A Markov Chain Approach to a Game of Chance

Maya Tsidulko

Department of Mathematical Sciences
Montana State University

May 2015

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Maya Tsidulko

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	Steve Cherry Writing Project Advisor
Date	Megan D. Higgs Writing Projects Coordinator

A Markov Chain Approach to a Game of Chance

Contents

1 Introduction

Stochastic processes is a study of how random systems evolve over time and/or space. One of the most important stochastic processes is a Markov process, named after the mathematician Andrey Markov. Markov's greatest insight was to show that complex phenomena can be described by the evolution of a "memoryless" system. Memoryless systems can be thought of as having finite memory: knowing the current state provides as much information about the future as knowing the entire past history. This property of conditional independence from the past and its extension to Markov chain Monte Carlo led to the development of some very powerful algorithms in computational science. Today, Markov chains are applied to solve many interesting problems in the fields of biology, engineering, information theory, physics, and much more.

In addition to scientific applications, we can use Markov chains to model games of chance. Such games, though intuitively completely random, can be fully described mathematically, leading to precise results. Moreover, analyzing such games can serve as a framework for learning the rich theory of Markov chains. As such, in the sections below, I provide and discuss the important definitions, theorems, and proofs that are studied in an introductory course on discrete time Markov chains. Focusing on one specific type of Markov chains, absorbing Markov chains, I build a theory of Chutes and Ladders, a simple board game that is popular among kids. Despite its simplicity, the mathematics of the game, with regards to Markov chains, are quite interesting. The objective of this project is to explore some of these mathematics, allowing us to answer some interesting questions.

Beyond an analytical exploration of the game for a single player, I turn to simulation techniques to evaluate some expected outcomes of the game for multiple players. Such simulation techniques are a powerful way to gain further insights into the Markov process of the game, especially when a more complicated theoretical framework is not as feasible.

2 Stochastic Processes

A stochastic process is a probabilistic counterpart to a deterministic process. Whereas a deterministic process can only evolve in one way, such as a solution to an ordinary differential equation, a stochastic process has inherent uncertainty. Even if the initial condition is known, there are several ways in which the process may evolve.

The mathematical definitions, theorems, and proofs that I provide in this paper can be found in Pinsky and Karlin's textbook "*An Introduction to Stochastic Modeling*" [1].

Definition 2.1 A *stochastic process* is a family of random variables X_t, where t is a parameter running over a suitable index set T. In a common situation, the index t corresponds to discrete units of time and the index set is $T = \{0, 1, 2, ...\}$. In this case, X_t might represent the outcomes at successive tosses of a coin, repeated responses of a subject in a learning experiment, or successive observations of some characteristics of a certain population. Stochastic processes for which $T = [0, \infty)$ are particularly important in applications. Here t often represents time, but different situations also frequently arise. For example, t may represent distance from an arbitrary origin, and X_t may indicate the number of defects in the interval $(0, t]$ along a thread, or the number of cars in the interval $(0, t]$ along a highway. Stochastic processes are distinguished by their *state space*, or by the range of possible values for the random variable X_t, by their index set T, and by the dependence relations among the random variables X_t.

3 Markov Chains

Markov chains are named after Andrey Andreevich Markov (1856 – 1922), a Russian mathematician and a student of the great Pafnuty Lvovich Chebyshev. Markov published hundreds of papers on number theory, continuous fraction theory, differential equations, probability theory, and statistics. He introduced his chains, to be named "Markov chains", in a 1907 paper, which was included in his textbook "Calculus of Probabilities". On January 23, 1913, Markov presented an extension to this paper at the Imperial Academy of Sciences in St. Petersburg, where he was a distinguished professor, describing his careful enumerations of vowels and consonants in Alexander Pushkin's very treasured poem "Evgeniy Onegin". This became Markov's first application of his own chains. In this example, Markov was interested in deriving the probabilities of encountering different sequences of letter pairs in the poem: vowel following a vowel, vowel following a consonant, consonant following a vowel, and consonant following a consonant. The description of his example, as well as the original page of the matrices that he drew are provided in Figure 1.

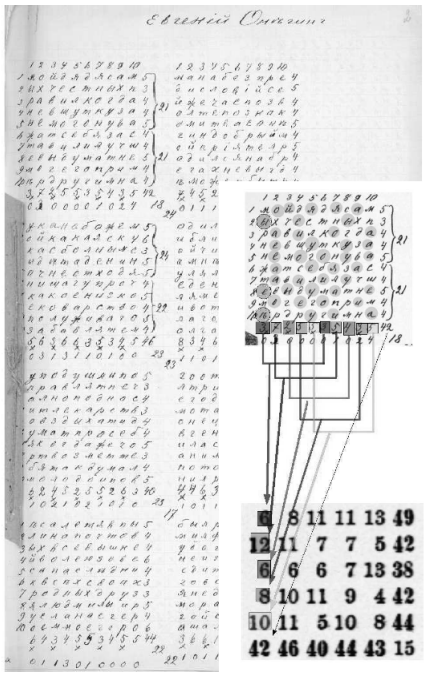


Figure 1: *Left background: The first 800 letters of 20,000 total letters compiled by Markov and taken from the first one and a half chapters of Pushkin's poem Evgeniy Onegin. Markov omitted spaces and punctuation characters as he compiled the cyrillic letters from the poem. Right foreground: Markov's count of vowels in the first matrix of 40 total matrices of 10x10 letters. The last row of the 6x6 matrix of numbers can be used to show the fraction of vowels appearing in a sequence of 500 letters. Each column of the matrix gives more information. Specifically, it shows how the sums of counted vowels are composed by smaller units of counted vowels. Markov argued that if the vowels are counted in this way, then their number proved to be stochastically independent.*

A Markov process is defined in greater detail below.

Definition 3.1 A Markov process (X_t) is a stochastic process with the property that, given the value of X_t, the values of X_s for $s > t$ are *not* influenced by the vales of X_u for $u < t$. That is, the probability of any particular future behavior of the process, when its current state is known exactly, is not altered by additional knowledge concerning its past behavior. A discrete-time Markov chain is a Markov process whose state space is a finite or countable set, and whose (time) index set is $T = (0, 1, 2, ...)$. In formal terms, the Markov property is that

$$Pr\{X_{n+1} = j | X_0 = i_0, .., X_{n-1} = i_{n-1}, X_n = i\} = Pr\{X_{n+1} = j | X_n = i\}$$

The probability of X_{n+1} being in state j given that X_n is in state i is called the one-step transition probability and is denoted by $P_{ij}^{n,n+1}$.

That is,

$$P_{ij}^{n,n+1} = Pr\{X_{n+1} = j | X_n = i\}$$

The notation emphasizes that the transition probabilities are functions not only of the initial and final states but also of the time of transition as well. When the one-step transition probabilities are independent of the time variable n, the Markov chain is said to have *stationary transition probabilities*. Then $P_{ij}^{n,n+1} = P_{ij}$ is independent of n, and P_{ij} is the conditional probability that the state value undergoes a transition from i to j in one trial. It is customary to arrange these numbers P_{ij} in a square *matrix*,

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & \cdots \\ P_{10} & P_{11} & P_{12} & P_{13} & \cdots \\ P_{20} & P_{21} & P_{22} & P_{23} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ P_{i0} & P_{i1} & P_{i2} & P_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

and refer to $\mathbf{P} = P_{ij}$ as the Markov matrix or *transition probability matrix* of the process.

The i^{th} row of $\mathbf{P}$, for $i = 0, 1, ...$, is the probability distribution of the values of X_{n+1} under the condition that $X_n = i$. If the number of states is finite, then $\mathbf{P}$ is a square matrix whose order is equal to the number of states. The quantities P_{ij} satisfy the following

$$P_{ij} \geq 0 \text{ for all } i, j = 0, 1, 2, ..., \sum_{j=0}^{\infty} P_{ij} = 1$$

A Markov process is completely defined once its transition probability matrix and the probability distribution of the initial state X_0 are specified. I outline the proof of this result below.

