Stat 400 Final Project

Jack Allen, Danny Laposata, Artie Palen

12/16/2021

Introduction

For our final project we wanted to replicate something that had to do with sports. After some digging we found an article called "Predict NBA Player Lines with Monte Carlo Simulation" by Carter Bouley. In his work he uses Monte Carlo to simulate thousands of NBA games and find not only a single estimate for how many points a player will score during the game, but a distribution of that players scoring over the thousands of games. All of us thought that it would be interesting as well as useful to simulate something like this.

How does sports betting work? When the oddsmakers release the betting line, they will try to accurately predict what is going to happen in a game. There are all types of things that you can bet on for a game. Take basketball for example, you can bet on things like the team to win, how much they will win by (this is called the spread), how many points and other stats players will produce, the total points for a game, the first player to score, etc.. For an example, we will look at the total points scored between two teams. When the oddsmakers release the betting line, you can bet on the total amount of points that they predict will be scored in that game, you can either predict the over or the under. In sports betting the odds will show as negative or positive. Negative means that it is favored to happen, while positive means that it is the underdog. For example, say the total points estimated to be scored in a basketball game is 200. This is a little low, so the odds that the score will be over that is -150, while the odds that the total score is under 200 will be +200. This means that if you bet $100 on the total points to be over 200, with -150 odds, you will win $66.67 and be returned with $166.67. If you bet $100 that the total points for the game will be under 200 with +200 odds, then you will win $200 and be returned with $300.

One would think, it is hard to make money in sports betting over the long term. The famous saying, "The house always wins" has a lot of truth to it. Given this, the motivation behind our project was that we might be able to use new these statistical methods to try and get an edge on the sportsbook. The rule of thumb is that a sport better needs to be winning at least 52.4% of their bets to just break even over the long run. There is a reason that the casinos and sports books have so much money, it is because most of the time they win. If we could use Monte Carlo simulation to estimate game lines, then maybe we can get a little bit of a mathematical edge on the sports book and boost winning percentages to over 52.4%. There are professional sports betters who are betting on these sportsbooks that consistently turn a profit. While they might be in the minority, it is at least possible to be wiser than the crowd.

