Project Report: Simulation of the San Diego Padres 2019 Baseball Season

The goal of our project was to use simulation in R to model the 2019 season of the San Diego Padres. Each of the nine players in the Padres lineup has a unique probability of either making an out or reaching base, and if they reach base, they further have their own individual probability of which of the four bases they reach. The more technical details of how the lineups were constructed can be found in the Construction of the Padres Lineup in the details folder of our repository. The event of interest in each inning is how many runs score, which occurs anytime a player crosses four total bases to reach home plate. An inning ends after three outs are made, and the team that wins the game is the one that scores the most runs after nine innings.

Because each of the nine batters have a unique probability of reaching base and they bat sequentially until twenty-seven outs are reached and the game ends, we calculated a large number of uniform random variables from 0 to 1. If the uniform random variable was less than the corresponding player's on base percentage then the player reached base, coded as a 1, and if not, he made an out, coded as a 0. We then calculated the same number of samples of the numbers 1, 2, 3, and 4 with probabilities corresponding to the probability of the player reaching each of those bases had they gotten on base. These two vectors were multiplied together to determine the outs and bases reached. The final vector was then split into nine intervals, separated by when three outs were reached, to represent the nine innings in which the Padres would have batted in a game.

To determine the outcome of each game, the runs scored in each inning were summed up and then compared to the average adjusted runs allowed per game that Padres pitchers gave up in 2019 of 5.87. The more technical details of how the adjusted runs were calculated can be found in the Construction of the Adjusted Runs in the details folder of our repository. Because this

value was a decimal and the Padres could only score a whole number of runs, if they scored more than the rounded up average adjusted runs allowed per game, or 6 runs, then they won the game. If they scored exactly 5 runs, then we ran a random binomial test, so that they won these games 13% of the time. And if they scored less than 5 runs, they lost the game. We then simulated 162 games to represent a season of baseball and added up the number of games that the Padres won. We then simulated 250 seasons to see the distribution of games won. Here is the resulting histogram:

Simulation of the Padres 2019 Season
Dashed Line at the Padres Actual Number of 2019 Wins of 70
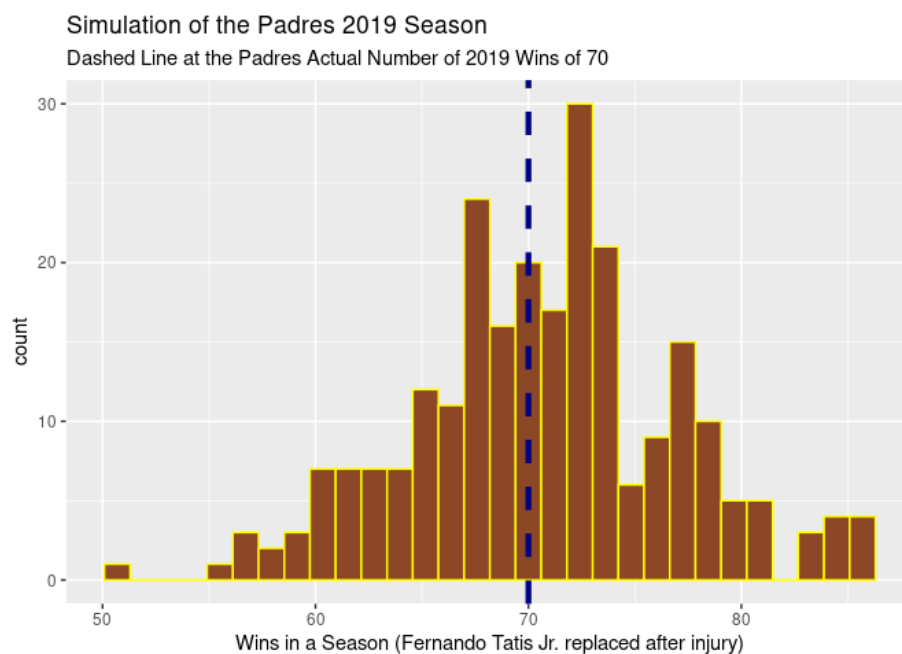
Our secondary question of interest was whether the San Diego Padres would have performed better had their best hitter Fernando Tatis Jr. not gotten injured halfway through the 2019 season and been replaced with a worse hitter. The simulation of seasons was rerun in the same manner 250 times; however, the first position in the lineup had only Tatis Jr.'s season statistics instead of the average of him and his replacement Luis Urias. The average number of wins of the original simulation and the second one were actually very similar, 70.7 compared to 71.1, but the simulation which included Tatis Jr for the entire season had a higher maximum

number of wins that would have put them pretty close to the 90 wins needed to make the playoffs in the National League in 2019. However, overall whichever lineup was used did not seem to make too much of a difference in terms of the average number of wins. Because the number of simulations run was on the smaller side due to time, we ran both several times to check for consistency and both simulations resulted in about 70 – 71 wins on average, very similar to the number of games that the Padres actually won during the 2019 season of 70.

References:

https://www.baseball-reference.com/teams/SDP/2019.shtml

https://www.mlb.com/padres/scores/2019-08-07

https://www.baseball-reference.com/leagues/MLB/2019.shtml