Simulation of the 2018-19 NBA Regular Season

Alemayehu Solomon Bogale
Department of Physics, University of Chicago, USA

Abstract
Sports provide very large datasets from numerous decades. Until recently much of this data had just been used as very informal measure of individual and team performance but now there seems to have been an explosion of data analytics with numerous statistical categories that have a large influence on how teams and players behave and play. The reason I choose the NBA rather than other professional sport organizations is because it provides a long enough schedule and playoff series which minimizes the role of "luck" and deviations in performance. I believe all this data will allow for possibly simulating how the season will end, predict how the playoffs will turn out , and possibly how individuals will perform. Using simulations we would use the initial state of the NBA such as: standings, performance, and roster to predict the final outcome of the season.

Theory

In order to simulate the season correctly we have to create probability distributions for each of the statistical categories such as: field goal percentage, field goal attempts, 3-point field goal percentage, 3-point field goal attempts, free throw percentage, free throw attempts. Then we would have to sample from these probability distributions. Ultimately calculate the scoring for each of these teams and iterating over the entire season

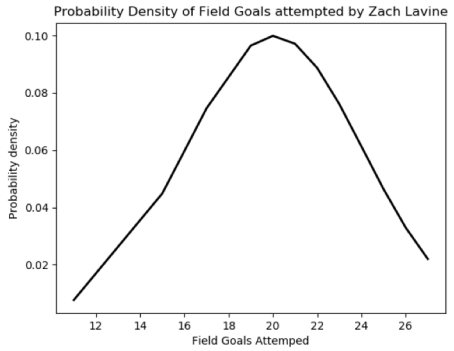
Team Per Game Stats * Playoff teams Share & more ▾ Glossary

Rk	Team	G	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%
1	New Orleans Pelicans	25	240.0	43.8	91.6	.479	10.0	28.8	.345	33.9	62.8	.540	20.5	26.6	.769
2	Toronto Raptors	25	242.0	43.9	89.1	.492	11.7	33.7	.346	32.2	55.4	.581	17.1	21.4	.801
3	Golden State Warriors	25	242.0	43.5	87.7	.496	11.3	29.4	.384	32.2	58.3	.552	18.1	22.0	.824
4	Philadelphia 76ers	25	244.0	40.3	87.2	.462	11.0	31.4	.352	29.2	55.8	.524	22.6	29.1	.775
5	New York Knicks	25	243.0	39.8	91.0	.438	10.0	29.2	.344	29.8	61.8	.482	19.3	24.6	.785

FIG 1. An example of statistics that we extract data from

Implementation

This simulation is a Python based object oriented approach and takes advantage of Python's open source, scalability, documentation system, powerful scientific computing packages, which include NumPy, SciPy, and matplotlib -- an extremely powerful and widely accessible plotting package. These packages make analysing experimental proton radiography data simpler and more efficient.



This project consists of 3 classes, Player, Team, and Game. The Game class takes advantage of the inheritance mechanism in Python and takes two of the Team classes and simulates the scoring for 100 possessions and then includes the free throws made by each team. Each Team class contains multiple Player classes. The scoring is simulated by using a probability distribution of each player for each statistical category and summing the sampling

Game Simulation

When two Teams go head to head the amount they score each possession is determined by the sampling the probability distribution of each player on the roster. However this alone would not be appropriate in determining the outcome of the game because the opponent's defense does affect with the efficiency one can score. In order to account for this I took the mean of their opponents defensive field goal percentage and field goal percentage.

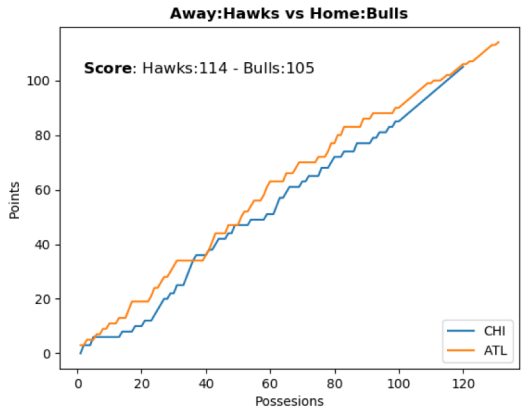


FIG 3. Visualization of the Chicago Bulls vs Atlanta Hawks by possession

Season Simulation

In order to simulate the whole NBA regular season with 1230 games we create 30 Team objects and iterate through all these games using statistics from games that have already been played. With each iteration the stats are updated and used for the following iterations.

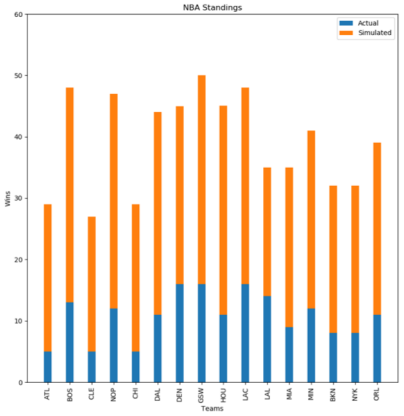


FIG 4. NBA Standings after 82 games for NBA team

Conclusion

When it comes to simulating human behavior there are many factors to consider. Basketball simulations are not exempt from this because there are many considerations with fluctuations in rosters such as injuries and trades and coaching adjustments. There are also many factors cannot be captured with statistics at this moment. However this Python project (https://github.com/AlemayehuB/250_final) allows for a skeleton for future projects that can account for more offensive, defensive statistics, and how these interact with each other.

References

[1] Seemethere. "Seemethere/nba_py." *GitHub*, 28 June 2017, github.com/seemethere/nba_py.

[2] "Basketball Statistics and History." *Basketball-Reference.com*, www.basketball-reference.com/.