Markov Chain Models for Baseball

and

Batting Order Improvement

Dale Garrett

Occidental College
Department of Mathematics
4/1/11

Table of Contents

Introduction

In the game of baseball, the manager determines the batting order based on theory and tradition that is widely accepted and has been quite consistent over the years. A typical lineup consists of a speedy contact hitter with a high on-base percentage in the leadoff spot, the best overall hitter with a high batting average and slugging percentage in the third spot, the best power-hitter batting cleanup (fourth), and the pitcher batting last with the other players filling in the remaining spots. But is this really the best lineup that produces the most runs and ultimately the most wins? It turns out that Markov chains are ideal for modeling baseball games because it is fairly easy to determine the probability of changing from one situation to another based on commonly kept statistics. Also, it is reasonable to assume that a batter's outcome is dependent only on the current situation, which is a key assumption in Markov models. According to a Markov chain model for baseball developed by Joel S. Sokol (2004), what is the most productive batting order and how different is this from what is traditionally accepted as best?

Markov Chains

A Markov chain is a mathematical model of a stochastic system where the changes involved are Markov processes. The two basic components of a Markov process are states and transitions. A state is a condition of the system that can be entirely explained by the values of the variables that classify that state. A system transitions between states when these variables change from the values particular to one state to those particular to another state (Howard). Ronald A. Howard describes a Markov process as a frog aimlessly jumping among lily pads in a pond. Each lily pad represents a possible state, and the current state is represented by the lily pad the frog is currently occupying. The transitions are represented by the jumps the frog makes from lily pad to lily pad (Howard).

Mathematically, a Markov chain is a discrete stochastic, or random, process for which the Markov property holds. The Markov property states that the probability of the next state is dependent only on the current state, and therefore knowledge of the previous state or states is irrelevant. For this reason, a Markov chain is said to be "memory-less." Formally, a Markov chain is defined as follows:

$$P\left(X_{n+1} = x \middle| X_1 = x_1, \ldots, X_n = x_n\right) = P(X_{n+1} = x | X_n = x_n) \tag{1}$$

where the potential values of X_i compose a finite set S, the state space of the chain. The probability of transitioning from state *i* to state *j*, $p_{ij}^{(n)}$, is given by

$$p_{ij}^{(n)} = P\left(X_n = j \middle| X_0 = i\right), \tag{2}$$

where *n* is the number of time steps taken to get from state *i* to state *j*. If it is possible to transition from state *i* to state *j*—in other words, if state *i* has a non-zero probability of transitioning to state *j* at any time—then state *j* is said to be accessible from state *i*. We write $i \Rightarrow j$, and say state *j* is accessible from state *i* if there exists an integer $n > 0$ such that

$$P\left(X_n = j \middle| X_0 = i\right) = p_{ij}^{(n)} > 0\,. \tag{3}$$

This means that state *j* can be reached from state *i* in *n* transitions or time steps. A state *i* where

$$p_{ii} = 1$$

and

$$p_{ij} = 0,$$

for $i \neq j$ is called a final or absorbing state because there are no states accessible from this state. As long as the state space, S, is finite, the distribution of transition probabilities can be represented by a transition matrix where the $(i,j)^{th}$ entry of the transition matrix is equal to

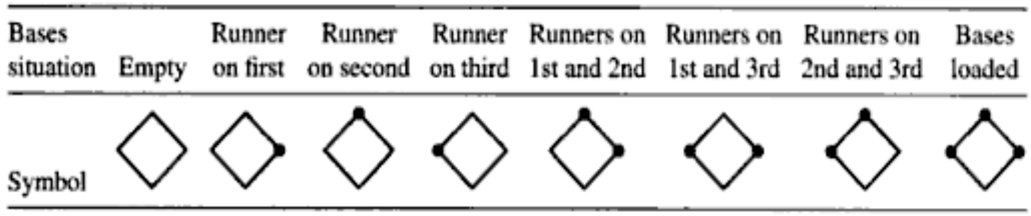
$$p_{ij} = P\left(X_{n+1} = j \middle| X_n = i\right) \tag{4}$$

("Markov Chain").

Baseball as a Markov Chain

Markov chains are appropriate for modeling the game of baseball because the game has a finite number of states and the probability of the next state occurring depends only on the current state. Thus, baseball satisfies both of the Markov properties and they are both reasonable assumptions for the game. A half-inning of baseball can, at any point, be represented by the number of outs and the number and location of runners on base. Thus, a state is defined by the number of outs and the distribution of base runners. There are 24 possible states for a half-inning of play. There are 3 possibilities (0, 1, and 2) for the number of outs, and 8 possible distributions for the runners on base (see **Figure 1**).

Figure 1: Distribution of Base Runners



There are 8 possibilities for the number and location of base runners in a baseball game.
Source: *Teaching Baseball Using Statistics* by Jim Albert (2003)

The 24 possible states of a half-inning can be represented by (D,O) where D is the distribution of base runners and O is the number of outs. The distribution of base runners are referred to using the following notation:

$$\varnothing, 1, 2, 3, 12, 13, 23, 123$$

where, for example, $\varnothing$ represents no runners on base and *23* represents runners on second base and third base. The state space of the Markov chain for a half-inning is depicted in **Figure 2.**

Figure 2: State Space

State space of the Markov chain; states are labeled in (D,O) format.
Source: *An Intuitive Markov Chain Lesson From Baseball* by Joel S. Sokol (2003)

The state space of the Markov chain for a whole baseball game contains 217 states. These come from the 24 possible states for a half-inning, the 9 offensive half-innings per game, and a final "game over" state yielding

$$24 \times 9 + 1 = 217$$

states. The 217 states for a whole baseball game can be represented by (I, D, O) where I is the inning, D is the distribution of base runners, and O is the number of outs.

Before and after each plate-appearance, the game is in a certain state determined by the number of outs and the distribution of runners on base. Transition probabilities are given by $p_{i,j}$, which represent the probability that the current batter will change the state of the game from state i to state j (Sokol 2003). For example, if a batter comes to bat with the bases loaded and one out $(123,1)$ and after the completion of his plate appearance changes the state of the game to runner on third base and one out $(3,1)$, then the batter must have hit a triple. Thus, the probability that the state of the game changed from bases loaded, one out $(123,1)$ to runner on third, one out $(3,1)$ is the probability that the batter hit a triple. The transition probabilities can be determined using commonly kept baseball statistics. The probability that a batter executes a certain action (walk, single, homerun, etc.), in a given plate-appearance can be thought of as how often these events have occurred for the batter over the course of a season, or, even better, a career. Thus, we assume the probability that a batter hits a triple in a given plate-appearance is the same as that player's number of career triples per plate-appearance. It is reasonable to assume that each plate-appearance is dependent only on the current situation, and independent of all previous situations. The second Markov property would be violated if these probabilities do not remain constant throughout a game (Sokol 2003). For example, it must be assumed that even if a batter has had a really good first half of a game and might be feeling more confident, his performance in the end of the game is unaffected. While it is possible that changes in batter confidence might lead to increased performance, we assume such changes mid-game are negligible. For these reasons, Markov chains are the most mathematically perfect way to model the game of baseball.

