When is an NBA Game Really Over?

Using Simulations to Discern When to Keep Tuning Into a Game

Chris Collins
Group 392
ISYE 6644
Georgia Tech

ABSTRACT

In this paper, we look at simulating outcomes of NBA games based on current in game state (score and time remaining), and historical team performance. Data for this project came from the nba_api, particularly the PlayByPlayV2 endpoint (swar, n.d.). This endpoint generated play by play data for NBA games that was used to build transition matrices that informed 3 iterations of simulation models, each with increasing specificity and success. Each of the models utilized a Markov Chain, built using increasingly more precise parameters, to create a transition matrix for simulating a sequence of events that the final score could then be calculated from. The best performing model, utilizing a Semi-Markov Chain to simulate transitions between possessions, was the most accurate, achieving 89% accuracy when starting the simulation from random fourth quarter events in NBA games and comparing the results to the actual outcome. The simulation reveals that 96% of games the Lakers are leading by more points than minutes remaining in the fourth quarter end up in a win.

1. Background and Description of the Problem

Basketball games are a dynamic affair, with many possible events occurring and each team adopting a unique style of play. For the busy NBA fan, knowing when to cash it in and flip to another game is an art form. Every fan wants to witness the key moments - the epic comebacks, the buzzer beater shots, and the heroic efforts in the final moments of a game that catapult a team to victory. But if you have watched the majority of a game and there appears to be no way for a team to come back, should you wait it out to see if something incredible happens or switch to another game or more productive activity than watching an NBA game?

For years, my wife has accused me of not being a true fan because I switch to another activity if my team, the Lakers, are down by a seemingly insurmountable margin. This drove the research question of this paper - when can an NBA fan most accurately deem that their team has won/lost a game before it is over?

Armchair Statisticians (a title I aspire to) have developed a rule of thumb that if a team is leading by more points than there are minutes remaining then they have an 80% chance of winning the game (Goel, 2012). Predictive basketball models previously made rely on Logistic Regression to calculate probabilities (FiveThirtyEight, 2021). This paper will explore how to use some of the concepts learned in ISYE 6644 to build simulations that allow us to understand when the Lakers are most likely to win the game and it is safe to switch to another channel.

2. Data Collection, Preparation, and Exploration

Many win-probability models utilize logistic regression on score differential and time remaining in the game. For this project, I used a data-driven Markov approach to drive event simulation. There is an abundance of data that is collected and made freely available through the NBA API. In order to dig into this question, I used the NBA API PlayByPlayV2 endpoint to pull data from just under 7,000 NBA games from the 2020/2021 season through March of the 2024/2025 season. Each game contained between 500 and 800 rows of data, with each observation signifying an event that transpired in the game and the timestamp it occurred, whether it's a shot made, shot missed, free throw, rebound, turnover, foul, violation, substitution, timeout,

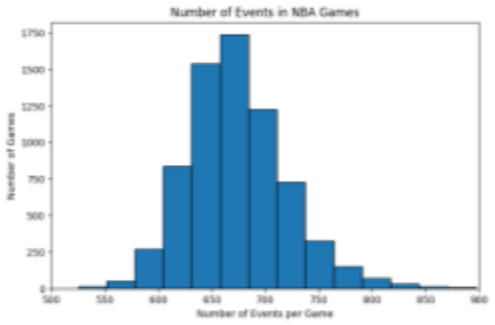

Figure 1. Histogram of Number of Events in NBA games.

jump ball, ejection, or the start/end of a period.

To understand how many events and what these events signified, I focused on a few key considerations - how many events occur in a game, how often do these events occur, and how often are these scoring events?

Figure 1 shows most NBA games have somewhere between 600 and 750 events that take place. For the sake of our analysis, only events marked as a 1 (Shot Made) or 2 (Shot Missed) are taken into consideration to build the simulation models. Figure 2 shows two important findings: first, games usually see between 169 and 184 shots taken throughout the course of a game, and second, that this dataset includes games that are not typical NBA games such as All Star Games or other exhibitions that might not follow normal behavior patterns of a typical NBA game.

In order to drive the possession component of the simulation, the time between each scoring event was taken into account. Figure 3 shows that most NBA games will see between 3 and 4 field goals (non free throws) taken per minute. This information will be used to inform the simulation models which will rely on simulating the amount of time between shots for each team.

