MONTE CARLO SIMULATIONS AND APPLICATIONS IN SPORTS

MATTHEW AKUZAWA

Abstract. Random walks and Brownian motion are two of the most important stochastic processes and are vital in creating models to understand the movement of prices, molecules, and more. This paper examines random walks and Brownian motion as a basis to understand how Monte Carlo simulations work. This paper focuses on how the Monte Carlo process can be used in sports data science to create models that more accurately demonstrate the performance of teams.

Contents

1. Introduction

In 2012, FC Barcelona was arguably the best team to ever play soccer. That year they faced off against Scottish side Celtic in the champions league which was a comparatively tiny team. By the end of the game Barcelona had recorded 23 shots to Celtic's 5, and had possessed the ball for 89% of the game to Celtic's 11%. On top of these outrageously one sided statistics, the Spanish side also completed nearly 6 times as many passes as their opponents. Yet the game ended 1-2 to the dismay of Barcelona players and supporters alike. On the surface, Celtic won and so many who hadn't watched the game might think they were the better team. However, looking at the statistics and watching the game would tell you that Barcelona controlled every aspect of the game. However, for all of their control they still lost.

Date: DEADLINES: Draft AUGUST 14 and Final version AUGUST 28, 2021.

At the end of the day, a lot of sports comes down to luck. Sometimes, luck goes in your favor and you score a 1 in a million goal. Other days it lets you down and you miss a shot that on any other day you can score with your eyes closed. This so called "luck" comes down to millions of small factors such as the weather, conditions of the soccer field, what the player ate for lunch, etc. Because there are so many factors that we could not hope to track and control, it is easier to attribute the variation in player performance to luck. With so much randomness needed to win games, how can we compare two teams and tell which team truly deserved to win? By this we mean that if we were somehow able to remove the element of randomness, which team would have won. While not perfect we can create a model that allows us to mitigate this randomness and create a statistic that allows us to compare one team with one another.

We explore this model by first introducing simple random walks to introduce the main principles behind stochastic processes. Then we move to a discussion of Brownian Motion and understand its construction and how it can be used to model movements in stocks and other natural phenomena. Next, we introduce the idea of Monte Carlo simulations as a way to further refine the model. Finally, we explore how we can use Monte Carlo simulations and a soccer statistics called *expected points* to better predict the performance of a team.

2. Simple Random Walk

To understand the simple random walk, first imagine a person standing at a point. We will call this starting location x. Now the person, who we will also call the walker, flips a fair coin. If the coin lands on heads, the walker takes a step to the right. If tails, they take a step to the left. They then keep flipping this coin over and over, taking one step in the appropriate direction every time.

Formally, we define the simple random walk on the probability space (Ω, $\mathcal{F}$, P) where Ω is the space of possible outcomes, $\mathcal{F}$ the set of all events, and P the probability function that maps each element of the event space to a number between 0 and 1. We see that $\Omega = \{Heads, Tails\}$ and P maps both outcomes to 1/2.

Definition 2.1. Let X_n be the random variable at time n such that

$$X_n = \begin{cases} 1 & \text{if Heads} \\ -1 & \text{if Tails.} \end{cases}$$

We define S_n to be the location of the walker at time n given that the walker starts at x:

$$S_n = x + X_1 + X_2 + ... + X_n.$$

Hence we see that $E[X_j] = 0$ for j $= 1, 2, ..., n$ because X_j has an equal chance of being positive and negative. Furthermore, $E[S_n] = x$ where x is the initial position.

Before we move on, we will quickly define four common terms that form the basis of our entire conversation on random variables

Definition 2.2. The *expected value* of the random variable X, otherwise written as $E[X]$, is the weighted average of all possible values of X. The *mean* of X which represents the average value of X is equal to $E[X]$. The *variance* of X which represents X's average deviation from its mean is equal to the expected value of $E[(X - E[X])^2]$. When the mean of X is zero, we see that the variance is simply $E[X^2]$. Lastly, a random variable is said to be *normally distributed* with mean μ and

variance σ if its distribution is symmetric and bell shaped with the corresponding mean and variance. A variable is said to have a *standard normal distribution* if it is normally distributed with mean 0 and variance 1.

Now that we have described what a simple random walk is, it is important that we now define a *martingale*. Martingales are important because they are one of the key examples of a stochastic process. In general, a martingale is a sequence of random variables where the expected value of the next variable in the sequence is equal to the current value. We will use martingales to show how simple random walks and Brownian motion display similar properties later on. In order to formally define Brownian motion, we first need to define a *filtration*.

Definition 2.3. The *filtration* F_n is equal to the set $\{X_1, X_2, ..., X_n\}$. It is also sometimes referred to as the information contained in $X_1, ..., X_n$.

Definition 2.4. A *martingale with respect to the filtration* $\{F_n\}$ is a sequence of random variables $M_1, M_2, ...$ that satisfies:

(1) For any time n, $E[|M_n|] < \infty$
(2) $E[M_n|F_m] = M_m$ when $m < n$.

If the second condition of the definition above is replaced with

$$E[M_n|F_m] \leq M_m,$$

then we refer to the process as a *submartingale*. If we replace it with

$$E[M_n|F_m] \geq M_m,$$

then the process is called a *supermartingale*.

With these two variables, we are able to define our first *stochastic process*. A *stochastic process* is a set of random variables that are indexed by time. Since each X_j is a random variable, the martingale is a stochastic process.