While there are many complex play possibilities and situations that could occur during the course of a baseball game, the Markov chain model represents a very simplistic view of the game. In the model, the play possibilities are limited to six events: walks (*w*), singles (*s*), doubles (*d*), triples (*t*), homeruns (*h*), and outs (*o*). The game of baseball is also complex in the sense that for some play possibilities (singles and doubles), runner advancement varies depending on where the ball is hit, the base runner's ability, the defense's ability, the score, the inning, and other factors. We assume a simple model of runner advancement that is summarized in **Figure 3.** This simple model of runner advancement does not specify which runners are on base, only which bases are occupied. This means that differences in base runner speed and ability are not accounted for, and that all base runners are identical because they each advance according to the probabilities described in **Figure 3.**

Figure 3: Play Possibilities and Runner Advancement Definitions

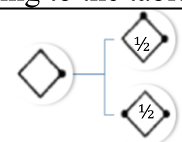
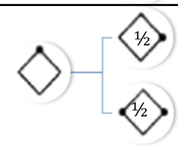
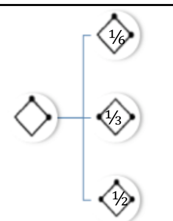
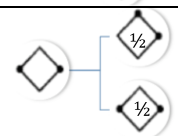
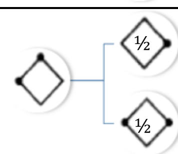
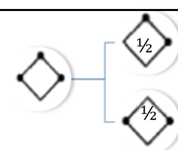
<table>
<tr><td>Walk</td><td>Batter reaches first base safely. All other runners advance one base, but only if they are forced.</td></tr>
<tr><td>Single</td><td>Batter reaches first base safely. Runner from first base never scores and runner from third base always scores. Other runners advance according to the table below.
<table>
<tr><td>Single with runner on first base</td><td>
</td><td>Given a single with a runner on first base:
$P(12) = ½$
$P(13) = ½$</td></tr>
<tr><td>Single with runner on second base</td><td>
</td><td>Given a single with a runner on second base:
$P(1) = ½$
$P(13) = ½$</td></tr>
<tr><td>Single with runner on first and second base</td><td>
</td><td>Given a single with a runners on first and second base:
$P(12) = ⅙$
$P(13) = ⅓$
$P(123) = ½$</td></tr>
<tr><td>Single with runner on first and third base</td><td>
</td><td>Given a single with runners on first and third base:
$P(12) = ½$
$P(13) = ½$</td></tr>
<tr><td>Single with runner on second and third base</td><td>
</td><td>Given a single with runners on second and third base:
$P(1) = ½$
$P(13) = ½$</td></tr>
<tr><td>Single with the bases loaded</td><td>
</td><td>Given a single with the bases loaded:
$P(12) = ½$
$P(13) = ½$</td></tr>
</table></td></tr>
<tr><td>Double</td><td>Batter reaches second base safely. Runners from second and third base always score. Runner on first scores with a probability of ½ and advances to third with probability of ½.</td></tr>
<tr><td>Triple</td><td>Batter reaches third base safely. All runners score.</td></tr>
<tr><td>Homerun</td><td>Batter scores. All runners score.</td></tr>
<tr><td>Out</td><td>Batter is out. Base runners do not advance.</td></tr>
</table>

Table defines each of the six play possibilities and how base runners advance.

Constructing the Mathematical Model

We can construct a mathematical model for the game of baseball now that we have defined the 217 possible states of a game and defined the probabilities for transitioning between these states. We let k represent the k^{th} plate-appearance of the game, where

$$k \in \{1, 2, 3, \ldots, end\}.$$

The game begins with the first plate appearance $(k = 1)$ and k continually increases as the game progresses until the game ends with the last batter. We let x_k represent the batter in the lineup batting in the k^{th} plate-appearance of the game, where

$$x_k \in \{1, 2, 3, \ldots, 9\}.$$

The possible values for x_k represent the nine players in a batting order. k and x_k differ in that k continually increases as the game progresses whereas x_k cycles as the lineup turns over after the ninth batter has hit. Next, we let A_K represent the action, or play possibility, performed by the batter in the k^{th} plate-appearance of the game, where

$$A_k \in \{w, s, d, t, h, o\},$$

and w, s, d, t, h, and o represent the six play possibilities previously defined. We let S_k represent the state of the game before the k^{th} plate-appearance, where a state is defined by the inning, the distribution of runners on base, and the number of outs. Thus,

$$S_k = (I_k, D_k, O_k)$$

where I_k is the inning before the k^{th} plate-appearance, D_k is the distribution of base runners before the k^{th} plate-appearance, and O_k is the number of outs before the k^{th} plate-appearance. Then, S_{k+1} represents the state of the game after the k^{th} plate-appearance and before the $k+1^{\text{th}}$ plate-appearance. Lastly, we let $R_{k,k+1}$ represent the number of runs scored in the transition between state S_k and state S_{k+1}.

The action the batter performs in the k^{th} plate-appearance of the game is dependent on which batter in the lineup is up. Thus,

$$A_k = h\left(x_k\right) \tag{5}$$

where the action performed in the k^{th} plate-appearance, is some function *h*, of the x_k^{th} batter in the lineup. The state of the game at any point during the game is dependent on the previous state and the action performed by the preceding batter. Thus,

$$S_{k+1} = g\left(S_k, A_k\right) \tag{6}$$

where the state of the game before the $k+1^{\text{th}}$ plate-appearance, S_{k+1}, is some function *g*, of the state before the k^{th} plate-appearance and the action performed by the batter in the k^{th} plate-appearance.

The number of runs scored in the transition from one state to the next is dependent only on those states. Thus,

$$R_{k,k+1} = f\left(S_k, S_{k+1}\right) \tag{7}$$

where the number of runs scored between state S_k and state S_{k+1} is some function *f*, of state S_k and state S_{k+1}. The number of runs scored in the transition between any two states is given by,

$$R_{k,k+1} = f\left(S_k, S_{k+1}\right) = 1 + \left(\left|D_k\right| + O_k\right) - \left(\left|D_{k+1}\right| + O_{k+1}\right) \tag{8}$$

where $\left|D_k\right|$ is the number of runners on base before the k^{th} plate-appearance and O_k is the number of outs before the k^{th} plate-appearance (Sokol 2004). This formula can be verified by an example. Take the transition from runners at second and third base, one out $(I,23,1)$ to runner at second base, one out $(I,2,1)$. We expect two runs to score as the runners from second and third base score on a double. Substituting into **Equation** (8) yields

$$R_{(I,23,1),(I,2,1)} = 1 + \left(|2| + 1\right) - \left(|1| + 1\right) = 2\,. \tag{9}$$

Then, the number of runs, *R*, scored per game is the sum of the runs scored in the transitions between all plate-appearances. Thus,

$$R = \sum_{k=1}^{ends} R_{k,k+1}\,. \tag{10}$$

In order to determine if traditional batting order strategy is best we need to be able to compare various batting orders according to their expected number of runs produced per game. Thus, we must derive the expected value of the random variable *R*, or the expected value of the number of runs scored per game. The expected value of a random variable is its mean or average value. The expected value of a sum of random variables is equal to the sum of the expected values of the random variables. For example, if Y_i is a random variable, then

$$E\left[Y_1 + Y_2 + Y_3 + \ldots + Y_n\right] = \sum_{i=1}^{n} E\left[Y_i\right]. \tag{11}$$