	Shots	Makes	Misses
Min	30	16	14
25%	169	77	88
50%	176	82	94
75%	184	88	101
Max	289	163	136

Figure 2. Table of Shots Per Game

3. Simulation Attempts

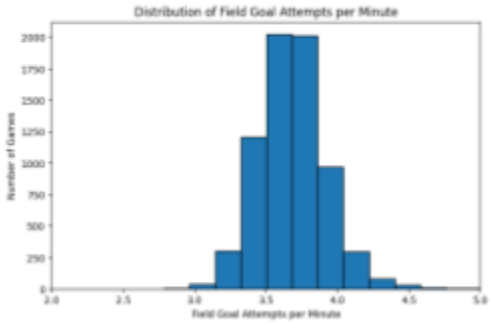

Figure 3. Histogram of Number Shots Taken Per Minute in an NBA Game

3.1 Model 1 - Standard Possession Time, Event to Event Transitions

Model 1 first sorted all events in a game sequentially and shifted the next event down so that each current event points to the future event to build a transition matrix. Any events that did not have a subsequent event were dropped (ie: last play of the game) and the remaining events built the transition matrix. The simulation function uses a dictionary that describes the

game state, defined as quarter, time remaining, score margin, and the transition matrix created based off all team data from the whole dataset. It then loops through the number of specified times to simulate that game, initializing the game time and score margin while using the Markov Chain and associated probabilities to simulate each subsequent event. For simplicity, the Markov Chain only includes states for events No Shot, a 2 Point Shot Make, a 2 Point Shot Miss, a 3 Point Shot Make, a 3 Point Shot Miss, a FT Shot Make, and a FT Shot Miss. The loop continues selecting subsequent events based on the probabilities in the transition matrix and updating each team's score until time remaining runs out, where it determines the team winner of that game. Each non-free throw shot attempt used a standard 20 seconds of clock to match

This model did not account for many team specific considerations, such as pace, shooting percentage, and breakdown of type of shots teams typically take, split between 2 point, 3 point, and free throws. The model was tested simulating each game 100 times and counting the number of times each team won to generate a win probability for the home team. The win probability for the home team correctly predicted the winner (giving them a greater than 50% chance of winning) 53% of the time. In other words, it only performed slightly better than a coin toss despite the data that informs it.

3.2 Model 2 - Dynamically Calculated Possession Time

While the first model used the same parameters for shooting percent and possession time for every team, this second model builds a transition matrix based on a team's shooting percentages over the last 5, 10, and 15 games to try to account for injuries, strategy tweaks a team may make, and recent hot/cold streaks. The model weights the last 5 games as 50% of the calculation while 10 and 15 games are each weighted as 25%. For possession time, the same weights were applied to the same time horizons in order to create a normal distribution with each team's weighted mean and standard deviation of possession time informs how much time each event should run off the clock.

This specificity led to meager gains in model accuracy, with 58% of games being correctly predicted using the model. One downside to making this many calculations is the amount of time that it takes to simulate. It took over 20x longer to simulate games in the second model than it did for the first model.

3.3 Model 3 - Holding Time Dictionary

The Markov Chain provided a nice foundation for the first two models, but it was falling short

because of its "memoryless" property that treats each subsequent event as independent from the previous. This means that, in our first two models, there could have been a run of shots in sequence or a long stretch with no shots occurring. The third model aimed to address this shortcoming by fitting each team's possession time to a distribution using the Kolmogorov-Smirnov (KS) Goodness of Fit Test, which compares a dataset with a known distribution to determine if they have the same distribution (Statistics How To, n.d.). Each team's possession distribution was tested and the KS test revealed that a lognorm distribution best fit possession times for all NBA teams.

Model 3 built in realistic dwell/holding times so that the process is not truly memoryless. The model first created a transition matrix for P to measure the state-to-state transition probabilities. For each transition (say, from s0 to s1), the time-difference between states was collected and the parameters for a lognormal distribution were stored in a dictionary F. This allows the simulation to select the next state based on P and run down the clock more accurately based on the dictionary F. The results of this increase in accuracy was the model correctly predicting a game's winner 89% of the time!

4. Main Findings

While basketball is a relatively simple sport, there is a surprising amount of complexity in trying to simulate it. There are 14 different game states that each use a different amount of game clock. In order to speed the simulations up, a simplified game state matrix was used that events No Shot, a 2 Point Shot Make, a 2 Point Shot Miss, a 3 Point Shot Make, a 3 Point Shot Miss, a FT Shot Make, and a FT Shot Miss. The best performing model was also the most complex model, accounting for the distribution of possession time between states in the Markov Chain.

