on what time in the season the game occurs. In its base form, all games throughout the season are equally important in the Massey Method. Any modification requires a precise intuition of the sport and a well-defined function to calculate our method of weighting. Langville and Meyer present a few common functional representations of temporalized weighting. See Figure 1.

Games in a season can be represented using a linear weighting where the weight of games increases at a constant rate as time goes on [**7**, p. 148]. This way, games earlier in the season are weighted less heavily, and progressively grow more important throughout the season. In Figure 1, the physical representation of the model is a linear equation intersecting the origin.

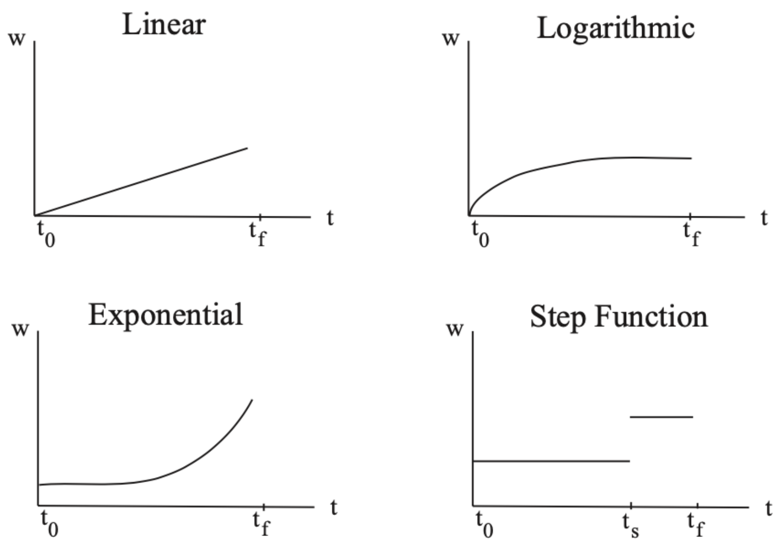

Figure 1: Four Basic Weighting Functions [**7**, p. 148].

The line is described by the equation,

$$f(t) = \frac{t - t_0}{t_f - t_0},$$

where t_0 is the start of the season, t is the time-in-season of the game, and t_f is the end of the season. Thus, $f(t)$ is the fraction of time elapsed over total time in the season. This method heavily emphasizes a constant rate of change, which proves useful to people who consider games to get steadily more important as the season goes on.

Another temporalized method is an exponential weighting scheme [**7**, p. 148]. Here, the weight of games increases at an exponential rate, indicating a sharper increase in weighting toward the end of the season. This method subscribes to the theory that recent games are more indicative of a team's true rating and strength. In Figure 1, this weighting scheme is depicted using an increasing exponential curve defined by the equation,

$$f(t) = e^{\frac{t - t_0}{t_f - t_0}},$$

where the exponent is increasing and variable between 0 and 1. This places a much higher emphasis on games much later in the season in comparison to the constant increase of a linear function. The exponential model provides valuable insight into a team's late season performance, which can prove useful in determining a team's post-season potential.

A logarithmic function provides a method of weighting games where beginning-of-season games are significantly less important than games in the rest of the season [**7**, p. 148]. The model attempts to weight early season games less heavily to improve accuracy. In Figure 1, the model is represented by a logarithmic curve defined by the equation,

$$f(t) = \log\left(\frac{t - t_0}{t_f - t_0} + 1\right).$$

The shift by an increment of 1 ensures that the weighting factor is never zero, which would completely disregard a game. This model increases accuracy if all teams start off slow, but remains unclear since this might not universally be the case for all teams.

If we desire a model that progresses weighting factors by constant increments, a step-function weighting proves advantageous [**7**, pp. 148-149]. In this method, weighting can be shifted at particular points in the season by defining a piecewise function. Here, specific stretches of the season can be defined as more important, providing a significant amount of flexibility to the model. In Figure 1, the physical representation of this system is an incremental step-function defined by the equation,

$$f(t) = \begin{cases} 1 & \text{if matchup occurs before } t_s \\ 2 & \text{otherwise,} \end{cases}$$

where any game after t_s is weighted more heavily. Different step-functions can also be defined to partition the season even further, such as bi-weekly or tri-weekly step-functions. This scheme provides flexibility for modelers to customize weights throughout the season based on perceived external factors.