Theorem 2.5. *The random variable S_n as defined above is a martingale.*

Proof. We saw earlier that $E[X_j] = 0$ for all j. Since each X_j is independent, we can find $E[S_j]$ for all j as follows:

$$\begin{aligned} E[S_n] &= E[X_1 + X_2 + ... + X_n] \\ &= E[X_1] + E[X_2] + ... + E[X_n] \\ &= \sum_{i=1}^{n} E[X_i] \\ &= \sum_{i=1}^{n} 0 \\ &= 0. \end{aligned}$$

Thus the first criteria is satisfied. Additionally:

$$\begin{aligned} E[S_{n+1}|F_n] &= E[S_n + X_{n+1}|F_n] \\ &= E[S_n|F_n] + E[X_{n+1}|F_n] \\ &= S_n + E[X_{n+1}] \\ &= S_n. \end{aligned}$$

By induction we can see that this property extends to any $m > n$. Thus we have that $E[M_n|F_m] = M_m$. Hence S_n is a martingale. □

Definition 2.6. A *square integrable* martingale is a martingale M_n for which $E[M_n^2] < \infty$ for all n.

A question one might ask is what is the expected distance from the origin at a given time n. We know that since S_n is a martingale, $E[S_n] = 0$. However, by definition the random walk cannot stay at the origin at every time n. Thus instead of looking at the expected value of S_n, we will look at the expectation of S_n^2. The reason we are looking at the square of the sum is that to find the typical distance traveled, we would want to look at either the $E[|S_n|]$ or $\sqrt{E[S_n^2]}$. The former is harder to calculate so we will use the latter to find the expected distance traveled. Furthermore, we will take advantage of the fact that our martingale is square integrable.

Theorem 2.7. *At time n, the expected distance between the random walker and the origin is* $\sqrt{n}$.

Proof. The random variable $X_j = \pm 1$ with equal probability of being positive or negative. Therefore the expected value of any given X_j is 0. Furthermore, since $(\pm 1)^2 = 1$, the expected value of X_j^2 is 1. Additionally:

$$\begin{aligned} E[S_n^2] &= E[(X_1 + X_2 + ... + X_n)^2] \\ &= E[X_1^2 + X_2^2 + ... + X_n^2 + \sum_{j \neq k} X_j X_k] \\ &= E[X_1^2] + ... + E[X_n^2] + \sum_{j \neq k} E[X_j X_k] \\ &= 1 + ... + 1 + \sum_{j \neq k} 0 \\ &= n. \end{aligned}$$

Therefore at a given time n, the expected distance from the origin is $\sqrt{n}$. □

One could picture the random walk as a function on a graph with time t on the x-axis and the location $S(t)$ on the y-axis. Thus the question arises: can we find the integral of this stochastic process and if so how would we do that? In order to define the integral of the walk, we must first define what it means for a random variable to be *predictable*. Doing so allows us to generalize this integral calculation to random processes other than a simple random walk.

Definition 2.8. A sequence of random variables $K_1, K_2, ...$ is called *predictable* with respect to a filtration F_n if for all $n = 1, 2, ...$, K_n is a function of F_{n-1}.

Definition 2.9. Let $K_1, K_2, ...$ be a sequence of predictable random variables that is also square integrable for all n. We define the integral of K_n with respect to the random walk S_n as follows:

$$Z_n = \sum_{i=1}^{n} K_i X_i = \sum_{i=1}^{n} K_i (S_i - S_{i-1}) = \sum_{i=1}^{n} K_i \Delta S_i.$$

Proposition 2.10. *The integral satisfies three key properties:*

- ***Martingale Property:*** *The integral Z_n is a martingale with respect to filtration F_n.*
- ***Linearity:*** *Let a, b be constants and K_n, L_n be predictable sequences. Then $aK_n + bL_n$ is a predictable sequence as well. Furthermore,*

$$\sum_{i=1}^{n}(aK_n + bL_n)\Delta S_i = a\sum_{i=1}^{n} K_i\Delta S_i + b\sum_{i=1}^{n} L_i\Delta S_i.$$

- ***Variance Rule:*** *Let σ^2 be the variance of each X_i. Then:*

$$Var\left[\sum_{i=1}^{n} K_i\Delta S_i\right] = E\left[(\sum_{i=1}^{n} K_i\Delta S_i)^2\right] = \sigma^2\sum_{i=1}^{n} E[K_i^2].$$

Proof. The martingale property of Z_n follows as in the proof of Theorem 2.5. Linearity is immediate by the definition of summations.

The first equality holds by definition. For the second equality, we must rewrite the summation as $(\sum_{i=1}^{n} K_iX_i)^2$. Since X_i and K_i are independent and $E[X_i] = 0$ for all i, we have $E[X_iK_i] = E[X_i]E[K_i] = 0$. Thus,

$$E\left[(\sum_{i=1}^{n} K_iX_i)^2\right] = \sum_{i=1}^{n} E[K_i^2X_i^2] + \sum_{i\neq j} E[X_iK_iX_jK_j] = \sum_{i=1}^{n} E[K_i^2X_i^2].$$

Since K_i is a function of F_{i-1} and X_i is independent of F_{i-1}, we see that:

$$\begin{aligned} E[X_i^2K_i^2] &= E[E[X_i^2K_i^2|F_{i-1}] \\ &= E[K_i^2(E[X_i^2|F_{i-1}])] \\ &= E[K_i^2(E[X_i^2])] \\ &= E[K_i^2\cdot\sigma^2] \\ &= \sigma^2E[K_i^2]. \end{aligned}$$

Thus plugging back into the earlier equality, we see

$$E\left[(\sum_{i=1}^{n} K_iX_i)^2\right] = \sigma^2\sum_{i=1}^{n} E[K_i^2].$$

□

3. Brownian Motion

Brownian motion, commonly referred to as the Wiener process, is a stochastic process with a continuous time index. Brownian motion is extremely helpful in modeling phenomena in the real world that change randomly and continuously. We can also view Brownian motion as the continuous analogous of the simple random walk. Whereas before we had random variables at discrete times, Brownian motion is randomly changing continuously. We will show later on how one can construct Brownian motion from a simple random walk.