Therefore, the expected value of the runs scored per game is the sum of the expected value of the runs scored between all plate-appearances. Thus,

$$E\left[R\right] = \sum_{k} E\left[R_{k,k+1}\right]. \tag{12}$$

The expected number of runs scored in the transition between the k^{th} and the $k+1^{\text{th}}$ plate-appearances depends on the possible state transitions between those appearances. Thus,

$$E\left[R_{k,k+1}\right]=\sum_{(s_k,s_{k+1})} f\left(s_k,s_{k+1}\right)\cdot P\left(S_k=s_k \,\&\, S_{k+1}=s_{k+1}\right). \tag{13}$$

The joint probability distribution of S_k and S_{k+1} can be calculated as the probability of S_{k+1} given S_k, multiplied by the probability of S_k. Thus,

$$P\left(S_k=s_k \,\&\, S_{k+1}=s_{k+1}\right)=P\left(S_{k+1}=s_{k+1}\middle|S_k=s_k\right)\cdot P\left(S_k=s_k\right). \tag{14}$$

The probability of a particular state transition is the sum of the probabilities that could cause that state transition. Thus,

$$P\left(S_{k+1}=s_{k+1}\middle|S_k=s_k\right)=\sum_{a_k:(s_k,s_{k+1})} P\left(A_k=a_k\right). \tag{15}$$

The probability that the batter in the k^{th} plate-appearance performs action a_k depends on who that batter is. Thus,

$$P\left(A_k=a_k\right)=h\left(x_k\right). \tag{16}$$

The marginal probability that the game is currently in any one of the 217 states, say s_{k+1}, can be calculated from the joint probabilities

$$P\left(S_{k+1}=s_{k+1} \,\&\, S_k=s_k\right) \tag{17}$$

as follows:

$$P\left(S_{k+1}=s_{k+1}\right)=\sum_{s_k} P\left(S_{k+1}=s_{k+1} \,\&\, S_k=s_k\right). \tag{18}$$

We initiate the Markov calculation by specifying an initial state. The game of baseball begins in the first inning, with nobody on base, and zero outs. Thus, we set the state of the game before the first plate-appearance to the first inning, nobody on base, and zero out state as follows:

$$s_1=\left(1,\varnothing,0\right).$$

The probability of this state occurring is given by

$$P\left(s_1 = \left(1,\varnothing,0\right)\right) = 1 \tag{19}$$

since we know the game must begin in this state. While the game could theoretically go on forever, we must stop the calculation. Therefore, we stop running the Markov chain when the probability of the 217th or "game over" state $\left(9,D,3\right)$ is above the threshold

$$P\left(s_k = \left(9,D,3\right)\right) > 0.99\,, \tag{20}$$

for some *k*.

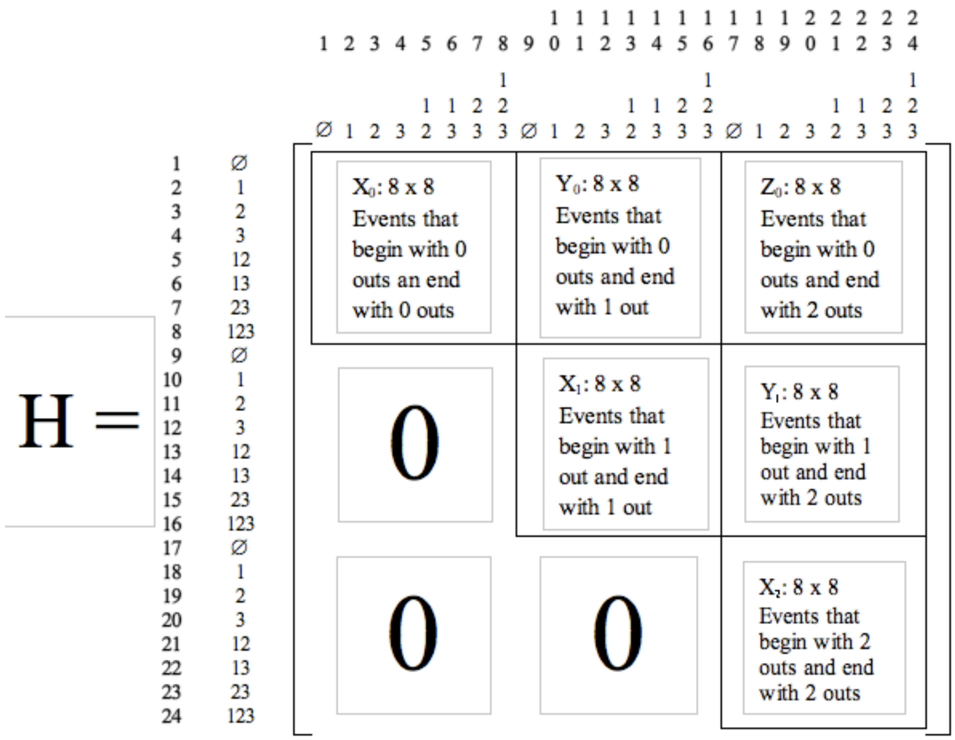
The Sokol Model

We can now use the mathematical model of a baseball game to fully understand the Sokol Markov chain model for baseball. Using the state space for a whole baseball game and the transition probabilities of moving between the 217 states, we can construct a transition matrix for a whole baseball game. But, first we must construct a generic transition matrix, ***H***, for a half-inning of a baseball game. This 24×24 generic transition matrix, ***H***, can be constructed as follows,

$$\mathbf{H} = \begin{pmatrix} X_0 & Y_0 & Z_0 \\ [0] & X_1 & Y_1 \\ [0] & [0] & X_2 \end{pmatrix} \tag{21}$$

where the *X*s, *Y*s, and *Z*s correspond to 8×8 block matrices and the $\left[0\right]$s refer to 8×8 matrices of all zeros. This matrix is summarized in **Figure 4**.

Figure 4: Generic Transition Matrix, *H*, for a Half-Inning



Generic transition matrix, ***H***, for a half-inning of a baseball game.

The X blocks represent events where the number of outs does not increase, the Y blocks represent events where the number of outs increase by one but do not result in the third out, and the Z block represents events where the number of outs increases from zero to two outs, such as double plays. However, double plays are not accounted for in this simple baseball Markov chain model so the Z_0 block contains all zeros. The subscript for each block refers to the number of outs at the beginning of the play. For example, X_0 represents transitions from situations with zero outs to situations with zero outs, X_1 represents transitions from situations with one out to situations with one out, and X_2

represents transitions from situations with two outs to situations with two outs. Similarly, Y_0 represents transitions from situations with zero outs to situations with one out, and Y_1 represents transitions from situations with one out to situations with two outs.

