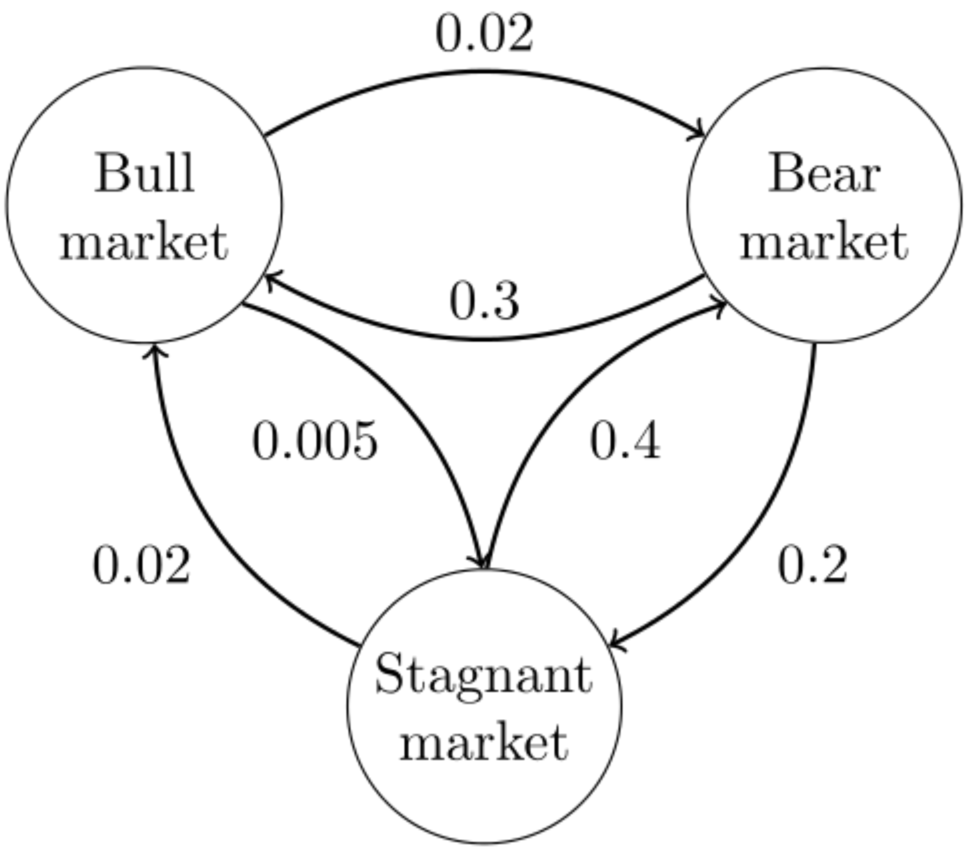
The Elegance of Markov Chains in Baseball

Lucas Calestini · Follow
Published in Sports Analytics
6 min read · Sep 10, 2018

Billy Beane (Brad Pitt) - Moneyball (2012). Source: Sony Pictures Home Entertainment

Sabermetrics has evolved considerably in the past decades. It continues to benefit from the AI obsession and the dissemination of cheap computing power. Yet, a few long-standing statistical models and algorithms continue to be the baseline approach for sports analytics (maybe the Lindy effect in play). It seems the more we [re]visit those models, the more we learn about the fundamentals of the game — and in the age of oversold skills, we should be more keen on fundamentals.

One of such models that we are focusing on at Torneo these days (together with our evaluation of the Elo Ratings) is Markov chains — a stochastic model for a sequence of events where the probability for each outcome only depends on the previous state of the system.



Standard example of Markov Chains. Source: Wikipedia

The idea is simple: the probability of the inning moving to a specific state is only a function of the current state, independent of how teams arrived at that place. If there are 2 outs and 2 players on scoring positions, all the preceding events and actions are of no importance (therefore stochastic systems are *memoryless*).