Now that the temporalized weighting functions have been defined, we need to find a way to incorporate them into the Massey Method. Langville and Meyer recommend solving a "weighted least squares" problem. To do this, we solve the equation,

$$M^T W M \vec{r} = M^T W \vec{d}, \tag{3.1}$$

where W is a diagonal matrix formed from a vector $\vec{w}$ containing weights for each game [**7**, pp. 149-150]. The previous equation ensures that weights are applied to individual game rows in the original matrix before taking the transpose and producing a rating. Since time-in-season is a fairly noticeable factor in game outcomes, it seems natural to formulate a weighting scheme to improve the accuracy of the Massey Method.

Returning to our previous example in Section 2, suppose game $r_1 - r_2 = 6$ is played on the first day of the season, games $r_2 - r_3 = 3$ and $r_4 - r_3 = 2$ occur at the midway point, and games $r_4 - r_1 = 7$ and $r_3 - r_1 = 1$ are played on the final day. Following a linear weighting scheme, a weighted matrix W is represented as follows,

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where the main diagonal of matrix W is populated by values of the function $f(t) = \dfrac{t - t_0}{t_f - t_0}$ evaluated at time-in-season the game occurred. Substituting W into Equation 3.1, the following matrix system is formed,

$$\begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 0.5 & -0.5 & 0 \\ -1 & -0.5 & 2 & -0.5 \\ -1 & 0 & -0.5 & 1.5 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -8 \\ 1.5 \\ -1.5 \\ 8 \end{bmatrix}. \tag{3.2}$$

On initial observation, System 3.2 looks significantly different from System 2.2. This is due to the fact that each game's contribution to the matrix $M^T W M$ is scaled by its corresponding value on the diagonal of W. Since game $r_1 - r_2 = 6$ occurred at the start of the season, it holds a weight of 0 and has no contribution to the weighted matrix $M^T W M$. As outlined in Section 2, all the proceeding steps of Massey's process can now be applied to System 3.2 to generate weighted team ratings. The resulting ratings are as follows: $r_1 = -3.25$, $r_2 = 1.75$, $r_3 = -1.25$, and $r_4 = 2.75$. The effect of the linear weighting scheme on team ratings is noticeable in comparison to the unweighted ratings in Section 2. In the context of linear weighting, team 1's decrease in rating from $r_1 = -0.5$ makes sense given that its one win occurs at the beginning of the season and its two losses occur on the final day. Thus, team 1's losses heavily outweigh its win, resulting in a lower rating. Team 2's increase in rating from $r_2 = -2.25$ is due to its one loss occurring at the beginning of the season.

Applying an exponential weighting method serves to further illustrate the affect of weighting schemes on ratings. Given each game's time-in-season as defined in the above example, an exponentially weighted matrix W is approximated in the following manner,

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.65 & 0 & 0 & 0 \\ 0 & 0 & 1.65 & 0 & 0 \\ 0 & 0 & 0 & 2.72 & 0 \\ 0 & 0 & 0 & 0 & 2.72 \end{bmatrix},$$

where components of matrix W are populated by the values of the function $f(t) = e^{\frac{t-t_0}{t_f-t_0}}$. Substituting W into Equation 3.1 yields the following matrix system,

$$\begin{bmatrix} 6.44 & -1 & -2.72 & -2.72 \\ -1 & 2.65 & -1.65 & 0 \\ -2.72 & -1.65 & 6.02 & -1.65 \\ -2.72 & 0 & -1.65 & 4.37 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -15.76 \\ -1.05 \\ -5.53 \\ 22.34 \end{bmatrix}.$$

Calculating the ratings, we obtain $r_1 = -1.44$, $r_2 = -1.51$, $r_3 = -0.92$, and $r_4 = 3.87$.