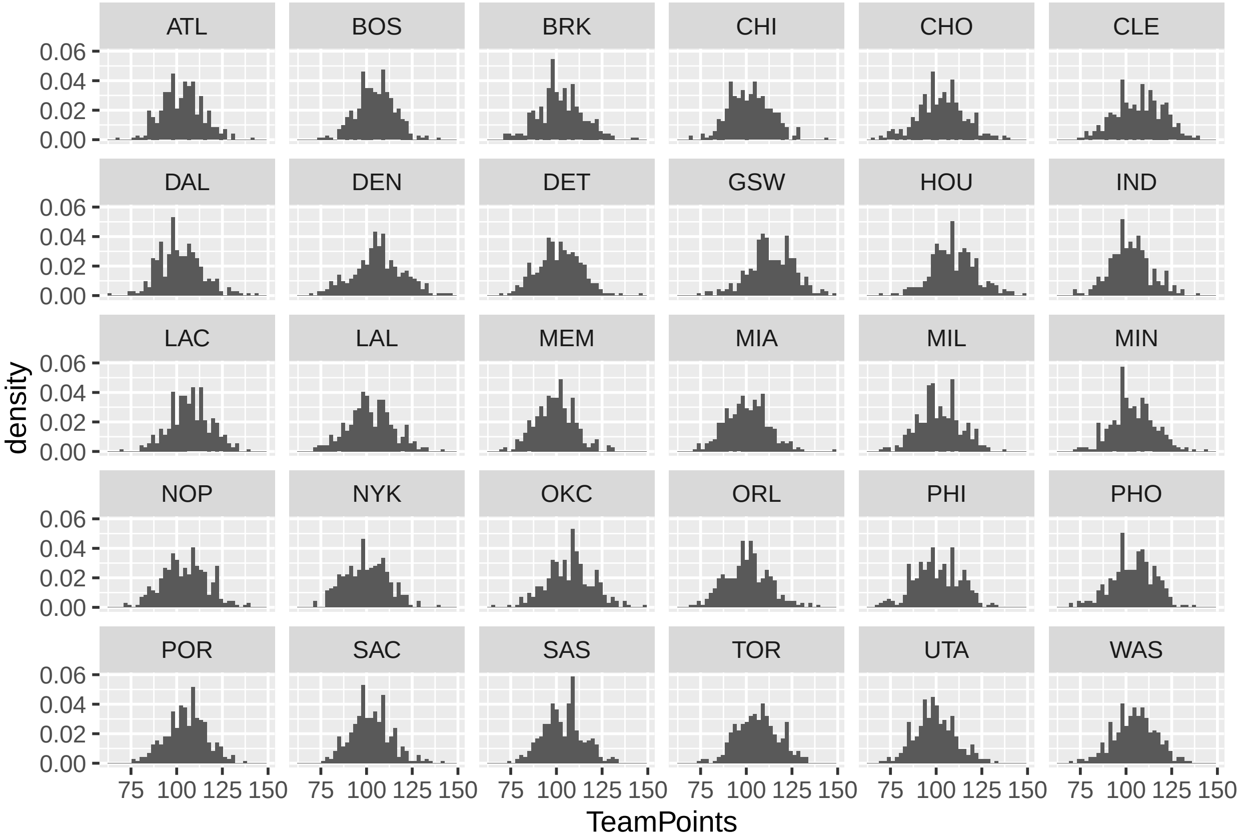
Most sports betters use what is called the "Efficient marketplace hypothesis". This suggests that all of the pricing within the sports book reflects all of the available information. In more simple terms, the prices on any event should always accurately reflect that events chance of occurring. Using this information, it is possible that we can build distributions to give us estimates that are not part of the efficient marketplace hypothesis and gain an edge over the sportsbook. Sports books try to accurately reflect what is going to happen in the game, but they also want the money on each side to be split, that way they are consistently turning a profit. If the majority of the public is betting money on one side of the game line, then the sports book will try and alter the line to have more people bet the other side. In order for this to work, we can only use it in markets where there is lots of people betting in the market. There are lots of markets with a lack of interest and people betting, so we do not want to try and estimate these because there is a lower chance that it is correctly priced.

Monte Carlo simulation is especially useful when there is a lot of randomness and variation when predicting an event. Instead of determining a result with a degree of uncertainty like most algorithms do, it instead uses a range of probabilities to give us a result. This is important because when we want to be able to accurately simulate games to be able to build fair lines that we can then compare against the sports book. There is so much randomness in sports such as injuries, lineup changes, coaching changes, illness, weather, fans, etc. The list really does go on and on, so looking at all of the historical data might not help us much. Instead, it will be more useful to predict game lines using Monte Carlo estimation instead of using the historical data. It is also going to be important for us to be able to be simulating a large number of games. Since there is so much randomness within games, we will need to be simulating at least a thousand games. Having a large sample size of games that we simulate will give us a much better estimation of the true outcome. It is important to remember that casinos make their money over the long run. They are not concerned with what happens with one game, but instead what happens over thousands of games.