Proof. By definition of conditional probability, *Proof.* By definition of conditional probability,

$$Pr\{X_n = i_n | X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\} = \frac{Pr\{X_n = i_n, X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\}}{Pr\{X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\}} \tag{1}$$

Rearranging the terms,

$$\begin{aligned} Pr\{X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i_n\} = {} & Pr\{X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\} \\ & \times Pr\{X_n = i_n | X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\} \end{aligned} \tag{2}$$

By definition of a Markov process,

$$Pr\{X_n = i_n | X_0 = i_0, X_1 = i_1.., X_{n-1} = i_{n-1}\} = Pr\{X_n = i_n | X_{n-1} = i_{n-1}\} = P_{i_{n-1}, i_n} \tag{3}$$

Substituting (3) into (2) gives

$$Pr\{X_0 = i_0, X_1 = i_1, ..., X_n = i_n\} = Pr\{X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}\}P_{i_{n-1},i_n} \tag{4}$$

Repeating the argument $n-1$ additional times, (4) becomes

$$Pr\{X_0 = i_0, X_1 = i_1, ..., X_n = i_n\} = p_{i_0}P_{i_0,i_1} \cdots P_{i_{n-2},i_{n-1}}P_{i_{n-1},i_n} \tag{5}$$

This shows that all finite-dimensional probabilities are specified once the transition probabilities and initial distribution are given. That is, we can easily find the joint probability of the process being at specific states at specific times.

The analysis of a Markov chain concerns mainly the calculation of the probabilities of the possible realizations of the process. Central in these calculations are the n-step transition probability matrices $\mathbf{P^{(n)}} = \left\| P_{ij}^{(n)} \right\|$, where $P_{ij}^{(n)}$ denotes the probability that the process goes from state i to state j in n transitions. Formally,

$$P_{ij}^{(n)} = \Pr\{\mathrm{X}_{m+n} = j | X_m = i\}$$

I show how to compute the n-step transition probabilities in section 3.1.1.

3.1 Overview of Markov Chains

Before delving into the theory of Chutes and Ladders, I classify the states of a Markov chain, look at the long run behavior of absorbing chains, and discuss the theorems that are needed to analyze the game analytically. In section 4.4, I examine these properties and theorems as they apply to the game.

Definition 3.1.2 An *absorbing state* is one in which, when entered, it is impossible to leave. That is, $p_{ii} = 1$. A state that is not an absorbing state is called a *transient* state. An *absorbing* Markov chain is a Markov chain with absorption states and with the property that it is possible to transition from any state to an absorbing state in a finite number of transitions.

Definition 3.1.3 State j is said to be *accessible* from state i if $P_{ij}^{(n)} > 0$ for some integer $n \geq 0$; i.e., state j is accessible from state i if there is positive probability that state j can be reached starting from state i in some finite number of transitions. Two states i and j, each accessible to each other, are said to *communicate*, and we write $i \leftrightarrow j$. If two states i and j do not communicate, then either

$$P_{ij}^{(n)} = 0 \text{ for all } n \geq 0$$

or

$$P_{ji}^{(n)} = 0 \text{ for all } n \geq 0$$

or both relations are true. We can partition the totality of states into equivalence classes. The states in an equivalence class are those that communicate with each other. The Markov chain is *irreducible* if the equivalence relation induces only one class. That is, the process is irreducible if all states communicate with each other.

Definition 3.1.4 A state i is *recurrent* if and only if

$$\sum_{n=1}^{\infty} P_{ii}^{(n)} = \infty$$

Equivalently, state i is *transient* if and only if

$$\sum_{n=1}^{\infty} P_{ii}^{(n)} < \infty$$

To examine the existence of a limiting probability distribution, I use the following definition of a *regular* Markov chain.

Definition 3.1.5 Suppose that a transition probability matrix $\mathbf{P} = \left\| P_{ij} \right\|$ on a finite number of states labeled $0, 1, ..., N$ has the property that when raised to some power k, the matrix $\mathbf{P}^k$ has all of its elements strictly positive. Such a transition probability matrix , or the corresponding Markov chain, is called *regular*. the most important fact concerning a regular Markov chain is the existence of a *limiting probability distribution* $\pi = (\pi_0, \pi_1, ..., \pi_N)$, where $\pi_j > 0$ for $j = 0, 1, ..., N$ and $\sum_j \pi_j = 1$, and this distribution is independent of the initial state. Formally, for a regular transition probability matrix $\mathbf{P} = \left\| P_{ij} \right\|$, we have the convergence

$$\lim_{n\to\infty} P_{ij}^{(n)} = \pi_{ij} > 0 \quad \text{for} \quad j = 0, 1, ..., N$$

or, in terms of the Markov chain X_n,

$$\lim_{n\to\infty} \{X_n = j | X_0 = i\} = \pi_{ij} > 0 \quad \text{for} \quad j = 0, 1, ..., N$$

This convergence means that, in the long run ($n \to \infty$), the probability of finding the Markov chain in state j is approximately π_j no matter which state the chain began in at time 0.

3.1.1 First Step Analysis

A significant number of functionals on a Markov chain can be evaluated by a technique called *first step analysis*. This method proceeds by breaking down the possibilities that can arise at the end of the first transition, and then invoking the law of total probability coupled with the Markov property to establish a characterizing relationship among the unknown variables. The derivations below, along with their details, can be found in section 3.7 of Pinsky and Karlin [1].

Proposition 3.1.1.1. The probability of going from state i to state j in precisely n steps, $p_{ij}^{(n)}$, is the i, j^{th} entry of $\mathbf{P^n}$.

Proof. The probability of going from state i to state j in two steps is the sum of the probability of going from step i to step 1, then from step 1 to step j, the probability of going from step i to step 2, then from step 2 to step j, and so on. Thus, letting P be a $w \times w$ matrix,

$$P_{ij}^{(2)} = p_{i1}p_{1j} + \mathrm{p}_{i2}p_{2j} + ... + p_{iw}p_{wj} = \sum_{r=1}^{w} p_{ir}p_{rj}$$

This parallels the definition of matrix multiplication. That is, it is evident that $p_{ij}^{(2)}$ is the i, j^{th} entry of $\mathbf{P^2} = \mathbf{P} \times \mathbf{P}$. This proves the proposition for the $n = 2$ case; the proofs for greater n follow by induction.

Often, we are interested in the *time* the system goes from some initial state to the absorbing state, or *time to absorption*. Time to absorption is a random variable since the probability of the Markov chain being absorbed at a given time varies according to the possibilities arising at the end of each transition. Consider the states $0, 1, ..., r-1$ to be transient in that $P_{ij}^{(n)} \to 0$ as $n \to \infty$, and the states $r, ..., N$ to be absorbing in that $P_{ii} = 1$ for $r \leq i \leq N$.

Definition 3.1.1.1. Let T be the time of absorption, the number of steps required to reach an absorbing state r through N. Formally,

$$T = min\{n \geq 0 : r \leq X_n \leq N\}$$

For an absorbing Markov chain, we can define a submatrix $\mathbf{Q}$ of $\mathbf{P}$ as the transition matrix between non-absorbing states. That is, $\mathbf{Q}$ is $\mathbf{P}$, but with the rows and columns corresponding to absorbing states removed.

The transition matrix has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

It is straight forward to show by induction that

$$\mathbf{P^{(n)}} = \begin{bmatrix} \mathbf{Q^{(n)}} & \mathbf{(I + Q + ... + Q^{n-1})R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

Let $\mathbf{W^{(n)}}$ be a matrix whose elements $W_{ij}^{(n)}$ contain the mean number of visits to state j up to state n for a Markov chain starting in state i. Formally,

$$W_{ij}^{(n)} = E[\sum_{i=0}^{n} \mathbf{1}\{X_l = j\} | X_0 = i\,]$$

where

$$\mathbf{1}\{\mathrm{X}_l = j\} = \begin{cases} 1 & \text{if } X_l = j, \\ 0 & \text{if } X_l \neq j. \end{cases}$$

Since $E[\mathbf{1}\{X_l = j\}|X_0 = i] = Pr\{X_l = j|X_0 = i\} = P_{ij}^{(l)}$ by definition of expected value, and since the expected value of a sum is the sum of the expected values, we can obtain the following

$$W_{ij}^{(n)} = \sum_{i=0}^{n} E[\mathbf{1}\{X_l = j\}|X_0 = i] = \sum_{i=0}^{n} Q_{ij}^{(l)} \text{ for transient states } i \text{ and } j$$