This model was used to simulate many game states to see how accurate the 80% win chance when leading by more points than minutes remaining "rule of thumb" was. While it's not a bulletproof mathematical way to look at the data, it was pretty fun to make it simulation based. I simulated 144 game states, from 1 to 12 minutes remaining and a score

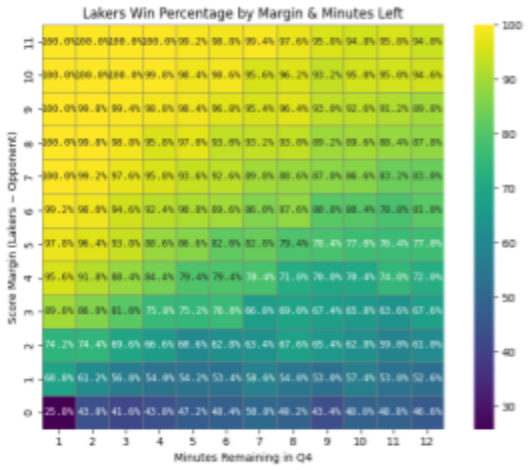

Figure 4. Simulated Predictions of Lakers Game Outcomes Depending on 4th Quarter Game States

margin from 0 to 11.

Figure 4 shows that for 100% of games where the Lakers were leading by more points than there were minutes remaining, the Lakers were predicted to win. The average winning chance for these states was 96%. When the score margin is only one point higher than the minutes remaining, the average winning percentage in the simulations is about 89%. This built the transition matrix and holding times distributions from the last 25 games in the dataset on the date the data was pulled, 3/14/21. For context, the Lakers had been playing especially well during that stretch and had traded for all world talent Luka Doncic on 2/2/25 where they continued their great stretch of play (Wikipedia, 2025). This is probably not the most representative time period to build a simulation off of, but I am ok living in this Lakers-optimistic world. Maybe that's why it's accounting for a higher win probability than the 80% that other "armchair statisticians" previously reported (Goel, 2012).

While Model 3 predicted games when starting in the fourth quarter well, it was not as accurate for games starting in the first, second, or third quarter. Model 3 only correctly predicted 59% of games when they were not required to pull their random starting moment from the fourth quarter.

5. Conclusions

In this study, three increasingly complex and specific simulation models were built - the first two based on the traditional "memoryless" Markov chains and the third based on a Semi-Markov framework that incorporates possession time distributions. These models were used to predict NBA game outcomes by starting a simulation at a random event in the fourth quarter and simulating all subsequent games until the game clock ran out. Model 1, the most basic, only did slightly better than a coinflip by correctly predicting 53% of game outcomes and, despite adding in more layers of complexity by calculating transition matrices specific to each team and pulling from a normal distribution for possession times between states, Model 2 only did slightly better at 58%. Model 3 fit a lognorlam distribution to the holding time between each combination of states. This proved to greatly increase the accuracy as Model 3 correctly predicted 96% of random NBA games.

Despite these exciting results, there is much more work to be done on this topic. The models, while they became team specific, never accounted for the players on the floor. Incorporating player lineup specific data, extending the dataset across more years, and including starting betting odds would likely improve the performance on the model and allow us to more accurately simulate game outcomes.

6. Acknowledgements

In this project, I used AI as a thought parter to help me in two ways. First, while coding in Visual Studio I used Github Copilot to turn pseudocode into workable code. It was rarely functioning correctly the first time, so I tested, documented, and would troubleshoot the code that Copilot helped produce.

I also used ChatGPT as a thought partner to give me suggestions on how I can be more specific in the models to produce greater results. Chatting through the results and the setup of the functions was what allowed ChatGPT to make the suggestion of building a dictionary of dwelling times that pulled from the lognormal distribution and much more accurately predicted the outcomes of games.

7. References

[1] FiveThirtyEight. (2021, May 5). How our March Madness predictions work. FiveThirtyEight. https://fivethirtyeight.com/methodology/how-our-march-madness-predictions-work-2/

[2] Goel, S. (2012, May 31). How close is close. Messy Matters. https://messymatters.com/moneyball/

[3] swar. (n.d.). nba_api (Version 1.9.0) [Computer software]. GitHub. Retrieved March 17, 2025, from https://github.com/swar/nba_api

[4] Statistics How To. (n.d.). Kolmogorov–Smirnov goodness of fit test. Statistics How To. Retrieved April 13, 2025, from https://www.statisticshowto.com/kolmogorov-smirnov-test/

[5] Wikipedia. (2025, April 22). Luka Dončić–Anthony Davis trade. In Wikipedia. Retrieved April 18, 2025, from https://en.wikipedia.org/wiki/Luka_Don%C4%8Di%C4%87%E2%80%93Anthony_Davis_trade