An analytic model for per-inning scoring distributions

By Keith Woolner

Introduction

One of the most invaluable sabermetric tools for analyzing strategic decisions has been the base-out expected runs matrix. Thorn & Palmer presented such a table in their classic book The Hidden Game Of Baseball. The table shows every baserunner situation (bases empty, runner on first, bases loaded, etc) and number of outs, and shows the average number of runs teams scored in actual games from that situation onward through the rest of the inning. Using such a tool enables one to calculate average break-even success rates for attempting stolen bases, the average value of the sacrifice hit, the costliness of a double play, and so on.

As useful as such as table is, there are limitations. The data are compiled from the play-by-play accounts of actual games, and as such are difficult for the ordinary fan to compute and independently verify. The results are applicable only to the average or typical team in the data set – as offense levels change, so does the base-out expected runs table, though it's not obvious exactly what the changes are. Such a table also can't be of help when considering the differences between average hitting teams and great/poor hitting teams. The Cleveland Indians have a higher expected run total for any base-out situation than do the Minnesota Twins, but you can't assess the magnitude of the difference from the table itself.

In addition, expected runs alone do not tell the whole story. Going into the bottom of the ninth inning, down by 2 runs, the key piece of information is not the expected number of runs, but the probability of scoring at two (to tie) or more (to win) runs in a single inning. The emphasis on end-game strategies involving closers, pinch hitters, pinch runners, and the like derives from the fact that all runs are not equal, when you have specific information about the state of the game.

The next step in the evolution of tools to analyze strategic decisions would be to have an analytic framework from which teams of differing offensive and defensive strengths could be studied. In addition, explicit consideration of the probability distributions of various outcomes would enable more sophisticated treatments of strategy. With these needs, in mind, I've attempted to develop new tools to aid in such analyses.

Analyzing the Per Inning Run Scoring Historical Record

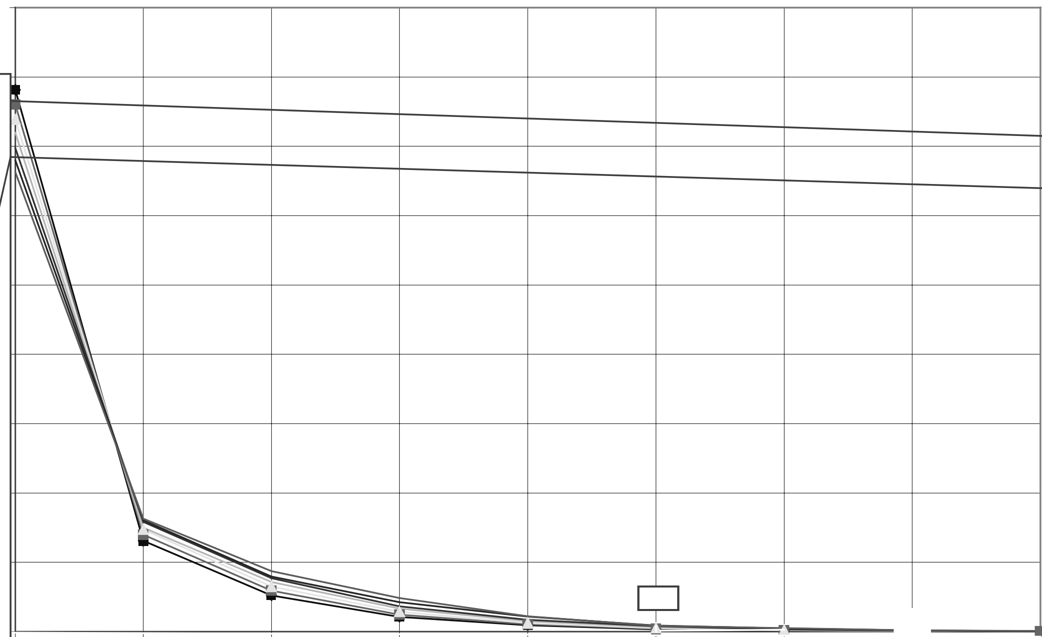
My first goal was to develop a formula for estimating the probability of a team scoring a certain number of runs in an inning (the Runs Per Inning, or RPI formula). The formula would take into account the overall offensive strength of the team.