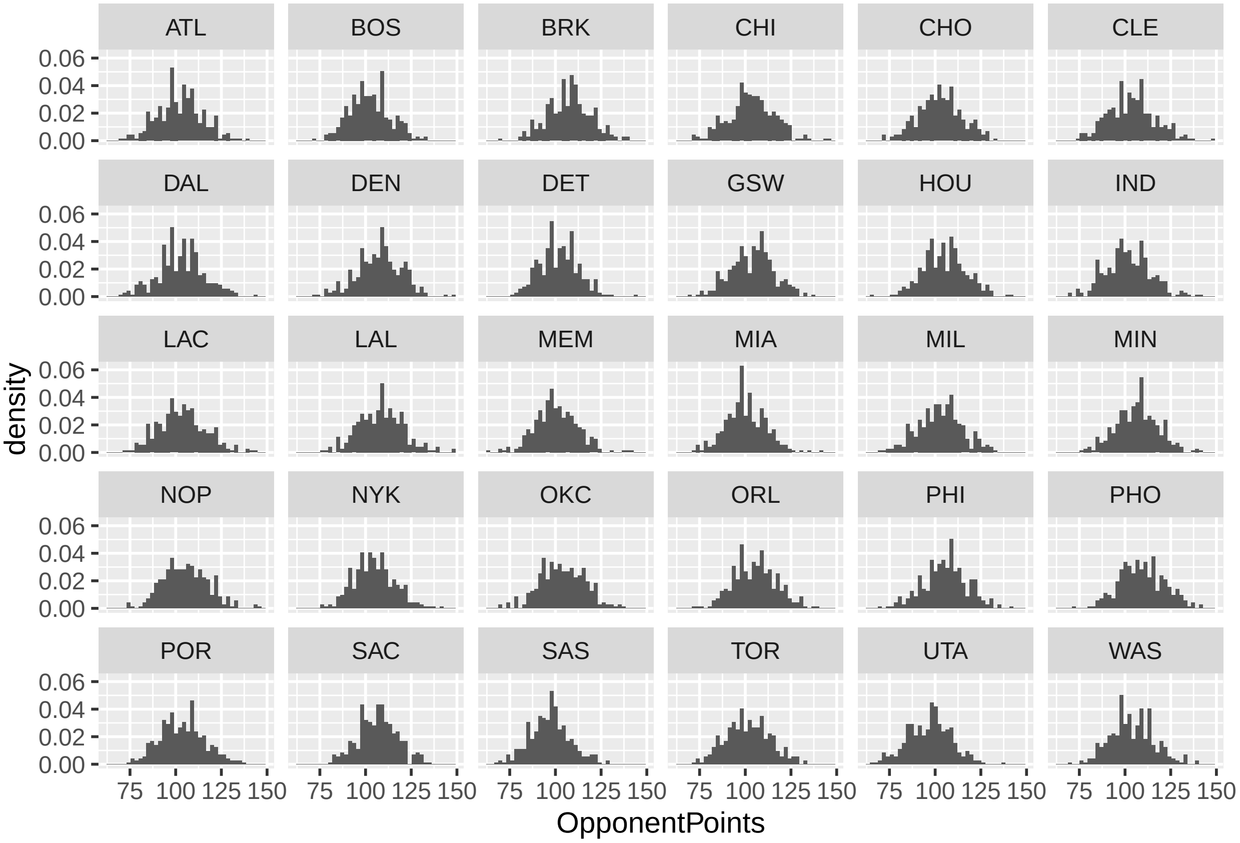
For this project we looked specifically at the National Basketball Association. The data for this project was easy to access and look at. There are lots of datasets that contain all of the data of box scores and game recaps from the last decade. The paper that we looked at had some great insights on what to be looking for when we build our model. In is paper, Bouley looks at multiple graphs. The first thing that he looked at was the disparity in the number of points that each position scores, and the home court advantage when it comes to scoring points. When the player is playing at home, they tend to score a little bit more than when they are on the road. This is to be expected, but it is something we will need to keep in mind when building our model. Bouley also looks at the points for and points against for teams as well as for players. He plots multiple histograms of the player and team points and points against, and all of the plots look relatively normal. This is important because since he was able to look at the data and conclude that they are relatively normal, we can do the same since the data is the same.