The reason I specifically like Markov chains lies in the simplicity of its definition, the insights it provides, and the vast reach of applications for which it can be used. For instance, we can analyze team performance, evaluate players contributions to a result, assess player trade values and optimize the batting line (it also helped us debug some of our retrosheet data-parsing code).

There are two key terms in Markov chains: **states** and **transitions**.

Game States

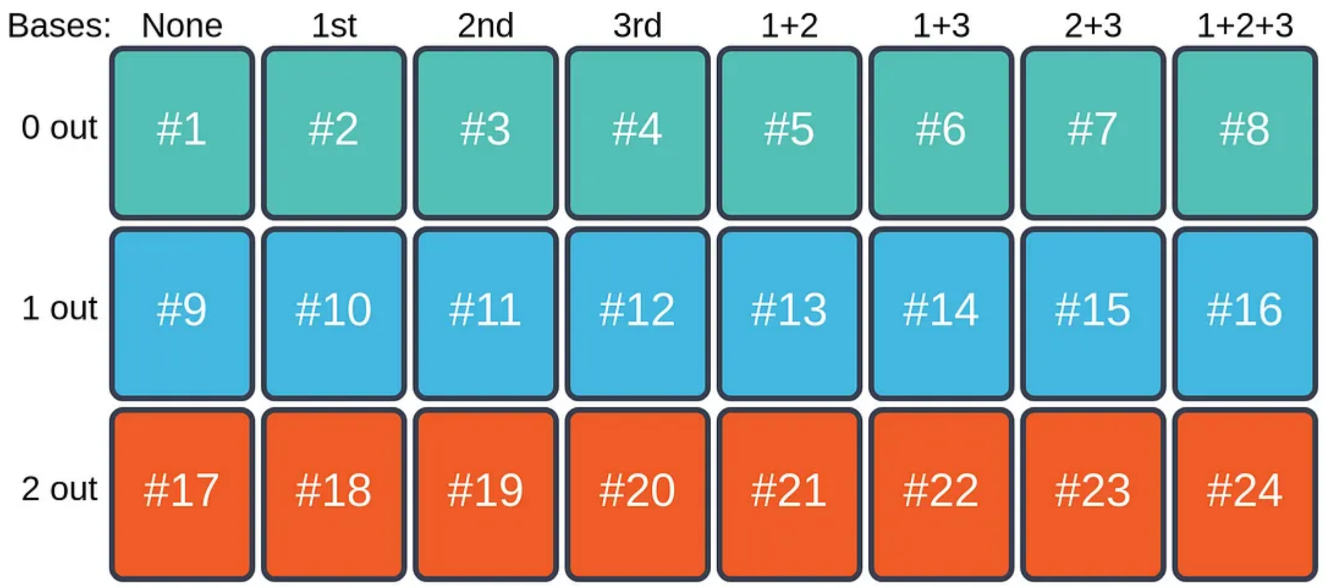


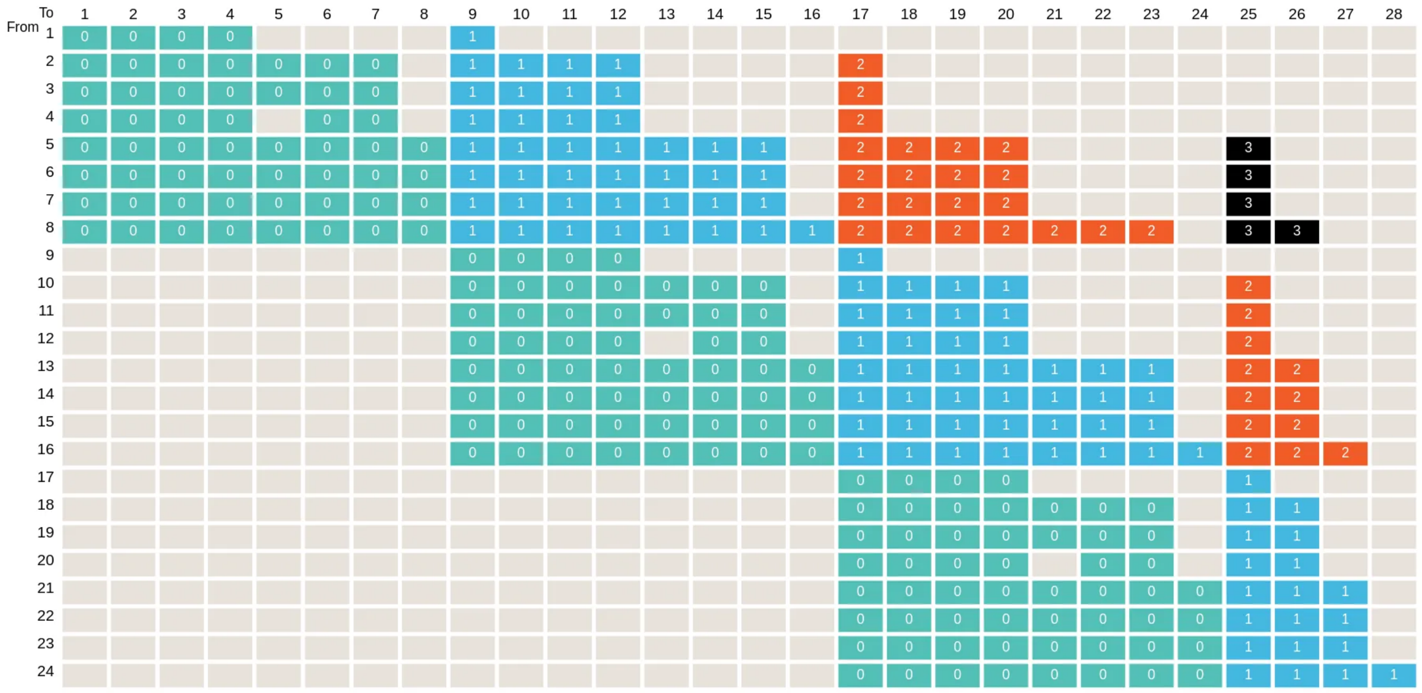
Figure 1: Non-absorbing states, numbered

Game states are all possible combinations of runners on base (0, 1, 2, 3, 12, 13, 23, and 123) and the number of outs (0, 1 and 2) at each play, totaling 24 states at any point in time of the inning (see Figure 1). The last state, the 25th, represents the end of the inning (or the absorbing state, as there is no possibility for a further transition). To account for all scenarios, we use 4 absorbing states, numbered from 25th to 28th, representing third-out situations with 0,1,2 and 3 runs, respectively.

These are not absolute. We could keep adding dimensions to create more relevant states using a third or fourth variable (with bases and outs currently being the only two variables). The more granular the states, the better our understanding, but the fewer data points we will have for those combinations. For instance, we could add plays or runs into the states, but it would easily expand the number of states and sparse the available transitions beyond desirable (Adding runs to each state, for instance, could exponentiate the number by states to 10–20. Let us not touch the curse of dimensionality).

Transitions

A transition is any movement of players for each plate appearance, or any change from one state to another. Singles, fly-outs, home runs or double plays, they all represent a specific transition, from a previous state (1–24) to a current state (1–28). In scoring perspective, a transition will always result in 0,1,2 or 3 outs and 0,1,2,3 or 4 runs. The figure below shows a transition matrix in terms of **outs**.