One external factor that comes into play is which team is playing at home for a particular matchup. This factor is known as home field advantage. Teams can play better at home for multiple reasons, including fan support, weather conditions, and facilities. These specific factors are out of the scope of this thesis, but a certain home field advantage rating can be incorporated into the Massey Method. Massey defines an additional variable to account for a home rating in his model [**8**, p. 40],

$$d = r_a - r_b + x_h r_h.$$

Massey makes clear that this home field advantage rating is constant for all teams, as "an assumption is made that the home team in any game benefits by a fixed number of points" [**8**, p. 40]. In the previous equation, r_h represents the home field rating and x_h indicates the game's location. For example $x_h = 1$, indicates the winning team is the host, while $x_h = -1$ indicates the winning team is the visitor. This method introduces a fixed variable that is inherently tied to the winning team and the matchup's location. When ratings are calculated according to the Massey Method, a home advantage rating is produced that represents a team's advantage when playing at home in the season. This home field advantage rating can then be used to properly weight home and away games in the model, following the steps of incorporating weights outlined previously in this section.

Section 4 Applying the Model

Besides the subject of weighting the model, there are several worthwhile topics to consider in a discussion about the Massey Method. Strength of schedule is very influential in accounting for a team's success in a season. However, determining the strength of a team's schedule requires a previous knowledge of the relative strengths of teams. Since the Massey Method depends on point differentials from previous matchup data, we must wait for games to be completed and ratings calculated before determining strength of schedule. Strength of schedule cannot realistically be incorporated into a weighting scheme to determine a team's rating because it relies on already knowing the relative strengths of teams, i.e. their ratings. Once initial ratings are calculated in the Massey Method, however, a strength of schedule rating can be produced to numerically represent the relative difficulty of a team's schedule. This section closely follows an article written by Massey titled "Massey Schedule Ratings" found on his ratings webpage [**9**].

Initially, one method of calculating the strength of schedule for a team would be to average the ratings of all the opposing teams it played throughout the season. However, Massey points out that strength of schedule is relative to the strength of the particular team in question [**9**]. For example, the same schedule that is relatively easy for a strong team might prove to be difficult for a weaker team. In response to this problem of relativity, Massey outlines 5 categories of teams and produces the following table of their ratings and probabilities of beating each other:

		Probability of Beating				
Team	Rating	Great	Good	Average	Bad	Pathetic
Great	20	50	75	90	97	100
Good	10	25	50	75	90	97
Average	0	10	25	50	75	90
Bad	-10	3	10	25	50	75
Pathetic	-20	0	3	10	25	50

Figure 2: Massey Categories and Probabilities [**9**].

This table categorizes specific teams based on their rating and provides probabilities of beating other teams of different categories. For example, a great team would have a 75% chance of beating a good team. To draw out his point on the relativity of schedule strengths, Massey outlines two distinct schedules as follows [**9**],

Schedule A: (Good, Good, Good, Average)

Schedule B: (Great, Great, Good, Pathetic)

In comparison, both schedules have the same average rating of opponents (7.5). So, a simple averaging-model of evaluating strength of schedule would conclude that these schedules are equally difficult. However, after applying the assumptions of Figure 2, Massey calculates the relative expected wins against the prospective schedule for each category of team.

	Expected Wins Against	
	Team A's Schedule	Team B's Schedule
Great	3.15	2.75
Good	2.25	1.97
Average	1.25	1.35
Bad	0.55	0.91
Pathetic	0.19	0.53

Figure 3: Expected Wins by Schedule [**9**].

If strength of schedule is defined by expected wins, the conclusion must be that the schedules vary in difficulty based on which category of team it belongs to. Massey provides the following principle in his observation of the data, “An above average team should prefer to play a less distributed schedule, while a below average team should prefer to play a more distributed schedule” [**9**]. In other words, weak teams have a better chance of winning if they play a more “distributed” schedule, namely, a schedule including more weak teams where there is more variance in team category. Strong teams have a better chance of winning if they play a less “distributed” schedule, namely, a schedule including fewer strong teams where there is less variance in team category. All of this serves to emphasize the point that schedule strength is dependent and relative to the quality of the team the schedule belongs to.