Definition 3.1. Let $0 = t_0 < t_1 < t_2 < ... < t_n$ be a set of integers. An *increment* of stochastic process B_t is the random variable $(B_{t_j} - B_{t_{j-1}})$ for all $j = 1, 2, ..., n$.

Definition 3.2. Let B_t, sometimes written as $B(t)$, be considered as *Brownian motion* with mean m and variance σ^2 if it satisfies the following:

- $B_0 = 0$.

- B_t is continuous with probability one.
- For all $j \neq k$, the increments B_{t_j} and B_{t_k} are independent.
- For $t_j < t_k$, $B_{t_k} - B_{t_j}$ is normally distributed with mean $m(t_k - t_j)$ and has a variance of $\sigma^2(t_k - t_j)$.

Furthermore, if $B_t \sim N(0,1)$, then B_t is considered *standard Brownian motion*. We can rewrite any Brownian motion in terms of the standard Brownian motion. Suppose we let Z_t be standard Brownian motion with mean 0 and variance 1, then for any non-standard Brownian motion Y_t with mean m and variance σ^2, we can write:

$$Y_t = \sigma Z_t + mt. \tag{3.3}$$

Apart from these four requirements for Brownian motion there are two key characteristics of Brownian motion that we will discuss. The first is that since Brownian motion is the continuous analogous of the random walk before, Brownian motion should also be a martingale.

Theorem 3.4. *Brownian motion is a martingale.*

Proof. Let B_t be a Brownian motion. From Definition 3.2, we know that each increment of B_t is normally distributed. Since the mean from standard Brownian motion is 0, every increment has an expected value of 0. From the definition of Brownian motion, we know that $B_0 = 0$. Thus every increment has an expectation of 0. Then

$$E[B_t] = E[B_t - 0] = E[B_t - B_0] = 0 < \infty.$$

for all t. Thus the first criterion in Definition 2.4 is fulfilled. For the second criterion in Definition 2.4, take any $k > j$. We know

$$E[B_{t_k} - B_{t_j}] = 0,$$

which implies

$$\begin{aligned} E[B_{t_k}|F_{t_j}] &= E[B_{t_k} - B_{t_j} + B_{t_j}|F_{t_j}] \\ &= E[B_{t_k} - B_{t_j}|F_{t_j}] + E[B_{t_j}|F_{t_j}]] \\ &= 0 + B_{t_j} \\ &= B_{t_j}. \end{aligned}$$

Therefore Brownian motion is a martingale. □

The second key fact about Brownian motion that we will cover is that Brownian motion is differentiable nowhere with probability one. At first glance this may come as a surprise as Brownian motion is continuous with probability one.

Theorem 3.5. *Brownian motion is differentiable nowhere with probability one.*

Proof. We can rewrite the derivative of Brownian motion at a given time t as follows:

$$B'(t) = \lim_{h \to 0} \frac{B(t+h) - B(t)}{h}.$$

Since we are working with stochastic variables, we cannot use the usual methods to calculate this limit. This is because regular calculus works with deterministic

variables whereas here we are working with fluctuating variables. So instead, we will first begin by defining X as such:

$$X = \frac{B(t+h) - B(t)}{h}.$$

We do this so that we can find the expectation and variance of the random variable X.

$$E[X] = E\left[\frac{B(t+h) - B(t)}{h}\right] = \frac{1}{h}E[B(t+h) - B(t)] = 0.$$

The second equality comes from the fact that h is a constant rather than a random variable. The last equality holds from the fact that $B(t+h) - B(t)$ is an increment and so its expectation is zero. We next need to find the variance so that we can rewrite X in terms of Z.

$$Var[X] = Var\left[\frac{B(t+h) - B(t)}{h}\right] = \frac{1}{h^2}Var[B(t+h) - B(t)] = \frac{1}{h}.$$

We come to this value for the variance of X using the same process we used to find the expectation. With an expectation of 0 and variance of $\frac{1}{h}$, we can see that $X \sim N(0, \frac{1}{h})$. Thus from equation 3.3, we can rewrite X as

$$X = \frac{1}{\sqrt{h}}Z$$

where Z is the standard Brownian motion. In order to finally prove that Brownian motion is non-differentiable with probability one, we must show that the absolute value of the derivative is equal to infinity with probability one. We will do this by setting an arbitrary $k > 0$. We want to show that for any k,

$$P[\,|B'(t)| > k] = 1.$$

We see that:

$$\begin{aligned} P[|B'(t)| > k] &= P[\lim_{h\to 0} |X| > k] \\ &= \lim_{h\to 0} P[\,|X| > k] \\ &= \lim_{h\to 0} P[\,|\frac{1}{\sqrt{h}}Z| > k] \\ &= \lim_{h\to 0} P[\,|Z| > k\sqrt{h}] \\ &= P[\,|Z| > 0] \\ &= 1. \end{aligned}$$

Thus for any k, we see that with probability one, the absolute value of the derivative is greater than k. Hence the derivative does not exist at any given time and therefore Brownian motion is nowhere differentiable. □

So far we have only discussed properties of Brownian motion and what it should look like. We have yet to confirm that some process truly exists that satisfies all of the requirements of Definition 3.2. In order to confirm that Brownian motion does in fact exist, we will construct it using the *dyadic rationals*.

Definition 3.6. The *dyadic rationals* is the set of rationals that can be written as:

$$\{\frac{k}{2^n} : n, k \in \mathbb{N} \cup \{0\} \text{ and } k \leq 2^n\}.$$

One last thing we will need before we begin the construction of Brownian motion is the following definition and a proposition that comes out from it.