Before we construct the transition matrix for a whole baseball game we must also construct an 8×8 fundamental block matrix, ***B***. This matrix represents the possible transitions between states that do not result in an out being recorded. The eight rows and eight columns of the ***B*** block matrix correspond to the eight different distributions for the runners on base,

$$\varnothing,\ 1,\ 2,\ 3,\ 12,\ 13,\ 23,\ 123.$$

The first cell in the ***B*** block matrix, $(\varnothing,\varnothing)$, corresponds to the probability that a plate-appearance beginning with nobody on base results in a state with nobody on base and no outs being recorded. The probability of this occurring, $p_{\varnothing,\varnothing}$, is the probability that the batter hits a home run, or p_h. This is the only option since any other play possibility results in a runner on base or an out being recorded. Therefore, p_h belongs in the first cell of the 8×8 ***B*** block matrix. The remaining cells of ***B*** can be filled in similarly using the six play possibilities previously defined, resulting in

$$\boldsymbol{B}=\begin{array}{c} \\ \varnothing \\ 1 \\ 2 \\ 3 \\ 12 \\ 13 \\ 23 \\ 123 \end{array}\begin{array}{c} \begin{array}{cccccccc} \varnothing & 1 & 2 & 3 & 12 & 13 & 23 & 123 \end{array} \\ \begin{pmatrix} p_h & p_w+p_s & p_d & p_t & 0 & 0 & 0 & 0 \\ p_h & 0 & \frac{p_d}{2} & p_t & p_w+\frac{p_s}{2} & \frac{p_s}{2} & \frac{p_d}{2} & 0 \\ p_h & \frac{p_s}{2} & p_d & p_t & p_w & \frac{p_s}{2} & 0 & 0 \\ p_h & p_s & p_d & p_t & 0 & p_w & 0 & 0 \\ p_h & 0 & \frac{p_d}{2} & p_t & \frac{p_s}{6} & \frac{p_s}{3} & \frac{p_d}{2} & p_w+\frac{p_s}{2} \\ p_h & 0 & \frac{p_d}{2} & p_t & \frac{p_s}{2} & \frac{p_s}{2} & \frac{p_d}{2} & p_w \\ p_h & \frac{p_s}{2} & p_d & p_t & 0 & \frac{p_s}{2} & 0 & p_w \\ p_h & 0 & \frac{p_d}{2} & p_t & \frac{p_s}{2} & \frac{p_s}{2} & \frac{p_d}{2} & p_w \end{pmatrix} \end{array}. \tag{22}$$

Cells with zeros represent transitions that are either impossible or assumed to be zero based on the model's play possibilities. In the game of baseball, it is impossible to increase the number of base runners by more than one after the completion of a plate-appearance. For example, the cell $(\varnothing,12)$ in the ***B*** block matrix corresponds to the transition from nobody on base, to runners on both first and second base. The probability of this transition is zero because it is impossible for the number of runners on base to increase by two after the completion of one plate-appearance. Also, consider the cell $(123,1)$ in ***B***. This entry corresponds to the transition from bases loaded and no outs, to a runner on first and no outs. Because the six play possibilities are defined so that the maximum number of bases a runner can advance on a single is two, and walks advance runners one base and only if they are forced, this transition is impossible as a base runner can't score from first (advancing three bases) on a single. While runners on first do occasionally score on singles, it is quite rare and not accounted for by this model.

Now with the ***B*** block matrix defined, we can construct the 217×217 transition matrix, ***T***, for each player. The dimensions for the ***T*** transition matrix are determined as follows. There are eight distributions for the runners on base, three out possibilities per half-inning, nine offensive half-innings in a game, and a final "game over" state, yielding

$$8 \cdot 3 \cdot 9 + 1 = 217$$

states. The framework for the 217×217 transitions matrix, ***T***, is developed as follows. Each of the nine offensive half-innings of a baseball game (for one team) is represented by a 24×24 block of cells along the diagonal of the 217×217 transition matrix, ***T*** (see **Appendix I**). The generic transition matrix, ***H***, for a half-inning can be placed within each of these nine 24×24 blocks of cells representing the nine half-innings (see

Appendix I). The rest of the cells in the 217×217 transition matrix, $\boldsymbol{T}$, contain zeros, except for the $(217, 217)$ cell that contains a 1. This cell represents the final "game over" state which only transitions to itself.

With the framework for the $\boldsymbol{T}$ transition matrix established, we can now further develop the $\boldsymbol{T}$ transition matrix by adding the transition probabilities based on the six play possibilities. The $\boldsymbol{T}$ transition matrix contains the 8×8 $\boldsymbol{B}$ block matrix along its diagonal (see **Appendix II**). The 8×8 blocks along the diagonal of the $\boldsymbol{T}$ transition matrix represent the transitions that do not result in an out being recorded (see X blocks in **Figure 5)**, which is exactly what is represented by the $\boldsymbol{B}$ block matrix.

Next, we define an 8×8 outs identity matrix, $\boldsymbol{I}$, as follows,

$$p_o I_8 \tag{23}$$

where p_o is the probability of an out and I_8 is the 8×8 identity matrix. The outs identity matrix $\boldsymbol{I}$ is constructed as follows,

$$\boldsymbol{I} = \begin{array}{c} \\ \varnothing \\ 1 \\ 2 \\ 3 \\ 12 \\ 13 \\ 23 \\ 123 \end{array} \begin{array}{c} \begin{array}{cccccccc} \varnothing & 1 & 2 & 3 & 12 & 13 & 23 & 123 \end{array} \\ \begin{pmatrix} p_o & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p_o & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_o & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_o & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_o & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_o & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_o & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_o \end{pmatrix} \end{array}. \tag{24}$$

The off-diagonal entries in the outs identity matrix $\boldsymbol{I}$ are zero because in this model base runners cannot advance on outs. The outs identity matrix $\boldsymbol{I}$ is placed in the Y_0 and Y_1 blocks (see **Appendix I**) for each of the nine half-innings in the transition matrix $\boldsymbol{T}$ (see

Appendix II). The Y_0 and Y_1 blocks represent the transitions that result in an out being recorded. The outs identity matrix ***I*** serves to transition back to the same inning and base runner distribution state—however, with one more out—when an out occurs that is not the third out.

Now, we must define an 8×1 outs column vector, ***V***, as follows,

$$\mathbf{V} = p_o v \tag{25}$$

where p_o is the probability of an out being recorded and v is the following 8×1 column vector,

$$v = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \tag{26}$$

The 8×1 outs column vector, ***V***, is constructed as follows:

$$\mathbf{V} = \begin{pmatrix} p_o \\ p_o \\ p_o \\ p_o \\ p_o \\ p_o \\ p_o \\ p_o \end{pmatrix}. \tag{27}$$

The outs column vector ***V*** is placed in the column adjacent to the X_2 blocks (see **Appendix I**) for each of the nine innings in the ***T*** transition matrix (see **Appendix II)**. The outs column vector, ***V***, serves to transition to the next innings zero out and no base runner state when the third out in the previous innings occurs, regardless of the base runner distribution when the third out occurs. The transition matrix, ***T***, is now complete and is summarized in **Appendix II**.

With the transition matrix, ***T***, complete we can now create the $9 \times 217 \times 217$ player matrices, ***M***. These are the nine transition matrices, ***T***, one for each batter in the lineup, grouped together in the order in which the lineup occurs. Each of these batter specific transition matrices, ***T***, is constructed with the player specific transition probabilities according to their career statistics for each of the six play possibilities.

We must now develop a run-value matrix that keeps track of how many runs score during each transition of a baseball game. The run-value matrix, ***R***, is a 217×217 matrix and its dimensions are defined according to the same logic used for determining the dimensions for the transition matrix, ***T***. There are eight base runner distributions, three out possibilities per half-inning, nine offensive half-innings in a game, and a final "game over" state, yielding

$$8 \cdot 3 \cdot 9 + 1 = 217$$

states.

