On Simulation as an additional framework for evaluating Baseball

Zachariah Zanger M.S.

Wareham Gatemen Baseball

Pushing for new Paradigms

In Baseball and Political forecaster Nate Silver's, *The Signal and the Noise: Why So Many Predictions Fail- But Some Don't*, Silver writes that, "Good innovators typically think very big and they think very small. New ideas are sometimes found in the most granular details of a problem where few others bother to look. And they are sometimes found when you are doing your most abstract and philosophical thinking, considering why the world is the way that it is and whether there might be an alternative to the dominant paradigm". Prior to the 2008 financial market collapse brought upon by the destruction of the United States housing market, all the major players in the global financial system subscribed to the paradigm that securities packaged with cash flows coming from mortgages experienced minimal defaults historically and therefore the future of mortgage cash flows must look the same. Mortgage cash flows could be sliced and diced by market makers to appeal to every investor risk profile. The existing and dominant paradigm for mortgage cash flows at the time brought the global financial system to its knees and changed our lives forever. (Trading mortgage cash flows never went away by the way. After working in the market and pricing mortgage backed securities for 2 years, I can tell you the market is alive and well, the money made on those things is just too good for market participants to pass up).

The nature of the consequences due to a failed paradigm in Baseball are not the same. But if today's overwhelming paradigm for evaluating Baseball teams and players on offense consists of attaching linear weights to outcomes derived from past empirical incidence (think WAR, wOBA), or manipulating stat lines by an equation, (think OPS, ISO) then it is borderline necessary to experiment with different frameworks and paradigms. One such framework is simulation. Games of Baseball move discretely (one at a time, with a set number of outcomes) between 24 combinations of baserunners and outs. If a model could simulate this process, and utilize player skillsets as inputs, users may be able to form inference from the model in ways that other frameworks cannot. This writing is an examination of a simulation framework for evaluating Baseball players and teams. The simulation model proposed is a first step in developing a more robust and intricate, forward looking evaluation framework.

Inputs

To begin, the model needs a simulation number. The simulation number is the number of times the model loops through the lineup. The model also needs a lineup to determine the order of batters to draw from. It looks like this:

```
Nine_Batters('Braiden_Ward','Darren_Baker','Dallas_Beaver','Adrian_Del_Castillo','Kameron_Guangorena','Jacob_Teter','Mike_Antico','Matt_Mclain','Matt_Rudick',3)
```

Each player in the order has his own probability distribution of Batted Ball Outs, Strikeouts, Walks, Singles, Doubles, Triples, and Homeruns. For example, the distributions of the 2019 Wareham Gatemen hitters using 2019 college season stats, would look like this:

	batted_ball_out_%	strikeout_%	walk_%	single_%	double_%	triple_%	homerun_%	
Adrian_Del_Castillo	0.52	0.09	0.11	0.16	0.08	0	0.04	1
Kameron_Guangorena	0.47	0.22	0.09	0.16	0.04	0.01	0.01	1
Andrew_Thomas	0.45	0.15	0.11	0.17	0.1	0	0.02	1
Darren_Baker	0.51	0.13	0.09	0.25	0.02	0	0	1
Benjamin_Sems	0.47	0.15	0.13	0.18	0.05	0	0.02	1
Chad_Stevens	0.53	0.14	0.08	0.17	0.06	0	0.02	1
Jacob_Teter	0.46	0.14	0.14	0.19	0.04	0	0.03	1
Braiden_Ward	0.45	0.18	0.11	0.19	0.05	0.02	0	1
Jack_Winkler	0.45	0.21	0.08	0.18	0.06	0	0.02	1
Mike_Antico	0.32	0.18	0.17	0.21	0.07	0.02	0.03	1
Dallas_Beaver	0.36	0.23	0.16	0.16	0.04	0	0.05	1
Matt_Rudick	0.55	0.07	0.11	0.22	0.04	0.01	0	1
Matt_Mclain	0.49	0.26	0.07	0.1	0.04	0.02	0.02	1
Ben_Sems	0.47	0.15	0.13	0.18	0.05	0	0.02	1
Tora_Otsuka	0.53	0.12	0.12	0.19	0.03	0	0.01	1

This table of distributions will be what the model draws from, according to what the probability distribution for the player is. Right now, there are no inputs regarding the opposing pitcher, lefty/righty splits, etc… These are items that can be modeled in with future versions of the model, now that the syntax is in place.