Model Assumptions

```r
game_stats <- read.csv("nba.games.stats.csv")
game_stats$Date <- substr(game_stats$Date,1,4)
## Distribution Plots of PTS For and PTS Against by Team in 2018
game_stats %>% group_by(Team) %>% ggplot() + geom_histogram(aes(TeamPoints,..density..),bins=40) + face
```



```
game_stats %>% group_by(Team) %>% ggplot() + geom_histogram(aes(OpponentPoints,..density..),bins=40) +
```

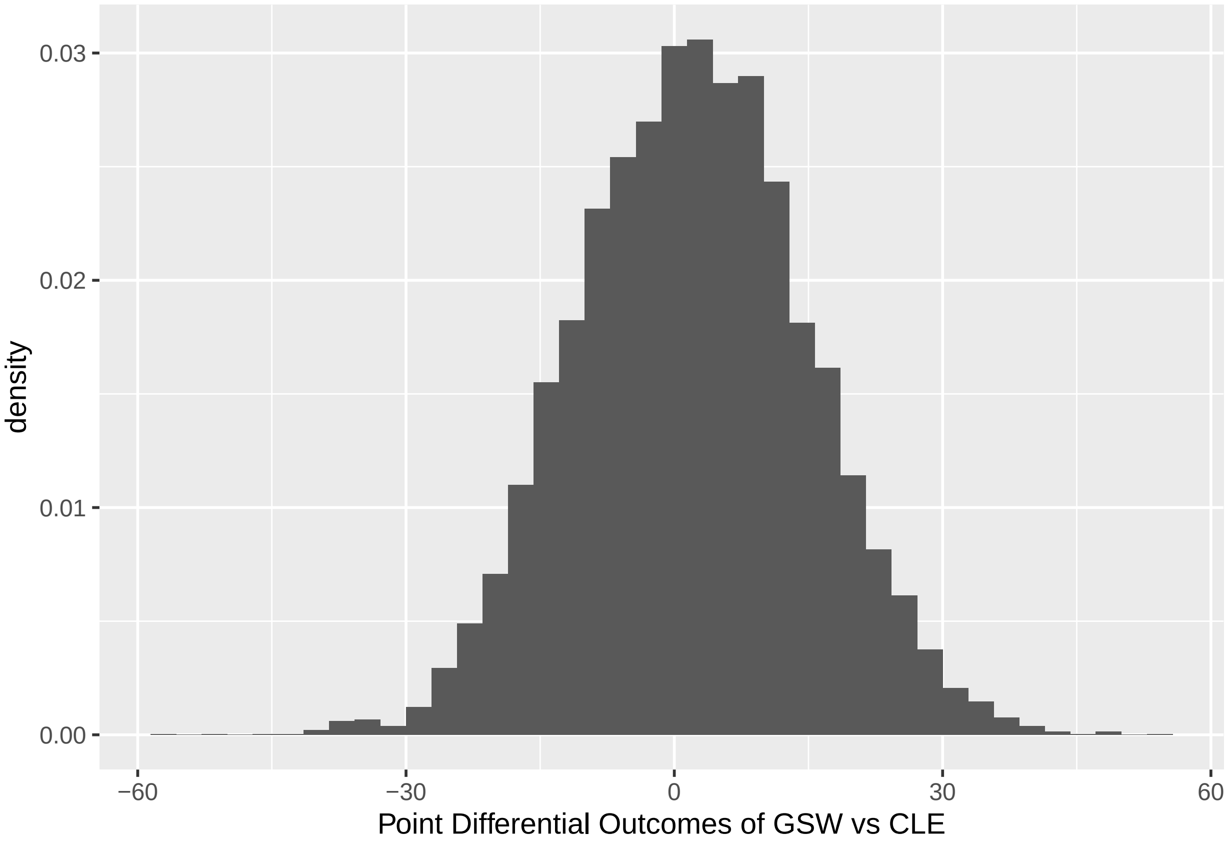


Carter Bouley builds his model to look at individual players statistics when they are playing against another team. Instead of doing that, we thought that it would be interesting to look at team stats instead of individual stats, as well as doing this will make us not have to account for all of the 0's that the Monte Carlo estimation will produce. We were able to find a data set on Kaggle.com that contained all of the team and individual player stats from 2014 to 2018. As you can imagine, this dataset is huge with over 100,000 observations. When we built our models, we wanted to take into account the historical data to model off of, but we do not want the data from 2014 to have as much influence as the data from 2018, so when we did build the models, we had to weigh the data by the recency of the year, where obviously the more recent data has a bigger impact than the data from 2014.