But, before we construct the run-value matrix, ***R***, we must first create an 8×8 runs matrix, ***N***, that keeps track of the number of runs that score during the transitions between states where no outs are recorded. Because we are assuming a simple model of runner advancement where runners cannot advance when outs are recorded, runs can only score in transitions between states where no outs are recorded. We can calculate the number of runs scored in the transition between any two states because, after each transition each base runner and the batter must be accounted for as either being on base, out, or having scored. Therefore, the number of runs scored, $R_{i,j}$, in the transition between states *i* and *j*, is a function of states *i* and *j* (see **Equation** (8)) and is given by

$$R_{i,j} = 1 + \left(\left|D_i\right| + O_i\right) - \left(\left|D_j\right| + O_j\right), \tag{28}$$

where $\left|D_i\right|$ is the number of base runners in state *i* and O_i is the number of outs in state *i*. For example, the transition from runners on first and second base, one out $(I,12,1)$ to runner on second base, one out $(I,2,1)$ results in 2 runs being scored. This can be verified by substituting into **Equation** (28) as follows:

$$R_{(I,12,1),(I,2,1)} = 1 + (2+1) - (1+1) = 2\,. \tag{29}$$

Using this logic we can construct the 8×8 runs matrix, ***N***, with the number of runs that score between all possible transitions that do not result in an out being recorded. The runs matrix, ***N***, is constructed as follows:

$$\boldsymbol{N} = \begin{array}{c} \\ \varnothing \\ 1 \\ 2 \\ 3 \\ 12 \\ 13 \\ 23 \\ 123 \end{array} \begin{array}{c} \begin{array}{cccccccc} \varnothing & 1 & 2 & 3 & 12 & 13 & 23 & 123 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 2 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 2 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 2 & 2 & 2 & 1 & 1 & 1 & 0 \\ 4 & 3 & 3 & 3 & 2 & 2 & 2 & 1 \end{pmatrix} \end{array}. \tag{30}$$

We can now construct the 217×217 run-value matrix, ***R***, which keeps track of the number of runs scored throughout a whole baseball game. The run-value matrix, ***R***, is a block-diagonal matrix that contains the 8×8 runs matrix, ***N*** (see **Equation** (30)), along its main diagonal (see **Appendix III**). This is because the 8×8 blocks that form along the main diagonal of the 217×217 run-value matrix, ***R***, represent the transitions that do not result in an out being recorded, and since we are assuming a simplified model where base runners cannot score on outs, these are the only transitions where runs could possibly score. The run-value matrix, ***R***, is independent of the number of outs and the current inning, which is why the 8×8 runs matrix, ***N***, can be repetitively placed along the diagonal of the run-value matrix, ***R***.

Next, we must define a 1×217 current state vector, ***C***, that keeps track of the probability that the game is in any given state (see **Equation** (18)). The current state vector, ***C***, will also be used to determine when the game is over and when the Markov chain reaches its final or absorbing state. The 1×217 current state vector, ***C***, is defined as follows,

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}. \tag{31}$$

To begin the game, the probability of the initial first inning, nobody on, nobody out state $(1, \varnothing, 0)$ is 1 and the probability of all other states is zero. The game is over when the probability that the game is over, or the entry in the 217th column of ***C***, is greater than 0.99.

For each batter in the lineup we recalculate the current state vector, ***C***, based on the transition matrix, ***T***, for the batter as follows,

$$C = C \cdot T\,. \tag{32}$$

We must also calculate how many runs are scored in each transition according to **Equation** (28). If ***R*** is the run-value matrix, containing the number of runs that score in each transition, and ***T*** is the transition matrix for a given player, then the expected number of runs each player will create in each transition is given by,

$$R.*T \ . \tag{33}$$

(Note: The operation .* refers to array multiplication which acts element-wise on the arrays, and • refers to standard matrix multiplication). Multiplying **Equation** (33) by the current state vector, ***C***, as follows

$$C\left(R.*T\right), \tag{35}$$

yields the vector of the expected number of runs the batter will generate if he were to bat in each state, weighted by the probability of each state. Therefore, the number of expected runs the batter creates in its current at bat can be calculated according to **Equation** (13) and is given by

$$\sum C\left(R.*T\right) \ . \tag{36}$$

As the game progresses from batter to batter, the number of expected runs the batter creates it its current at bat is added to the expected number of runs produced thus far. This occurs until the game is over and the expected number of runs produced per game is obtained.

Data

I use the 2010 Los Angeles Dodgers to determine if improvements can be made to traditional batting order strategy. Baseball-reference.com provides the batting order used most often by Dodgers manager Toe Torre during the 2010 season (see **Figure 5**).

Figure 5 Los Angeles Dodgers Most Often Used Batting Order

1.	Rafael Furcal	SS
2.	Matt Kemp	CF
3.	Andre Ethier	RF
4.	Manny Ramirez	LF
5.	James Loney	1B
6.	Casey Blake	3B
7.	Blake DeWitt	2B
8.	Russell Martin	C
9.	(Pitcher's Spot)	P

The batting order used most often (10 times) by manager Joe Torre of the Los Angeles Dodgers in 2010

Source: http://www.baseball-reference.com/teams/LAD/2010-batting-orders.shtml

I use the lineup used most often by the 2010 Dodgers as a benchmark because this lineup accurately reflects traditional batting order strategy. Rafael Furcal is a typical leadoff hitter because he is very fast, a good base stealer, a switch hitter, a contact hitter, and has a high on-base percentage. Andre Either is a typical third batter in that he is an excellent overall hitter who hits for both average and power. Manny Ramirez—who is arguably one of the best, active offensive players in the game (although he might be past his prime)—epitomizes the cleanup batter because he is a great homerun hitter and has a very high on-base plus slugging percentage. Using career statistics obtained from baseball-reference.com to create transition probabilities and to construct transition matrices for each of the nine players in the lineup, I investigate whether Joe Torre's favorite lineup is in fact the best.

It should be noted that because Major League Baseball teams usually use a rotation of five starting pitchers who each pitch every fifth game, and the pitchers are almost always the worst hitters on the team, I use the combined statistics of all the Dodgers' pitchers in 2010. This is because, while the defensive lineup—excluding the pitcher—remains relatively consistent from game to game, a pitcher will only pitch in one of five games. Also, the batting order is created without regard to who is pitching, as the pitcher, except for rare occasions, bats last. Baseball-reference.com provides offensive season statistic totals based on position. I use the season statistic totals for all of the pitchers on the Dodgers to create the transition matrix for the "average" Dodgers starting pitcher.