Output

As of right now, the model subsets all the half innings that are played with the number of plate appearances stated in inputs. The output takes the form of lists of games. This is the output for rolling through the lineup 50 times:

```
[[0, 0, 0, 1, 0, 1, 0, 0, 0], [4, 0, 0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 0, 5, 0, 1, 0],
[0, 0, 2, 0, 0, 0, 1, 1, 3], [0, 5, 3, 1, 1, 1, 1, 0, 0], [2, 0, 0, 0, 3, 4, 0, 0, 3],
[0, 0, 2, 0, 0, 1, 0, 0, 0], [2, 0, 1, 0, 3, 3, 2, 0, 0], [0, 2, 0, 0, 0, 1, 4, 0, 0],
[0, 3, 1, 0, 0, 0, 0, 0, 1], [0, 0, 1, 0, 0]]
[2, 5, 6, 7, 12, 12, 3, 11, 7, 5, 1]
Min: 1.000
Q1: 4.000
Average: 6.455
Median: 6.000
Q3: 9.000
Max: 12.000
```

The red box represents the half innings segmented into 9 innings plus 5 innings leftover. The blue box represents the sum of runs in each game, and the green box represents the 5-number statistical summary. Given this is the first version of the model, it is also very likely the model output will become more intricate and robust with more advanced statistical inference.

Applications

This tool has many applications because most good simulation frameworks can be tinkered and adjusted. On the face of the current model, one potential application is evaluating player worth by evaluating substitutes in their place. For instance, if Adrian Del Castillo was replaced by Chad Stevens in the lineup in Game 1 vs Orleans, the model evaluates (in 10,000 iterations) the lineup WITH Adrian Del Castillo to score on average 6.16 runs in 2,100 simulated games. If Chad Stevens hit in the same spot in the order for that lineup, the lineup would score on average 5.75 runs in approximately 2,100 simulated games. The difference of .46 runs may sound minimal, but an average of .46 runs over approximately 2,100 observations is substantial.

In addition, it is very natural for Baseball people to imagine how different lineup combinations work. The simulation tool can do this. If for the lineup in Game 1 vs Orleans, if Dallas Beaver and Adrian Del Castillo were flipped 3-4 in the lineup, the model evaluates the new lineup to score on average 6.13 runs in approximately 2,100 games. The difference is very small.

Going forward, additional code will need to be developed to form the optimal lineup. This would mean designing the code to loop through every potential combination lineup (via recursion most likely) and selecting the optimal lineup per the model.

In the Model

For each player, the model essentially reaches into a jar of 100 marbles (plate appearances) and draws a marble that will either be a Batted Ball Out, Strikeout, Walk, Single, Double, Triple, or Homerun. For Chad Stevens, his marble jar contains 53 Batted Ball Outs, 14 Strikeouts, 8 Walks, 17 Singles, 6 Doubles, 0 Triples, and 2 Homeruns. In the Python script, it looks like this for each batter:

```
'''SIMULATE/DRAW SIMS# PLATE APPEARANCES'''
batter_a_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_a_prob_dist)
batter_b_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_b_prob_dist)
batter_c_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_c_prob_dist)
batter_d_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_d_prob_dist)
batter_e_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_e_prob_dist)
batter_f_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_f_prob_dist)
batter_g_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_g_prob_dist)
batter_h_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_h_prob_dist)
batter_i_outcome = np.random.choice(scorebook_outcomes, size=sims, p= batter_i_prob_dist)
```

For instance, if *sims* were set to 100, the model will make 100 draws from each batter's probability distribution. Setting the model to 3, the output temporarily is set to a data frame with each batter getting 3 draws. The lineup represented is the lineup used in game 1 vs Orleans on June 23, 2019:

Braiden_Ward	Darren_Baker	Dallas_Beaver	Adrian_Del_Castillo	Kameron_Guangorena	Jacob_Teter	Mike_Antico	Matt_Mclain	Matt_Rudick
triple	battedballout	single	homerun	battedballout	battedballout	walk	single	single
strikeout	battedballout	strikeout	battedballout	walk	battedballout	battedballout	strikeout	double
walk	battedballout	battedballout	homerun	double	battedballout	battedballout	battedballout	battedballout

At this point, the model is ready to take plate appearance outcomes and turn them into a game. Empty lists for baserunners, half inning outs, half inning runs, total outs, and total runs are created. Once again, a Baseball game moves in between the 24 Base Out states, and the model executes this. The model iterates over every plate appearance generated and will move the baserunners, runs scored, or outs accordingly. The model resets to no baserunners and 0 outs after 3 outs are made and will pick up with the next plate appearance for the next batter in the lineup.

Assumptions

This model acts almost like a game of Strat-O-Matic. The baserunning assumptions are probably a bit bold and will need to be adjusted. For instance, the model has code built in that all runners on 2nd Base score on singles. This is not reality, but just an assumption. Another assumption is that all Batted Ball Outs do not produce any runs scored. This model clearly has many areas for improvement in its assumptions. Luckily, with the vast amount of Synergy data available, Synergy could be utilized to form better assumptions and therefore more accurate model.

In Conclusion

This Simulation model is a first step in an alternative framework for evaluating. In all areas of analysis, as Nate Silver writes about in his book, it is a healthy and necessary step to explore for deeper meaning of truth. This often entails proposing new and even radically perceived ideas, but that is part of the process. This model is nowhere near close to perfect, but no model is perfect. Proponent of Bayesian inference and Statistician George Box said, “All models are wrong, but some are useful”. Users of models should never make decisions only because of what a model says. But if a model could be an input to a decision made, it is very possible for a better decision to be made. This Simulation model is a start in the process.