Definition 3.7. If every random variable in a finite sequence $(X_1, X_2, ..., X_n)$ is a linear combination of independent standard normally distributed random variables, then the sequence has a *joint normal* distribution.

Proposition 3.8. *Let X and Y be independent random variables and $X, Y \sim N(0,1)$. Furthermore, let*

$$A = \frac{X+Y}{\sqrt{2}},\ B = \frac{X-Y}{\sqrt{2}}.$$

Then both A and B are independent random variables and $A, B \sim N(0,1)$.

Proof. Suppose (A, B) has a joint normal distribution by definition. Since both X and Y have mean 0, so do both A and B. Making use of $E[X^2] = E[Y^2] = 1$ and $E[XY] = 0$, we get the following equations:

$$E[A^2] = E\left[\frac{X^2 + XY + Y^2}{2}\right] = \frac{E[X^2] + E[XY] + E[Y^2]}{2} = \frac{2}{2} = 1,$$

$$E[B^2] = E\left[\frac{X^2 - XY + Y^2}{2}\right] = \frac{E[X^2] - E[XY] + E[Y^2]}{2} = \frac{2}{2} = 1,$$

$$E[AB] = E\left[\frac{X^2 - Y^2}{2}\right] = \frac{E[X^2] - E[Y^2]}{2} = \frac{0}{2} = 0.$$

From this we can see that the covariance matrix of (A, B) is just a 2-by-2 identity matrix. This is in fact the same covariance matrix as the one for two independent $N(0,1)$ random variables. Therefore we can see that A and B are independent random variables and are normally distributed with mean 0 and variance 1. □

3.1. **The Construction of Brownian Motion.** This construction of Brownian motion follows the construction in [2]. We will start by defining B_t on the interval $[0,1]$, and then from there we can take a countable union of these Brownian motions to construct B_t on $[0,\infty)$. To do this we first define D_n to be the subset of the dyadic rationals whose denominator is 2^n:

$$D_n = \{\frac{k}{2^n} : k = 0, 1, 2, ...2^n\}.$$

We can then rewrite the dyadic rationals as $D = \bigcup_{i=0}^{\infty} D_i$. We will further define a sequence of random variables:

$$\{X_k\}_{k\in D}\ , X_k \sim N(0,1), \forall k \in D.$$

Next in order for the construction to work we must prove that the dyadic rationals are dense.

Theorem 3.9. *The dyadic rationals are dense in the real numbers*

Proof. It suffices to show that the dyadic rationals are dense on $[0,1]$. Let $x \in [0,1]$ and $\epsilon > 0$. By the Archimedean property, there must exist some $n \in \mathbb{N}$ such that $\frac{1}{2^n} < \epsilon$. Let $\lfloor\cdot\rfloor$ be the floor function, otherwise know as the greatest integer less than function. Furthermore let $y = \lfloor x \cdot 2^n \rfloor$. Note that by definition of the floor

function y is an integer. In this case since x is positive, y is a positive integer. Then we have:

$$y = \lfloor x \cdot 2^n \rfloor \leq x \cdot 2^n < \lfloor x \cdot 2^n \rfloor + 1 = y + 1,$$

$$\frac{y}{2^n} \leq x < \frac{y+1}{2^n},$$

$$0 \leq x - \frac{y}{2^n} < \frac{1}{2^n} < \epsilon.$$

Thus for any $\epsilon > 0$ and $x \in [0, 1]$, there exists some number k, that can be written in the form $\frac{y}{2^n}$, such that k is less than ϵ away from x. By definition, k is a dyadic rational and so the dyadic rationals are dense on $[0, 1]$. As we stated before, this also further implies that the dyadic rationals are dense on $[0, \infty)$ □

Now that we have shown that the dyadic rationals are dense, we can construct Brownian motion B_t over D using the random variables X_t as defined above. We will begin by defining $B_0 = 0$ and $B_1 = X_1$. From here we will define $B_{\frac{1}{2}}$, then $B_{\frac{1}{4}}$ and $B_{\frac{3}{4}}$, then so on and so forth. Essentially, we will inductively define the Brownian motion on the set D_0, then from there on on the set $\{D_{n+1} \backslash D_n\}$. For example, we start with D_0 and D_1. We then define $D_{\frac{1}{2}}$ based on the expectation of B_1.

We will use the following equation to define B_k. For $P = 0, 1, 2, ..., 2^n - 1$, we set $k = \frac{2p+1}{2^{n+1}} \in D_{n+1} \backslash D_n$. We will assume that for the inductive hypothesis that D_j is defined for all $j \in D_n$. Then we can define B_k as follows:

$$B_k = B_{\frac{p}{2^n}} + \frac{B_{\frac{p+1}{2^n}} - B_{\frac{p}{2^n}}}{2} + \frac{X_k}{2^{\frac{n+2}{2}}}.$$

Now, using Proposition 3.8 several times, we see that for all n, every element of the set of random variables

$$\{B_{\frac{p}{2^n}} - B_{\frac{p-1}{2^n}} : P = 0, 1, 2, ..., 2^n\}.$$

is independent and has a distribution of $N(0, 2^{-n})$. Thus we can see that for any $p \in D$, B_p satisfies all of the requirements for Brownian motion as stated in its definition except for continuity. Our construction so far is only limited to the dyadic rationals. For a full proof of how to take Brownian motion from the dyadic rationals to the set of reals, see [2]. However for our purposes, we will use the fact that the dyadic rationals are dense as sufficient proof that Brownian motion extends to the reals.