The career statistics for each of the players in the 2010 Dodgers' lineup obtained from baseball-reference.com are provided in **Figure 6.** The career statistics for each of

Figure 6: Career Statistics for 2010 Dodgers' Lineup

Order	Player's Name	Position	Home Runs	Triples	Doubles	Singles	Walks	Outs	Plate App.
1	Rafeal Furcal	SS	100	65	278	1165	569	4122	6299
2	Matt Kemp	CF	89	24	107	425	176	1648	2469
3	Andre Ethier	RF	98	18	165	441	270	1820	2812
4	Manny Ramirez	LF	555	20	547	1451	1329	5855	9757
5	James Loney	1B	55	19	125	435	203	1599	2436
6	Casey Blake	3B	163	17	254	701	419	3298	4852
7	Blake DeWitt	2B	16	7	40	159	95	653	970
8	Russell Martin	C	54	7	115	462	319	1756	2713
9	(Pitcher's Spot)	P	0	0	2	21	16	316	355

Career statistics for the six play possibilities and plate appearances for the 2010 Dodgers' Lineup

Source: www.baseball-reference.com

the six play possibilities can then be converted into transition probabilities by dividing the career total for each of the six play possibilities by the number of career plate-appearances for each player in the lineup (see **Figure 7**). For example, the transition probability for homeruns, p_h, is obtained by taking the ratio of career homeruns per plate-appearance as follows,

$$p_h = \frac{Homeruns}{PlateAppearance}. \tag{18}$$

Figure 7: Transition Probabilities

Order	Player's Name	Position	P_h	P_t	P_d	P_s	P_W	P_o
1	Rafeal Furcal	SS	0.01587554	0.0103191	0.04413399	0.18494999	0.0903318	0.65438959
2	Matt Kemp	CF	0.03604698	0.00972053	0.04333738	0.17213447	0.07128392	0.66747671
3	Andre Ethier	RF	0.03485064	0.00640114	0.0586771	0.15682788	0.09601707	0.64722617
4	Manny Ramirez	LF	0.05688224	0.00204981	0.05606231	0.14871374	0.1362099	0.60008199
5	James Loney	1B	0.022578	0.00779967	0.05131363	0.17857143	0.08333333	0.65640394
6	Casey Blake	3B	0.03359439	0.00350371	0.05234955	0.1444765	0.08635614	0.6797197
7	Blake DeWitt	2B	0.01649485	0.00721649	0.04123711	0.16391753	0.09793814	0.67319588
8	Russell Martin	C	0.01990417	0.00258017	0.0423885	0.17029119	0.11758201	0.64725396
9	(Pitcher's Spot)	P	0	0	0.0056338	0.05915493	0.04507042	0.89014085

Transition probabilities for the six play possibilities for the 2010 Dodger's Lineup

Each of the transition probabilities can be thought of as the probability that a given player will execute a certain action (homerun, triple, double, single, walk, or out) in a given plate-appearance. This is because over the course of a player's career he has executed these actions with the determined probabilities.

Matlab

I use the technical computing software Matlab to determine if traditional batting order strategy is optimal. Matlab is an appropriate tool for performing Markov chain calculations because of its ability to easily perform matrix manipulations. The model that I've presented and the Matlab program that I use are derived from the Matlab code that accompanies Sokol (2004). The Sokol Matlab program that I use is presented in **Appedix IV** and the Matlab program operates as follows. First, a .data file must be created containing the statistical totals for the six play possibilities for each of the nine players that comprise a lineup. Then, given this set of statistics, the program takes a specified batting order as input and, after performing the Markov chain calculations, provides the average number of runs this lineup will score per game as output. For example, the players are ordered one through nine according to the order in which their statistics are entered into the .data file. Therefore, the inputted batting order "123456789" would yield the average number of runs per game for the lineup specified in the .data file. By rearranging these nine hitters, the average number of runs per game for any lineup combination can be calculated, for example, "456123789." The program will also provide the average number of runs scored per game for a lineup consisting of nine identical players, such as "333333333."

Results

First, in order to validate the model I test its accuracy by comparing the average number of runs per game predicted by the model for the benchmark lineup to the number of runs per game actually scored by the 2010 Dodgers. The model predicts that the lineup used most often by manager Joe Torre of the Los Angeles Dodgers in 2010 will score an average of 4.162438 runs per game. According to baseball-reference.com the 2010 Dodgers scored 667 runs in 162 games, yielding

$$\frac{667}{162} = \mathbf{4.117284}$$

runs per game. Thus, the difference between the number of runs per game for Joe Torre's preferred lineup predicted by the model and the number of runs per game actually scored by the Dodgers in 2010 is

$$4.162438 - 4.117284 = \mathbf{0.045154}$$

runs. The percentage error is

$$\frac{4.162438 - 4.117284}{4.117284} \times 100\% = \mathbf{1.096694}\% .$$

Comparing the average number of runs predicted by the model for the lineup used most often by Dodgers' manager Joe Torre in 2010 to the actual number of runs per game scored by the Dodgers in 2010, however, can be slightly misleading. This is because the actual number of runs per game scored by the Dodgers in 2010 includes contributions by players other than the nine players considered by the model, since these nine players didn't play every inning of the 162 game season. However, the fact that these figures are so close provides evidence that the model is accurate. The model's accuracy is also validated by the fact that it predicts a greater number of runs per game than were actually score by the Dodgers in 2010. This is because the model considers the Dodger's "starting lineup" with their nine best players, whereas the actual number of runs per game scored by the Dodgers in 2010 takes into account games where one or more of these nine players is replaced by a bench player of lower ability due to injury, a day off, or other such reasons.

In the 10 games that the Joe Torre started his most often used lineup (see **Figure 5**) the Dodgers scored 63 runs (baseball-reference.com). Therefore, in these 10 games the Dodgers scored 6.3 runs per game. Thus, the difference between the number of runs per game for Joe Torre's preferred lineup predicted by the model and the number of runs per game actually scored by the Dodgers in the 10 games that Joe Torre started his most used lineup is

$$6.3 - 4.162438 = \mathbf{2.137562}$$

runs per game. The percentage error is therefore,

$$\left|\frac{4.162438-6.3}{6.3}\right|\times 100\% = \mathbf{33.92955}\% .$$

There appears to be quite a large discrepancy between the model's prediction and the actual number of runs per game scored by Joe Torre's preferred lineup in the 10 games in which they started. However, 10 games is such a small sample that the actual number of runs per game scored by this lineup might be significantly less if more games were played. In looking at these 10 games, the Dodgers scored 9 runs twice, 10 runs once, and 14 runs once. These would all be considered high-scoring Major League Baseball games, as National League teams in 2010 scored only 4.33 runs per game (baseball-reference.com). The 2.14 difference in runs scored per game between the model's prediction and the actual number of runs scored per game is quite large, however, it is likely due to the small sample of games started by Joe Torre's preferred lineup.

The most interesting applications of this model, however, are in comparing the number of runs scored per game by different lineup combinations of the same nine players. I use Joe Torre's most used lineup in 2010 (see **Figure 5**) as a benchmark, as this is a lineup that clearly reflects traditional batting order strategy. This lineup is inputted into the Matlab program as "123456789". I use this lineup to determine if traditional batting order strategy is best, or if there is another lineup that is superior in its number of runs scored per game.

Since each lineup consists of nine hitters, there are 9! = 362,880 possible batting orders. Because there are so many possible batting orders, it is unreasonable to enter each lineup possibility into the Matlab program to determine which lineup produces the most runs per game. Therefore, I am unable to find a true optimal batting order. I instead use a "guess and check" method for determining the "optimal" batting order in which I make educated changes to the benchmark lineup based on the following criteria. First, the better hitters should hit towards the top of the lineup and the poorer hitters should hit toward the bottom of the lineup. This is because hitters at the top of the lineup will obtain significantly more plate-appearances per game over the course of a season than hitters at the bottom of the lineup. Second, altering the placement of the best and worst hitter will have the biggest effect on the number of runs scored per game. Lastly, more runs will be scored when there are runners on base as the best hitters come to bat. Therefore, having solid hitters batting in the five spots before the best hitters will lead to more runs scored per game. This "guess and check" method is a reasonable strategy for determining the "optimal" batting order because there are numerous batting orders that are clearly less productive than the benchmark lineup and differ so greatly from traditional batting order strategy that a Major League manager would not consider them.

