EVALUATING APBA CARDS

DONALD M. DAVIS

1. Introduction

I played APBA avidly as a boy from 1956 to 1966. By 1964, I had learned enough sophisticated mathematics (Markov chains) to try to perform an analysis of APBA cards, but was limited primarily by limited access to computers. Since 1971, I have been a professor of mathematics. The August, 2009, *New York Times* article about APBA caused me to try to redo this analysis more carefully, which I have now accomplished. This article is written for APBA players. Another article will be written for mathematicians unfamiliar with APBA. The first four sections of this article can be read without any understanding of higher mathematics. The optional Section 5 will explain many of the mathematical details. The long final section is an annotated version of the computer program.

Analyzing, or evaluating, cards means telling exactly how valuable each number, from 1 to 41, is on a player's card. By adding the values of the 36 numbers on a player's card, you can tell, on average, how much will each roll for the player increase (or decrease) the number of runs that you expect to score in the inning. Here you are averaging, not just over the 36 numbers on the player's card, but also over the 24 (base,out) situations that occur, and over all the things that might happen during the rest of the inning. This takes into account the different sorts of pitchers and fielders that you might be facing.

For a specific team, in which you know exactly what players will come to bat next, you could do a more accurate analysis. For example, getting on base could be more valuable if you know that the next batter is a slugger. But any such analysis would require running a special computer program for each such situation.

I know that there is now a Master Game, with some more sophisticated aspects. It is unlikely that my results would be very applicable to the Master Game. The analysis which I performed, using rules with which I am very familiar, was hard enough (6 weeks of spending most evenings working on it) that I am not interested in adapting it to unfamiliar rules. My analysis is based on the 1964 "boards." Changes

in the boards that were made after that could have some small effect on my results. What could have a much larger effect is a change in the cards being used. I used a sample of 350 cards from 1956 to 1964. This affects greatly the average number of runs scored in an inning, which will be one of the important numbers arising in, and affecting, the analysis. It would also have some effect on the relative value of the numbers on the cards.

My analysis also tells the value of each pitching grade and adornment, and of each fielding grade. It tells how you should align your outfielders to obtain the best results on average. It tells, and utilizes, strategies about playing it safe on singles, doubles, and fly balls. It is all based on averaging. For example, it tells whether you should play it safe with an average batter coming up, but with a specific batter coming up, your strategy might be different. We also analyze, and incorporate, hit and running. We find that if the runner on first has an 11, it is, on average, advantageous to hit and run, but if the runner only has a 10, there is no real advantage to hit and running, on average, and so we do not incorporate this (hit and running unless the runner has an 11) into our analysis. We do not analyze sacrifices at all; they will not increase your expected number of runs scored, except in exceptional situations. Also, we do not incorporate playing your infield "close." It seems that, on average, this will always lead to a less advantageous outcome, although I realize that in certain situations it is imperative to keep a run from scoring.

The primary use that might be made of my results is to try to equalize "fantasy" teams. We used to call them "all star" teams. Using my rudimentary 1964 "system," we would have a league in which we would say, for example, that your team could total no more than 2000 points, under the point system that we used then. A team's points includes that of their batters and also fielding ratings and pitching ratings. We will discuss in Section 4 how to evaluate teams' scores.

For a reader who just wants to see the main results without any regard for how they were obtained:

- A batter's value is the sum of the values from Table 1 of the 36 numbers on his card.
- To obtain a team's total value, add the values of the batters in their starting lineup, and add to this the numbers from Tables 8 and 9 for the speed ratings of their starters[1], their five starting pitchers, their three relievers, and their fielders.

_____

[1]The extra value of an 11 is for hit and running.

- Strategy about playing it safe is given by bullet points in Section 2. This is the strategy with all average players coming up to bat.
- The best way to align your outfield against average hitters is given in Table 7.
- On average, hit-and-running is advantageous with an 11 on first. With a 10 on first, it is borderline.

2. OFFENSE NUMBERS

Most readers will probably be most interested in knowing the values of the numbers on a card. This is the average increase in runs scored in an inning if this number is rolled. (I find it convenient to use the inaccurate term that the number (from 1 to 41) is "rolled." The dice are rolled and then the number appears on the card; it is the result of the roll, but I will say it is "rolled.) For every (base,out) situation, we determine the expected[2] number of runs scored from that situation. This information will be given in Table 2. We also determine the fraction of the time that each (base,out) situation will occur. This information will be given in Table 3. Then the value of a number is the sum, weighted by the numbers in Table 3, of the values $E_2 - E_1 + R$, where E_1 is the expected number of runs scored (i.e., the numbers in Table 2) from the initial situation (before the roll), E_2 is the expected number runs scored from the situation resulting after the number is rolled, and R is the number of runs scored on that roll. The results appear in Table 1. If a player's card has 0's and then an extra column for extra base hits, then a 2 or 6 sometimes behaves differently than an ordinary 2 or 6. We call 2A and 6A these special 2's and 6's that occur after rolling a 0.

The important numbers in Tables 2 and 3 associated to the (base,out) situations which were used, along with the boards, to obtain the numbers in Table 1, were obtained by mathematical analysis involving Markov chains. Most of the mathematical details are discussed in the optional Section 5. A reader should be able to understand the results and methodology without trying to understand Markov chains. In Table 2 is the expected number of runs scored in the remainder of the inning if you are in the specified situation. The most interesting of these is the expected number of runs scored when no one is on and no one out, since that tells the average number of runs scored in an inning. This value, 0.4327, when multiplied by 9, gives the average number of runs scored by a team in a 9-inning game. This value, 3.8943, is quite

---

[2]"Expected" is a mathematical term meaning "average."

TABLE 1. Values of numbers on APBA cards

#	Value	#	Value	#	Value
1	1.4101	14	0.2597	28	−.1860
2	0.9843	15	0.2533	29	−.2281
2A	0.9620	16	0.0875	30	−.2305
3	0.9350	17	0.4045	31	−.2195
4	0.8699	18	0.3353	32	−.2297
5	1.0023	19	0.3264	33	−.2484
6	0.7723	20	0.3175	34	−.2484
6A	0.7021	21	0.3691	35	−.2327
7	0.4708	22	0.1669	36	0.1055
8	0.2257	23	0.0346	37	−.0870
9	0.0904	24	−.3026	38	0.0374
10	0.3535	25	−.3363	39	−.1743
11	0.5476	26	−.2158	40	−.0051
12	−.2679	27	−.2298	41	0.0217
13	−.2317				

consistent with real baseball figures during the early 1960's on which this analysis is based. It would be quite a bit higher if we based our analysis on the cards from the late 1990's.

TABLE 2. Expected number of runs from different situations

		outs		
		0	1	2
	0	0.4327	0.2254	0.0792
	1	0.8071	0.4737	0.1892
Bases	2	0.9833	0.6055	0.2862
	3	1.2053	0.8693	0.3481
	12	1.3294	0.8288	0.4042
	13	1.6274	1.0519	0.4450
	23	1.7479	1.1924	0.5279
	123	2.1726	1.4705	0.7098

Table 3 tells the proportion of the time that a batter will appear in each (base,out) situation. This is probably of less general interest than Table 2, but is essential to evaluation of the numbers on the cards. I wish to emphasize that the numbers in these tables are not based on empirical observations. They are a direct mathematical consequence of only the 350 cards in our sample and the rules of the game (the "boards").

TABLE 3. Fraction of the time a batter is in each situation

		outs		
		0	1	2
	0	.24239	.17284	.13518
	1	.06248	.07462	.07359
	2	.01671	.03318	.04347
Bases	3	.00189	.00728	.01491
	12	.00924	.01664	.01895
	13	.00584	.01128	.01744
	23	.00496	.01013	.01305
	123	.00233	.00515	.00645

Other offensive attributes are speed ratings S and F, and having an 11 so that hit-and-running will be effective. The amount by which having an S, F, or 11 runner on base improves the expected value on a single roll is given in Table 4. For 11, this is just the additional value that being on first base with an 11 gives to the next batter; it has nothing to do with the value of 11 as a batting number. These numbers, for S and F, are obtained by considering all numbers that might be affected by the S or F or by playing it safe, and taking the difference between expected number of runs after the number is rolled, with runs scored on the play taken into account, weighted by probabilities of the numbers being rolled and by the probabilities of the (base,out) situation. For hit-and-running with an 11 on, we take the weighted sum, over all numbers j that might be rolled, of the differences $E_j - E_j'$, where E_j and E_j' are the expected values of the outcome of rolling a j either hit-and-running or not. These numbers are also multiplied by the probabilities of being in a situation in which hit-and-running is possible. Roughly one fourth of the time you are in a hit-and-runnable situation, and if you are, on average, hit-and-running with an 11 on first increases the expected number of runs by about .06.

TABLE 4. Value of S, F, and 11 (for H&R)

	Value
S	$-.01898$
F	0.00387
11	0.01389

These small Value amounts should be added appropriately onto the batter's batting value. They are analogous to a single number on the

batter's card, in that they tell the increase in expected number of runs scored. However, some care is required here, for two reasons.

One is that the player with the S (or F or 11) is not going to always be on base. The more frequently the player is on base, the more advantageous (or disadvantageous) his running number is. But my analysis cannot measure such a fine distinction. We must assume that each player on the team is equally likely to be on base. Suppose a team has, in its lineup, two S players. Then two ninths of the time a specific baserunner (such as the runner on second) would have an S. So the average loss to the team on any roll due to the S's would be $\frac{2}{9} \cdot .01898$. But a given player with an S will only be causing half of this loss. Thus the average loss caused by the player's S is $.01898/9$.

But this analysis is happening every play of the game (while your team is at bat). An S runner could be on an affected base while several batters are up. The previous paragraph takes this into account. In Section 4, we will consider these matters more fully. If you average the values of the 36 numbers on a batter's card, this gives the average amount by which he increases the team's expected number of runs on a single roll. On average, a batter will be up 4.5 times per game, and so the average of the values of the numbers on his card should be multiplied by 4.5 to give the amount by which his batting numbers increase the team's expected number of runs during a game. The value $.01898/9$ that a person's S hurts you on every roll of the game should be multiplied by 40.5, for the 40.5 rolls during a game, on average. Since $40.5/9 = 4.5$, over the course of a game the .01898 negative value of an S is exactly comparable to the average value of the numbers on the players card. If comparing it with a single number on the player's card, its .01898 should be multiplied by 36, yielding .683, since the numbers on the card are divided by 36 in forming the average.