Now that the relativity of schedule strength is established, how do we calculate a rating? Massey defines the following function,

$$n * \mathrm{EW}(X) = \mathrm{EW}(\text{actual schedule played}),$$

where the input is the actual schedule played and the output is a single team that would result in an equivalent number of expected wins if continuously played throughout the whole season. In the equation, n represents the number of games played in the season, $EW()$ is expected wins of playing the contained schedule calculated from Figure 2, and X represents the team whose rating would produce an equivalent number of expected wins as the actual schedule. In essence, this function seeks to create a single value or rating that encapsulates a schedule's strength. It does so by first finding expected wins of a schedule, then finding the average expected win per game, and finally, finding the rating of the opposing team that would produce that many expected wins on average. The result is a strength of schedule rating that takes into account relative team categories based on the classifications set forward in Figure 2.

Strength of schedule ratings provide value in their predictive capabilities. In a matchup between teams with similar Massey ratings, the team possessing the stronger schedule during the season often has the advantage. Thus, a strength of schedule rating can provide a strong indication of future matchup outcomes. This insight proves useful in the context of bracketology and March Madness.

Section 5 Modern Applications of the Massey Method

Technological advances in computing power have led sports analytics companies to rely on computer-assimilated models in formulating insights and predictions. As fans, teams, and managers begin to realize the impact of data and analytics on sports, demand for analytical services has grown. In modern times, there is a clear emergence of computer models to meet this demand. Particularly in the realm of sports, companies like ESPN, FiveThirtyEight, and even AWS are industry-leading organizations in the realm of sports analytics. These companies are continually perfecting deeply complex analytical models that can assimilate a range of both human and computer models. Sports ranking models produced by the Bowl Championship Series and FiveThirtyEight include applications of the Massey Method in their combined ratings for teams. Incorporation of a wide variety of models improves the accuracy of team ratings, but there is creative freedom in which models are included.

Computer rating models are incorporated into many team ranking methods for sports leagues, including NCAA college football. The Bowl Championship Series (BCS), active from 1998-2013, was a post-season selection system that nominated teams to compete in bowl games [**4**]. A nationwide selection system that had postseason implications required precision and accuracy. As such, the BCS factored the results of several models and polls into its rankings. In an article about the BCS selection process, Brad Edwards of ESPN described the rankings as, "an intricate combination of two polls, eight computer ratings, schedule strength and number of games lost" [**4**]. Among the computer rating methods factored into the system were mentioned the Massey Method, New York Times model, Sagarin Method, and several others [**4**]. The Massey Method's unique dependence on point differential provided a valuable perspective on relative team strength when incorporated into the BCS system.

The average of multiple computer models introduced a variation of factors into the system to improve accuracy. Human voting and polls were also included in the system, reinforcing the diversification of the model. Another major factor in the BCS ranking system was schedule strength [**4**], where teams' ratings were affected by the difficulty of their schedule.

A team with good results against a strong schedule maintained a better ranking than one with good results against a weaker schedule, taking into account the relative strength of teams previously discussed in Section 4. Applying this principle provided an additional indicator of relative team strength to the BCS selection system. The incorporation of the Massey Method and strength of schedule into BCS rankings is evidence of their practical application to the world of sports today.

For the NCAA women's basketball tournament, FiveThirtyEight has a power rating model that takes into account pre-tournament ratings and in-tournament bonuses. The following figure provides a process map of the rating system.

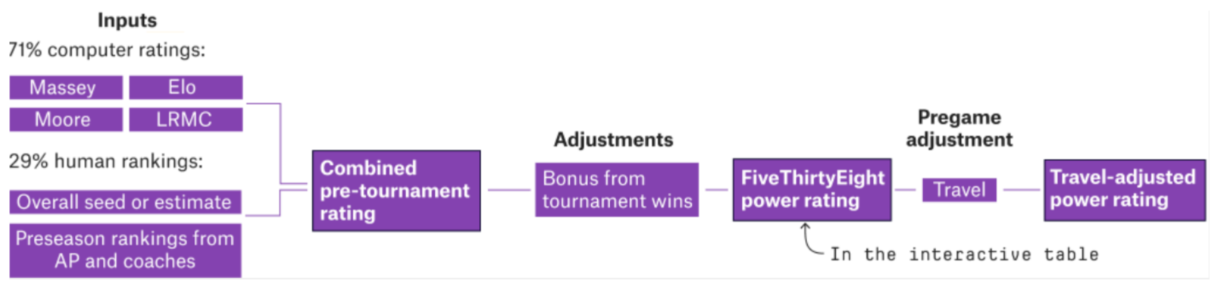

Figure 4: Women's Team Ratings [**5**].