In matrix notation, $\mathbf{Q^{(0)}} = \mathbf{I}$, and since $\mathbf{Q^{(n)}} = \mathbf{Q^n}$, the nth power of $\mathbf{Q}$, then it can be shown that

$$\mathbf{W^{(n)}} = \mathbf{I} + \mathbf{QW^{(n-1)}}$$

This equation asserts that the mean number of visits to state j in the first n stages starting from the initial stage i includes the initial visit if $i = j$ plus the future visits during the $n - 1$ remaining stages weighed by the appropriate transition probabilities. Taking the limit,

$$\lim_{n\to\infty} W_{ij}^{(n)} = E[\text{Total visits to } j | X_0 = i], \qquad 0 \leq i,\ j < r,$$

and the matrix equation

$$\mathbf{W} = \mathbf{I} + \mathbf{Q} + \mathbf{Q^2} + \cdots$$

and

$$\mathbf{W} = \mathbf{I} + \mathbf{QW}$$

Rewriting this equation in the form

$$\mathbf{W} - \mathbf{QW} = (\mathbf{I} - \mathbf{Q})\mathbf{W} = \mathbf{I}$$

We see that $\mathbf{W} = (\mathbf{I} - \mathbf{Q})^{-\mathbf{1}}$, the inverse matrix to $\mathbf{I} - \mathbf{Q}$. The matrix $\mathbf{W}$ is often called the *fundamental matrix* associated with $\mathbf{Q}$.

It then follows that to compute the expected time to absorption given that the chain starts in state i, we sum across the columns j for each row i of $\mathbf{W}$. Formally,

$$\sum_{i=0}^{n} W_{ij} = \text{ Expected time to absorption for transient states } j = 0, 1, ..., r - 1$$

We see that, once we have $\mathbf{W}$, it is easy to compute the expected time to absorption for all starting states of the Markov process.

4 Chutes and Ladders

4.1 Rules

Chutes & Ladders, or Snakes & Ladders, originated in India, where the game represented a player's journey through life that was complicated by virtues (ladders), and vices (snakes). The game of Chutes & Ladders is played on a 10x10 gridded board which is numbered, sequentially, in a zig-zag pattern from 1 (the start), in the lower left corner, to 100 (the end), in the top left corner, as shown below. There can be as many players as desired with the actions of each player being independent from the actions of the other player. The player begins off of the board, at a figurative square zero. The player then rolls a six-sided die, and advances the number of spaces shown on the die. If after advancing a player lands on a chute or a ladder, they slide down or climb up the chute or the ladder, respectively.

Furthermore, the player must land exactly on square 100 to win. If a player rolls a die that would advance them beyond square 100, they stay at the same place.

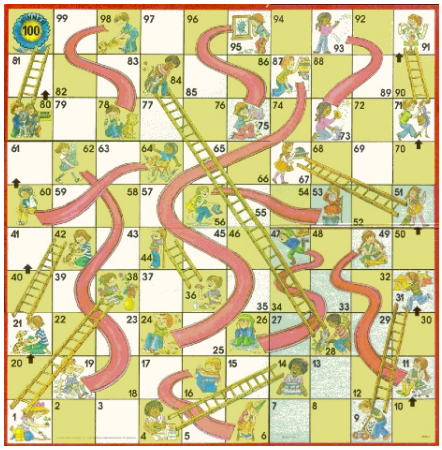


Figure 2: *A standard Chutes & Ladders game board.*

Chutes and Ladders can be represented as an absorbing Markov chain, where each square is a state, with the final square of 100 as the only absorbing state. This absorbing state is *recurrent*, while the remaining 99 states are *transient*. Furthermore, we can break the chain into *communicating classes*. Since not all 100 states communicate with each other, this Markov chain is *reducible*. Finally, this Markov chain is a *regular* chain with a limiting probability distribution given by

$$\pi_i = (0, 0, ..., 1) \text{ for states } i = 0, 1, ..., 100$$

The standard board of the game can be denoted as $\mathbf{R}$, representing all of the transitions. Specifically,

$\mathbf{R} = \{(1, 38), (4, 14), (9, 31), (21, 42), (28, 84), (36, 44), (51, 67), (71, 91), (80, 100), (16, 6),$
$(47, 26), (49, 11), (56, 53), (62, 19), (64, 60), (87, 24), (93, 73), (95, 75), (98, 78)\}$.

For example, if a player lands on square 1 upon first roll, they instantaneously transition to state 38, gaining 37 squares. If, however, upon getting closer to the winning square 100, a player lands on square 95, they move back to square 75, losing 20 positions.

4.2 Transition Matrix

To apply the theory of Markov chains, we first need to define the transition matrix. Since figurative square zero is a state, the first row in the transition matrix is the row representing state 0. Below is a summary of a *naive* transition matrix, of order 101 by 101.

$$\mathbf{P}_{101\times101} = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ 97 \\ 98 \\ 99 \\ 100 \end{array} \begin{array}{c} \begin{array}{ccccccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & ... & 14 & ... & 38 & ... & 78 & ... & 97 & 98 & 99 & 100 \end{array} \\ \begin{pmatrix} 0 & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & ... & 0 & ... & 0 & ... & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & ... & 0 & ... & \frac{1}{6} & 0 & \frac{3}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & ... & 0 & ... & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & ... & 0 & ... & 0 & ... & 0 & 0 & \frac{5}{6} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & ... & 0 & ... & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

It is important to note that the states, or grid cells, representing the bottom of ladders or top of chutes have zero probability associated with them, since upon landing there, the player instantaneously transitions to another state. Thus, in order to be considered a proper transition matrix., the matrix needs to be reduced to a 82×82 dimension. Computationally, though, the two would lead to the same results. Moreover, defining a naive transition matrix of 101×101 makes interpretation of states easier, as they match the game board squares.

4.3 Questions of Interest

Applying theory to computation, I would like to answer the following questions regarding the Markov process of Chutes & Ladders.

1. How does the Markov chain *behave over time*?
2. What is the probability distribution for *time to absorption*?
3. What is the *expected time to absorption* and *variance of time to absorption*?
4. What is the *expected number of visits to transient states* & the *expected time to absorption at each state*?

These questions are answered analytically using theoretical results discussed in section 3.1.1. In addition, I have two questions of interest that are answered using simulation techniques.

5. What is the probability distribution for *time to absorption* using simulation?
6. How does the distribution of the difference between the state of the winner of the game, necessarily 100, and the state of the second place player change as as number of players increases?

Finally, I would like to answer the following question.

7. What is the estimated approximating distribution to the discrete random variable time to absorption?

To answer the analytical questions, I discuss the theory as it applies to the game. In section 5, I discuss my simulation results. Answering the last question would allow us to find summaries of the distribution of time to absorption with little computation.

4.4 Theory

The probability mass function for T could be computed as follows. First, by Proposition 3.1.2, the probability that the chain is in state i at time t is the i^{th} entry in the vector

$$\mathbf{u^{(t)}} = \mathbf{uP^t}$$

where $\mathbf{u}$ is defined as the *initial state*, or initial probability distribution of all states at the start of the game. Since the player is off the board at state 0, $\mathbf{u}$ is defined as the following vector

$$\mathbf{u} = (1, 0, ..., 0) \text{ for } i = 0, 1, ..., 100$$

Note that $\mathbf{u^{(t)}}$ gives us the chain behavior over time given that the game starts off the board at state 0, answering question 1. Further, we define a *final distribution of states* $\mathbf{e}$ representing the probability distribution of the states required to end the game. Since the player must reach state 100 to win the game, $\mathbf{e}$ is defined as the following vector

$$\mathbf{e} = (0, 0, ..., 1) \text{ for } i = 0, 1, ..., 100$$

Then,

$$F(t) = P(T < t) = \mathbf{u^{(t)}e} \text{ for } t = 1, 2, ..., n,$$

and we get the *cumulative distribution function* for T. We can then compute the *probability mass function* as following

$$f(t) = F(t) - F(t-1) \text{ for } t = 1, 2, ..., n$$

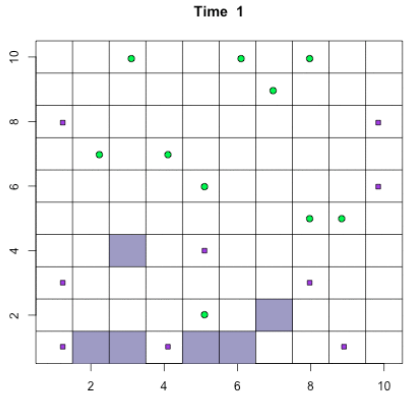
giving us the probability distribution for T, and providing the computation necessary to answer question 2.

