Markov Chains in Baseball

Jake Singleton

3/15/2022

Introduction

Baseball is a relatively static game. Players are not always in motion; a clock isn't always ticking. Instead, the action on the field takes place one step at a time–the leadoff hitter hits a single, yielding a configuration of runner on first with no outs. Then the second batter strikes out, changing the configuration to runner on first with one out. The third hitter comes up and hits a two-run homerun, leaving us again with bases empty, but still with one out. The next two batters each make outs, yielding 3 outs total, and the *half inning* ends. Then, the batting and fielding teams switch, and they do so for 18 half innings (i.e. 9 innings total).

There are three bases which a baserunner may or may not occupy at any given time, and the batting team gets to keep batting until the fielding team makes three outs. Since each base may be occupied or unoccupied with 0, 1, or 2 outs, there are $2 * 2 * 2 * 3 = 24$ *base-out states*, with the three out state being an *absorbing state*. In this project, we model transitions between these states using Markov chains.

Research Focus and Methods

The primary focus of this project is to use Markov chains to understand run scoring in Major League Baseball. In particular, we seek to answer how many runs an average team can expect to score for each of the 24 base-out states. Given these figures, we may better understand how to make smarter in-game decisions regarding plays like the bunt and steal. We also seek to answer if the home team's Markov chain differs at all from the away team's, which may help us better understand home field advantage in baseball.

The Markov chains we define here are empirically computed from data, and they allow us to simulate half innings of baseball repeatedly in order to estimate run scoring distributions. Therefore Monte Carlo estimation also plays a key role in this project.

Data

We use a data set of all 2021 Major League Baseball (MLB) regular season games. The data is at the play-by-play level, and the metadata is here. By play-by-play, we mean that each observation is either an "Event made by the batter at the plate" or "Base-running event not involving the batter", as they're called in the metadata. Most commonly, the batter will make the event such as a batted ball, strikeout, or walk. Much more rarely, a baserunner may steal a base or the pitcher may balk. Either type of event results in a change or stay in base-out state. Prior to cleaning, the data consists of 187,210 observations on 97 variables.

Example Data

As an example, we consider the first observation from the data set below, selecting the important columns:

```
##        GAME_ID AWAY_TEAM_ID OUTS_CT EVENT_OUTS_CT BASE1_RUN_ID BASE2_RUN_ID
## 1 ANA202104010          CHA       0             1         <NA>         <NA>
##   BASE3_RUN_ID BAT_DEST_ID RUN1_DEST_ID RUN2_DEST_ID RUN3_DEST_ID
## 1         <NA>           0            0            0            0
```

The GAME_ID and AWAY_TEAM_ID fields tell us this was a game that took place between the Los Angeles Angels of Anaheim (ANA), the home team, and the Chicago White Sox (CHA), the away team, on April 1, 2021. The rest of the selected fields are crucial for defining the base-out state:

- OUTS_CT is the number of outs before the play occurred
- EVENT_OUTS_CT is the number of outs that occurred on the play
- BASE1_RUN_ID - BASE3_RUN_ID tells us which bases were occupied before the play
- BAT_DEST_ID and RUN1_DEST_ID - RUN3_DEST_ID tell us which bases were occupied at the end of the play

For this example, the initial base-out state is 0 outs, bases empty, which we will call 0000, and the final state is 1 out, bases empty, which we denote 1000. So the first X in the string XXXX is the number of outs, and the next three are binary indicators for whether a baserunner occupies that respective base. For instance, the 2 outs, runners on first and second state is denoted 2110.

Markov Chain Computation

Upon computation of the Markov chain the data consists of $n = 178,738$ observations.

Let $P \in \mathbb{R}^{25\times 25}$ be the matrix of transition probabilities between the 24 base-out states plus the absorbing three out state. Let STATE0 $= i$ be the state before the play and STATE1 = j be the state after the play. Then we estimate the probability of transitioning from i to j by the proportion of times this transition took place in the data. That is, $P_{ij} = \frac{\sum_{k=1}^{n} I(\text{STATE0}_k=i)*I(\text{STATE1}_k=j)}{\sum_{k=1}^{n} I(\text{STATE0}_k=i)}$ where $I()$ is an indicator function.