This system is an example of a more complex model that incorporates human and computer rankings. It also contains additional adjustments and updates as the tournament progresses. The Massey Method comprises roughly 18% of a team's combined pre-tournament rating [**5**]. Pre-tournament ratings are a combination of computer models such as the Massey, Elo, Moore, and LRMC methods, as well as human ratings including seeding and pre-season polls. The additional factors of travel and tournament wins serve as adjustments to overall team power rating, with the inclusion of FiveThirtyEight's signature power rating [**5**]. FiveThirtyEight's incorporation of the Massey Method establishes the Massey Method as a prominent model in modern analytics. Pre-tournament ratings can be a major indicator of matchup outcomes in the tournament. We now proceed to build a rating model that utilizes the principles of the Massey Method to gain insight into the 2024 Men's March Madness Tournament.

Section 6 Building a Model

With the principles of the Massey Method firmly in place, our focus shifts toward formulating a computer representation of the model. For our purposes, Python serves as a versatile object-oriented programming language with immense data-processing capabilities. Python can import data via CSV or API, load that data into a dataframe, populate matrices, and perform matrix operations. In other words, Python has the computing power necessary to carry out the processes detailed in the Massey Method. The key is identifying what Python libraries provide these functionalities. Designing a computer program involves step-by-step instructions that must be carried out, each with a particular goal in mind. The details of designing a computer program for the Massey Method that generates ratings for NCAA men's basketball are outlined as follows.

The first task is finding and importing the data. The Massey Method requires matchup entries with final scores and team names for every game in the season. To calculate temporalized weighting, dates for each game must also be included. For NCAA men's basketball, SportsData.IO is a highly reputed and reliable source of sports data for both commercial and personal use [**11**]. For purposes of the model, a CSV file of game and team data for the 2023-24 season is downloaded to the computer and placed in an easily accessible folder. Utilizing the pandas library, the CSV file is imported into the program as a pandas dataframe. Now that the data is in the program as a dataframe, it can be filtered and cleaned to include only necessary information for the model.

The next segment of the program is directed toward manipulating the dataframe to include only data necessary for the model. This is accomplished by a series of filters applied to the data, which specify that games must be in the regular season and must be completed, as opposed to preseason, postseason, or canceled games. New team ID's are assigned to each team to provide a standardized process of identifying teams that ensures there are no gaps in game information. Once the dataframe has been filtered and modified, the data is readily available for use.

Utilizing game data from the 2023-24 season, the matrix system $M\vec{r} = \vec{d}$ needs to be populated with matchup information according to Massey's method. The necessary dimensions of the matrices are calculated by a count of the number of teams and the number of games in the season. Blank matrices with the corresponding dimensions are created using the numpy library. Components of the matrices are properly populated by looping through the dataframe and obtaining the corresponding pieces of information, such as final scores and teams in the matchup, by using a conditional clause. Reference Section 2 and System 2.1 for what information populates this system.

Once the appropriate matrices are formed, the numpy library provides the capability of performing matrix operations. Because the system $M\vec{r} = \vec{d}$ has no solutions, the program must calculate and solve $M^TM\vec{r} = M^T\vec{d}$, where the last row of matrix M^TM is replaced by a row of 1's and the corresponding entry of $M^T\vec{d}$ replaced by a 0 using a direct call for numpy arrays. This is Massey's method that ensures one solution to the matrix system. After using numpy to solve for $\vec{r}$, the rating vector $\vec{r}$ is populated with ratings for each corresponding team. Now that the desired output is attained, it needs to be presented visually.

The last segment of code provides school name and team rating for each team in NCAA men's basketball. By way of team ID's and a series of dictionaries, the school name of each team is recovered through the original dataframe. School-rating pairs are then alphabetized via dictionary and printed to the console using a Python loop, yielding the final result. This model serves as a foundation that can be built upon with multiple weighting schemes. The Python code used to build the model can be found in the Appendix. Once calculated, team ratings in the Massey Method provide many practical capabilities in the world of sports analytics.