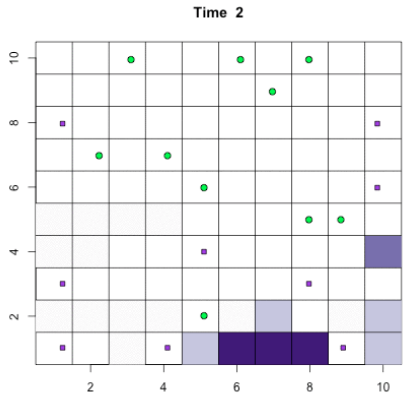
I then define the $\mathbf{Q}$ matrix. Since only square 100 is an absorbing state, the $\mathbf{Q}$ matrix is simply the transition matrix, with the row 100 and column 100 removed. Thus, $\mathbf{Q}$ is a 100×100 matrix. Once I have defined the $\mathbf{Q}$ matrix, I can compute $\mathbf{W}$, answering questions 3 & 4. I answer questions 5 & 6 by simulating the game for 1 ,2, 3, 5, 10, 15, and 20 players. To answer question 7, I write a loss function to find the best parameters of a proposed approximating distribution of T.

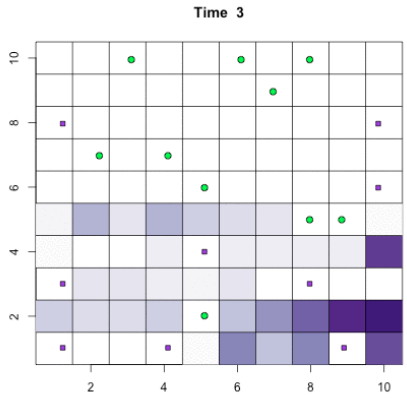
5 Results

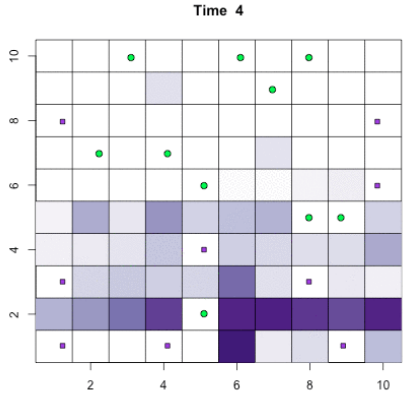
5.1 Analytical Results

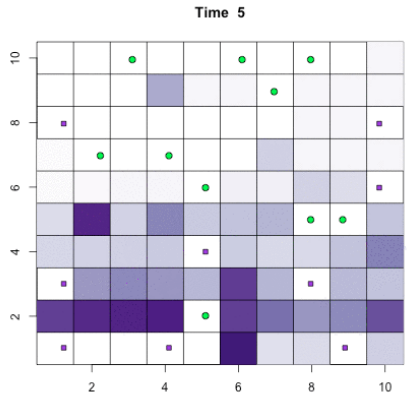
I answer most of my questions of interest by applying the above theory and exploring the results graphically. First, I analyze how the Markov chain behaves over time for a single player (Figure 3).

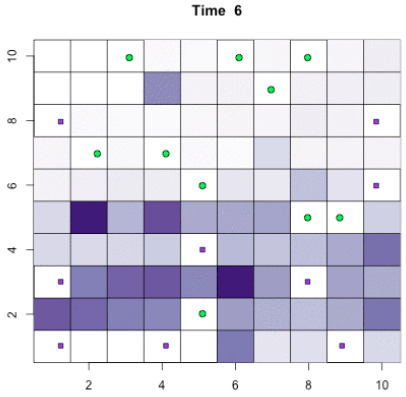

Time 1
2
4
6
8
10

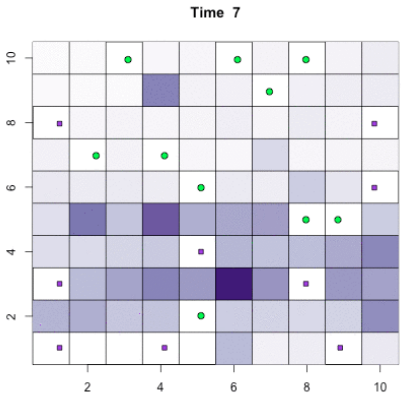

Time 2
2
4
6
8
10

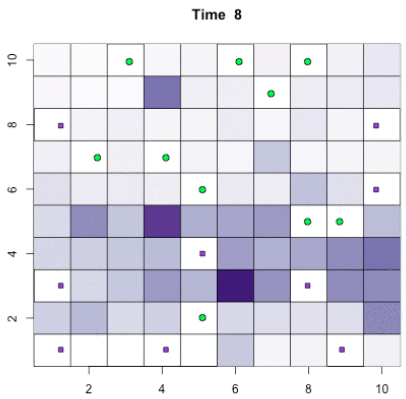

Time 3
2
4
6
8
10


Time 4
2
4
6
8
10


Time 5
2
4
6
8
10


Time 6
2
4
6
8
10


Time 7
2
4
6
8
10


Time 8
2
4
6
8
10

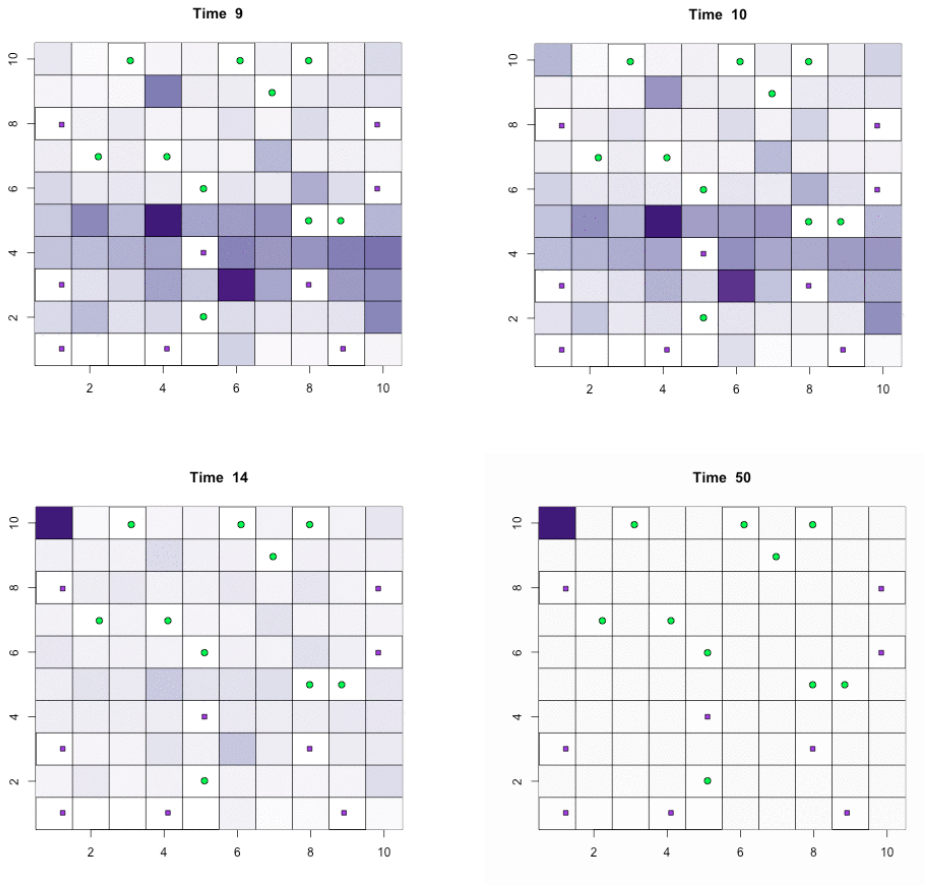


Figure 3: *The probability distribution over all 100 states for times* $1, 2, ..., 14$ *and* 50. *Times denote turns, or rolls of the die. The small circles and squares represent the chutes and ladders on the board, while the depth of the shaded squares represents the magnitude of the probability that a player lands at that state at the specific time.*

At time 1, the process is uniformly distributed over states 2, 3, 5, 6, 14, and 38. We then see how the process evolves over time. Specifically, as we progress in time, the probability of being in the transient states gets smaller (fainter squares), while the probability of being at the absorbing state 100 gets larger (darker squares), supporting the form of the limiting probability distribution stated in section 5. We note that the minimum number of turns needed to reach state 100 is 7. The probability of the game being over at time 7, however, is very low.