Finally, we built a couple models. The first being a spread model and the second being an over under model. Our model has a few more assumptions, the first being that the game simulation can be done simply. The second is that there are hundreds of factors that can influence the outcome of a game. It is not possible to model something like that without it becoming a black box. So, we will keep the model simple and this way we will be able to understand what it is doing and avoiding the black box. Our model also assumes the points scored by a team is equivalent to their average points for added to their opponent's average points against, divided by 2. First we needed to subset all of our data by year and then add a weight to the data for each year. We weighted the year 2018 by 0.85, the games in 2017 by 0.1, the games in 2016 and 2015 by 0.02, and the games in 2014 by 0.01. For the spread model, we looked at games between the Golden State Warriors and the Cleveland Cavaliers. We picked these because our most recent data is from 2018, and in 2018, the Warriors and the Cavaliers were the two teams in the finals. First we built a function called game_outcome that takes parameters of the home team and away team and simulates the points for and against using the rnorm() function in r. We also do the same thing for away points for and against. Then we take the outcomes and average the home and away scores to find the different between the two games.

Spread Model

```
## Spread line mc prediction function
game_stats_14 <- game_stats %>% filter(Date == "2014")
team_14 <- game_stats_14 %>% group_by(Team) %>% summarise("AvgPtsFor" = mean(TeamPoints),"AvgPtsAgt" =

game_stats_15 <- game_stats %>% filter(Date == "2015")
team_15 <- game_stats_15 %>% group_by(Team) %>% summarise("AvgPtsFor" = mean(TeamPoints),"AvgPtsAgt" =

game_stats_16 <- game_stats %>% filter(Date == "2016")
team_16 <- game_stats_16 %>% group_by(Team) %>% summarise("AvgPtsFor" = mean(TeamPoints),"AvgPtsAgt" =

game_stats_17 <- game_stats %>% filter(Date == "2017")
team_17 <- game_stats_17 %>% group_by(Team) %>% summarise("AvgPtsFor" = mean(TeamPoints),"AvgPtsAgt" =

game_stats_18 <- game_stats %>% filter(Date == "2018")
team_18 <- game_stats_18 %>% group_by(Team) %>% summarise("AvgPtsFor" = mean(TeamPoints), "AvgPtsAgt" =

## Weighting the data based on recency

wgt_pts_for <- team_18$AvgPtsFor*.85 + team_17$AvgPtsFor*.1 + team_16$AvgPtsFor*.02 + team_15$AvgPtsFor
wgt_pts_agt <- team_18$AvgPtsAgt*.85 + team_17$AvgPtsAgt*.1 + team_16$AvgPtsAgt*.02 + team_15$AvgPtsAgt
wgt_sd_for <- team_18$SDPtsFor*.85 + team_17$SDPtsFor*.1 + team_16$SDPtsFor*.02 + team_15$SDPtsFor*.02
wgt_sd_agt <- team_18$SDPtsAgt*.85 + team_17$SDPtsAgt*.1 + team_16$SDPtsAgt*.02 + team_15$SDPtsAgt*.02

wgt_pts_stats <- cbind(wgt_pts_for,wgt_pts_agt,wgt_sd_for,wgt_sd_agt)
row.names(wgt_pts_stats) <- team_18$Team

game_outcome <- function(dat,home_team, away_team,m){
  outcomes <- rep(NA,m)
  for (i in 1:m){
    home_pts_for <- rnorm(1,dat[home_team,1],dat[home_team,3])
    home_pts_agt <- rnorm(1,dat[home_team,2],dat[home_team,4])

    away_pts_for <- rnorm(1,dat[away_team,1],dat[away_team,3])
    away_pts_agt <- rnorm(1,dat[away_team,2],dat[away_team,4])

    outcomes[i] <- ((home_pts_for + away_pts_agt) / 2) - ((away_pts_for + home_pts_agt)/2)
  }
  outcomes
}
## 10000 MC samples of Golden State vs Cleveland Point Differential
gsw_cle <- game_outcome(wgt_pts_stats,"GSW","CLE",10000)
ggplot() + geom_histogram(aes(gsw_cle,..density..),bins=40) + xlab("Point Differential Outcomes of GSW
```