Section 7 Conclusion

The Massey Method provides a system to gauge relative strengths of teams. At its core, it focuses on providing a numerical representation of past performance or data, strictly characterizing it as a descriptive analytical method. However, introducing the objective of predicting future outcomes of games based on ratings gives the Massey Method clear potential for predictive analytics. This is also evidenced by its direct connection to bracketology, an application of predictive analytics. The ratings generated by the model can also be incorporated into a ranking system for teams in a league, since their inherent representation is the relative strength of teams. In this thesis, however, Massey ratings are created for the purpose of their predictive capabilities and comparison with 2024 NCAA Men's March Madness results.

After careful calculation and research, the results are in. March Madness brackets were formed for each weighting method purely based on the previously calculated Massey ratings. By rule, matchup selections are dictated by which team has the higher rating [**3**, p. 68]. According to the standards of ESPN's bracket challenge, total amassed points for each type of bracket are as follows: unweighted (770), exponential (920), logarithmic (1210), linear (1220), and bi-weekly step (1220). When compared to all brackets submitted to ESPN's Men's Tournament Challenge in 2024, the bi-weekly step and linear brackets both ranked in the 93rd percentile with 1220 total accumulated points. The 50th percentile for March Madness brackets submitted to the challenge landed at 670 points. For reference, the number one bracket accumulated 1810 total points. If the bi-weekly step and linear brackets were submitted to the Christendom College bracket challenge in 2024, they would have come in 6th place out of 83 total brackets. In comparison, the 1st place bracket amassed 1440 total points. As seen by the results, the Massey Method produces ratings that have clear predictive capabilities, especially in the case of bi-weekly step and linear weighting schemes for 2024 March Madness. But, complete certainty is not guaranteed. Sports analytics can come close to predicting real-world results, but an inherent unpredictability of sports is always present. Of course, this is what keeps fans on the edge of their seats.