Further, the full probability distribution of *time to absorption*, computed using theory is shown below.

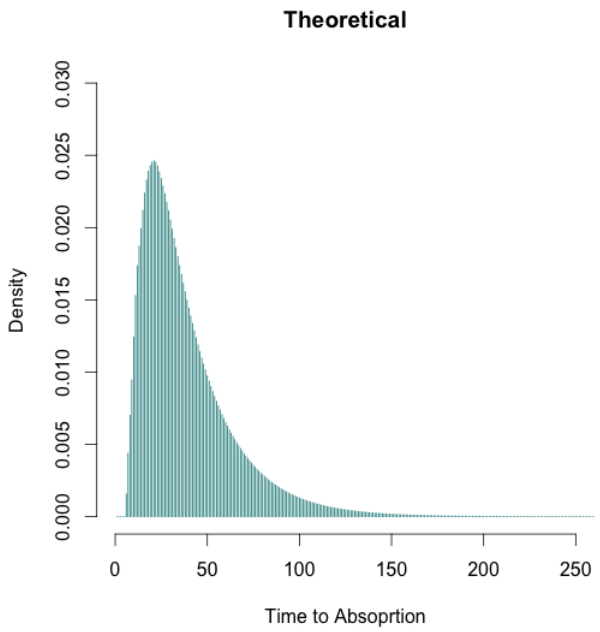


Figure 4: *The probability mass function of time to absorption computed analytically.*

The distribution of time to absorption is right skewed, with *expected time to absorption* of 39.59, *variance of time to absorption* of 636.87, and *median* of 32. Summaries of the distribution, including quantiles, are provided below. Theoretically, there is no upper bound to the game. The probability of the game going beyond ≈ 200 turns for a single player, however, is practically zero.

	Time to Absorption
Expected	≈ 39.5
Variance	≈ 636.9
Median	32
Mode	26
Minimum	7
10^{th} %ile	14
25^{th} %ile	21
75^{th} %ile	49
95^{th} %ile	89

The *expected time to absorption at each state* is shown below. We see that as the game evolves, the expected time to absorption generally decreases, with the exception of a few spikes occurring at the states of the chutes. The most noticeable spike starts at state 75, a state which is within reachable distance of the longest chute in the game, $\{81, 24\}$.

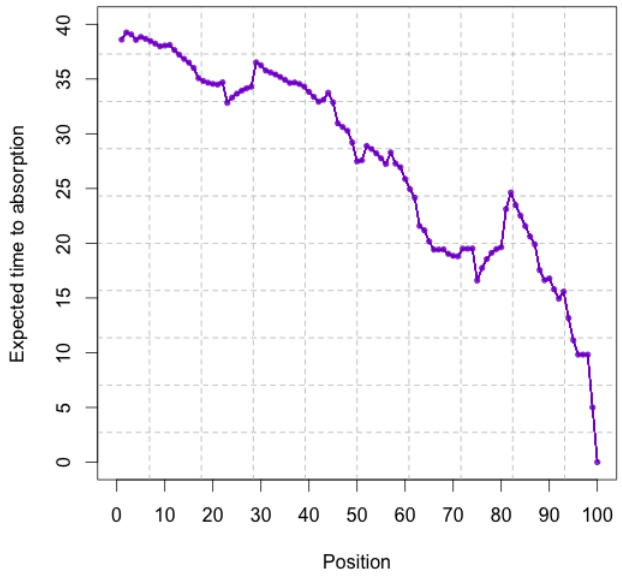


Figure 5: *The expected time to absorption at each state (position) of the player.*

We can also examine the standard deviation of time to absorption at each state.

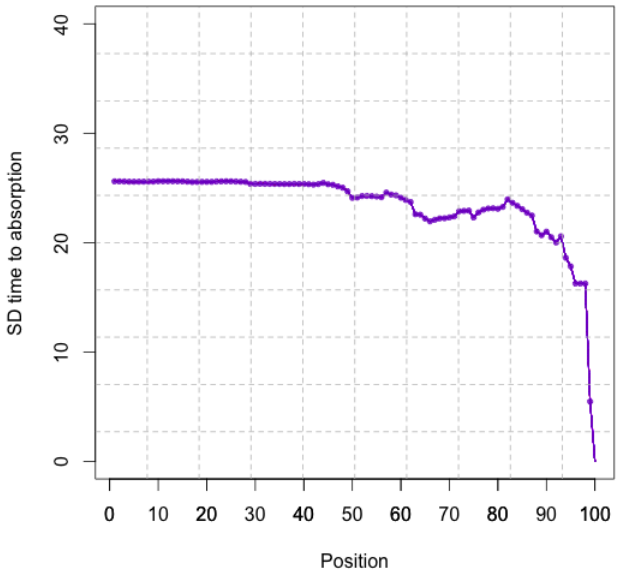


Figure 6: *The standard deviation of time to absorption at each state (position) of the player.*

5.2 Simulation Results

The full probability distribution of *time to absorption*, computed through simulating 10,000 games is shown below. We see that it fits the theoretical distribution in Figure 4 well.

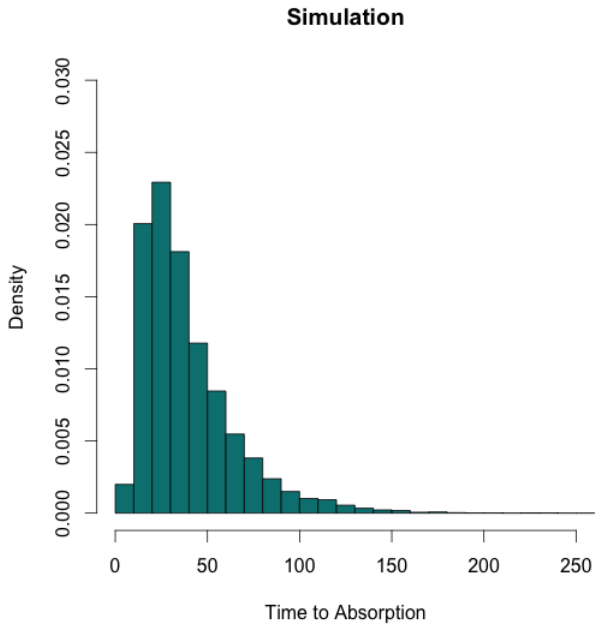
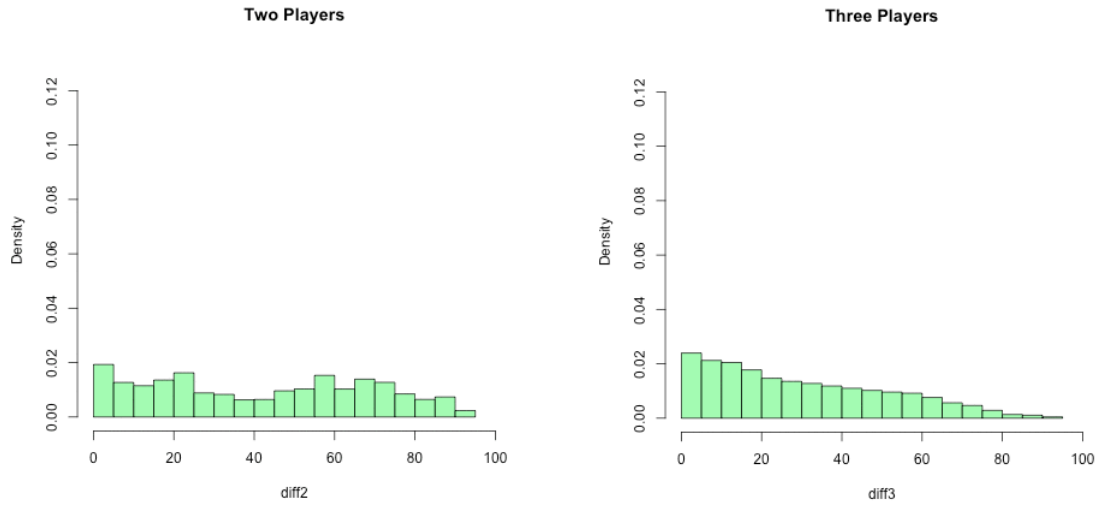


Figure 7: *The probability mass function of time to absorption computed through simulation of 10,000 games.*

Through simulations of the game, we consider a distribution of the difference between the state of the winner of the game, necessarily at 100, and the state of the second place player change as we increase the number of players in the game. Intuitively, the games should get closer in "scores" as the number of players increases since there is a higher probability of each player's own Markov chain evolving in a different manner. For two players, this distribution is close to being uniform over states 1 to 100. As the number of players increases, the distribution appears to have more and more of its mass between differences of 0 and 5.