Thus having an S is roughly equal to the difference between one of your batting numbers being a 7 rather than a 26. For another comparison, I have an old Harmon Killebrew slugger card for which the total of its batting numbers is 2.527. He is S, and this brings its value down to 1.844, quite a decrease. A similar analysis applies to F and 11's for hit-and-running.

The considerations for "playing it safe" were tedious. For each (base,out) situation, I took the weighted sum, summed over all affected numbers, of the difference in your expected number of runs after that roll if you played it safe minus that if you didn't, assuming an S was on base in an affected way. The results were, **assuming that** an S runner is on the affected base

- Runner on first: play it safe against C or D , with any number of outs. It improves your expectation by at least .03.
- Runner on second: play it safe against B, C, or D, with any number of outs. With less than two outs or against a D, it is a significant improvement; with two outs against a B or C, it is borderline. In these cases, I said to play it safe mostly to ease my programming.
- Runner on third: whether to play it safe on a fly ball. This was very complicated because of the various possible outfield configurations. As a compromise to ease my programming, I ended up saying that with none out you should play it safe unless there was an A pitcher and WLF and WCF. (This notation means "Worst Left Fielder" and "Worst Center Fielder". We will continue to use that sort of abbreviation. By this, I mean that the outfielder has fielding rating 1, which puts him in fielding column 3. It is because of this confusion that I chose not to say it with numbers.) There were several other situations where it was borderline. With 1 out, it was even more complicated, and I ended saying to play it safe if (B or C or D) and MCF. Of course, you never play it safe on a fly ball with two outs. In game situations, your play-it-safe strategies will often be different than those I have used here, depending on the actual numbers on the batter's card. Remember that all my analysis is for an "average" player. You should find the numbers in Table 2 useful for helping you decide whether to play it safe in a given situation.
- Runners on first and second: play it safe on a single unless there are two outs and there is a C or D pitcher. (i.e. always against an A or B).
- Runners on first and second: always play it safe on a double. This is a no-brainer.
- Runners on first and third: always play it safe on a single. This is a very clear decision.
- Runners on first and third: play it safe on a fly ball only with none out, MLF, and not an A pitcher. Here is how we figure this; it is rather typical. With none out, playing it safe saves .86 runs $(1.05 - .19)$ over an out at home, and loses .42 runs $(1.47 - 1.05)$ compared to the runner scoring. The relevant numbers 7, 8, 9, 30, 31, and 32 all occur roughly equally often (with 8 occurring more often than the others). The only S-out-at-home is 30 against MLF. As we will see in Table 7, with MLF, you should also have MCF, and so runners hold on 31 and

32. So you should play it safe unless three (borderline with 2) of the hit numbers are outs with the runner scoring, and that only happens against an A. With one out, the .86 and $-.42$ change to .44 and $-.75$, so you should never play it safe.
- With second and third or bases loaded, there is no need whatsoever to play it safe on a fly ball.
- Runners on second and third: always play it safe on a single against an A, never play it safe against a B, and against a C or D play it safe with less than 2 outs.
- With bases loaded, play it safe on a double, and play it safe on a single against a C or D pitcher with none out.

As has already been noted, our analysis depends on a sample of cards, not only for the batting numbers, but also for pitchers, fielders, and speed. In Section 6, we will offer an annotated version of the `Maple` code that was used for most of our work. The percentages for batting numbers, pitchers, fielders, and speed will appear there. These all play a role in the analysis. In particular, 9% of the cards in our sample have an 11. We estimate that with an 11 on first, you hit and run 80% of the time. So we say that with a runner on first or first and third that you hit and run 7% of the time. We have a parameter H to which we give the value .07. All outcomes with runners on first or first and third are of the form H times the hit-and-run outcome plus $(1 - H)$ times the usual outcome.

3. Defense values

The values for pitching and fielding are evaluated under a different system than those for batting. Whereas the batting values involved how much is the expected number of runs increased by a single occurrence of that number, the pitching and fielding values are best expressed as how much is the average number of runs that the opposing team expects to score in an inning affected by the choice of pitcher or fielder. The expected number of runs in an inning against an average pitching/fielding opposition is 0.43273. This is saying that the grades of the opposing pitcher will be as described in the next paragraph. For example, the program knows that if you roll an 8 with a runner on second and nobody out, the resulting (base,out) situation will be (3,1) with probability $A + B$ (i.e. the probability that there is an A or B pitcher, and so it is an out, runner to third), will be (13,0) with probability $S(C + D)$ (since you will be playing it safe if S is on base), and will be (1,0) with 1 run scored with probability $(1 - S)(C + D)$. The parameters A, B, C, D, and S all have numerical values for each run of the

program. For example, to obtain the main numbers, $A = .0941$ and $B = .2788$. We can run the program with parameters $A = 1$, $B = 0$, $C = 0$, and $D = 0$ to see what is the expected number of runs scored in an inning against an A pitcher. The difference between that number, .25186, and the average value of .43273 is the value (in an inning) of an A pitcher compared to an "average" pitcher. We do the same thing for pitching adornments, W, X, Y, and Z, except for them we should compare the difference between the expected number of runs scored with the adornment and that without the adornment. These outcomes appear in Table 5. We say that it is positive if the expected number of runs scored is less than .43273. In Section 4, we will explain how to use these numbers in evaluating your team.

TABLE 5. Pitching values

	runs in inning	difference from average			runs in inning	difference from not having
A	.25186	0.18087		W	.51065	$-.08219$
B	.35543	0.07730		X	.43063	0.00223
C	.46128	$-.02855$		Y	.43140	0.00197
D	.64560	$-.21377$		Z	.41033	0.04036

The program assumes that there will be an A pitcher 9.41%, B pitcher 27.88%, C pitcher 46.38%, and D pitcher 16.33% of the time. Here is how those numbers were obtained. In my sample of cards, the fractions of pitchers which were A, B, C, and D were .04, .16, .41, and .39, respectively. But your better pitchers will be pitch more often than the poorer pitchers. I assumed that you had seven non-relief pitchers and that they combined to pitch, on average, 7 innings per game. I assumed that your best pitcher pitched 23% of those innings, second best pitched 21%, and then 19%, 17%, 15%, 4%, down to 1% for the worst pitcher. It was a fairly straightforward computation to compute that if you were drawing 7 pitchers from a huge sample of which 4% were A's, 16% B's, etc., as above, and you pitched them according to the seven fractions in the preceding sentence, then during these seven innings, you would have an A 6.37%, a B 23.22%, a C 51.13%, and D 19.28% of the time. Relief pitchers tend to have better grades than starters. Among the 1962 set, of the relievers, the fraction in the four grades were $(A, B, C, D) = (.14, .45, .31, .10)$. I assumed that a team had four relievers, of which the best pitches 38% of the time and the others 27%, 21%, and 14% of the two relief innings. An analysis similar to the one done for the non-relief pitchers

gives $(A, B, C, D) = (.2004, .442, .2974, .0602)$ for the relievers. Scaling the non-reliever and reliever fractions by 7/9 and 2/9 yields the figures stated in the first sentence of this paragraph.

Our main analysis of the value of batting numbers was predicated on the other team having average fielders. For infielders, this was easy. For example, .319 of our sample second basemen were in the worst fielding column, and so we used W2b=.319. Other values can be seen in the `Maple` program in Section 6. The probabilities for team and infield being in the various columns were determined using our card sample to say what fraction of each position has each fielding rating, and then combining all the ways that a total can be achieved for the infield or the team, and summing these probabilities. It was easier to divide them into subsets, computing probabilities first for the sums (2b+3b+SS), (1b+C+P), and outfield, and then for the combinations of these. For outfielders, we will explain later in this section how the "average fielder" values were obtained.

Now to determine the value to the team of having an infielder, infield, or team in a certain column, we set the appropriate fielding probability to equal 1, and ran the program to see what would now be the average runs scored in an inning by an average team. For example, if the program is run with W2b=1, M2b=0, and B2b=0 (and everything else with its average values), the expected number of runs scored is .44158, compared to the .43273 against an average defense. Thus having a second baseman in the worst fielding column costs you .00885 ($= .44158 - .43273$) runs per inning. In Table 6, we list the average number of runs scored against a team having an infielder or a team or infield rating in a certain column, and also the difference between that and .43273. This difference will be negative when the fielder hurts you by allowing the other team to score more than .43273 runs. In Section 4, we will tell how to use this in evaluating a team.

TABLE 6. Fielding values

	Worst		Middle		Best	
	expected runs	difference from ave	expected runs	difference from ave	expected runs	difference from ave
2b	.44158	−.00885	.43567	−.00294	.42423	.00850
3b	.43512	−.00239	.43388	−.00115	.41947	.01326
SS	.43617	−.00344	.43394	−.00121	.42206	.01067
IF	.43635	−.00362	.42365	0.00908	.42316	.00957
Team	.43706	−.00433	.42894	0.00379	.42112	.01161

For outfielders, our sample of cards had 88 with fielding rating 1 (the worst), 80 with fielding rating 2, and 27 with fielding rating 3 (the best). Elementary combinatorics computes the numbers listed in the second column of Table 7 as the probabilities that if three cards are selected at random from this batch, their fielding numbers will be in each of the listed configurations. We determine the best way to align your outfielders by running the program with each of the various configurations, and seeing which configuration gives the lowest expected number of runs scored. For example, if you have two 2's and a 3 (best) in the outfield, the opponents' expected number of runs in an inning is .42463 if the 3 is in left, .43322 if the 3 is in center, and .42234 if the 3 is in right, so you should put your 3 in right field. A similar analysis is done for each configuration, yielding the strategy in the third column of Table 7, with the expected number of runs for an average opposing team in column 4, and its difference from the overall average, .43273, in the last column. The number is positive if the opponents' expectation is less than .43273. Many APBA players may prefer to put their outfielders in the position that they are accustomed to play. That could change expectations slightly, but the differences are small. The probabilities that we use for the various outfielders being of each fielding type in our main `Maple` run are obtained using the probabilities and strategies in Table 7. For example, the only time you will have a 2 in right field is if all your outfielders are 2-fielders, and so MRF=.0675, from the table.