Appendix

The following appendix provides a Python coding script that generates weighted and unweighted Massey ratings for 2023-24 NCAA men's basketball teams.

```
#import necessary packages
import numpy as np
import pandas as pd
import datetime as datetime
from itertools import islice
import math

#read CSV files into the program
game_data = pd.read_csv('Game.2024.csv')
team_data = pd.read_csv('Team.2024.csv')

#filter the dataframe to include only necessary columns
filtered1_data = game_data[['SeasonType', 'Status', 'Day', 'AwayTeamID',
    'HomeTeamID', 'AwayTeamScore', 'HomeTeamScore']]

#filter the dataframe to include only regular season games and games that
#have been completed
filtered2_data = filtered1_data[filtered1_data['SeasonType']==1]
filtered3_data = filtered2_data[(filtered2_data['Status']=='Final') |
    (filtered2_data['Status']=='F/OT')]
filtered_data = filtered3_data

#create a dictionary of new ID's for teams so that there are no gaps in
```

```
#ID data
new_ids = {}
idcount = 0
for index, row in filtered_data.iterrows():
    if row['AwayTeamID'] not in new_ids.keys():
        new_ids[row['AwayTeamID']] = idcount
        idcount += 1
    if row['HomeTeamID'] not in new_ids.keys():
        new_ids[row['HomeTeamID']] = idcount
        idcount += 1

#modify the dataframe to incorporate new ID system
#this step responds to the problem of gaps in ID data that eventually
#leads to a singular matrix_MTM when populating matrix_M by team IDs
for index, row in filtered_data.iterrows():
    filtered_data.loc[index, 'AwayTeamID'] = new_ids[row['AwayTeamID']]
    filtered_data.loc[index, 'HomeTeamID'] = new_ids[row['HomeTeamID']]

#determine the number of columns of matrix M
max_awayteamid = filtered_data['AwayTeamID'].max()
max_hometeamid = filtered_data['HomeTeamID'].max()
def get_greater(a, b):
    if a >= b:
        return a
    else:
        return b
total_id = get_greater(max_awayteamid, max_hometeamid) + 1
```

```
#determine the number of rows of matrix M
total_games = len(filtered_data)

#create a dictionary of new indices for each row so that there are no
#gaps in the dataframe index
#this step responds to the problem of gaps in index values for the
#dataframe filtered_data that eventually leads to an index-out-of-range
#error when populating matrix_M by dataframe index values
index_to_newindex = {}
indexcount = 0
for index, row in filtered_data.iterrows():
    index_to_newindex[index] = indexcount
    indexcount += 1

#form the matrix M from the previous dataframe using new indices
matrix_M = np.zeros((total_games, total_id))
for index, row in filtered_data.iterrows():
    if int(row['AwayTeamScore']) > int(row['HomeTeamScore']):
        matrix_M[index_to_newindex[index],int(row['AwayTeamID'])]=1
        matrix_M[index_to_newindex[index],int(row['HomeTeamID'])]=-1
    elif int(row['AwayTeamScore']) < int(row['HomeTeamScore']):
        matrix_M[index_to_newindex[index],int(row['AwayTeamID'])]=-1
        matrix_M[index_to_newindex[index],int(row['HomeTeamID'])]=1
    else:
        raise Exception("there is a game resulting in a tie")
```

```
#form vector d from the previous dataframe using new indices
vector_d = np.zeros((total_games, 1))
for index, row in filtered_data.iterrows():
    vector_d[index_to_newindex[index],0]=abs(int(row['AwayTeamScore'])-
        int(row['HomeTeamScore']))

#create a list of all game days during the season
date_list = []
for index, row in filtered_data.iterrows():
    if datetime.strptime(row['Day'], "%m/%d/%Y %H:%M:%S %p") not in
        date_list:
        date_list.append(datetime.strptime(row['Day'], "%m/%d/%Y
            %H:%M:%S %p"))

#find the first and last days of the season
max_date = max(date_list)
min_date = min(date_list)

#create a linear function that takes gameday as input and produces a
#time-in-season ratio as output
#add 0.01 to ensure matrix_MTWM is singular
def linear_function(game_date):
    datetime_game_date = datetime.strptime(game_date, "%m/%d/%Y
        %H:%M:%S %p")
    delta1 = datetime_game_date - min_date
    delta2 = max_date - min_date
    total_seconds_delta1 = delta1.total_seconds()
```

```
    total_seconds_delta2 = delta2.total_seconds()
    return (total_seconds_delta1 / total_seconds_delta2) + 0.01

#form weighted linear_W from the previous dataframe using
#linear_function()
linear_W = np.zeros((total_games, total_games))
for index, row in filtered_data.iterrows():
    linear_W[index_to_newindex[index],index_to_newindex[index]]=
        linear_function(row['Day'])

#create an exponential function that takes a float as input and produces
#a float as output
def exponential_function(value):
    return math.exp(value)

#form weighted exponential_W from the previous dataframe using
#linear_function() called by the exponential_function()
exponential_W = np.zeros((total_games, total_games))
for index, row in filtered_data.iterrows():
    exponential_W[index_to_newindex[index],index_to_newindex[index]]=
        exponential_function(linear_function(row['Day']))

#create a logarithmic function that takes a float as input and produces
#a float as output
def logarithmic_function(value):
    return math.log(value+1)
```

```
#form weighted logarithmic_W from the previous dataframe using
#linear_function() called by the logarithmic_function()
logarithmic_W = np.zeros((total_games, total_games))
for index, row in filtered_data.iterrows():
    logarithmic_W[index_to_newindex[index],index_to_newindex[index]]=
        logarithmic_function(linear_function(row['Day']))

#form weighted stepfunction_W from the previous dataframe using
#linear_function() and defining a step function using for loops
stepfunction_W = np.zeros((total_games, total_games))
for index, row in filtered_data.iterrows():
    for number in range(1,12):
        if linear_function(row['Day']) <= (number/11):
            stepfunction_W[index_to_newindex[index],
                index_to_newindex[index]]=(number/11)

#form matrix MTM by performing matrix operations
#option: edit the following matrix operations to include linear,
#exponential, logarithmic, or step function W for different weighting
#schemes
matrix_MTM = np.matmul(matrix_M.transpose(),matrix_M)

#form the Massey Matrix MTM by replacing the last row with 1s
list_of_ones = [1] * total_id
matrix_MTM[total_id -1] = list_of_ones

#form vector MTd by performing matrix operations
```

```
vector_MTd = np.matmul(matrix_M.transpose(),vector_d)

#form the adjusted vector MTd by replacing the last row with a 0
vector_MTd[total_id -1] = [0]

#calculate the inverse of Massey Matrix MTM
matrix_MTM_inverse = np.linalg.inv(matrix_MTM)

#multiply the inverse of Massey Matrix MTM with adjusted vector MTd
vector_final_ratings = np.matmul(matrix_MTM_inverse,vector_MTd)

#create a dictionary mapping old ID's to school names
id_to_school = {}
for index, row in team_data.iterrows():
    if row['TeamID'] not in id_to_school.keys():
        id_to_school[row['TeamID']] = row['School']

#recover ratings with the original school name and place school-rating
#pairs into a dictionary
def get_key_from_value(my_dict, target_value):
    for key, value in my_dict.items():
        if value == target_value:
            return key
    return None
team_ratings = {}
for index, value in enumerate(vector_final_ratings):
    team_ratings[id_to_school[get_key_from_value(new_ids, index)]] =
```

```
        value.item()

#alphabetize the dictionary by school name
alphabet_team_ratings = dict(sorted(team_ratings.items()))

#option 1: print the alphabetized final results to the output
for key, value in sorted_team_ratings.items():
    print(f"Team Name: {key}, Rating: {value}")

#sort the dictionary by rating highest -> lowest
hi_to_low_ratings = dict(sorted(team_ratings.items(), key=lambda item:
    item[1], reverse=True))

#option 2: print the top 100 final results from highest -> lowest to
#the output
for key, value in islice(hi_to_low_ratings.items(), 100):
    print(f"Team Name: {key}, Rating: {value}")
```