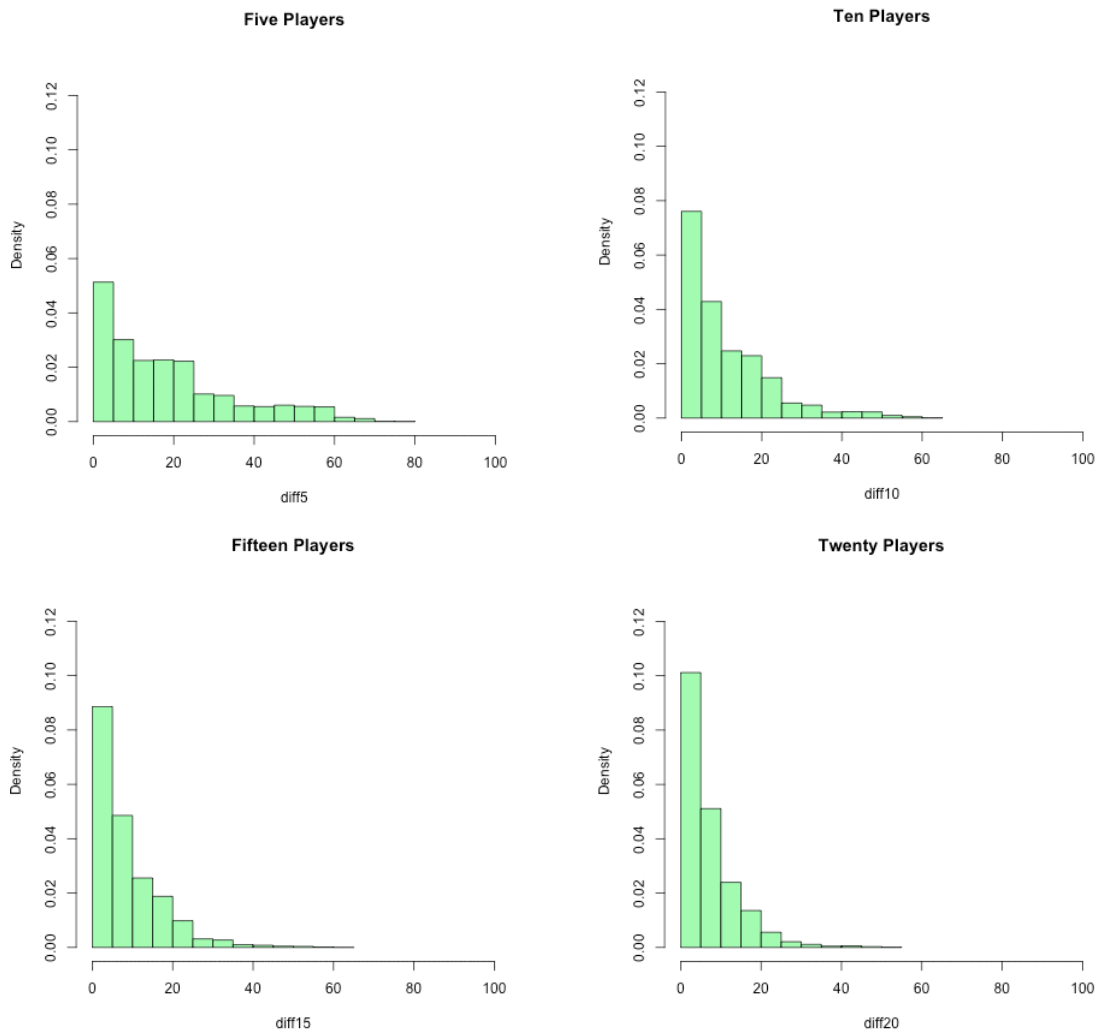


Figure 8: *The distribution of the difference between the state of the winner of the game, necessarily at 100, and the state of the second place player for 10,000 games.*

5.3 Approximating Parametric Distribution for T

Finally, I wrote a loss function to estimate an approximating distribution for T. I explored two discrete distributions, the Geometric and Negative Binomial, and two continuous distributions, the Gamma and Log-normal. I was unable to estimate parameters for the discrete distributions due to problems with the zero probability for $t \in (0, 7)$. I chose the log normal family of continuous distributions because of its parameter support on $(0, \infty)$ and flexibility in skewness. The plot of the approximating curve, along with the summaries of the distribution, including quantiles, are provided below. Comparing the summaries of the Log-normal distribution with the summary of the computed theoretical distribution, we see that they match pretty closely, with the exception of mode and minimum.

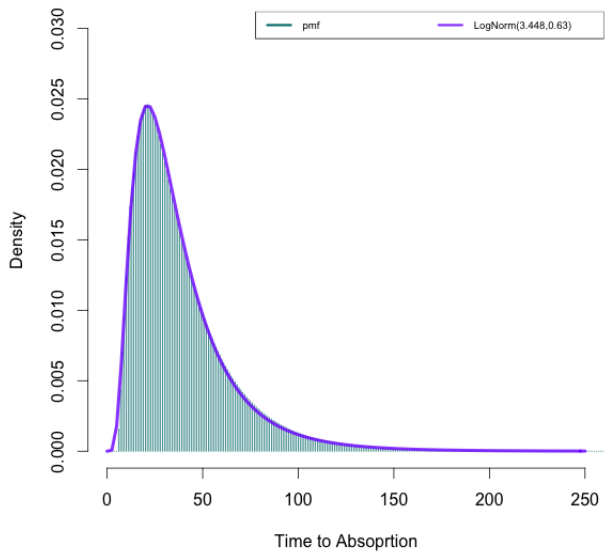


Figure 9: *The approximating continuous distribution to the random variable time to absorption was found to be the log normal distribution with parameters (3.448 ,0.63).*

	Theoretical	**Log-Norm(3.448,0.63)**
Expected	≈ 39.5	38.33832
Variance	≈ 636.9	716.1103
Median	32	31.43745
Mode	26	21.13858
Minimum	7	0.7184793
10^{th} %ile	14	14.02193
25^{th} %ile	21	20.55433
75^{th} %ile	49	48.08299
95^{th} %ile	89	88.61116

6 Reflection

This project allowed me to explore a topic that has interested me for quite some time. In many regards, I have only gotten a taste of Markov chains' rich theory and application, leaving me wanting to learn more. Beyond the project's focus on absorbing Markov chains, it let me think of and research other types of Markov chains and situations where they are powerful. For example, an area of Markov chains that I find very interesting is *Markov chain Monte Carlo* (McMC), which is a method for integrating a function that might not have a closed form solution, and is largely used in Bayesian inference. Having used McMC methods before, I would like to explore the details of its theory in the future.