From To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	0	0	0					1																			
2	0	0	0	0	0	0	0		1	1	1	1					2											
3	0	0	0	0	0	0	0		1	1	1	1					2											
4	0	0	0	0		0	0		1	1	1	1					2											
5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2	2	2					3			
6	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2	2	2					3			
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2	2	2					3			
8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2		3	3		
9									0	0	0	0					1											
10									0	0	0	0	0	0	0		1	1	1	1					2			
11									0	0	0	0	0	0	0		1	1	1	1					2			
12									0	0	0	0		0	0		1	1	1	1					2			
13									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2		
14									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2		
15									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		2	2		
16									0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	
17																	0	0	0	0					1			
18																	0	0	0	0	0	0	0		1	1		
19																	0	0	0	0	0	0	0		1	1		
20																	0	0	0	0		0	0		1	1		
21																	0	0	0	0	0	0	0	0	1	1	1	
22																	0	0	0	0	0	0	0	0	1	1	1	
23																	0	0	0	0	0	0	0	0	1	1	1	
24																	0	0	0	0	0	0	0	0	1	1	1	1

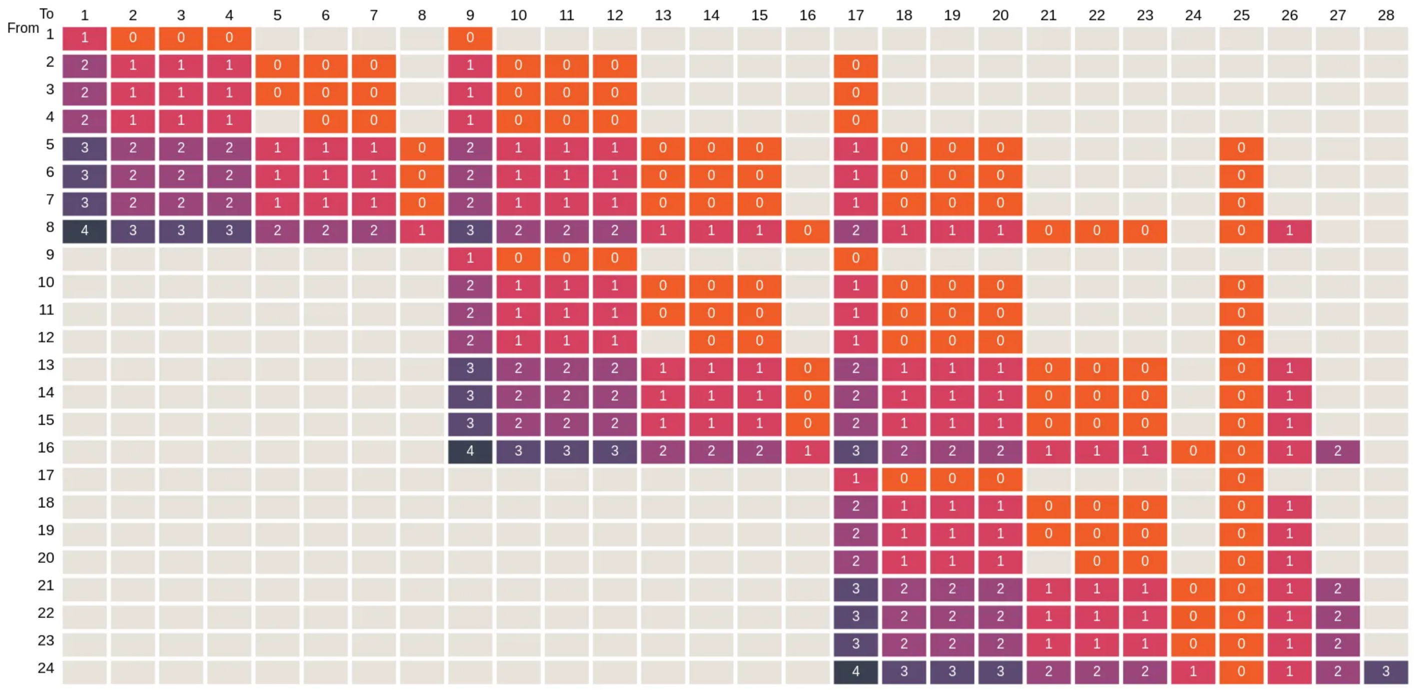
Figure 2: Outs for every transition in baseball (rows = **From**, columns = **To**)