Bibliography

[**1**] Chartier, Tim. *Big Data: How Data Analytics Is Transforming the World*. Chantilly, VA: The Teaching Company, 2014.

[**2**] ——, Erich Kreutzer, Amy Langville, and Kathryn Pedings. "Bracketology: How can math help?" In *Mathematics and Sports*, edited by Joseph A. Gallian and Mathematical Association of America, 55-70. Washington, DC: Mathematical Association of America, 2010.

[**3**] ——. *Math Bytes: Google Bombs, Chocolate-Covered Pi, and Other Cool Bits in Computing*. Princeton: Princeton University Press, 2014.

[**4**] ESPN. "Figuring out the BCS isn't as hard as it looks." Accessed December 11, 2024. https://www.espn.com/ncf/preview00/s/2000/0811/679200.html#:~:text=The%20BCS%20rankings%20are%20not,are%20more%20important%20than%20others

[**5**] FiveThirtyEight. "How Our March Madness Predictions Work." Accessed December 11, 2024. https://fivethirtyeight.com/methodology/how-our-march-madness-predictions-work-2/

[**6**] Hayduk, Ted. "On the use of quantitative data in the sports context." In *Statistical Modelling and Sports Business Analytics*, edited by Vanessa Ratten and Ted Hayduk, 10-25. Abingdon, Oxon: Routledge, 2020.

[**7**] Langville, Amy N., and Carl D. Meyer. *Who's #1?: The Science of Rating and Ranking*. Princeton: Princeton University Press, 2012.

[**8**] Massey, Kenneth. "Statistical Models Applied to the Rating of Sports Teams." Undergraduate honors thesis, Bluefield College, 1997.

[**9**] MasseyRatings. "Massey Schedule Ratings." Accessed March 7, 2025. https://masseyratings.com/theory/sched.htm

[**10**] Minton, Roland B. *Sports Math: An Introductory Course in the Mathematics of Sports Science and Sports Analytics*. Boca Raton, FL: CRC Press LLC: Chapman and Hall/CRC, 2016.

[**11**] SportsData.IO. "REAL-TIME SPORTS DATA FOR GAMING, MEDIA & BEYOND." Accessed March 21, 2025. https://sportsdata.io

[**12**] Vaziri, Baback, Shaunak Dabadghao, Yuehwern Yih, and Thomas L. Morin. "Properties of Sports Ranking Methods." *The Journal of the Operational Research Society* 69, no. 5 (2018): 776-87. JSTOR (accessed December 11, 2024).

[**13**] Winston, Wayne L., Scott Nestler, and Konstantinos Pelechrinis. *Mathletics: How Gamblers, Managers, and Fans Use Mathematics in Sports*. Princeton: Princeton University Press, 2022.