TABLE 7. Outfielding strategies and values

fielding ratings	Probability	Best way LF-CF-RF	Expected runs	Difference from ave
111	.0902	1-1-1	.44965	$-.01692$
222	.0675	2-2-2	.43143	0.00130
333	.0024	3-3-3	.41762	0.01511
112	.2517	1-2-1	.43768	$-.00495$
113	.0849	3-1-1	.43613	$-.00340$
122	.2285	2-2-1	.43130	0.00143
223	.0701	2-2-3	.42234	0.01039
233	.0231	3-2-3	.41570	0.01703
133	.0254	3-3-1	.42631	0.00642
123	.1562	3-2-1	.42448	0.00825

4. Evaluating a team

We wish to see how the offensive and defensive values combine, in order to compare teams. Since on average a batter will come up 4.5 times per game, we would multiply the average of the 36 numbers on a card by 4.5 to see how much a player increases the team's expected number of runs per game as compared to an average player. Equivalently, we could take the sum of the 36 numbers times $4.5/36 = 1/8$. The sum of the 36 numbers on some cards, probably all from the same year in the early 1960's is: Mickey Mantle 3.79, Hank Aaron 3.45, Al Kaline 2.60, Harmon Killebrew 2.53, Eddie Mathews 1.68, Ernie Banks 1.29, and Boog Powell $-.55$. Divide these numbers by 8 to obtain their value over an average player in a game, and multiply this by 162, hence multiplying the sum on their card by about 20, to obtain their seasonal Batting Runs Above Average. These numbers of about 70 to 75 for Mantle and Aaron are about the values that sabermetric people would give for them in a very good season.

As we discussed before, the numbers for F, S, and 11 as a running number given in Table 4 should be multiplied by 36 if we wish to add them to the total of the 36 numbers on the players card. We also multiply the 11-value by 0.8 to reflect the perception that one only hit-and-runs 80% of the possible times when there is a runner on first with an 11.

The "difference" values for pitchers given in Table 5 are runs-per-inning compared to an average pitcher. The best way to compare them to batting runs is probably in a runs-per-game scenario. If we estimate that a starting pitcher pitches 7 innings every fifth game, hence 1.4 innings per game, then the difference numbers in the table should be multiplied by 1.4 to give Runs Above Average Pitcher per game. For an A pitcher with a Z, this is $.22 \cdot 1.4 = .308$ per game (out of all the team's games, not just the one the pitcher pitched). Over a 162 game season, it is nearly 50, again quite comparable with sabermetric figures for top pitchers. For a relief pitcher who might pitch 1 inning every second game, we would multiply the value of the difference number in Table 5 by 0.5 to give this pitcher's value per game.

The "difference" numbers for fielding in Tables 6 and 7 should be multiplied by 9 to give the "runs against average" per game, since the numbers in those tables are "per inning."

To make it so we don't have to divide batting numbers by 8, and to make the numbers a little less small, we find it convenient to multiply everything by 8. Thus a team's value can be obtained by first adding,

for each of its 8 or 9 batters[3], the sum of the values given in Table 1 of the 36 numbers on their card. Add these together, and include also baserunning values listed in Table 8 for each of their starters who has an S, F, or 11. Then add to this the pitching values listed in Table 8 for their top five starters and three relievers, including their W, X, Y, and Z, and also the fielding ratings for the team's starters given in Table 9. The values in Table 8 are obtained by multiplying S and F values from Table 4 by 36, the 11-value by $.8 \cdot 36$, the starting pitcher values from Table 5 by $8 \cdot 1.4$ and the relief pitcher values by $8 \cdot 0.5$. The values in Table 9 are obtained by multiplying the values from Table 6 and 7 by $8 \cdot 9$. Since the team and infield fielding ratings include the pitcher, you might proportionalize these if some of the team's five starting pitchers will put you in a different fielding column than the others.

TABLE 8. Speed and pitching values contributed to team total

		Starter		Reliever	
S	−.6833	A	2.0257	A	0.7235
F	0.1393	B	0.8658	B	0.3092
11	0.4000	C	−0.3198	C	−.1142
		D	−2.3942	D	−.8551
		W	−0.9205	W	−.3288
		X	0.0250	X	0.0089
		Y	0.0221	Y	0.0079
		Z	0.4520	Z	0.1614

TABLE 9. Fielding values contributed to team total

	Worst	Middle	Best	OF	
2b	−.6372	−.2117	0.6120	111	−1.2182
3b	−.1721	−.0828	0.9547	222	0.0936
SS	−.2477	−.0871	0.7682	333	1.0879
IF	−.2606	0.6538	0.6890	112	−0.3564
Team	−.3118	0.2729	0.8359	113	−0.2448
				122	0.1030
				223	0.7481
				233	1.2262
				133	0.4622
				123	0.5940

---

[3]depending upon whether you are using a Designated Hitter

We emphasize that the values in Tables 8 and 9 go along with the sums of the numbers from Table 1 of numbers on the batting cards in your starting lineup. If you prefer not working with decimals, you could multiply everything by 100 or 1000 and round off.

5. Mathematics underlying our analysis

In this optional section, we explain the mathematical techniques used in our analysis. Our main assumption is that the reader knows how to multiply matrices.

The numbers on the cards are considered to be 1 to 43, with 42 and 43 representing 2 and 6, respectively, in the extra column of a player with 0's. For $1 \le k \le 43$, let p_k denote the fraction of all the numbers on all the cards which equal k. The values of these are listed near the beginning of the `Maple` program in the next section. The (base,out) situations are numbered 1 to 25, with 1 to 8 being when none are out and the base runners are, respectively, 0, 1, 2, 3, 12, 13, 23, and 123. Situations 9 to 16 are when 1 is out, and 17 to 24 with two out. Situation 25 is the terminal situation of three out.

For $1 \le k \le 43$, let M_k be the 25-by-25 transition matrix whose ith row gives the probabilities of the various situations if k is rolled from situation i. The 25th row is $[0, \ldots, 0, 1]$. It is included for theoretical reasons. These probabilities involve the strategies such as playing-it-safe and hit-and-running as discussed in the preceding sections, and the probabilities of parameters such as pitcher, fielder, and baserunner. The values of these parameters are included near the beginning of the `Maple` program in the next section. Let

$$M_0 = \sum_{i=1}^{43} p_k M_k.$$

This is the transition matrix for an average batter. Let $\widetilde{M}$ denote the 24-by-24 matrix obtained from M_0 by removing the last row and column. The matrix M_0 is the transition matrix of a Markov chain with one absorbing state, and $\widetilde{M}$ is its transient submatrix.

Let $\mathbf{q} = [q_1, \ldots, q_{24}]$ be a row vector of probabilities of being in the various states prior to a roll, with $s := \sum_{i=1}^{24} q_i \le 1$. This sum might be less than 1 because of the probability that the inning has already ended. Then $\mathbf{q}\widetilde{M}$ is the row vector of probabilities of being in the various states after the roll, and the sum of its entries is $\le s$. Let $\mathbf{e} = [1, 0 \ldots, 0]$, a row vector of length 24. It represents the state at

the beginning of the inning. Then

(5.1) $$\mathbf{p} := \mathbf{e} + \mathbf{e}\widetilde{M} + \mathbf{e}\widetilde{M}^2 + \mathbf{e}\widetilde{M}^3 + \cdots$$

has as its ith entry the expected number of times that situation i will occur during an inning. Let $\mathbf{pr}$ denote the vector obtained from $\mathbf{p}$ by dividing each entry of $\mathbf{p}$ by the sum of the entries of $\mathbf{p}$. Then $\mathbf{pr}$ gives the probabilities $\mathrm{pr}(i)$ of being in each of the 24 states. These are the probabilities listed in Table 3.

The easiest way to compute $\mathbf{p}$ is obtained by first multiplying (5.1) on the right by $\widetilde{M}$, obtaining

$$\mathbf{p}\widetilde{M} = \mathbf{e}\widetilde{M} + \mathbf{e}\widetilde{M}^2 + \mathbf{e}\widetilde{M}^3 + \cdots.$$

Combining this with (5.1) yields that $\mathbf{p}\widetilde{M} = \mathbf{p} - \mathbf{e}$, and so

(5.2) $$\mathbf{p}(I - \widetilde{M}) = \mathbf{e},$$

where I is the 24-by-24 identity matrix. This (5.2) is a system of equations which can be solved for $\mathbf{p}$. The system of equations is perhaps more commonly thought of as the transposed version of (5.2), i.e., $(I - \widetilde{M})^T \mathbf{p}^T = \mathbf{e}^T$. Here you are working with column vectors and the equations line up more naturally. At any rate, it is by solving this system of equations that $\mathbf{p}$ and hence $\mathbf{pr}$ is obtained. `Maple` easily solves this system.

Now we explain how we found the numbers in Table 2. Let $E(j)$ denote the expected number of runs scored in an inning subsequent to being in situation j. Let $E(25) = 0$, and let $\mathbf{E}$ be the column vector of length 24 with entries $E(j)$. If you are in situation i and roll a k, the number of runs that you expect to score in the remainder of the inning, including on that roll, is

$$\sum_{j=1}^{25} M_k(i, j)(r_{k,i,j} + E(j)),$$

where $M_k(i, j)$ is the (i, j)-entry of the matrix M_k and $r_{k,i,j}$ is the number of runs scored on that roll (rolling k and going from i to j). Then

$$E(i) = \sum_{k=1}^{43} p_k \sum_{j=1}^{25} M_k(i, j)(r_{k,i,j} + E(j)),$$

or equivalently

(5.3) $$\mathbf{E} = \widetilde{M}\mathbf{E} + \mathbf{b},$$

where $\mathbf{b}$ is a column vector of length 24 whose ith entry is $\sum_{k=1}^{43} p_k \sum_{j=1}^{25} M_k(i, j) r_{k,i,j}$.