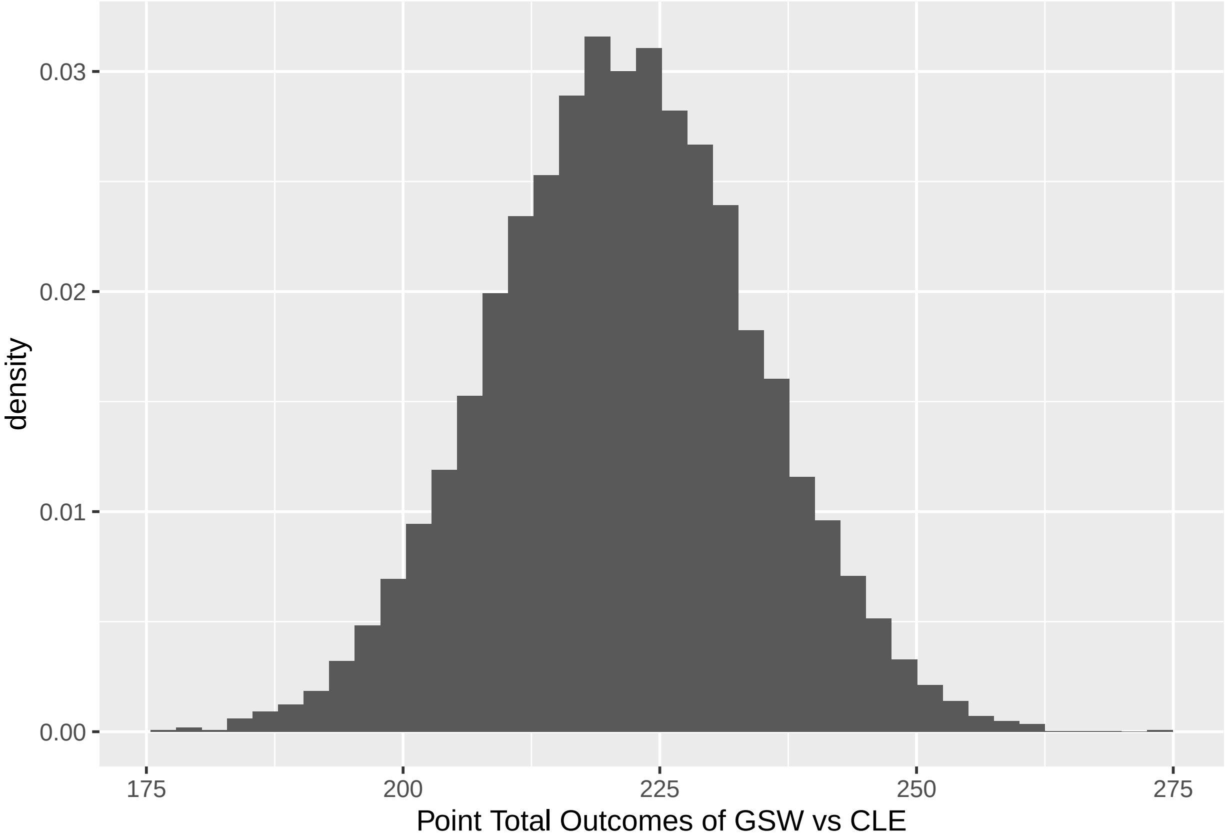
```r
mean(gsw_cle)
```

```
## [1] 1.987117
```