3.2. **Brownian Motion as the Limit of Random Walks.** Brownian motion can also be seen as the limit of a random walk as both time and space increments approach zero. To show this we will use the same definitions of X_i and S_i as defined in Definition 2.1. Before when we defined these random variables, we chose $\Delta t = \Delta x = 1$ for simplicity. However, we can also choose $\Delta t = \frac{1}{N}$ where N is a large integer that we will eventually take to infinity. Whereas before we indexed the random variables with the integers from 0 to n, we will instead now use:

$$S_0, S_{\frac{1}{N}}, S_{\frac{2}{N}}, ...S_{\frac{N-1}{N}}, S_1. \tag{3.10}$$

Like before, we will define X_i to be equal to 1 or -1 with equal probability and keep the same indexing as before, but now from 1 to N. Thus at time 1, we can write the location of the random walker as follows:

$$S_1 = \Delta x(X_1 + X_2 + ... + X_N).$$

If we want this random walk to resemble Brownian motion, then we need to choose a Δx such that the $\text{Var}[S_1] = 1$. Knowing that $\text{Var}[X_i] = 1$, we can do the calculations and we see that:

$$\begin{aligned}
\text{Var}[S_1] &= \text{Var}[\Delta x(X_1 + X_2 + ... + X_N)] \\
&= (\Delta x)^2 \cdot [\text{Var}(X_1) + \text{Var}(X_2) + ... + \text{Var}(X_N)] \\
&= (\Delta x)^2 \cdot [1 + 1 + ... + 1] \\
&= (\Delta x)^2 \cdot N.
\end{aligned}$$

Therefore we have that

$$\Delta x = \sqrt{\frac{1}{N}} = \sqrt{\Delta t}.$$

Thus we have the requirements for a random walk to approach Brownian motion.

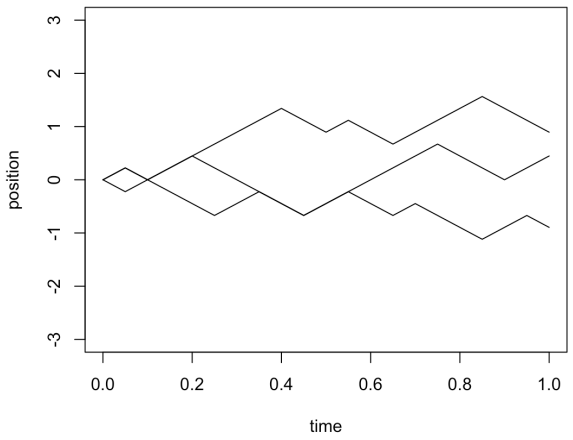

FIGURE 1. 3 Random Walks with Time Increments of $\frac{1}{20}$

As an illustration, we show simulations as $N \to \infty$ in 3.10. Here we set $N = 20$. For this smaller number, the random walk does not closely resemble Brownian Motion. For the next few diagrams, we will represent the number of paths with the variable n and N will remain the time increment.

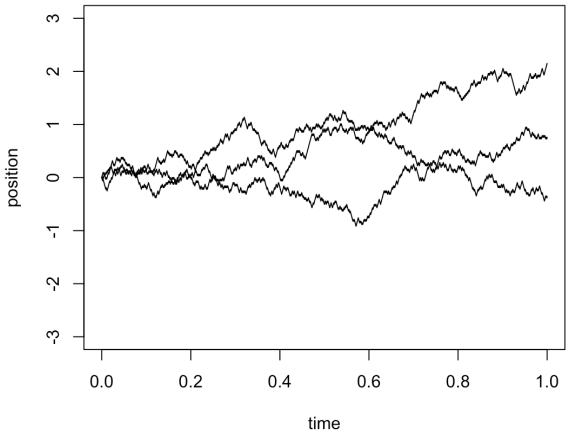

FIGURE 2. 3 Random Walks with $N = \frac{1}{1000}$

As N gets much larger, we start to see something more akin to Brownian motion take form. Here $N = 1000$. It also becomes more intuitive that Brownian motion is non-differentiable everywhere as stated in Theorem 3.5 as the graph has sharp turns at every point.

4. Monte Carlo Simulation

In the previous section we have shown how to get from random walks to Brownian motion. We know that our newly constructed motion is normally distributed with mean zero and variance one. However, for more more complicated models, it will not be as easy to figure out the distribution. Especially with models that aren't grounded by simple distributions, coming up with an equation for the expectation becomes harder. Thus we will introduce the Monte Carlo simulation as a computational solution to this problem.

The idea of the Monte Carlo method is to repeated random sampling to obtain a numerical result to an algorithm. Take for example the Brownian Motion as formed above. From running just three simulations, we don't get a good idea of what the mean or variance of the final S_1 value. However, if we were to take more simulations, we might start to see its true distribution.

We run the random walk 10000 times using the same N value of 1000 from before. We then take the final location of these random walks and graph them in a histogram along with the normal distribution. As you can see below, this distribution closely follows the normal distribution outlined in red, implying that the increment $B_1 - B_0 \sim N(0, 1)$. Thus we have confirmation that the expectation and variance are what we expected.