First, we needed empirical data to verify a model against. To that end, I collected inning by inning scoring data for all teams between 1980 and 1998 (courtesy of Retrosheet and Total Sports/Baseball Workshop). I categorized each team by their season's average runs-per-game (3.0-3.5, 3.5-4.0, 4.0-4.5, etc.), and determined the number of times the teams in each category scored 1, 2, 3, … runs per inning. In the following, every notation of X runs per game refers to the class of teams scoring between X and (X + 0.5) runs/game. The results are shown in the following two tables:

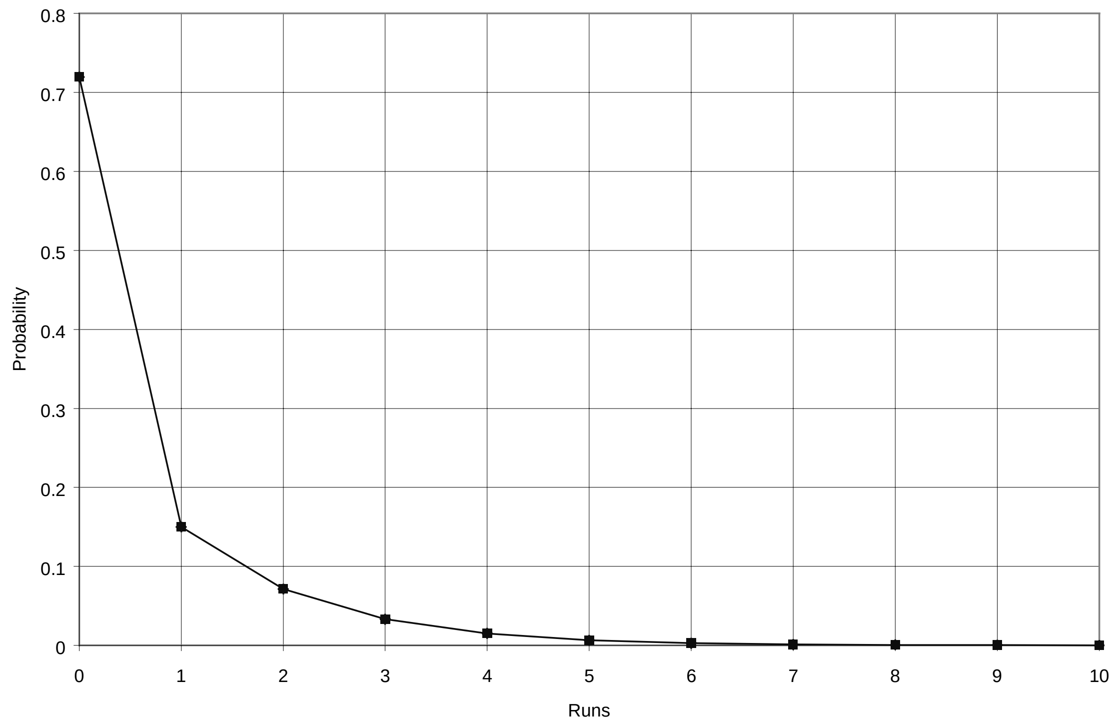
Frequency	**Runs/Inning**																	
Average Runs/Game	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Total**
3.0	11821	1978	790	323	141	46	18	11	1									15129
3.5	90333	16590	7022	2904	1254	484	174	55	24	13	3		2		1			118859
4.0	191653	38186	16891	7452	3207	1291	535	206	90	40	10	3		3				259567
4.5	151833	31632	15073	6990	3190	1368	573	205	97	30	11	13	3	2				211020
5.0	53400	12136	5897	2794	1326	594	264	107	45	18	10	4		2				76597
5.5	17560	4148	2049	1091	563	229	113	50	15	6	2	5					1	25832
6.0	1628	400	214	119	55	21	12	5	2									2456
Total	**518228**	**105070**	**47936**	**21673**	**9736**	**4033**	**1689**	**639**	**274**	**107**	**36**	**25**	**5**	**7**	**1**	**0**	**1**	**709460**