For our second model we wanted to look at predicting the over under of a game. Again, the over under refers to the number of total points scored by a team. It is called the over under because you can either bet the over of the number of points that the sports book predicts, or you can bet the under of the total points by that team. We first build a function to take inputs of home team data and away team data. The first thing that we did was create a blank vector of 10,000 observations. Then we simulate the home points for and against the team using the rnorm() input using the data from the home team. We then do the same thing for the away team. Then we take the average of home team points for and away teams for and the take average, then we do the same thing and take the average of home team and away team points against. We then add these two quantities and store them in our vector of outcomes. For this example, we are looking at the totals for the Cleveland Cavaliers versus the Golden State Warriors.

Over/Under Model

```r
## Over/Under line mc prediction function
over_under_outcome <- function(dat,home_team, away_team,m){
  outcomes <- rep(NA,m)
  for (i in 1:m){
    home_pts_for <- rnorm(1,dat[home_team,1],dat[home_team,3])
    home_pts_agt <- rnorm(1,dat[home_team,2],dat[home_team,4])

    away_pts_for <- rnorm(1,dat[away_team,1],dat[away_team,3])
    away_pts_agt <- rnorm(1,dat[away_team,2],dat[away_team,4])
    outcomes[i] <- ((home_pts_for + away_pts_agt) / 2) + ((away_pts_for + home_pts_agt)/2)
  }
  outcomes
}
gsw_cle2 <- over_under_outcome(wgt_pts_stats,"GSW","CLE",10000)
ggplot() + geom_histogram(aes(gsw_cle2,..density..),bins=40) + xlab("Point Total Outcomes of GSW vs CLE
```



```
mean(gsw_cle2)
```

```
## [1] 221.502
```

Comparison

```
gsw_spread <- c(22, 19, 5, -21, 9, 10, 19, 8, 23)
mean(gsw_spread)
```

```
## [1] 10.44444
```

```
sd(gsw_spread)
```

```
## [1] 13.52878
```

```
gsw_over <- c(204, 245, 231, 253, 249, 238, 225, 212, 193)
mean(gsw_over)
```

```
## [1] 227.7778
```

```
sd(gsw_over)
```

```
## [1] 21.00463
```

For some context, we looked at the 2017 and 2018 NBA Finals. These two series were played against the Cleveland Cavaliers and the Golden State Warriors. The amount of games played between the two teams in total was 9 games where the Warriors won 8 of the 9 and won the championship in both seasons. If there were to be a hypothetical 10th game between the two teams, the recent games would suggest to take the Warriors on the spread and the over for the over-under bet. Our model shows that these values are still very likely to be closer to what is expected and closer to what Vegas would set as the odds for each scenario. The standard deviation of both values would also suggest that the sample size for both cases is too small.

Conclusions

The model is very much in line with what we would expect to see from Vegas when they give the spread and over-under for each game. That is because in the end, our model is still very simple. There are many other factors that go into whether a team will have better or lesser odds than the game prior. Injuries, shifting each player's amount of minutes played, how well each player sets up their teammates, rebound efficiency, etc. all have sizable factors when it comes to how a game will end. The model we used was built on points scored and how well the defense of each opponent. There are even ways to break up those components into smaller increments. For example, we could look at each individual player when it comes to defense and how well each defender defends against certain positions on offense.

There are still many positives that can come from this model. It shows that Monte Carlo can be viable option when it comes to the world of sports betting. It also shows that using points to evaluate NBA scores is a very good place to start and gives a jumping off point for making a even more complete model. It would be in the best interest of anyone reading to not use our model to bet on sports games as this is not the most complete evaluation. There is still plenty to take away from this model and could shape for better models down the line for sports betting.