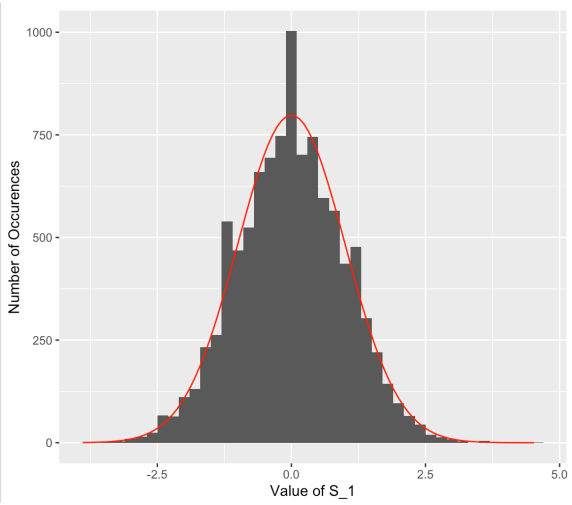

FIGURE 3. Histogram of Random Walk Outcomes

5. Expected Points Model

Finally, we will move to an application of both random walks and the Monte Carlo method. One soccer statistic of particular interest to fans, coaches, and bookies is called *expected goals*, otherwise known as the expected value of a given shot. This is the probability that a shot is scored primarily given its location, but further developments have taken into account other information such as the number of players in between the goal and the ball and what part of the body is used to shoot. The number is important because it helps to quantify whether a player is good at getting into goal scoring positions, regardless of their actual ability to shoot. In general, the expected goals of a shot is calculated by taking every recorded shot in history from that location and seeing with what probability the shot resulted in a goal. We will not cover the specifics of how this number is calculated as it is not specifically linked to the previous stochastic processes we discussed earlier. However, a different statistic called expected points can be extracted from expected goals. As suggested by the name, it is the expected amount of points a team receives in a game. We will represent expected points with the variable $xPts$. In a normal league game, a team is awarded three points for a win, one for a draw, and zero for a loss. Thus we can write the following formula:

$$xPts = 3 \cdot P(win) + 1 \cdot P(tie) + 0 \cdot P(loss).$$

It should be noted that if we were to represent points with the variable Pts, then $xPts = E[Pts]$. For this model we will make heavy use of open data sets from FBref [3]. FBref has data for every shot taken in each game played in the 2021-22 season of the Premier league. While they do not supply expected goals values for each shot, they do offer this value per player in a given match. They do this by summing up the expected goals of each shot taken by said player and giving one larger value. We can not see how this value is distributed between each shot. Thus while not exact, we take the average expected goals per shot for a given player in a match.

5.1. **Setting up the Model.** For each match, the FBref database contains a match report that lists every shot taken in said match. The reason we are taking the expected goals for each shot and not over the course of the match as a whole is that we want to treat each game itself as a random walk in order to more accurately determine how many points a team could have expected from a given match. We will treat each of these shots as a random variable. Therefore we can think of each game as a sequence of random variables with varying probabilities, and thus in a way they can be seen as a random walk. However, our model will be slightly different. We will define each random variable as follows:

Definition 5.1. Let n be the number of shots in a given match. For every random variable X_i^j, we will let the subscript represent the index of the shot in the sequence and the superscript represent which team the shot is for. The team will be denoted by h or a representing home and away respectively. Furthermore let P_i be the corresponding probability or expected goal value for a given shot i. Then we will define X_i^j as

$$X_i^h = \begin{cases} 1, & P_i \\ 0, & 1 - P_i \end{cases} \quad \text{and } X_i^a = \begin{cases} -1, & P_i \\ 0, & 1 - P_i. \end{cases}$$

In our random walk, if the shot is for the home team, then the chance of the difference between both teams' goals increasing by one in favor of the home team increases from zero to the expected goal value of that shot. This is equivalent to there being a chance to step right but not left, and vice versa for away teams. Next we will define S_n similarly to how we defined it before. We will let $S_0 = 0$ and define S_n as follows:

$$S_n = X_1^j + X_2^j + X_3^j + ... + X_n^j,$$

where j is equal to h or a depending on who is taking the shot.

We can see then that the random variable S_n represents the goal difference between the teams and thus determines who has scored more and who wins. We will then define a new variable to represent the amount of points each team is awarded. We will again use the notation h and a to define which team we are assigning the points to.

Definition 5.2. Given random variable S_n, we define Y_m^h and Y_m^a as follows:

$$Y_m^h = \begin{cases} 3, & S_n > 0 \\ 1, & S_n = 0 \\ 0, & S_n < 0 \end{cases} \quad ,Y_m^a = \begin{cases} 0, & S_n > 0 \\ 1, & S_n = 0 \\ 3, & S_n < 0 \end{cases} \quad .$$

Where m refers to which trial the random variable corresponds to.

5.2. **Applying the Model to an Individual Game.** Now we have our random walk, but it is harder here to find the expected value because the random variables, the shots, are not uniform. Each game has a different number of shots for each team and furthermore each shot has a different probability of going in. Thus it would be hard to calculate an expected points for each game with so much variability. Take for example the Premier league game Manchester City vs Leicester City which in real life ended 6-3. We have chosen this game as an example for its larger than normal shot volume. We can simulate the random walk and get a sort of "goal timeline" where the score corresponds to the relative goal difference between two

teams. We can also see the y-value as corresponding to the home team's goals minus the away team's. Below is an example of such simulation.

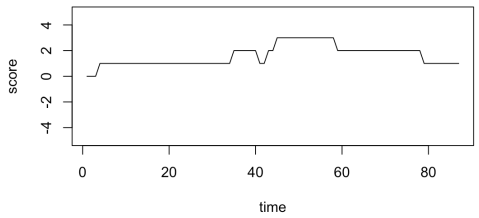

FIGURE 4. Example one of Game Simulation

Above, you can see that for a majority of the time increments, there is no chance of an increase or decrease. However, at the time of each shot, we see the possibility of taking a step to the right or left depending on who is shooting. In this simulation the home team have won by one goal, but the final score was in fact 4-3.