As we can see, some transitions are impossible due to the rules of the game. An inning cannot go from 2-out to 1-out, or from 0 on base to 2 players on base. Those transitions are represented in grey in the picture, whereas green are for transitions with no outs, blue for 1, orange for 2 and black for 3. Figure 2 also helps us see a pattern in the transition matrix, and such patterns are what makes it possible to write the matrix T as a **block matrix,** based on the number of outs (the last row of the matrix represents states 25–28, and it is represented by 0s and 1 matrices, as those are the absorbing states, and it is not shown on Figure 2).

$$\mathrm{P} = \begin{pmatrix} A_0 & B_0 & C_0 & D_0 \\ \mathbf{0} & A_1 & B_1 & E_1 \\ \mathbf{0} & \mathbf{0} & A_2 & F_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}$$

Transition matrix represented as a block matrix

Similarly, we can see transitions based on the run outcome. A transition from state 8 (bases loaded) to state 1 (aka grand slam) represents 4 runs and 0 out. Therefore by combining states to transitions, we can fit all possible plays from state 1 (0 outs, 0 on base) to states 25–28 (3 outs). It makes it easier, and more relevant, that time is not a factor on baseball.



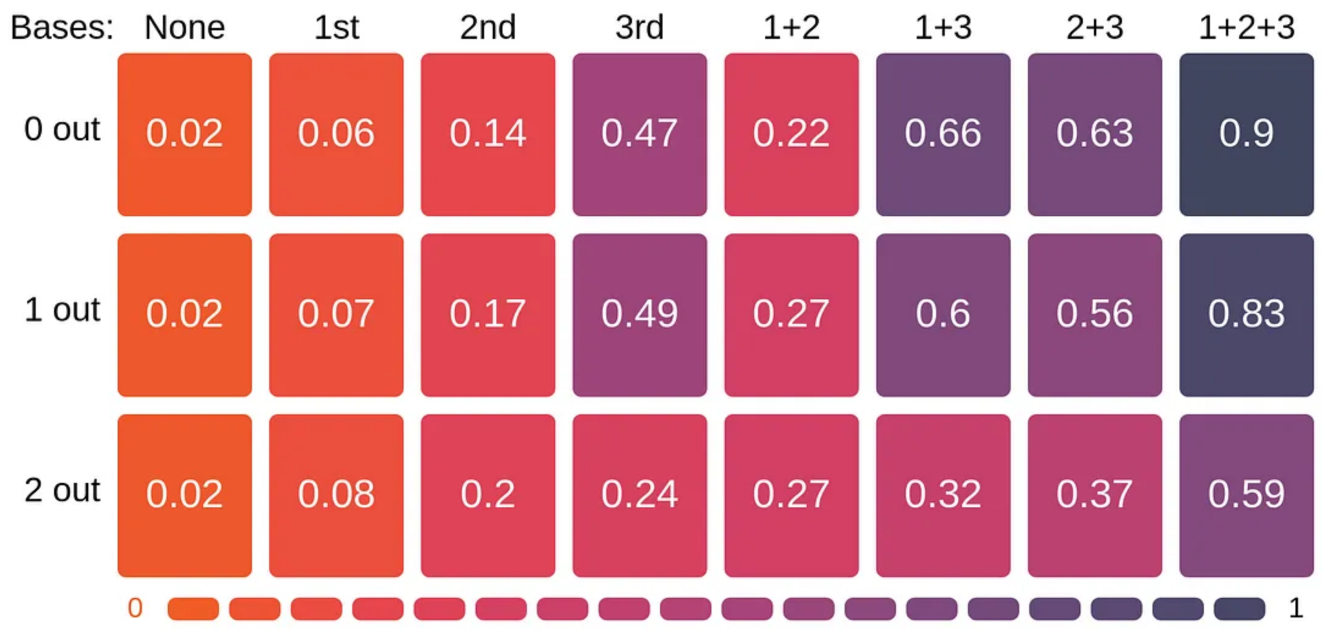
To From	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	0	0	0					0																			
2	2	1	1	1	0	0	0		1	0	0	0					0											
3	2	1	1	1	0	0	0		1	0	0	0					0											
4	2	1	1	1		0	0		1	0	0	0					0											
5	3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		1	0	0	0					0			
6	3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		1	0	0	0					0			
7	3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		1	0	0	0					0			
8	4	3	3	3	2	2	2	1	3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		0	1		
9									1	0	0	0					0											
10									2	1	1	1	0	0	0		1	0	0	0					0			
11									2	1	1	1	0	0	0		1	0	0	0					0			
12									2	1	1	1		0	0		1	0	0	0					0			
13									3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		0	1		
14									3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		0	1		
15									3	2	2	2	1	1	1	0	2	1	1	1	0	0	0		0	1		
16									4	3	3	3	2	2	2	1	3	2	2	2	1	1	1	0	0	1	2	
17																	1	0	0	0					0			
18																	2	1	1	1	0	0	0		0	1		
19																	2	1	1	1	0	0	0		0	1		
20																	2	1	1	1		0	0		0	1		
21																	3	2	2	2	1	1	1	0	0	1	2	
22																	3	2	2	2	1	1	1	0	0	1	2	
23																	3	2	2	2	1	1	1	0	0	1	2	
24																	4	3	3	3	2	2	2	1	0	1	2	3