P (its first 10 rows and 10 columns) was found to be

	0000	1000	2000	1100	2100	0100	2010	2110	1010	1110
0000	0.04	0.68	0.00	0.00	0.00	0.23	0.00	0.00	0.00	0.00
1000	0.00	0.03	0.69	0.23	0.00	0.00	0.00	0.00	0.04	0.00
2000	0.00	0.00	0.03	0.00	0.23	0.00	0.04	0.00	0.00	0.00
1100	0.00	0.03	0.00	0.00	0.48	0.00	0.07	0.00	0.01	0.20
2100	0.00	0.00	0.03	0.00	0.00	0.00	0.02	0.19	0.00	0.00
0100	0.04	0.00	0.11	0.48	0.00	0.00	0.00	0.00	0.08	0.00
2010	0.00	0.00	0.03	0.00	0.08	0.00	0.05	0.15	0.00	0.00
2110	0.00	0.00	0.03	0.00	0.00	0.00	0.02	0.04	0.00	0.00
1010	0.00	0.03	0.00	0.06	0.01	0.00	0.47	0.00	0.05	0.13
1110	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.40	0.01	0.04

Note P is valid as each of its rows sum to 1, which is shown in the appendix.

Markov Chain Connection to Run Scoring

Having obtained our Markov chain P, we are now able to envision a half inning of baseball. Since all innings begin in the state 0000, we see that 68% of the time, we transition to the 1000 state. 4% of the time, we transition to the 0000 state, meaning a solo home run was hit. A run scored! In this section, we first define the mapping from state transitions to runs.

Let NRunners_k = the number of runners in STATEk, O_k = the number of outs in STATEk, $k = 0, 1$ (as before STATE0 = the initial state before the play and STATE1 = the final state after the play). Then the number of runs for any transition* is RUNS $= (\text{NRunners}_0 + O_0 + 1) - (\text{NRunners}_1 + O_1)$, i.e. the number of runs scored on a play is the sum of number of runners and outs before the play PLUS 1 minus the sum of runners and outs after the play. Although adding 1 may seem unintuitive, think of the solo home run case: going from 0000 to 0000 means a home run must have occurred, resulting in one run scored.

*This definition only applies after filtering out non-batted ball events. More in the "Data" section of the appendix.

With the definition of RUNS in hand, we are able to create the following run matrix $R \in \mathbb{R}^{24\times 25}$, where each entry R_{ij} = the number of runs scored in a transition from i to j. Note the matrix is of this dimension because we cannot start in the 3 out, absorbing state, but we can end in it. The first 10 rows and columns of R are the following:

	0000	1000	2000	1100	2100	0100	2010	2110	1010	1110
0000	1	0	-1	-1	-2	0	-2	-3	-1	-2
1000	2	1	0	0	-1	1	-1	-2	0	-1
2000	3	2	1	1	0	2	0	-1	1	0
1100	3	2	1	1	0	2	0	-1	1	0
2100	4	3	2	2	1	3	1	0	2	1
0100	2	1	0	0	-1	1	-1	-2	0	-1
2010	4	3	2	2	1	3	1	0	2	1
2110	5	4	3	3	2	4	2	1	3	2
1010	3	2	1	1	0	2	0	-1	1	0
1110	4	3	2	2	1	3	1	0	2	1

Although the negative entries look wrong and we also see an entry for 5 runs, they are not problematic since these transitions correspond to 0 probability transitions in P.

With R and P ready, we can now simulate a half inning of 2021 baseball, keeping track of the number of runs scored. While the code can be found in the appendix, the point of the simulation is to learn about the run scoring environment for each base-out state. We pick one state as the starting state and simulate repeatedly. Repeating this for each state, we are left with a run distribution for each one.

Run Distributions

Below we have run distributions for each of 24 base-out states as the starting state. $N = 10,000$ half innings were simulated for each state.

One comparison worth making is the one between the `0000` traditional starting state and the `0010` starting state. In 2020 and 2021, MLB decided to start extra innings in the `0010` state in an effort to promote scoring, helping the game finish faster. Extra innings occur when the game is tied after 9 innings, and traditionally, extra innings have been played no differently than normal. By observation, we can see that starting from the `0010` state yields generally far more runs than starting from the standard `0000` state, and we will quantify this difference more deeply in the next section.