Most of the time, $r_{k,i,j}$ does not depend on k and equals

(5.4) $r(i,j) := 1 + BR(i) + \left[\frac{i-1}{8}\right] - BR(j) - \left[\frac{j-1}{8}\right],$

where

$$BR(i) := \begin{cases} 0 & i \equiv 1 \ (8) \\ 1 & i \equiv 2,3,4 \ (8) \\ 2 & i \equiv 5,6,7 \ (8) \\ 3 & i \equiv 0 \ (8) \end{cases}$$

is the number of base runners in situation i, and $\left[\frac{i-1}{8}\right]$, which denotes the integer part of the fraction, is the number of outs in situation i. The 1 in (5.4) is for the batter. The formula (5.4) does not work when $j = 25$, when $r_{k,i,j}$ is usually 0. Let R be a column vector of length 24 whose ith entry is $\sum_{j=1}^{24} \widetilde{M}(i,j)r(i,j)$. This gives, except for the exceptional situations noted below, the average number of runs scored on a single roll from situation i. Note that $r(i,j)$ is sometimes negative, which is nonsensical, but this will never occur if $\widetilde{M}(i,j) \neq 0$.

There are two types of exceptions when the formulas of the preceding paragraph do not give the number of runs scored on the play. Those with $j = 25$ are the rare cases, such as 4 with a runner on third, or 15 with runners on first and second and BLF, in which runs score on the play, but then a base runner makes the third out. Let $d(k,i)$ denote the expected number of runs which are scored on inning-terminating plays when rolling a k from situation i. For example, $d(9,20) = B + D$ and $d(4,20) = 1$. Let $\mathbf{d}$ be a column vector of length 24 whose ith entry is $d_i := \sum_{k=1}^{43} p_k d(k,i)$.

A bigger exception is the cases in which the batter stays up after the roll, such as rolling a 37 with the bases empty if the pitcher does not have a W. We have $M_{37}(1,1) = 1 - W$. The formula (5.4) would incorrectly say that one runner scored on the play. We define a column vector No of length 24 which has ith entry $\text{No}_i := \sum_{k=1}^{43} p_k U(k,i)$, where $U(k,i)$ is the probability that the batter stays up if you roll k in situation i. The above example for 37 says $U(37,1) = 1 - W$ and contributes $p_{37}(1-W)$ to No_1.

Now we can write $\mathbf{b} = R + \mathbf{d} - \text{No}$, and (5.3) becomes

$$(I - \widetilde{M})\mathbf{E} = R + \mathbf{d} - \text{No},$$

and this is what was solved by `Maple` to obtain Table 2.

The values $V(k)$ of the numbers on a card are now easily obtained as

$$
\begin{aligned}
V(k) \;=\; & \sum_{i=1}^{24} \mathrm{pr}(i) \bigg(\sum_{j=1}^{24} M_k(i,j)\big(r(i,j) + E(j) - E(i) - U(k,i)\big) \\
& + M_k(i,25)\big(d(k,i) - E(i)\big) \bigg).
\end{aligned}
$$

This is obtained by averaging, over all initial situations i and all subsequent situations obtained when rolling a k, the number of runs obtained on that roll plus the change in expected number of subsequent runs to be obtained later in the inning. This yields the numbers in Table 1.

6. The `Maple` program

In this section, we present the `Maple` program that performed most of our computations. The reader can obtain from this the input probabilities that we used, and our transcription into computer code of the "boards." We will annotate the program with explanatory remarks.

The first line inputs the number of occurrences of each number on our sample of 350 APBA cards. There are 43 numbers. The last two are 2 and 6 in the extra column (on batters having a 0). For batters with a 0, the numbers in their extra column are proportionalized. For example, if a player has two 0's and 7 1's in the extra column, that counts as $2 \cdot \frac{7}{36}$ 1's. This is the reason that all the extra-base hit numbers appear as decimals here. We add all these and divide by that number to get the fraction of the time that the numbers come up, called P[i].

```
>  P:=Vector([166.13,0,15.44,85.06,150.43,220,625,1108,
>  690,122,44,350,1149,982,45,42,41,73,47,78,20,
>  25,18,546,350,518,510,545,468,814,580,723,
>  260,240,350,52,78,69,77,124,38,25.11,134.55]);
>  T:=add(P[i],i=1..43);
```
$$T := 12597.72$$
```
>  for i from 1 to 43 do P[i]:=P[i]/T od:    evalm(P);
```

$[0.01319, 0., 0.001226, 0.006752, 0.01194, 0.01746, 0.04961, 0.08795, 0.05477,$
$0.009684, 0.003493, 0.02778, 0.09121, 0.07795, 0.003572, 0.003334, 0.003255,$
$0.005795, 0.003731, 0.006192, 0.001588, 0.001984, 0.001429, 0.04334, 0.02778,$
$0.04112, 0.04048, 0.04326, 0.03715, 0.06461, 0.04604, 0.05739, 0.02064,$
$0.01905, 0.02778, 0.004128, 0.006192, 0.005477, 0.006112, 0.009843, 0.003016,$
$0.001993, 0.01068]$

The next few lines establish 43 matrices M[k], each 25-by-25, which will be the transition matrices for the various rolled numbers, and fill them with 0's, and also a matrix No, which will tell when a batter stays up at bat after the number is rolled. This information is important because the number of runs scored on a roll is determined by a formula which assumes that the batter either got on base or made an out, and so the information in No will override this. At this point in the program, the matrix No is just filled with 0's.

The 25 situations are: 1 to 8 are none out and runners on the following bases 0,1,2,3,12,13,23,123. Then 9 through 16 are these with 1 out, 17 through 24 with two out, and 25 is when three are out.

```
>  for k from 1 to 43 do M[k]:=matrix(25,25) od:
>  for k from 1 to 43 do for i from 1 to 25 do
>  for j from 1 to 25 do
>  M[k][i,j]:=0 od od od:
>  No:=matrix(25,43):  for i from 1 to 25 do
>  for j from 1 to 43 do
>  No[i,j]:=0 od od:
```

Now we input the values of the parameters. This is one part a reader may want to look at even if he is not interested in all the `Maple` code. D pitcher must be called De because `Maple` has a reserved use for the letter D.

```
>  S:=.22:  F:=.167:  A:=.0941:  B:=.2788:  C:=.4638:  De:=1-A-B-C:
>  W:=.052:
>  X:=.059:  Y:=.325:  Z:=.445:
>  W2b:=.319:  M2b:=.264:  B2b:=1-W2b-M2b:
>  W3b:=.435:  M3b:=.4491:  B3b:=1-W3b-M3b:
>  WSS:=.385:  MSS:=.4425:  BSS:=1-WSS-MSS:
>  WLF:=.342:  MLF:=.366:  BLF:=1-WLF-MLF:
>  WCF:=.175:  MCF:=.797:  BCF:=1-WCF-MCF:
>  WRF:=.837:  MRF:=.067:  BRF:=1-WRF-MRF:
>  WT:=.4893:  MT:=.4929:  BT:=1-WT-MT:
>  WIF:=.7176:  MIF:=.2818:  BIF:=1-WIF-MIF: H:=.07:
```

Next we fill in the matrix No with situations and probabilities in which the batter stays up. For example, No[1,37]= 1−W says that if you roll 37 with nobody on, nobody out, the batter stays up unless the pitcher has a W. The program inserts the information that W=.052 to

say that No[1,37]=.948, filling in that entry of the matrix No. We just write these out for the "No out" situations and then, in the final lines of this group we just add 8 to the situation number and put the same probability in, to account for what happens when 1 or 2 are out. The very last line refers to the fact that 23 with a runner on third is a balk if two are out rather than a single. The formulas involving H are due to the possibility of hit and running.

```
>  No[1,36]:=1:  No[1,37]:=1-W: No[1,38]:=1-W: No[1,39]:=1-W:
>  No[1,40]:=1-W: No[2,23]:=1:  No[2,36]:=1:  No[2,37]:=1:
>  No[2,38]:=1:  No[2,39]:=1:
>  No[2,40]:=BT*(1-H):
>  No[3,22]:=1:  No[3,23]:=1:  No[3,36]:=1:  No[3,37]:=MT:
No[3,39]:=1:
>  No[3,41]:=1:
>  No[4,36]:=1:  No[4,41]:=1:
>  No[2,14]:=Z: No[5,14]:=Z: No[8,14]:=Z:

>  No[5,23]:=1:  No[5,36]:=1:  No[5,37]:=1:
>  No[6,21]:=BIF*(1-H)+H: No[6,22]:=1:  No[6,23]:=1-H: No[6,36]:=1:
>  No[6,37]:=1:  No[6,38]:=1:  No[6,39]:=1:
>  No[6,40]:=1-WT*(1-H): No[6,41]:=1-H: No[6,13]:=H: No[6,14]:=H:

>  No[7,36]:=1:  No[7,41]:=1:

>  No[8,23]:=1-BT: No[8,36]:=1:  No[8,39]:=1:  No[2,13]:=H:
>  No[2,21]:=H:
>  No[2,22]:=H: No[2,35]:=H:
>  for i from 1 to 8 do for j from 1 to 43 do No[i+8,j]:=No[i,j]:
>  No[i+16,j]:=No[i,j] od od:
>  No[20,23]:=1:
```

The next batch of lines is a transcription of what happens when nobody is on and nobody out (base situation 1). For example, the line M[8][1,2]:=C+De says that against a C or D pitcher rolling an 8 gives you a runner on first (base situation 2). The values of C and De inputted earlier in the program are inserted to give the numerical probability of this happening.

```
>  M[1][1,1]:=1:
>  M[2][1,4]:=1:  M[42][1,4]:=1:
>  M[3][1,4]:=1:
>  M[4][1,3]:=1:  M[5][1,3]:=1:  M[6][1,3]:=1:  M[11][1,3]:=1:
>  M[43][1,3]:=1:
>  M[7][1,2]:=1:  M[10][1,2]:=1:  M[14][1,2]:=1:  M[18][1,2]:=1:
>  M[20][1,2]:=1:  M[23][1,2]:=1:
```

```
>   M[8][1,2]:=C+De:  M[9][1,2]:=B+De:
>   M[8][1,9]:=A+B: M[9][1,9]:=A+C:
>   M[12][1,9]:=1:  M[13][1,9]:=1:
>   M[15][1,2]:=WLF: M[15][1,9]:=1-WLF:
>   M[16][1,2]:=WCF: M[16][1,9]:=1-WCF:
>   M[17][1,3]:=WRF: M[17][1,2]:=MRF: M[17][1,9]:=BRF:
>   M[19][1,2]:=1-B3b:  M[19][1,9]:=B3b:
>   M[21][1,2]:=1-BIF: M[21][1,9]:=BIF:
>   M[22][1,2]:=WIF: M[22][1,9]:=1-WIF:
>   for i from 24 to 35 do M[i][1,9]:=1 od:
>   M[36][1,1]:=1:
>   for i from 37 to 40 do M[i][1,2]:=W: M[i][1,1]:=1-W od:
>   M[41][1,3]:=1-BT: M[41][1,2]:=BT:
```