After making educated changes to the benchmark lineup using the "guess and check" strategy described above, I find the "optimal" batting order for the 2010 Dodgers to be "154326987". This lineup is given in **Figure 8**.

Figure 8 **Optimal Lineup**

1.	Rafael Furcal	SS
2.	James Loney	1B
3.	Manny Ramirez	LF
4.	Andre Ethier	RF
5.	Matt Kemp	CF
6.	Casey Blake	3B
7.	(Pitcher's Spot)	P
8.	Russell Martin	C
9.	Blake DeWitt	2B

The "optimal" batting order determined by the Matlab model for the 2010 Dodgers

The "optimal" batting order, as predicted by the model, produces 4.190908 runs per game. Thus, the difference between the model's prediction of the number of runs per game scored by the "optimal" batting order and the batting order used most often by the 2010 Dodgers is

$$4.190908 - 4.162438 = \mathbf{0.02847}$$

runs per game. Over the course of a 162 game season the "optimal" lineup will produce

$$0.02847 \times 162 = \mathbf{4.61214}$$

more runs than the lineup used most often by the 2010 Dodgers. This suggests that improving on traditional batting order strategy can result in the production of approximately 4.6 more runs per season.

In analyzing the "optimal" batting order predicted by the model, it is apparent why this lineup is more productive than traditional batting order strategy represented by the benchmark batting order. The most interesting deviation from traditional batting order strategy is the pitcher batting in the seventh spot as opposed to the ninth spot. This is more productive because it allows two much better hitters with significantly higher on base percentages to bat in the eighth and ninth spots. This allows solid hitters to be batting in the five spots preceding the best hitters, making it more likely that runners will be on base as the best hitters come to bat. Placing the pitcher in the seventh spot as opposed to the ninth spot in the lineup suggests that having runners on base as the best hitters come to bat affects run production more than placing the worst hitter last so that the worst hitter comes to bat least often. It is also interesting that the "optimal" batting order has James Loney in the second spot and Matt Kemp in the fifth spot, whereas the benchmark batting order has these players reversed. Loney is more of a contact hitter than Kemp, as Loney has a higher on base percentage and a lower slugging percentage. This suggests that having contact hitters with high on base percentages at the top of the lineup and power hitters with high slugging percentages in the middle of the order is most effective in producing runs.

Conclusion

Because Markov chains are the most mathematically perfect way to model the game of baseball, we can develop a Markov chain model of baseball that preforms quite well in predicting the expected number of runs scored per game by a given batting order. We can then use this model to compare various batting orders in terms of their run production to determine if we can improve on traditional batting order strategy. It turns out that according to a Markov chain model of baseball developed by Joel S. Sokol (2004) and data from the 2010 Los Angeles Dodgers, traditional batting order strategy is not optimal. The Sokol model predicts that an improved lineup will score almost 0.03 more runs per game than a benchmark batting order representing traditional batting order strategy. While this improved lineup will only result in approximately 4.6 more runs produced per season, if one run can make the difference between a win or a loss and one win can make the difference between making the playoffs and not, then the additional 4.6 runs a season is significant.