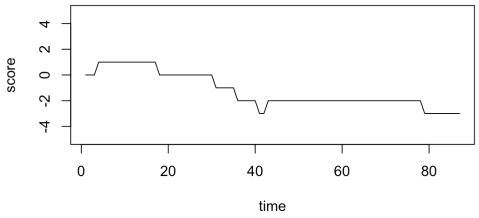

FIGURE 5. Example two of Game Simulation

In this simulation of the same game, we now see that it is the away team that have won. In fact the final scoreline would have read 2-5 if the game had played out as shown here. From just this random walk it is hard to tell how many points each team should have expected from this game. Thus we will utilize the Monte Carlo method as defined previously. In order to do so, we run this random walk simulation N times. In this case we will set $N = 1000$.

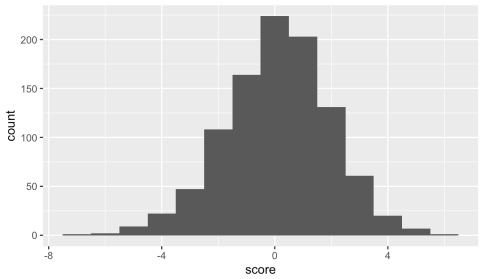

FIGURE 6. Histogram of Possible Games

In the histogram above we have collected the final score difference for all 1000 simulations. We can see that the graph resembles a bell curve that is skewed left.

We can take the random variable Y of each of these trials to get the equation:

$$\text{home team:} \quad xPts = \frac{1}{N}\sum_{i=1}^{N} Y_i^h$$

$$\text{away team:} \quad xPts = \frac{1}{N}\sum_{i=1}^{N} Y_i^a.$$

In this Monte Carlo simulation, we ended up with a final expected points value of 1.493 for the home team and 1.283 for the away team. Hence from our simulation we are able to find an expected value from non-uniform random variables.

5.3. **Full Model.** The next logical step is to apply this model to every game in a season. By doing this, we are essentially simulating an alternate version of the 2021-22 season. By combining the expected points of every game, we can recreate a league table based on how many points each team would've expected to receive based on their performance. Taking the full 2021-22 Premier league season and running 1000 simulations of each game, we end up with the following table:

	Team	Expected_Points	Actual_Points	Points_Difference	Place_Difference
1	Manchester City	81.351	93	11.649	0
2	Liverpool	71.836	92	20.164	0
3	Chelsea	71.331	74	2.669	0
4	Tottenham	61.865	71	9.135	0
5	Arsenal	61.603	69	7.397	0
6	Manchester Utd	57.319	58	0.681	0
7	Crystal Palace	57.169	48	-9.169	-5
8	West Ham	57.143	56	-1.143	1
9	Brentford	54.096	46	-8.096	-4
10	Brighton	53.694	51	-2.694	1
11	Aston Villa	50.138	45	-5.138	-3
12	Leicester City	49.704	52	2.296	4
13	Southampton	47.608	40	-7.608	-2
14	Everton	45.940	39	-6.940	-2
15	Newcastle Utd	44.968	49	4.032	4
16	Leeds United	41.753	38	-3.753	-1
17	Wolves	41.220	51	9.780	7
18	Burnley	35.348	35	-0.348	0
19	Watford	33.300	23	-10.300	0
20	Norwich City	29.280	22	-7.280	0

FIGURE 7. Premier League 2021-22 Table Sorted by Expected Points

In the table above, the expected points column represents the expected points won per team as determined by our model. The teams are sorted based on their expected points. The next column represents the actual amount of points each team accumulated throughout the season. Finally, the last two columns display the difference between each team's actual points and place in the leader board and their expected points and place. Thus we have created a model that outputs an alternative table. The advantage to this expected points versus the actual points is our new model minimizes the randomness from taking shots.

Looking at the results of this model, we can see that the top 6 finished exactly where they expected to be. However, we see that second place Liverpool performed

way better than expected resulting in what should've been a 10 point gap to first place becoming a 1 point gap. We can also see teams that performed well but were unlucky. Teams like 7th place Crystal Palace and 9th place Brentford can be confident in their playing ability and, should thing regress towards the mean, can expect a better 2022-23 season. Another interesting observation is Everton in 14th place. Throughout the season they were the subject of negative press around their performances and the fact that they were almost relegated to the lower divisions. However, from here we can see that they were not as bad as they were portrayed in the media, and that this season represents an outlier more so than a trend. Finally we see a team like Wolves in 17th place. They performed extremely better than our model would have predicted. This could be down to them being extremely lucky, but it is also possible that other factors resulted in their low expected points. For example, they might be a team that tended to score first, thus instead of going out and scoring more they elected to sit back and defend their lead. We would see a lower expected points value than is realistic.

5.4. **Flaws of the Model.** This model is simplified to show how random walks can be applied to create models of real world situations. There are several flaws with this model. As mentioned earlier, the database used did not contain the expected goals value per shot. It instead only had the total expected goals per each player and so we had to estimate the value per shot. The second is that for our case we treated each game as a martingale with pre-set random variables. In reality, if one of the shots earlier on had been successful, it could have potentially changed the course of the rest of the game. It is possible that scoring a goal earlier could increase the confidence of the player, inducing them to score more. At the same time, it is possible that scoring earlier increases the other teams awareness of the goal scorer and so could increase the chances that they block the next shot. However, with a larger sample size and running each game 1000 times we should be able to mitigate the effect that each shot has on one another. Lastly, it does not take into account game state. As mentioned at the end of the discussion on the final results, scoring first and protecting that lead would result in less shots, meaning less expected goals and less expected points. The model does not take something like this into account, and there are many other examples of situations where the expected points might hide some important information. Furthermore, unlike an actual random walk where each random variable is independent, All of these scenarios are possible and while we could never hope to account for all of them, it should be noted that they may cause inaccuracies in the model.