The previous lines took care of nobody on, nobody out. The next
two lines adapt this to one out and two out. Then we start doing the
analogous things for "runner on first."

```
>   for i from 1 to 43 do  for k from 1 to 9 do M[i][9,k+8]:=M[i][1,k]:
>   M[i][17,k+16]:=M[i][1,k] od od:
>   M[1][2,1]:=1:  M[5][2,1]:=1:
>   M[2][2,4]:=1:  M[4][2,4]:=1:  M[42][2,4]:=1:
>   M[3][2,3]:=1-S: M[3][2,11]:=S:
>   M[6][2,7]:=1-F: M[6][2,3]:=F: M[43][2,7]:=1-F: M[43][2,3]:=F:
>   M[7][2,7]:=B+(1-S)*(C+De):  M[7][2,11]:=A: M[7][2,5]:=S*(C+De):
>   M[8][2,6]:=(C+De)*(1-S): M[8][2,11]:=A+B: M[8][2,5]:=(C+De)*S:
>   M[9][2,11]:=C: M[9][2,5]:=1-C:
>   M[10][2,7]:=1-S*(C+De):  M[10][2,5]:=S*(C+De):  M[11][2,7]:=1-S*(C+De)
>   M[11][2,5]:=S*(C+De):
>   M[12][2,5]:=W: M[12][2,10]:=(1-W)*WIF: M[12][2,17]:=(1-W)*(1-WIF):
>   M[13][2,10]:=1:  M[26][2,10]:=1:
>   M[14][2,2]:=Z: M[14][2,5]:=1-Z:
>   M[15][2,5]:=1:
>   M[16][2,10]:=BCF*(1-F): M[16][2,11]:=BCF*F:
>   M[16][2,5]:=(1-BCF)*(1-F): M[16][2,6]:=(1-BCF)*F:
>   M[17][2,11]:=BRF*(1-S*(C+De)):
>   M[17][2,7]:=WRF*(1-S*(C+De))+MRF*(1-F-S*(C+De)):  M[17][2,3]:=MRF*F:
>   M[17][2,5]:=S*(C+De):
>   M[18][2,11]:=BSS: M[18][2,6]:=1-BSS:
```

```
>  M[19][2,5]:=1:  M[22][2,5]:=1:
>  M[20][2,11]:=B2b:  M[20][2,6]:=1-B2b:
>  M[21][2,7]:=1-F*WIF: M[21][2,3]:=F*WIF:
>  M[23][2,9]:=1:
>  M[24][2,17]:=1:  M[25][2,17]:=1:
>  M[27][2,10]:=X: M[27][2,17]:=(1-X)*B3b:
>  M[27][2,11]:=(1-X)*(1-B3b):
>  M[28][2,11]:=1-BSS: M[28][2,10]:=BSS:
>  M[29][2,10]:=Y+(1-Y)*MIF: M[29][2,11]:=(1-Y)*(1-MIF):
>  for i from 30 to 34 do M[i][2,10]:=1 od:
>  M[35][2,5]:=W: M[35][2,10]:=1-W:
>  M[36][2,3]:=1:
>  M[37][2,9]:=1:  M[39][2,9]:=1:
>  M[38][2,3]:=1-S: M[38][2,2]:=S:
>  M[40][2,11]:=WT*(1-F): M[40][2,12]:=WT*F: M[40][2,10]:=MT:
>  M[40][2,9]:=BT:
>  M[41][2,17]:=WT+BT: M[41][2,10]:=MT:
```

In the next few lines, we adapt the above information to the situation when one or two are out. If adding 16 to the situation number of the outcome makes it ≥ 25, so that three are out, the probability is added onto outcome 25. Then there are a few lines which override this, for special situations that happen when two are out. For example, the first line will have filled in M[6][18, 23] := 1−F and M[6][18, 19] :=F, but since anyone scores with two outs this is overridden in the next line.

```
>  for i from 1 to 43 do for k from 1 to 17 do M[i][10,k+8]:=M[i][2,k]:
>  if M[i][2,k]<>0 then if k+16<=25  then M[i][18,k+16]:=M[i][2,k]
>  else
>  M[i][18,25]:=M[i][18,25]+M[i][2,k] fi end if od  od:
>  M[6][18,19]:=1:  M[6][18,23]:=0:  M[3][18,19]:=1:  M[3][18,25]:=0:
>  M[16][18,22]:=WCF+MCF: M[16][18,21]:=0:
>  M[23][18,25]:=0:  M[23][18,19]:=1:
```