Figure 3: Runs scored for every transition in baseball (rows = **From**, columns = **To**)

Expected Runs

One of the by-products of Markov chains is a matrix of expected runs (or outs) for each state in the game. The number of expected runs is the sum of the transition probabilities times the run vector (the run each transition generates, see Figure 3). Because every play can only start from one single state, the row probabilities need to add to 1 (T is a right stochastic matrix).

Hence we have a matrix of 24 values, for each possible state the game is in once the batter comes to plate . It makes intuitive sense to think of it from a batter perspective, as if his chances of scoring an RBI would be influenced by how the runners and outs happen to be once he is up to bat.



Bases:	None	1st	2nd	3rd	1+2	1+3	2+3	1+2+3
0 out	0.02	0.06	0.14	0.47	0.22	0.66	0.63	0.9
1 out	0.02	0.07	0.17	0.49	0.27	0.6	0.56	0.83
2 out	0.02	0.08	0.2	0.24	0.27	0.32	0.37	0.59

Figure 4: Expected run at each state since 1921

We can control for teams, seasons, players and innings, and here is where the power of Markov chains unfold. For instance, Figure 4 shows us the aggregated expected run for all MLB games from 1921 to 2017. On average, a team with the bases loaded and no out will likely score one run.

Controlling by players, we can create a direct comparison of batters expected runs for all scenarios. Figure 5 exemplifies such comparison using Josh Donaldson and Nelson Cruz*. Naively analyzing the chart, we see that Josh tends to perform better with less outs in the game, and Cruz hits his prime with 2 outs.