Given these distributions, now we can make a *run expectancy* (RE) matrix that shows how many runs an average team can expect to score from each of the 24 base-out states.

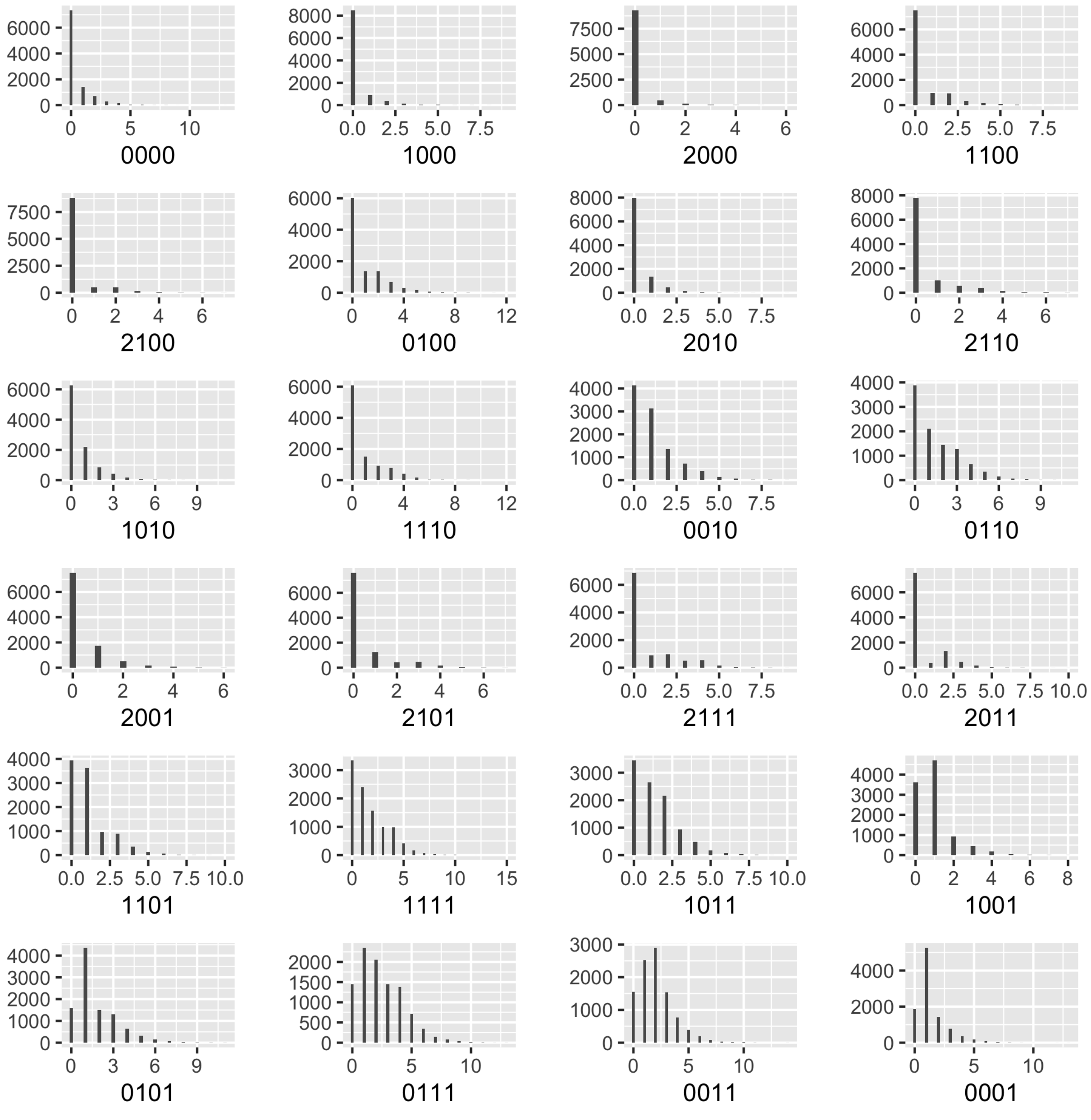

Figure 1: Run distributions for each of 24 base-out states; N = 10,000

Run Expectancy

We define run expectancy as merely the average number of runs scored through the end of the inning, given the current base-out state. That is, RE is the conditional expectation $RE = E[\text{number of runs scored in half inning} \mid \text{base-out state}]$. These conditional expectations are quite easy to compute using our simulated run distributions above. They yield the RE matrix