We continue as above, filling in the matrices for the other six base situations.

```
>  M[1][3,1]:=1:  M[3][3,1]:=1:  M[5][3,1]:=1:
>  M[2][3,4]:=1:  M[4][3,4]:=1:  M[42][3,4]:=1:
>  M[6][3,3]:=1:  M[43][3,3]:=1:
>  M[7][3,2]:=1-S*(1-A): M[7][3,5]:=S*(1-A):
```

```
> M[8][3,12]:=A+B: M[8][3,2]:=(C+De)*(1-S): M[8][3,6]:=(C+De)*S:
> M[9][3,12]:=A+C: M[9][3,2]:=(B+De)*(1-S): M[9][3,6]:=(B+De)*S:
> M[10][3,12]:=A+B: M[10][3,6]:=(C+De)*(1-F): M[10][3,2]:=(C+De)*F:
> for i from 12 to 13 do M[i][3,11]:=1 od:  M[11][3,7]:=(1-A)*S:
> M[11][3,11]:=1-(1-A)*S:
> M[14][3,5]:=1:  M[19][3,5]:=1:
> M[15][3,2]:=WLF+MLF: M[15][3,11]:=BLF:
> M[16][3,3]:=WCF*(1-(1-A)*S): M[16][3,11]:=(1-WCF)*(1-(1-A)*S):
> M[16][3,6]:=(1-A)*S:
> M[17][3,11]:=(1-F)*(WRF+BRF): M[17][3,12]:=F*(WRF+BRF):
> M[17][3,6]:=MRF*(1-F): M[17][3,2]:=MRF*F:
> M[18][3,5]:=WSS+BSS: M[18][3,12]:=MSS*(1-F): M[18][3,9]:=MSS*F:
> M[20][3,3]:=W2b+M2b:  M[20][3,12]:=B2b:
> M[21][3,6]:=1:
> M[22][3,3]:=S: M[22][3,4]:=1-S:
> M[23][3,9]:=1:
> M[24][3,11]:=1:  M[30][3,11]:=1:  M[31][3,11]:=1:  M[27][3,11]:=1:
> M[33][3,11]:=1:  M[34][3,11]:=1:
> M[25][3,17]:=1:
> M[26][3,12]:=W2b+M2b:  M[26][3,11]:=B2b:
> M[28][3,11]:=1-WSS: M[28][3,12]:=WSS:
> M[29][3,12]:=1:
> M[32][3,11]:=BRF+MRF*(1-F): M[32][3,12]:=WRF+MRF*F:
> M[35][3,11]:=1-W: M[35][3,5]:=W:
> M[36][3,4]:=1:
> M[37][3,10]:=WT: M[37][3,9]:=MT: M[37][3,17]:=BT:
> M[38][3,5]:=WT+MT: M[38][3,6]:=BT:
> M[39][3,9]:=1:
> M[40][3,6]:=WT: M[40][3,12]:=MT*(1-S): M[40][3,17]:=MT*S:
> M[40][3,10]:=BT:
> M[41][3,1]:=WT+MT: M[41][3,9]:=BT:
> for i from 1 to 43 do for k from 1 to 17 do M[i][11,k+8]:=M[i][3,k]:
> if M[i][3,k]<>0 then if  k+16<=25 then M[i][19,k+16]:=M[i][3,k]
else
> M[i][19,25]:=M[i][19,25]+M[i][3,k] fi end if od  od:
> M[17][19,18]:=MRF: M[17][19,22]:=0:
> M[18][19,25]:=1-WSS: M[18][19,21]:=0:  M[18][19,22]:=WSS:
> M[38][19,22]:=1:  M[38][19,21]:=0:  M[11][19,23]:=(1-A)*S:
> M[11][19,25]:=0:  M[11][19,19]:=1-(1-A)*S:
```

```
>  M[1][4,1]:=1:  M[3][4,1]:=1:  M[6][4,1]:=1:

>  M[2][4,4]:=1:  M[42][4,4]:=1:

>  M[5][4,3]:=1:  M[43][4,3]:=1:  M[10][4,3]:=1:  M[11][4,3]:=1:

>  M[4][4,9]:=1:  M[41][4,9]:=1:

>  M[7][4,2]:=1-A: M[7][4,9]:=A*(1-S*WLF*WCF):
>  M[7][4,12]:=A*WLF*WCF*S:

>  M[8][4,2]:=1-B: M[8][4,9]:=B*(1-S): M[8][4,12]:=B*S:

>  M[9][4,9]:=1-(A+C)*S: M[9][4,12]:=(A+C)*S: M[26][4,9]:=1:
>  M[28][4,9]:=1:

>  M[12][4,12]:=X: M[12][4,17]:=1-X:

>  M[13][4,12]:=1:  M[29][4,12]:=1:  M[33][4,12]:=1:  M[34][4,12]:=1:

>  M[14][4,6]:=1:  M[16][4,6]:=1:

>  M[15][4,3]:=1:  M[19][4,2]:=1-B3b:  M[19][4,11]:=B3b:

>  M[17][4,2]:=1-MRF: M[17][4,9]:=MRF*(1-(1-A)*S):
>  M[17][4,12]:=MRF*(1-A)*S:

>  M[18][4,2]:=1:  M[23][4,2]:=1:  M[20][4,2]:=1:  M[22][4,2]:=1:

>  M[21][4,9]:=1-WIF: M[21][4,3]:=WIF:

>  M[24][4,17]:=BIF: M[24][4,10]:=MIF: M[24][4,11]:=WIF:

>  M[25][4,17]:=1:

>  M[27][4,12]:=1:

>  M[30][4,9]:=1-S-BLF*(1-F-S):  M[30][4,12]:=S+BLF*(1-F-S):

>  M[31][4,9]:=1-S-BCF*(1-F-S):  M[31][4,12]:=S+BCF*(1-F-S):

>  M[32][4,9]:=1-S-(1-WRF)*(1-F-S):  M[32][4,12]:=S+(1-WRF)*(1-F-S):

>  M[35][4,6]:=W: M[35][4,12]:=1-W:

>  M[36][4,1]:=1:

>  for i from 37 to 39 do M[i][4,9]:=1-(1-A)*S: M[i][4,12]:=(1-A)*S
>  od:

>  M[40][4,9]:=1-WT: M[40][4,6]:=WT:

>  for i from 1 to 43 do for k from 1 to 17 do M[i][12,k+8]:=M[i][4,k]:
>  if M[i][4,k]<>0 then if k+16<=25 then M[i][20,k+16]:=M[i][4,k]
else
>  M[i][20,25]:=M[i][20,25]+M[i][4,k] fi end if od  od:
```

```
>  M[15][20,18]:=WLF:  M[15][20,19]:=1-WLF:
>  M[30][12,25]:=MLF*S*(1-MCF*(1-A)):
>  M[30][12,20]:=BLF*(1-F)+(1-BLF)*S*MCF*(1-A):
>  M[30][12,17]:=1-M[30][12,25]-M[30][12,20]:  M[31][12,25]:=S*MCF*A:
>  M[31][12,17]:=WCF+BCF*F+MCF*(1-S):
>  M[31][12,20]:=BCF*(1-F)+MCF*S*(1-A):
>  M[32][12,17]:=WRF*(1-S)+F*(1-WRF):  M[32][12,25]:=S*(1-MCF*(1-A))*WRF:
>  M[32][12,20]:=1-M[32][12,17]-M[32][12,25]:  M[7][12,17]:=A:
>  M[7][12,20]:=0:  M[8][12,17]:=B*(1-S*MCF):  M[8][12,20]:=B*S*MCF:
>  M[9][12,20]:=C*S*MCF:  M[9][12,17]:=A+C*(1-S*MCF)+B+De:

>  M[22][20,25]:=1:  M[22][20,18]:=0:
>  M[40][20,18]:=WT:  M[40][20,22]:=0:  M[23][20,17]:=1:
>  M[23][20,18]:=0:

>  M[1][5,1]:=1:  M[2][5,1]:=1:  M[4][5,1]:=1:
>  M[5][5,4]:=1:  M[42][5,4]:=1:
>  M[3][5,3]:=1-S: M[3][5,7]:=S: M[6][5,3]:=1-S: M[6][5,7]:=S:
>  M[43][5,3]:=1-S: M[43][5,7]:=S:
>  M[11][5,7]:=1-S: M[11][5,8]:=S:
>  M[7][5,6]:=1-S: M[7][5,8]:=S:
>  M[8][5,15]:=A+B: M[8][5,5]:=(C+De)*(1-S): M[8][5,8]:=(C+De)*S:
>  M[9][5,15]:=A+C: M[9][5,8]:=B+De:
>  M[10][5,8]:=C+De:  M[10][5,15]:=A+B:
>  M[12][5,8]:=W: M[12][5,20]:=1-W:
>  M[13][5,13]:=1:  M[26][5,13]:=1:  M[30][5,13]:=1:  M[33][5,13]:=1:
>  M[34][5,13]:=1:
>  M[14][5,5]:=Z: M[14][5,8]:=1-Z:
>  M[15][5,7]:=(1-BLF)*(1-S): M[15][5,12]:=BLF*(1-S): M[15][5,8]:=S:
>  M[16][5,15]:=1:  M[17][5,8]:=1:  M[20][5,8]:=1:
>  M[18][5,6]:=1-S: M[18][5,8]:=S:
>  M[19][5,6]:=1-M3b:  M[19][5,15]:=M3b:
>  M[21][5,8]:=1-BIF: M[21][5,15]:=BIF:
>  M[22][5,8]:=1:  M[40][5,8]:=1:
>  M[23][5,5]:=1:
>  M[24][5,20]:=1:  M[25][5,20]:=1:
>  M[27][5,13]:=X: M[27][5,15]:=1-X:
>  M[28][5,14]:=1-B3b:  M[28][5,20]:=B3b:
>  M[29][5,15]:=WIF: M[29][5,20]:=1-WIF:
>  M[31][5,13]:=1-F*WCF: M[31][5,14]:=F*WCF:
>  M[32][5,13]:=1-F*(1-BRF): M[32][5,14]:=F*(1-BRF):
```

```
>  M[35][5,8]:=W: M[35][5,13]:=1-W:

>  M[36][5,7]:=1:  M[37][5,7]:=1:

>  M[38][5,15]:=1-MT: M[38][5,14]:=MT:

>  M[39][5,20]:=WT: M[39][5,19]:=MT: M[39][5,18]:=BT*(1-F):
>  M[39][5,14]:=BT*F:

>  M[41][5,25]:=1:
>  for i from 1 to 43 do for k from 1 to 25 do if M[i][5,k]<>0
then if
>  k+8<=25  then M[i][13,k+8]:=M[i][5,k]
>  else M[i][13,25]:=M[i][13,25]+M[i][5,k] fi:
>  if k+16<=25 then M[i][21,k+16]:=M[i][5,k] else
>  M[i][21,25]:=M[i][21,25]+M[i][5,k] fi end if od  od:

>  M[8][21,22]:=C+De:  M[8][21,21]:=0:  M[7][21,22]:=1-S:

>  M[7][21,25]:=S:

>  M[7][21,24]:=0:

>  M[8][21,24]:=0:

>  M[11][21,24]:=0:  M[11][21,23]:=1:  M[15][21,24]:=0:

>  M[15][21,23]:=1-BLF: M[15][21,25]:=BLF: M[18][21,24]:=0:

>  M[18][21,25]:=S: M[18][21,22]:=1-S:

>  M[1][6,1]:=1:  M[5][6,1]:=1:

>  M[2][6,4]:=1:  M[3][6,4]:=1:  M[4][6,4]:=1:  M[42][6,4]:=1:

>  M[6][6,3]:=1:  M[43][6,3]:=1:

>  M[7][6,5]:=(1-B)*S: M[7][6,6]:=(1-B)*(1-S): M[7][6,10]:=B*(1-S*MLF):
>  M[7][6,14]:=B*S*MLF:

>  M[8][6,10]:=A+C*(1-S*MLF): M[8][6,6]:=(B+De)*(1-S):

>  M[8][6,5]:=(B+De)*S: M[8][6,14]:=C*S*MLF:

>  M[9][6,10]:=A: M[9][6,5]:=(1-A):

>  M[10][6,10]:=A: M[10][6,11]:=(1-A)*(1-S): M[10][6,5]:=(1-A)*S:

>  M[11][6,5]:=1-F: M[11][6,6]:=F: M[12][6,17]:=1:

>  M[13][6,14]:=1:  for i from 32 to 35 do M[i][6,14]:=1
od:

>  M[14][6,8]:=1:  M[18][6,8]:=1:

>  M[15][6,5]:=1-BLF: M[15][6,10]:=BLF:

>  M[16][6,5]:=WCF*(1-F-S*MLF): M[16][6,6]:=WCF*F:
>  M[16][6,10]:=MCF*(1-S*MLF): M[16][6,9]:=BCF*(1-S):
>  M[16][6,19]:=BCF*S*(1-MLF): M[16][6,14]:=S*MLF:

>  M[17][6,3]:=(1-MRF)*(1-S): M[17][6,11]:=MRF*(1-S): M[17][6,5]:=S:

>  M[19][6,6]:=1:

>  M[20][6,11]:=1-F-S: M[20][6,7]:=F: M[20][6,5]:=S:

>  M[21][6,14]:=1-BIF: M[21][6,3]:=BIF:

>  M[22][6,3]:=1:
```

```
> M[23][6,6]:=1:
> M[24][6,17]:=1:  M[25][6,17]:=1:
> M[26][6,17]:=1-W2b:  M[26][6,9]:=W2b:
> M[27][6,14]:=Y: M[27][6,15]:=1-Y:
> M[28][6,17]:=BSS: M[28][6,9]:=1-BSS:
> M[29][6,14]:=X: M[29][6,15]:=1-X:
> M[30][6,14]:=BLF+S*MLF*(1-A):  M[30][6,18]:=MLF*S*A:
> M[30][6,10]:=WLF+MLF*(1-S):
> M[31][6,14]:=MCF: M[31][6,10]:=(1-MCF):
> M[36][6,7]:=(1-MT*S): M[36][6,6]:=MT*S:
> M[37][6,7]:=WT*(1-S): M[37][6,6]:=WT*S: M[37][6,9]:=MT:
> M[37][6,12]:=BT:
> M[38][6,7]:=1-BT*S: M[38][6,6]:=BT*S:
> M[39][6,11]:=1-WT: M[39][6,7]:=WT*(1-S): M[39][6,6]:=WT*S:
> M[40][6,10]:=WT*(1-S): M[40][6,18]:=WT*S: M[40][6,11]:=1-WT:
> M[41][6,3]:=1:
> for i from 1 to 43 do for k from 1 to 20 do if M[i][6,k]<>0
then if
> k+8<=25 then M[i][14,k+8]:=M[i][6,k] else
> M[i][14,25]:=M[i][14,25]+M[i][6,k] fi:
> if k+16<=25 then M[i][22,k+16]:=M[i][6,k] else
> M[i][22,25]:=M[i][22,25]+M[i][6,k] fi end if od  od:
> M[9][22,22]:=1-A: M[9][22,21]:=0:  M[7][14,18]:=B: M[8][14,18]:=A+C:
> M[8][14,22]:=0:  M[7][14,22]:=0:  M[30][14,25]:=S*MLF:
> M[30][14,22]:=BLF:
> M[15][22,22]:=1-BLF: M[15][22,21]:=0:
> M[21][22,24]:=MIF: M[21][22,25]:=0:  M[21][22,21]:=WIF:
> M[1][7,1]:=1:  M[4][7,1]:=1:
> M[2][7,4]:=1:  M[5][7,4]:=1:  M[42][7,4]:=1:
> M[6][7,3]:=1:  M[43][7,3]:=1:
> M[3][7,9]:=1:
> M[7][7,11]:=A: M[7][7,3]:=(1-A)*(1-S)+B*S: M[7][7,6]:=(1-A-B)*S:
> M[8][7,12]:=B: M[8][7,2]:=(1-B)*(1-S): M[8][7,6]:=(1-B)*S:
> M[9][7,12]:=B+C: M[9][7,11]:=A: M[9][7,6]:=De:
> M[10][7,12]:=C: M[10][7,11]:=A: M[10][7,6]:=B+De:
> M[11][7,7]:=1:
> M[12][7,15]:=1-BT: M[12][7,20]:=BT:
> M[13][7,15]:=1:  M[27][7,15]:=1:  M[29][7,15]:=1:  M[33][7,15]:=1:
> M[34][7,15]:=1:
```

```
>  M[14][7,8]:=1:  M[21][7,8]:=1:
>  M[15][7,3]:=WLF*(1-S*(1-B)): M[15][7,7]:=MLF*(1-S*(1-B))+BLF:
>  M[15][7,6]:=(1-BLF)*S*(1-B):
>  M[16][7,3]:=(1-BCF)*(1-S*(1-B)): M[16][7,12]:=BCF:
>  M[16][7,6]:=(1-BCF)*S*(1-B):
>  M[17][7,11]:=1-F-S*(1-B): M[17][7,3]:=F: M[17][7,6]:=S*(1-B):
>  M[18][7,8]:=1-BSS: M[18][7,15]:=BSS:
>  M[19][7,6]:=1-W3b*F: M[19][7,2]:=W3b*F:
>  M[20][7,6]:=1-B2b:  M[20][7,15]:=B2b:
>  M[22][7,14]:=1:
>  M[23][7,25]:=1:
>  M[24][7,20]:=1-WIF: M[24][7,15]:=WIF:
>  M[25][7,20]:=1:
>  M[26][7,12]:=1:  M[28][7,12]:=1:
>  M[30][7,15]:=1-WLF: M[30][7,11]:=WLF:
>  M[31][7,12]:=WCF: M[31][7,15]:=MCF: M[31][7,11]:=BCF:
>  M[32][7,15]:=1-MRF: M[32][7,11]:=MRF*(1-F): M[32][7,12]:=MRF*F:
>  M[35][7,8]:=W: M[35][7,15]:=1-W:
>  M[36][7,4]:=1:
>  M[37][7,6]:=1:
>  M[38][7,12]:=1:  M[39][7,12]:=1:
>  M[40][7,12]:=1-BT*S: M[40][7,20]:=BT*S:
>  M[41][7,4]:=1:
>  for i from 1 to 43 do for k from 1 to 25 do if M[i][7,k]<>0
then if
>  k+8<=25 then M[i][15,k+8]:=M[i][7,k] else
>  M[i][15,25]:=M[i][15,25]+M[i][7,k] fi:
>  if k+16<=25 then M[i][23,k+16]:=M[i][7,k] else
>  M[i][23,25]:=M[i][23,25]+M[i][7,k] fi end if od  od:
>  M[10][23,18]:=B+De:  M[10][23,22]:=0:  M[11][23,19]:=1:
>  M[11][23,23]:=0:  M[7][23,19]:=1-A: M[7][23,22]:=0:  M[8][23,22]:=0:
>  M[8][23,25]:=S*(1-B)+B: M[15][23,22]:=0:  M[15][23,23]:=MLF:
>  M[15][23,19]:=1-MLF: M[16][23,19]:=1:  M[16][23,25]:=0:
>  M[16][23,22]:=0:  M[17][23,22]:=0:  M[17][23,25]:=1-F:
>  M[18][23,22]:=MSS+WSS*(1-F): M[18][23,18]:=WSS*F: M[18][23,24]:=0:
>  M[1][8,1]:=1:  M[15][23,19]:=1-MCF: M[15][23,23]:=MCF:
>  M[2][8,4]:=1:  M[42][8,4]:=1:  M[3][8,4]:=1:  M[6][8,4]:=1:
>  M[4][8,7]:=1:  M[43][8,7]:=1:  M[7][8,6]:=1-S*(C+De):
>  M[7][8,5]:=S*(C+De):  M[10][8,7]:=1-S*(C+De):
>  M[10][8,5]:=S*(C+De):  M[11][8,7]:=1-S*(C+De):  M[11][8,5]:=S*(C+De):
```

```
>  M[5][8,3]:=1-S: M[5][8,7]:=S:
>  M[8][8,13]:=A+B: M[8][8,6]:=(C+De)*(1-S): M[8][8,5]:=(C+De)*S:
>  M[9][8,13]:=A+C: M[9][8,8]:=B+De:
>  M[12][8,20]:=1:
>  M[13][8,16]:=1:  M[33][8,16]:=1:  M[34][8,16]:=1:
>  M[14][8,8]:=1:  M[20][8,8]:=1:
>  M[15][8,6]:=(1-BLF*S)*(1-S*(C+De))-WLF*F: M[15][8,2]:=WLF*F:
>  M[15][8,14]:=BLF*S*(1-S*(C+De)):  M[15][8,5]:=S*(C+De):
>  M[16][8,6]:=WCF*(1-(C+De)*S): M[16][8,12]:=MCF+BCF*(1-(C+De)*S):
>  M[16][8,5]:=S*(C+De)*(1-MCF):
>  M[17][8,8]:=1:  M[21][8,8]:=1:
>  M[18][8,6]:=1:
>  M[19][8,7]:=W3b:  M[19][8,6]:=M3b*(1-S): M[19][8,14]:=M3b*S:
>  M[19][8,15]:=B3b:
>  M[22][8,22]:=1:
>  M[23][8,8]:=1-BT: M[23][8,15]:=BT:
>  M[24][8,20]:=1:
>  M[25][8,23]:=1:
>  M[26][8,14]:=1-B2b:  M[26][8,20]:=B2b:
>  M[27][8,16]:=X+(1-X)*M2b:  M[27][8,13]:=(1-X)*W2b:
>  M[27][8,20]:=(1-X)*B2b:
>  M[28][8,14]:=1:
>  M[29][8,16]:=1:
>  M[30][8,16]:=1-MLF: M[30][8,13]:=MLF:
>  M[31][8,16]:=1-WCF: M[31][8,13]:=WCF:
>  M[32][8,14]:=1:
>  M[35][8,8]:=W: M[35][8,16]:=1-W:
>  M[36][8,7]:=1:
>  M[37][8,15]:=1-WT*F: M[37][8,11]:=WT*F:
>  M[38][8,15]:=1:
>  M[39][8,15]:=1:
>  M[40][8,8]:=1:
>  M[41][8,25]:=1:
>  for i from 1 to 43 do for k from 1 to 25 do if M[i][8,k]<>0
then if
>  k+8<=25 then M[i][16,k+8]:=M[i][8,k] else
>  M[i][16,25]:=M[i][16,25]+M[i][8,k] fi:
>  if k+16<=25 then M[i][24,k+16]:=M[i][8,k] else
>  M[i][24,25]:=M[i][24,25]+M[i][8,k] fi end if od  od:
```

```
>  M[9][24,22]:=B+De:  M[9][24,24]:=0:  M[7][16,14]:=1:  M[7][24,22]:=1:
>  M[7][16,13]:=0:  M[7][24,21]:=0:  M[10][16,15]:=1:  M[11][16,15]:=1:
>  M[10][16,13]:=0:  M[11][16,13]:=0:  M[10][24,21]:=0:  M[11][24,21]:=0:
>  M[10][24,23]:=1:  M[11][24,23]:=1:

>  M[5][8,3]:=1-S:  M[5][8,7]:=S:

>  M[8][16,18]:=(C+De)*S:  M[8][24,25]:=(C+De)*S+A+B:  M[8][16,13]:=0:
>  M[8][24,21]:=0:

>  M[15][16,14]:=1-WLF*F-BLF*S:  M[15][16,10]:=WLF*F:
>  M[15][16,22]:=BLF*S:  M[15][16,13]:=0:
>  M[15][24,22]:=1-WLF*F-BLF*S:  M[15][24,18]:=WLF*F:  M[15][24,25]:=BLF*S:
>  M[15][24,21]:=0:

>  M[16][16,14]:=WCF:  M[16][16,20]:=1-WCF:  M[16][24,22]:=WCF:
>  M[16][24,25]:=1-WCF:  M[16][16,13]:=0:  M[16][24,21]:=0:

>  M[17][24,22]:=1:  M[17][24,24]:=0:

>  M[38][24,24]:=1:  M[38][24,25]:=0:
```