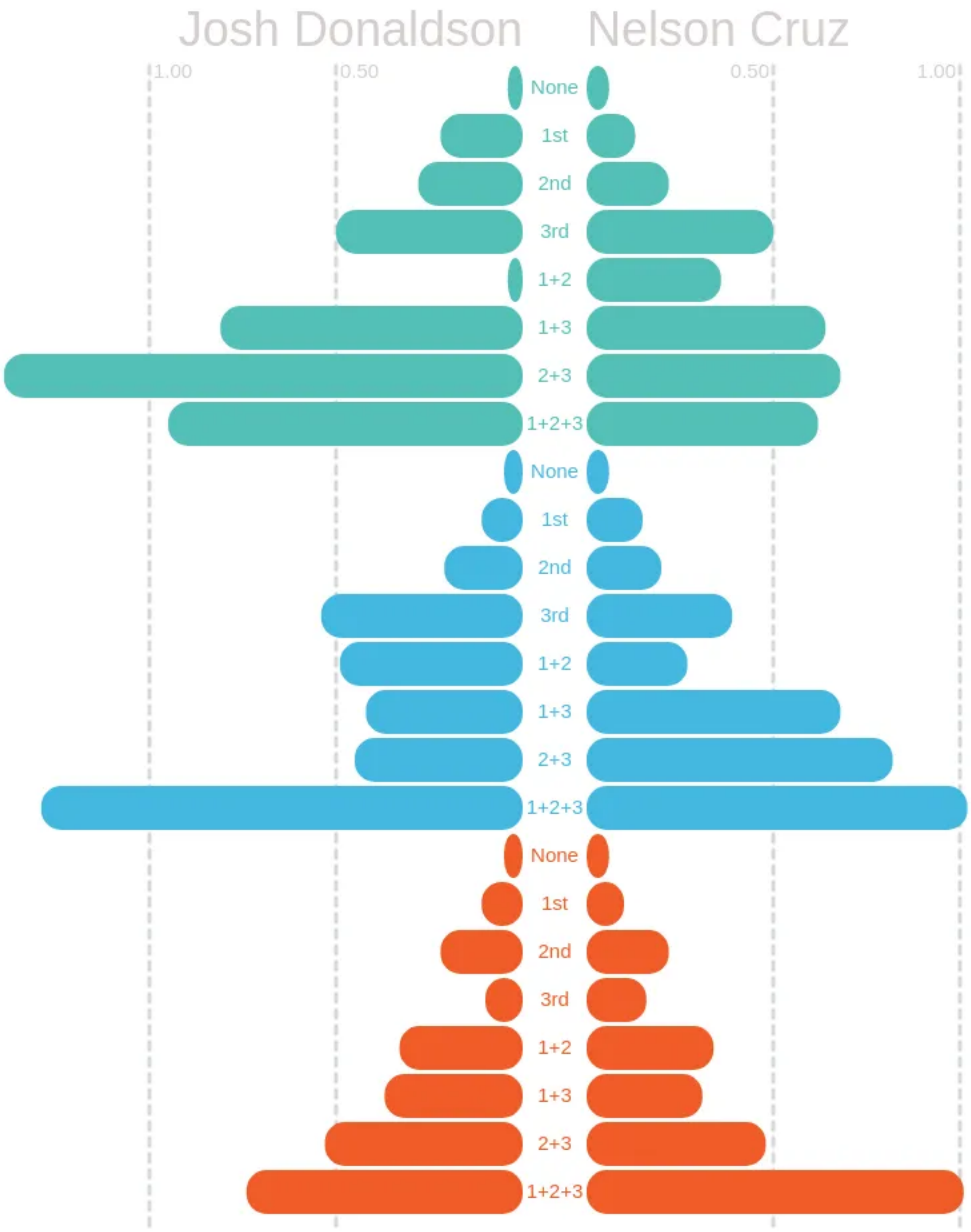


Figure 5: Expected runs. Colors represent the outs (0, 1 and 2, respectively)

Other Applications

Beyond expected runs, Markov chains is used to simulate outcomes for a line of players (a batting line), or to calculate the exact impact of different players in the current team. If we multiply the transition matrix by a initial state and by the batting sequence, for 9 innings, we can then estimate the expected runs scored for a head-2-head. There are plenty of ways in which Markov Chain models can be further expanded, and an understanding of its structure is only the bare backbone for applying it to real-life situations. But let us get a good grip of fundamentals before heading to the deep waters of prediction.

. . .

**There are limitation to the number of available observations. Therefore we might come across numbers that are not necessarily intuitive because the player did not appear many times in that transition during a game.*

Notes

- Kudos to Retrosheet for the data.
- Data used for this analysis ranges from 1921 to 2017
- We removed plays that do not move the batter, such as SB, CS, PO, Balk
- Visualizations done in D3.js and calculations in Python

Further Reading

- Markov Chain Models: Theoretical Background
- A Markov Chain Approach to Baseball
- Stochastic Matrix
- The Markov Chain Model of Baseball

Sports