	0	1	2
000	0.49	0.25	0.10
100	0.89	0.50	0.22
010	1.10	0.64	0.30
001	1.36	0.93	0.36
110	1.51	0.89	0.43
101	1.72	1.09	0.46
011	2.01	1.34	0.56
111	2.41	1.65	0.78

We interpret this matrix in the following way: if the state is no outs and bases empty (the (1, 1) entry), the batting team can expect to score 0.49 runs to the end of the inning. If runners are on first and third with 1 out (the (6, 2) entry), the batting team can expect to score 1.09 runs to the end of the inning.

Therefore, from the traditional `0000` starting state, the average team can expect to score 0.49 runs to the end of the inning. Under the new extra innings rule mentioned just above, batting teams start in the `0010` state and can expect to score 1.10 runs to the end of the inning. This RE is 2.24 times as many runs as normal, so MLB's decision changes the game drastically.

Decision-Making in Baseball

One important application of this matrix to professional baseball is decision-making–for example, batting teams have traditionally employed the strategy of bunting with no outs and a runner on first. The bunt is a strategically weakly-hit batted ball designed to move a runner one base over at the cost of the batter (the bunter) being thrown out at first. Essentially, the batting team sacrifices an out for an advanced baserunner. Using this matrix, we can see that with a runner on first and no outs, the batting team can expect to score 0.89 runs in the inning. However, after a successful bunt, resulting in a runner on second with 1 out, the run expectancy actually decreases to 0.64. This realization has made the bunt much less popular in modern baseball.

Another often-scrutinized play is the stolen base. For example, consider the `0100` state again with run expectancy 0.89. If the runner successfully steals second base, RE increases to 1.10, a gain of 0.21 RE points. If he fails, we are left in the `1000` state with RE 0.25, a loss of 0.64 points. Then, if $E(s)$ is the expected gain in RE from the steal and p is the probability the runner successfully steals second, then $E(s) = 0.25p - 0.64(1-p)$, and we should only be wiling to steal if $E(s) > 0 \iff p > 0.72$. That is, we should only try to steal if we are at least 72% sure we will be successful.

Home and Away Markov Chains

One important point to remember is that, thus far, these results have held for a perfectly average baseball team in 2021. This section seeks to answer if run scoring is meaningfully different between home and away teams. Upon obtaining two Markov chains, one for home teams and one for away, we can simulate again $N = 10,000$ half innings for each from the traditional starting state `0000` and plot the run distributions. Results are shown below. As we can see, the distributions look quite similar at first glance. However, we do observe that we can expect home teams to score about 0.5246 runs per half inning (empirical standard deviation 1.05 runs) and away teams to score 0.4652 runs per half inning (empirical SD 0.97 runs). This means, on average, we can expect the home team to score 0.06 (0.5246-0.4652) more runs more per half inning than the away team (pooled sample SD 1.01 runs). Over 9 innings, this difference is 0.54 runs (SD 9.09 runs). Hence, in 2021, the average home team had about a half run advantage over the average away team.