The matrices have now all been filled in, under the assumption of not hit-and-running with an 11 on first. In the next lines, the matrices N1 and N2 are filled in with the outcomes of hit-and-running with a runner of first or first and third, respectively, and nobody out. For example, $N1[7,6] := 1-A$ says that if hit-and-running with a runner on first and the opposing pitcher is not an A, then 7 will put runners on first and third (base situation 6). The entry will be filled in with the numerical value that A= .0941.

```
>  N1:=matrix(43,25):  for k from 1 to 43 do
>  for j from 1 to 25 do N1[k,j]:=0 od od:
>  N1[1,1]:=1:  N1[2,9]:=1:  N1[42,9]:=1:  N1[3,4]:=1:
>  N1[4,6]:=1:  N1[5,6]:=1-F:  N1[7,6]:=1-A:  N1[8,6]:=C+De:
>  N1[9,6]:=De:  N1[16,6]:=1:  N1[17,6]:=1:  N1[31,6]:=1:
>  N1[40,6]:=1:  N1[41,6]:=1:  N1[5,2]:=F:  N1[6,3]:=1:
>  N1[43,3]:=1:  N1[7,11]:=A:  N1[8,11]:=A+B:  N1[9,11]:=1-De:
>  N1[10,11]:=1-De:  N1[11,11]:=A+B:  N1[12,11]:=1:  N1[18,11]:=1:
>  N1[20,11]:=1:  N1[24,11]:=1:  N1[26,11]:=1:  N1[27,11]:=1:
>  N1[28,11]:=1:  N1[29,11]:=1:  N1[30,10]:=1:  N1[32,10]:=1:
>  N1[10,7]:=De:  N1[11,7]:=C+De:  N1[13,3]:=1:  N1[21,9]:=1:
>  N1[22,9]:=1:  N1[23,9]:=1:  N1[35,9]:=1:  N1[37,9]:=1:
>  N1[39,9]:=1:  N1[14,3]:=1:  N1[38,3]:=1:  N1[15,5]:=1:
>  N1[19,5]:=1:  N1[25,17]:=1:  N1[33,17]:=1:  N1[34,17]:=1:
>  N1[36,4]:=1:
```

```
>  N2:=matrix(43,25):
>  for k from 1 to 43 do for j from 1 to 25 do N2[k,j]:=0
od od:
>  N2[1,1]:=1:  N2[2,4]:=1:  N2[42,4]:=1:  N2[3,3]:=1:
>  N2[6,3]:=1:  N2[43,3]:=1:  N2[4,6]:=1-F: N2[5,6]:=1:
>  N2[7,6]:=1-A: N2[8,6]:=C+De:  N2[11,6]:=C+De:  N2[9,6]:=De:
>  N2[10,6]:=De:  N2[15,6]:=1:  N2[16,6]:=1:  N2[7,11]:=A:
>  N2[17,6]:=1:  N2[31,6]:=1:  N2[8,11]:=A+B:  N2[11,11]:=A+B:
>  N2[9,11]:=1-De:  N2[10,11]:=1-De:  N2[24,11]:=1:  N2[25,11]:=1:
>  N2[26,11]:=1:  N2[28,11]:=1:  N2[12,15]:=1:  N2[19,15]:=1:
>  N2[20,15]:=1:  N2[27,15]:=1:  N2[29,15]:=1:  N2[13,7]:=1:
>  N2[14,7]:=1:  N2[36,7]:=1:  N2[18,8]:=1:  N2[21,11]:=1:
>  N2[37,11]:=1:  N2[40,11]:=1:  N2[22,3]:=1:  N2[38,3]:=1:
>  N2[39,3]:=1:  N2[23,19]:=1:  N2[30,14]:=1:  N2[35,14]:=1:
>  N2[32,17]:=1:  N2[33,20]:=1:  N2[34,20]:=1:  N2[41,25]:=1:
>  N2[4,2]:=F:
```