6. Appendix

The following section contains the code used to create the full model as outlined in 5.3.

```r
library(rvest)
library("dplyr")

n = 100000;
time = 10;
url_top = "https://fbref.com/en/comps/9/schedule/Premier-League-Scores-and-Fixtures"

#compiling links
links = read_html(url_top) %>%
  html_elements(".stats_table a")%>%
  html_attr("href")
links <- lapply(links, unlist)
for(x in length(links):1) {
  if ((nchar(links[x]) < 30) || (substring(links[x],5,5) != "m")) {
    links[[x]] = NULL
  }
}
for(x in seq(length(links),2, -2)) {
  links[[x]] = NULL
}

#creating fixture list and standings
full_table = read_html(url_top) %>%
  html_element(".stats_table")%>%
  html_table
fixtures <- data.frame(home = full_table$Home, away = full_table$Away)
fixtures = fixtures[!apply(fixtures == "", 1, all),]
standings <- data.frame(team = sort(unique(fixtures$home)), points = 0)

#moving through fixture list and running expected points simulation
for(k in 1:length(links)) {
  Sys.sleep(time)

  #Setting individual game variables
  score = 0;
  home_pts = 0;
  away_pts = 0;

  #scraping individual game information
  url = toString(paste("https://fbref.com", links[k], sep = ""))
  tables = read_html(url) %>%
    html_elements(".stats_table") %>%
    html_table()
  home = tables[[1]]
  away = tables[[8]]
  shots = read_html(url) %>%
    html_element("table#shots_all") %>%
    html_table()

  #home data
  home_dat = data.frame(players = home[,1], PK = home[,10], shots = home[,11], xG = home[,20])
  colnames(home_dat) <- c("player", "PK", "shots", "xG")
  home_dat = home_dat[-1,]
  home_dat = home_dat[-nrow(home_dat),]
  home_dat$shots = as.numeric(home_dat$shots) + as.numeric(home_dat$PK)
  home_dat$xG = as.numeric(home_dat$xG)
  home_dat$xG_per_shot = ifelse(home_dat$shots == 0, 0, home_dat$xG/home_dat$shots)

  #away data
  away_dat = data.frame(players = away[,1], PK = away[,10], shots = away[,11], xG = away[,20])
  colnames(away_dat) <- c("player", "PK", "shots", "xG")
  away_dat = away_dat[-1,]
  away_dat = away_dat[-nrow(away_dat),]
  away_dat$shots = as.numeric(away_dat$shots) + as.numeric(away_dat$PK)
  away_dat$xG = as.numeric(away_dat$xG)
  away_dat$xG_per_shot = ifelse(away_dat$shots == 0, 0, away_dat$xG/away_dat$shots)

  #shot data
  timeline = data.frame(time = shots[,1], player = shots[,2])
  timeline = timeline[-1,]
  colnames(timeline) <- c("minute", "player")
  timeline = timeline[!apply(timeline == "", 1, all),]
```

```
  for (x in 1:nrow(timeline)) {
    if(nchar(timeline$minute[x]) <= 2) {
    } else if (substring(timeline$minute[x],1,1) == 4) {
      timeline$minute[x] = 45;
    } else {
      timeline$minute[x] = 90 + as.numeric(substring(timeline$minute[x],4));
    }
  }
  timeline$minute = as.numeric(timeline$minute)
  timeline$xG = 0
  timeline$team = 1

  #assigning xG to a shot
  for (x in 1:nrow(timeline)) {
    #remove (pen) tag
    if (substring(timeline$player[x], nchar(timeline$player[x]) - 5) == "(pen)") {
      timeline$player[x] = substring(timeline$player[x], 1, nchar(timeline$player[x]) - 6)
    }

    #check in home team
    for(h in 1:nrow(home_dat)) {
      if (timeline$player[x] == home_dat$player[h]) {
        timeline$xG[x] = home_dat$xG_per_shot[h];
      }
    }

    #check in away team
    for(a in 1:nrow(away_dat)) {
      if (timeline$player[x] == away_dat$player[a]) {
        timeline$xG[x] = away_dat$xG_per_shot[a];
        timeline$team[x] = -1
      }
    }
  }

  #expected points for a match calculation
  for(y in 1:n){
    score = 0;
    for (x in 1:nrow(timeline)) {
      score = score + timeline$team[x] * sample(c(1,0), size = 1, prob = c(timeline$xG[x], 1- timeline$xG[x]))
    }
    if(score == 0) {
      home_pts = home_pts + 1;
      away_pts = away_pts + 1;
    } else if (score > 0) {
      home_pts = home_pts + 3;
    } else {
      away_pts = away_pts + 3;
    }
  }
  home_pts = home_pts/n;
  away_pts = away_pts/n;

  #adding points to scoreboard
  for(x in 1:length(standings$team)) {
    if (standings$team[x] == fixtures$home[k]) {
      standings$points[x] = standings$points[x] + home_pts;
    } else if (standings$team[x] == fixtures$away[k]) {
      standings$points[x] = standings$points[x] + away_pts;
    }
  }
}

#Sorting final standings
expected_points = standings[sort(standings$points,decreasing = TRUE), ]
rank = order(standings$points, decreasing = TRUE)
for(x in 1:nrow(standings)) {
  expected_points[x,] = standings[rank[x],];
}
row.names(expected_points)<- c(1:nrow(expected_points))
View(expected_points)

```

Acknowledgments

I would like to give a special thanks to my mentor Katie Gravel for introducing me to this topic and helping me through every step of this process. I would also

like to thank Peter May for setting up the whole program and Gregory Lawler for leading the probability sequence.

References

[1] Gregory F. Lawler. Random Walk and the Heat Equation. http://www.math.uchicago.edu/~lawler/reu.pdf
[2] Gregory F. Lawler. Stochastic Calculus: An Introduction with Applications. http://www.math.uchicago.edu/~lawler/finbook.pdf
[3] FBref. https://fbref.com/en/