7 Appendix

```
#####################################################################
###Function to simulate the game for one or multiple player(s)####
#####################################################################

chutes<-function(nplayers=1){
  transitions <- rbind(c(1, 38),c(4, 14),c(9, 31),c(16, 6),c(21, 42),c(28, 84),c(36, 44),
                       c(48, 26),c(49, 11),c(51, 67),c(56, 53),c(62, 19),c(64, 60),c(71, 91),
                       c(80, 100),c(87, 24),c(93, 73),c(95, 75),c(98, 78))

    ## set up the board, accounting for overshooting by including 6 extra squares

  transmat <- 1:106

  ## transition the player accordingly when they hit a chute or a ladder

  names(transmat) <- as.character(1:106)
  transmat[transitions[,1]] <- transitions[,2]

  ## initial position is off the board at 0
  last <- 0

  ## setup a storage vector of current position and history to keep track of
  current <- NULL
  history <- NULL

   ## while statement to keep playing until a player reaches state 100

  while(all(current < 100)) {

    current <- last + sample(1:6, nplayers, repl=TRUE)

     ## assign current position based on the transitions

    current <- transmat[current]

      ## if a player overshoots, they stay at their last position

    if(any(current > 100))
      current[current > 100] <- last[current > 100]

     ## last position at time t-1 is the current position at time t

    last <- current

     ## history of positions by row

    history<-rbind(history,current)
  }

  ## return history vector

  return(history)
}
```

```
## repeat function for 1,2,3,5,10,15 players 10000 times

sim1<-replicate(10000,chutes(1))
sim2<-replicate(10000,chutes(2))
sim3<-replicate(10000,chutes(3))
sim5<-replicate(10000,chutes(5))
sim15<-replicate(10000,chutes(15))
sim20<-replicate(10000,chutes(20))

## storage vector for game length
length.game<-NULL

## loop to compute length of simulated games

for (i in 1:10000){
  length.game[i]<-length(sim1[[i]])
}

## plot histogram of game lengths

hist(length.game,freq=FALSE,nclass=25,col=rgb(0.5, 1, 0.5, 0.7),
     main="Simulation of 10,000 games",ylim=c(0,0.03),
     xlab="Time to Absorption")

sim2<-replicate(10000,chutes(2))
sim3<-replicate(10000,chutes(3))
sim5<-replicate(10000,chutes(5))
sim10<-replicate(10000,chutes(10))
sim15<-replicate(10000,chutes(15))
sim20<-replicate(10000,chutes(20))


max2<-matrix(NA,nrow=10000,ncol=2)
diff2<-NULL
storage2<- sim2
sort2<-matrix(NA,nrow=10000,ncol=2)

for (i in 1:10000){
max2[i,1]<-max(storage2[[i]][nrow(storage2[[i]]),1])
max2[i,2]<-max(storage2[[i]][nrow(storage2[[i]]),2])


sort2[i,]<-sort(max2[i,])
diff2[i]<-sort2[i,2]-sort2[i,1]
}

par(mfrow=c(1,1))
dir.create("difference")
setwd("difference")

png(file="differences%02d.png", width=450, height=450)
hist(diff2,col=rgb(0.5, 1, 0.5, 0.7),freq=FALSE,ylim=c(0,0.13),main="Two Players",xlim=c(0,100))
#lines(density(diff2))
hist(diff3,col=rgb(0.5, 1, 0.5, 0.7),main="Three Players",ylim=c(0,0.13),freq=FALSE,xlim=c(0,100))
hist(diff5,freq=FALSE,ylim=c(0,0.13),col=rgb(0.5, 1, 0.5, 0.7),xlim=c(0,100),main="Five Players")
```

```
hist(diff10,col=rgb(0.5, 1, 0.5, 0.7),ylim=c(0,0.13),freq=FALSE,xlim=c(0,100),main="Ten Players")
hist(diff15,col=rgb(0.5, 1, 0.5, 0.7),ylim=c(0,0.13),freq=FALSE,xlim=c(0,100),main="Fifteen Players")
hist(diff20,col=rgb(0.5, 1, 0.5, 0.7),ylim=c(0,0.13),freq=FALSE,xlim=c(0,100),main="Twenty Players")

dev.off()
system("convert -delay 80 *.png difference_1.gif")
#####################################################################
###Markov chain theory to compute probability distribution########
#####################################################################

calcTM <-
  function()
  {
    transitions <- rbind(c(1, 38),c(4, 14),c(9, 31),c(16, 6),c(21, 42),c(28, 84),c(36, 44),
                         c(48, 26),c(49, 11),c(51, 67),c(56, 53),c(62, 19),c(64, 60),c(71, 91),
                         c(80, 100),c(87, 24),c(93, 73),c(95, 75),c(98, 78))


    # simple transition matrix, not accounting for chutes and ladders
    tm <- matrix(0,ncol=107, nrow=107)
    dimnames(tm) <- list(as.character(0:106), as.character(0:106))
    tm[col(tm) > row(tm) & col(tm) < row(tm) + 7] <- 1/6

    # account for the chutes and ladders
    for(i in 1:nrow(transitions)) {
      tm[,transitions[i,2]+1] <- tm[,transitions[i,2]+1] + tm[,transitions[i,1]+1]
      tm[,transitions[i,1]+1] <- 0
    }

    # can't go past 100
    for(i in 1:nrow(tm)) {
      tm[i,i] <- tm[i,i] + sum(tm[i,102:107])
      tm[i,102:107] <- 0
    }

    tm[1:101,1:101]
  }

# calculate transition matrix
tm <- calcTM()
# start vector
start <- rbind(c(1, rep(0, 100)))
# vector to pull out end probability
end <- cbind(c(rep(0, 100), 1))

# calculation of full distribution at each step
maxsteps <- 400
probs <- matrix(nrow=maxsteps, ncol=ncol(tm))
rownames(probs) <- as.character(1:maxsteps)
probs[1,] <- start %*% tm

for(i in 2:maxsteps){
  probs[i,] <- probs[i-1,] %*% tm
}

cdf <- probs %*% end
```

```
pdf<-NULL

for(i in 1:399){
  pdf[i]<-cdf[i+1]-cdf[i]
}

probs<-probs[,-1]
par(mfrow=c(1,1))

##Plot pdf##

x<-seq(0,250,50)
y=seq(0,0.03,0.005)
plot(pdf,xlim=c(0,250),pch=19,cex=0.5,col=rgb(0,0.5,0.5,1),xlab="Time to Absoprtion",
     ylim=c(0,0.03),type='h',main="6 sided die",ylab="Density",axes=F)
axis(1, at = x)
axis(2, at = y)

##Plot cdf and 1-cdf###

plot(1-cdf,xlim=c(0,250),pch=19,cex=0.5,col=rgb(0,0.5,0.5,1),xlab="Time",
     ylab="Pr(Still Playing by Time k)", main="")

plot(cdf,xlim=c(0,250),pch=19,cex=0.5,col=rgb(0,0.5,0.5,1),xlab="Time",
     ylab="Pr(Game Over by Time k)", main="")

require(ggplot2)

##Create an array of 10x10 probability matrices for times 1 through 50##

array<-array(NA, dim=c(10,10,50))

for (i in 1:50){
  array[1,1:10,i]<-matrix(probs[i,1:10])
  array[2,1:10,i]<-matrix(rev(probs[i,11:20]))
  array[3,1:10,i]<-matrix(probs[i,21:30])
  array[4,1:10,i]<-matrix(rev(probs[i,31:40]))
  array[5,1:10,i]<-matrix(probs[i,41:50])
  array[6,1:10,i]<-matrix(rev(probs[i,51:60]))
  array[7,1:10,i]<-matrix(probs[i,61:70])
  array[8,1:10,i]<-matrix(rev(probs[i,71:80]))
  array[9,1:10,i]<-matrix(probs[i,81:90])
  array[10,1:10,i]<-matrix(rev(probs[i,91:100]))
  array[array == 0] <- NA
}

require(grid)
require(RColorBrewer)
mypalette<-brewer.pal(9,"Purples")

##Plot probability matrices as a game board##

for (i in 1:50){

  Sys.sleep(0.1)
  image(1:ncol(array[,,50]), 1:nrow(array[,,i]), t(array[,,i]),
   col = rev(heat.colors(50,alpha=0.7)),main=paste("Time ", i),
```

```
   xlab="",ylab="")
  grid(10,10, col = "black",lty=1)
  grid.rect(x=0.19, y=0.2, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.rect(x=0.42, y=0.2, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.rect(x=0.805, y=0.2, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.circle(x=0.5,y=0.27,r=0.007,gp=gpar(fill="green",color=NA))
  grid.rect(x=0.19, y=0.34, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.rect(x=0.73, y=0.34, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.rect(x=0.5,y=0.41,width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.circle(x=0.73,y=0.48,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.8,y=0.48,r=0.007,gp=gpar(fill="green",color=NA))
  grid.rect(x=0.88,y=0.55,width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.circle(x=0.5,y=0.55,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.27,y=0.62,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.42,y=0.62,r=0.007,gp=gpar(fill="green",color=NA))
  grid.rect(x=0.88,y=0.69,width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.rect(x=0.19, y=0.69, width=0.01, height=0.01,gp=gpar(fill="purple",color=NA))
  grid.circle(x=0.65,y=0.76,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.73,y=0.83,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.58,y=0.83,r=0.007,gp=gpar(fill="green",color=NA))
  grid.circle(x=0.34,y=0.83,r=0.007,gp=gpar(fill="green",color=NA))
}


## Create Q and I matrix##
Q<-tm[-101,-101]
I<-diag(100)

##Compute (I-Q)^-1; this will give the expected number of visits to each transient state
##given the player starts at a certain state##

n.ij<-solve((I-Q))

##Expected time to absorption starting at state 0##
sum(n.ij[1,])

##Expected time to absorption using simulated results##

pdf<-table(length.game)/10000
sum(as.numeric(names(table(length.game)))*as.matrix(pdf))

##Compute expected time to absorption at each state (there's a much
quicker way to do it using the n.ij matrix)##

pdf<-array(NA, dim=c(1,399,100))
expected.time.to.absorption<-NULL

for (i in 1:100){
  maxsteps <- 400
  probs <- matrix(nrow=maxsteps, ncol=ncol(tm))
  end <- cbind(c(rep(0, 100), 1))
  start<-c(rep(0,101))
  start[i] <- 1
  probs[1,] <- start %*% tm

  for(j in 2:maxsteps){
    probs[j,] <- probs[j-1,] %*% tm
```

```
  }

  cdf <- probs %*% end

  for(k in 1:399){
    pdf[1,k,i]<-cdf[k+1]-cdf[k]
  }
}

x<-c(1:399)
sd.time.to.absorption<-NULL

for(i in 1:100){
  expected.time.to.absorption[i]<-sum(pdf[,,i]*x)
  expected.time.to.absorption[100]<-0
  sd.time.to.absorption[i]<-sqrt(sum(pdf[,,i]*x^2)-(sum(pdf[,,i]*x))^2)
  sd.time.to.absorption[100]<-0
}

##Plot expected time to absorption at each state##

par(mfrow=c(1,1))

plot(c(),c(), ylim=c(0,40), xlim=c(0,100),xaxt='n',yaxt='n',xlab='Position',
ylab='Expected time to absorption')
axis(1, at = c(0,10,20,30,40,50,60,70,80,90,100))
axis(2, at = c(0,5,10,15,20,25,30,35,40))
grid(10,10,col = "gray",lty=2,lwd=1)

for (i in 1:100){
  points(i,expected.time.to.absorption[i],pch=20,col=rgb(0.5,0,0.8,0.8),cex=0.9)
  lines(1:100,expected.time.to.absorption,col=rgb(0.5,0,0.8,0.2),cex=0.7)
}

##Plot standard deviation of time to absorption at each state##

plot(c(),c(), ylim=c(0,40), xlim=c(1,100),xlab='Position', ylab='SD time to absorption')
axis(side = 1, at = c(0,10,20,30,40,50,60,70,80,90,100))

for (i in 1:99){
  points(i,sd.time.to.absorption[i],pch=19,col=rgb(0.5,0,0.8,0.2),cex=0.5)
  lines(1:100,sd.time.to.absorption,col=rgb(0.5,0,0.8,0.2),cex=0.7)
}

#####################################################################
###Estimate approximating continuous distribution#####################
#####################################################################

loss.fun <- function(ab.vec,
                     quant.50 = 32,quant.95=89,
                     quant.75=49,quant.10=14,quant.20=19,quant.05=11,
                     quant.30=23,quant.60=37,quant.40=27,quant.70=45,
                     quant.15=17,quant.25=25,quant.45=29,quant.55=34,
                     quant.65=41,quant.80=55,quant.85=62,
                     quant.01=8, quant.02=9, quant.03=10, quant.005=7,
                     quant.82=57, quant.73=47,quant.46=30,quant.47=40, quant.48=31,
                     quant.49=31, quant.51=32, quant.52=33, quant.53=33, quant.54=34)
```

```
  {

  dist.1 <- abs(qlnorm(0.25, ab.vec[1], ab.vec[2]) - quant.25)
  dist.2 <- abs(qlnorm(0.95, ab.vec[1], ab.vec[2]) - quant.95)
  dist.3<- abs(qlnorm(0.5, ab.vec[1], ab.vec[2]) - quant.50)
  dist.4<- abs(qlnorm(0.75, ab.vec[1], ab.vec[2]) - quant.75)
  dist.5<- abs(qlnorm(0.1, ab.vec[1], ab.vec[2]) - quant.10)
  dist.6<- abs(qlnorm(0.05, ab.vec[1], ab.vec[2]) - quant.05)
  dist.7<- abs(qlnorm(0.20, ab.vec[1], ab.vec[2]) - quant.20)
  dist.8<- abs(qlnorm(0.30, ab.vec[1], ab.vec[2]) - quant.30)
  dist.9<- abs(qlnorm(0.60, ab.vec[1], ab.vec[2]) - quant.60)
  dist.10<- abs(qlnorm(0.40, ab.vec[1], ab.vec[2]) - quant.40)
  dist.11<- abs(qlnorm(0.70, ab.vec[1], ab.vec[2]) - quant.70)
  dist.12<- abs(qlnorm(0.15, ab.vec[1], ab.vec[2]) - quant.15)
  dist.13<- abs(qlnorm(0.45, ab.vec[1], ab.vec[2]) - quant.45)

  dist <-dist.1 + dist.2+dist.3+dist.4+dist.5+dist.6+dist.7+dist.8+dist.9+dist.10+
    dist.11+dist.12
  return(dist)
}
a.vals <- seq(0, 5, length = 1000)
b.vals <- seq(0, 5, length = 1000)
ab.grid <- expand.grid(a.vals,b.vals)
ab.dist <- apply(ab.grid, 1, loss.fun)
ab.grid[ab.dist == min(ab.dist), ]

plot(pdf,xlim=c(0,150),pch=19,cex=0.5,col=rgb(0.8, 0.5, 0.8, 1),
     xlab="Time to Absoprtion",ylim=c(0,0.037),main="6 sided die",type='h',lwd=2)
curve(dlnorm(x,3.448 ,0.63), type="l",xlim=c(0,200),add=TRUE,col=rgb(0.5,0,1,0.8),lwd=4)
legend(120,0.03)
```