The next group of lines refills the matrices M[k] with $(1 - H)$ times their previous value plus H times the hit-and-run outcome from N1 and N2. When one or two are out, the numbers from N1 and N2 must be shifted appropriately. The line after that checks to make sure that the sum of the entries in each row is 1, to make sure that I haven't overlooked something. Because we are working with decimals, we don't require that it be exactly 1, but within .01 should be close enough.

```
>  for k from 1 to 43 do for j from 1 to 25 do
>  M[k][2,j]:=(1-H)*M[k][2,j]+H*N1[k,j]:
>  M[k][6,j]:=(1-H)*M[k][6,j]+H*N2[k,j] od:
>  for j from 9 to 24 do M[k][10,j]:=(1-H)*M[k][10,j]+H*N1[k,j-8]:
>  M[k][14,j]:=(1-H)*M[k][14,j]+H*N2[k,j-8] od:
>  for j from 17 to 24 do
>  M[k][18,j]:=(1-H)*M[k][18,j]+H*N1[k,j-16]:
>  M[k][22,j]:=(1-H)*M[k][22,j]+H*N2[k,j-16] od:
>  M[k][10,25]:=(1-H)*M[k][10,25]+H*add(N1[k,j],j=17..25):
>  M[k][18,25]:=(1-H)*M[k][18,25]+H*add(N1[k,j],j=9..25):
>  M[k][14,25]:=(1-H)*M[k][14,25]+H*add(N2[k,j],j=17..25):
>  M[k][22,25]:=(1-H)*M[k][22,25]+H*add(N2[k,j],j=9..25)
od:
>  for k from 1 to 43 do for i from 1 to 24 do
>  sm:=add(M[k][i,j],j=1..25):
>  if sm<.99 or sm>1.01 then print(k,i):
>  for j from 1 to 25 do
>  if M[k][i,j]<>0 then print(j,M[k][i,j]) fi od fi od od:
```

The next line determines the all-important average transition matrix M0, weighting all the matrices M[k] by the probability P[k] that k is rolled. The matrix Mt is obtained from this by deleting the last row and column, and IQ=I-Mt, where I is the 24-by-24 identity matrix.

```
>  M0:=matrix(25,25):
>  for i from 1 to 25 do for j from 1 to 25 do
>  M0[i,j]:=add(P[k]*M[k][i,j], k=1..43) od od:

>  Mt:=matrix(24,24):  Mp:=delrows(M0,25..25):  Mt:=delcols(Mp,25..25):

>  IQ:=matrix(24,24):
>  for i from 1 to 24 do for j from 1 to 24 do IQ[i,j]:=-Mt[i,j]
od od:
>  for i from 1 to 24 do IQ[i,i]:=1+IQ[i,i] od:
```

In the next two lines, we solve the matrix equation (5.2), in its transposed form which is mentioned just after (5.2). Then we form **pr**, by dividing by the sum of the entries. The values of these are listed; they also appear in Table 3.

```
>  e1:=Vector(1..24):  e1[1]:=1:
>  for i from 2 to 24 do e1[i]:=0 od:

>  pr1:=linsolve(transpose(IQ),e1):

>  pr:=Vector(1..24):    sisum:=add(pr1[i],i=1..24):
>  for i from 1 to 24 do pr[i]:=pr1[i]/sisum od:  evalm(pr);
```

$[0.2423890034, 0.06248443545, 0.01671058123, 0.001893042703, 0.009243561222,$
$0.005839090998, 0.004955699184, 0.002333018095, 0.1728442492,$
$0.07462130587, 0.03317918805, 0.007281017757, 0.01663683033,$
$0.01128006927, 0.01012563566, 0.005149180089, 0.1351818428,$
$0.07358632211, 0.04346591278, 0.01490600232, 0.01894785962,$
$0.01743811744, 0.01305482755, 0.006453206979]$

The next few lines are implementing the equation $\mathbf{b} = R + \mathbf{d} - \mathrm{No}$ which appears near the end of Section 5. What the computer calls No is U in the text; its values were filled in early in the program. The computer's Nov is the text's No.

```
>  Nov:=Vector(1..24):  for i from 1 to 24 do
>  Nov[i]:=add(P[k]*No[i,k],k=1..43) od:

>  BR:=proc(x) if x mod 8=0 then 3 elif x mod 8=1 then 0
>  elif x mod 8<5 then 1 else 2  end if end:

>  r:=matrix(24,24):  for i from 1 to 24 do for j from 1
to 24 do
>  r[i,j]:=BR(i)+iquo(i-1,8)-BR(j)-iquo(j-1,8)+1 od od:

>  R:=Vector(1..24):  for i from 1 to 24 do R[j]:=add(Mt[i,j]*r[i,j],
>  j=1..24) od:
```

```
>   dd:=matrix(25,43):
>   for i from 1 to 25 do for j from 1 to 43 do dd[i,j]:=0
od od:
>   dd[20,9]:=B+De:  dd[20,4]:=1:  dd[21,15]:=BLF:
>   dd[22,17]:=MRF: dd[22,20]:=1-F: dd[23,3]:=2:  dd[23,17]:=1-F:
>   dd[24,16]:=2*(1-WCF):
>   dd[23,8]:=S*(1-B): dd[21,18]:=S: dd[21,16]:=1-MCF: dd[18,2]:=H:

>   d:=Vector(1..24):  for j from 1 to 24 do
>   d[j]:=add(P[k]*dd[j,k],k=1..43) od:  evalm(d);
```

$[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.03096663523,$
$0.002994668877, 0.005375655278, 0.01911716661, 0.005500995418]$

```
>   b:=Vector(1..24):  for i from 1 to 24 do b[i]:=R[i]-Nov[i]+d[i]
>   od:evalm(b);
```

$[0.01318730690, 0.06755231178, 0.2022774308, 0.5362808575, 0.2337684441,$
$0.6647757428, 0.6159840615, 0.8929346851, 0.01318730690, 0.06755231178,$
$0.2022774308, 0.5428298960, 0.2337684441, 0.5398836088, 0.6159840615,$
$0.7887469993, 0.01318730690, 0.06643071418, 0.1941808881, 0.2712750401,$
$0.2494084088, 0.2959106829, 0.4145913099, 0.5462440625]$

The next line implements (5.3), and the final line of the program
implements the last equation of Section 5.

```
>   E:=linsolve(IQ,b);
```

$E := [0.4327328120, 0.8071345411, 0.9833401650, 1.205260236, 1.329354858,$
$1.627401708, 1.747861964, 2.172555469, 0.2253746362, 0.4737191857,$
$0.6055246798, 0.8693486467, 0.8287713991, 1.051854981, 1.192384015,$
$1.470533299, 0.07916201079, 0.1891991804, 0.2861697727, 0.3481051674,$
$0.4042025272, 0.4450268558, 0.5278651998, 0.7097821340]$

```
>   V:=Vector(1..43):  for k from 1 to 43 do
>   V[k]:=add(pr[i]*(add(M[k][i,j]*(r[i,j]+E[j]-E[i]-No[i,k]),i=1..24)+M[k
>   ][i,25]*(dd[i,k]-E[i])),i=1..24) od:  evalm(V);
```

[1.410111439, 0.9842784188, 0.9349642877, 0.8699303615, 1.002306045, 0.7723095383, 0.4708125788, 0.2256815027, 0.09041707370, 0.3534896890, 0.5475563823, −0.2678709132, −0.2316908951, 0.2596733634, 0.2532921455, 0.08745100579, 0.4044730484, 0.3353422988, 0.3264475335, 0.3174884833, 0.3691107363, 0.1669102842, 0.03464534326, −0.3026386221, −0.3362869255, −0.2158365730, −0.2297909098, −0.1860367963, −0.2280979868, −0.2304891868, −0.2195396500, −0.2296933543, −0.2483663980, −0.2483663980, −0.2326765559, 0.1055085711, −0.08697165441, 0.03737658931, −0.1742790794, −0.005117658231, 0.2016673951, 0.9620400826, 0.7020508705]

Department of Mathematics, Lehigh University, Bethlehem, PA 18015, USA
E-mail address: `dmd1@lehigh.edu`