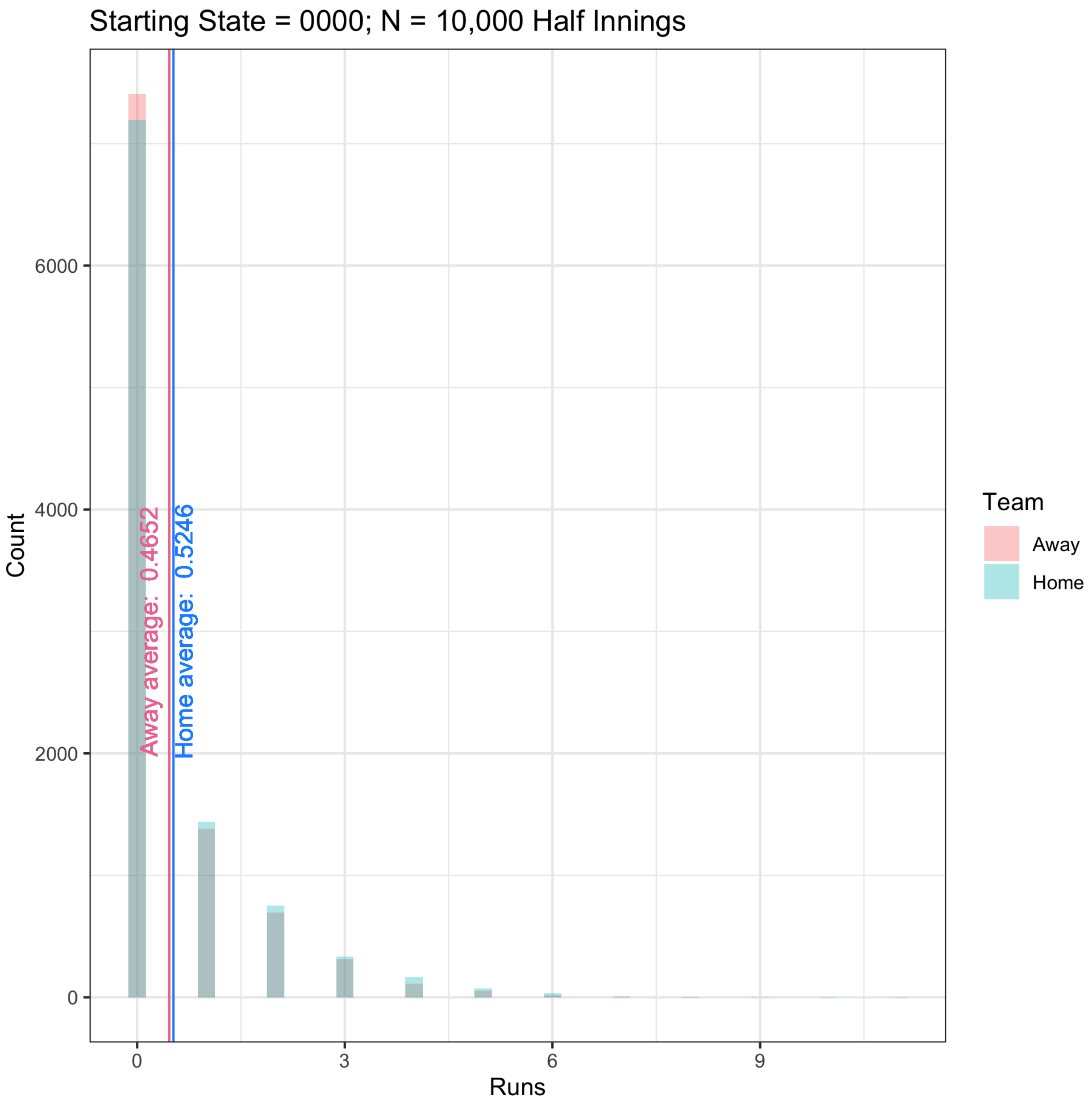

Figure 2: Run Distribution for Home and Away Teams; Starting State 0000

Conclusion

In this project, we've computed Markov chains and a run-scoring matrix between all base-out states to understand how each state has its own run scoring distribution. We compared two particularly important starting states for extra innings, the `0000` and `0010` states, and found the latter drastically promotes more scoring. We then used each of the 24 distributions to compute a run expectancy matrix, which enable us to make smarter decisions amidst a baseball game. Finally, we computed separate Markov chains for home and away teams and found that, on average, home teams have a half run advantage over away teams.

In further work, I would like to explore extensions of this project that may take into account specific teams and specific players. For example, the best team in baseball will almost certainly have a different Markov chain than the worst team in baseball. Hence, the in-game decisions certain teams may want to make may differ than the averages I've presented here.

Non-Class References

1. Absorbing Markov Chains Wikipedia
2. Marchi, Max; Albert, Jim; Marchi, Max; Albert, Jim; Baumer, Benjamin S.. Analyzing Baseball Data with R, Second Edition (Chapman & Hall/CRC The R Series). CRC Press. Kindle Edition.
3. Winston, Wayne. Mathletics. Princeton University Press.