Appendix

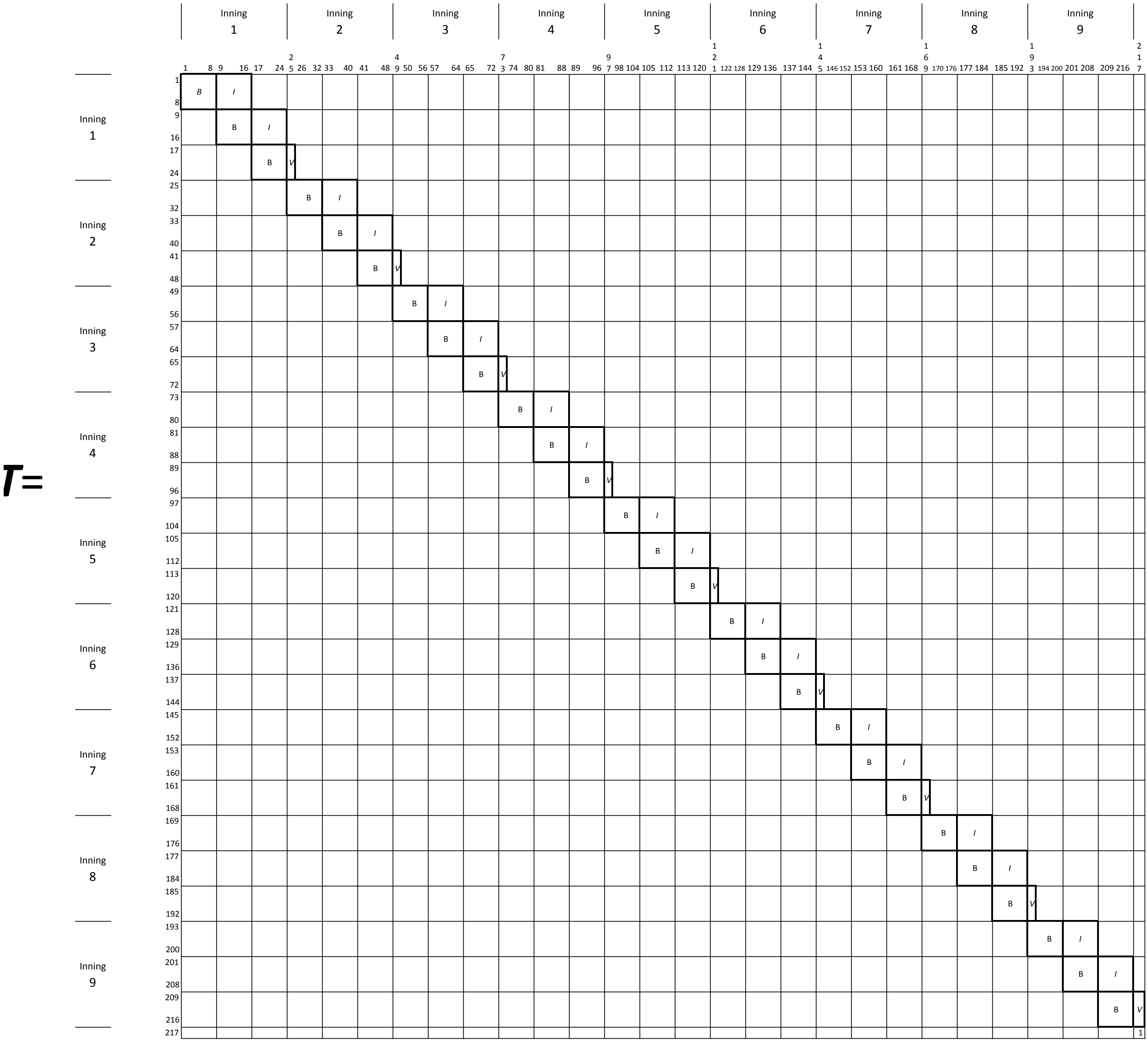
I. Transition Matrix, *T*

T =

		Inning 1			Inning 2			Inning 3			Inning 4			Inning 5			Inning 6			Inning 7			Inning 8			Inning 9			
		1–8	9–16	17–24	25–32	33–40	41–48	49–56	57–64	65–72	73–80	81–88	89–96	97–104	105–112	113–120	121–128	129–136	137–144	145–152	153–160	161–168	169–176	177–184	185–192	193–200	201–208	209–216	217
Inning 1	1–8	X_0	Y_0	Z_0																									
	9–16	[0]	X_1	Y_1																									
	17–24	[0]	[0]	X_2																									
Inning 2	25–32				X_0	Y_0	Z_0																						
	33–40				[0]	X_1	Y_1																						
	41–48				[0]	[0]	X_2																						
Inning 3	49–56							X_0	Y_0	Z_0																			
	57–64							[0]	X_1	Y_1																			
	65–72							[0]	[0]	X_2																			
Inning 4	73–80										X_0	Y_0	Z_0																
	81–88										[0]	X_1	Y_1																
	89–96										[0]	[0]	X_2																
Inning 5	97–104													X_0	Y_0	Z_0													
	105–112													[0]	X_1	Y_1													
	113–120													[0]	[0]	X_2													
Inning 6	121–128																X_0	Y_0	Z_0										
	129–136																[0]	X_1	Y_1										
	137–144																[0]	[0]	X_2										
Inning 7	145–152																			X_0	Y_0	Z_0							
	153–160																			[0]	X_1	Y_1							
	161–168																			[0]	[0]	X_2							
Inning 8	169–176																						X_0	Y_0	Z_0				
	177–184																						[0]	X_1	Y_1				
	185–192																						[0]	[0]	X_2				
Inning 9	193–200																									X_0	Y_0	Z_0	
	201–208																									[0]	X_1	Y_1	
	209–216																									[0]	[0]	X_2	
	217																												1

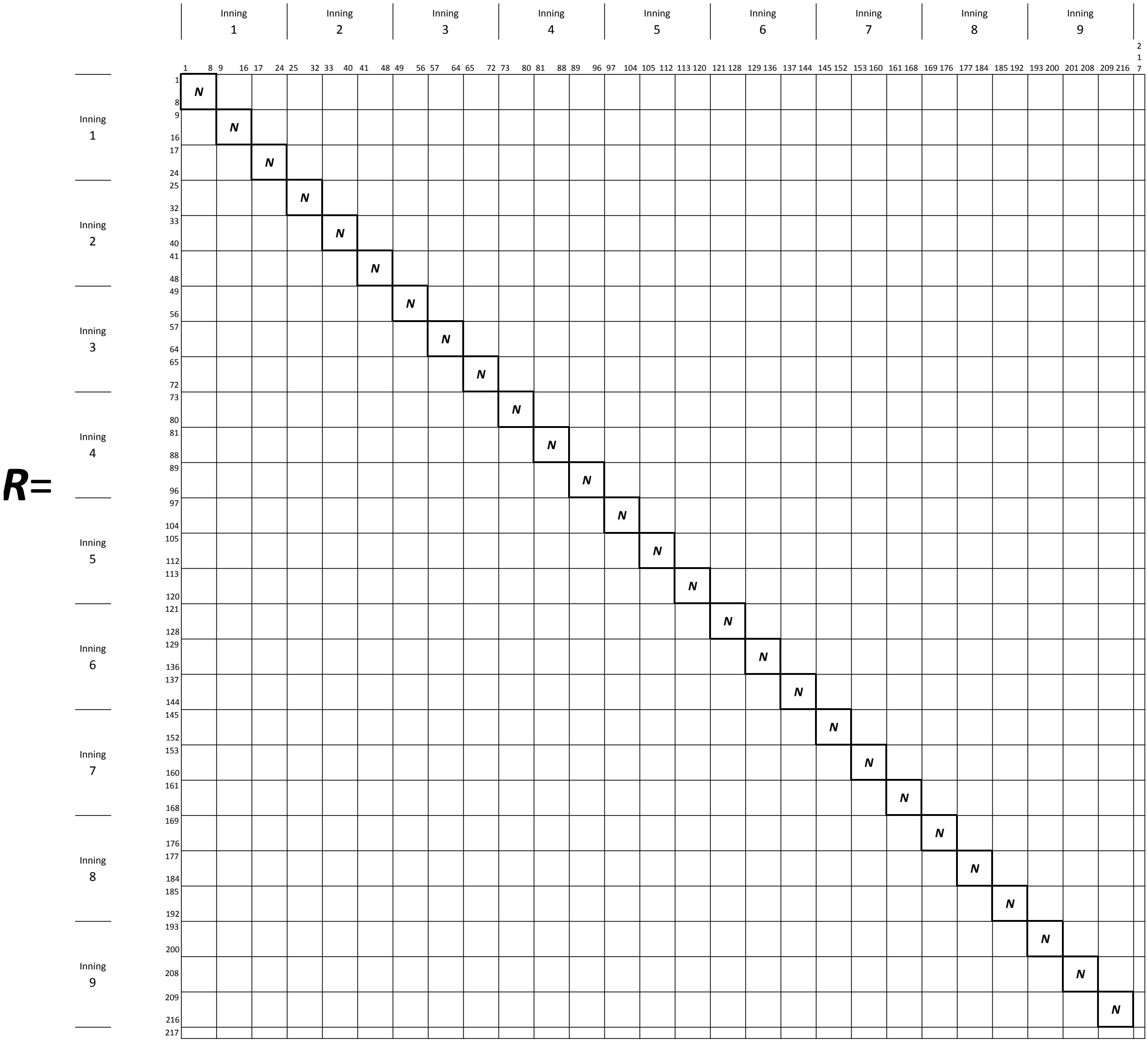
Note: All cells or blocks of cells that are empty it the figure above contain zeros.

II. Completed Transition Matrix, *T*



Note: All cells or blocks of cells that are empty it the figure above contain zeros.

III. Run-Value Matrix, *R*



Note: All cells or blocks of cells that are empty it the figure above contain zeros.

IV. Matlab Code

See following pages.

Works Cited

1. Albert, Jim. Teaching Statistics Using Baseball. Washington, DC: Mathematical Association of America, 2003. Print.

2. *Baseball-Reference.com - Major League Baseball Statistics and History*. Web. 01 Apr. 2011. <http://www.baseball-reference.com/>.

3. Bukiet, B., E. R. Harold, and J. L. Palacios. "A Markov Chain Approach to Baseball." Operations Research 45.1 (1997): 14-23. Print.

4. Howard, Ronald A. Dynamic Programming and Markov Processes. [Cambridge]: Technology of Massachusetts Institute of Technology, 1960. Print.

5. Kiyoshi Osawa, and Kento Aida. "Speed-up Techniques for Computation of Markov Chain Model to Find an Optimal Batting Order."

6. Ladany, Shaul P., and Robert Engel Machol. Optimal Strategies in Sports. Amsterdam: North Holland Pub., 1977. Print.

7. "Markov Chain." Wikipedia, the Free Encyclopedia. Web. 19 Oct. 2010. <http://en.wikipedia.org/wiki/Markov_chain>.

8. *MATLAB*. Natick, MA: MathWorks, 2006. Computer software.

9. Sokol, Joel S. "A Robust Heuristic for Batting Order Optimization Under Uncertainty." *Journal of Heuristics* 9 (2003): 353-70. Web.

10. Sokol, Joel S. "An Intuitive Markov Chain Lesson From Baseball." *INFORMS Transactions on Education* 5.1 (2004). Web.