References

[1] Pinsky, Mark A., Karlin, Samuel, *An Introduction to Stochastic Modeling*, 4th edition, Academic Press, 2012.

[2] Broman, Karl, *R simulation of Chutes & Ladders*, https://gist.github.com/kbroman/5600209, University of Wisconsin Madison.

[3] Broman, Karl, *More on Chutes & Ladders*, http://www.r-bloggers.com/more-on-chutes-ladders/, University of Wisconsin Madison.

[4] Hochman, Michael, *Chutes & Ladders*, http://math.uchicago.edu/ may/REU2014/REUPapers/Hochman.pdf, University of Chicago.

[5] Volfovsky, Alexander, *Markov Chains and Applications*, http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALFULL/Volfovsky.pdf, University of Chicago, 2007.

[6] Von Hilgers, Philipp, Langville, Amy N., *The Five Greatest Applications of Markov Chains*, http://langvillea.people.cofc.edu/MCapps7.pdf

[7] Humpherys, Jeffrey, *Chutes and Ladders*, http://www.math.byu.edu/ jeffh/mathematics/games/chutes/chutes.html, BYU Mathematics Department

[8] Gely P. Basharin, Amy N. Langville, *The life and work of A.A. Markov*, http://www.sciencedirect.com/science/article/pii/S0024379504000357

[9] Hayes, Brian, *First Links in the Markov Chain*, http://www.americanscientist.org/libraries/documents/201321152149545-2013-03Hayes.pdf, American Scientist

[10] Leslie A. Cheteyan, Stewart Hengeveld, Michael A. Jones, $\widehat{\text{C}}$hutes and Ladders for the Impatient The Mathematical Association of America, http://www.maa.org/sites/default/files/pdf/upload$_l ibrary/22/Polya/Cheteyan - 2012.pdf$