Probability	**Runs/Inning**																
Average Runs/Game	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
3	78.1%	13.1%	5.2%	2.1%	0.9%	0.3%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3.5	76.0%	14.0%	5.9%	2.4%	1.1%	0.4%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4	73.8%	14.7%	6.5%	2.9%	1.2%	0.5%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4.5	72.0%	15.0%	7.1%	3.3%	1.5%	0.6%	0.3%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	69.7%	15.8%	7.7%	3.6%	1.7%	0.8%	0.3%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5.5	68.0%	16.1%	7.9%	4.2%	2.2%	0.9%	0.4%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
6	66.3%	16.3%	8.7%	4.8%	2.2%	0.9%	0.5%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Total	**73.0%**	**14.8%**	**6.8%**	**3.1%**	**1.4%**	**0.6%**	**0.2%**	**0.1%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**

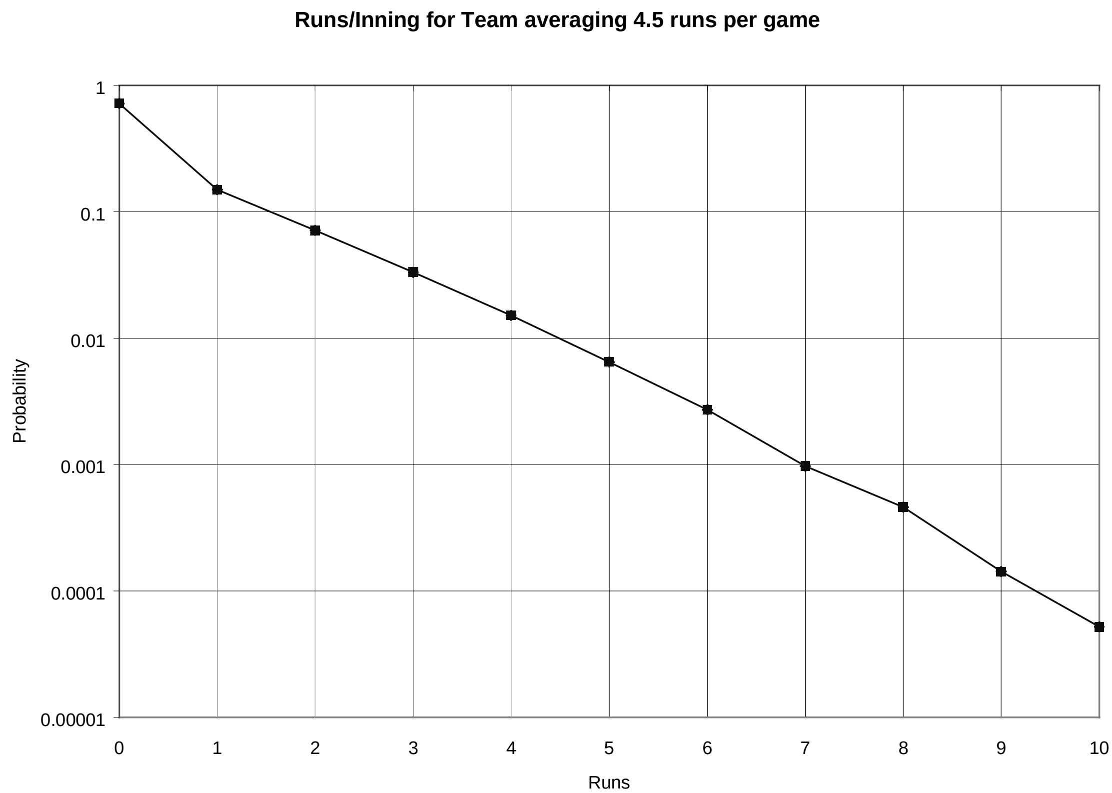
Having determined the empirical distributions, what we then needed was an RPI formula to predicting the likelihood that a team who averages X runs per game scores Y runs in an inning. Start by graphing the results to get a visual representation of what the distribution looks like:



Runs/Inning for Team averaging 4.5 runs per game

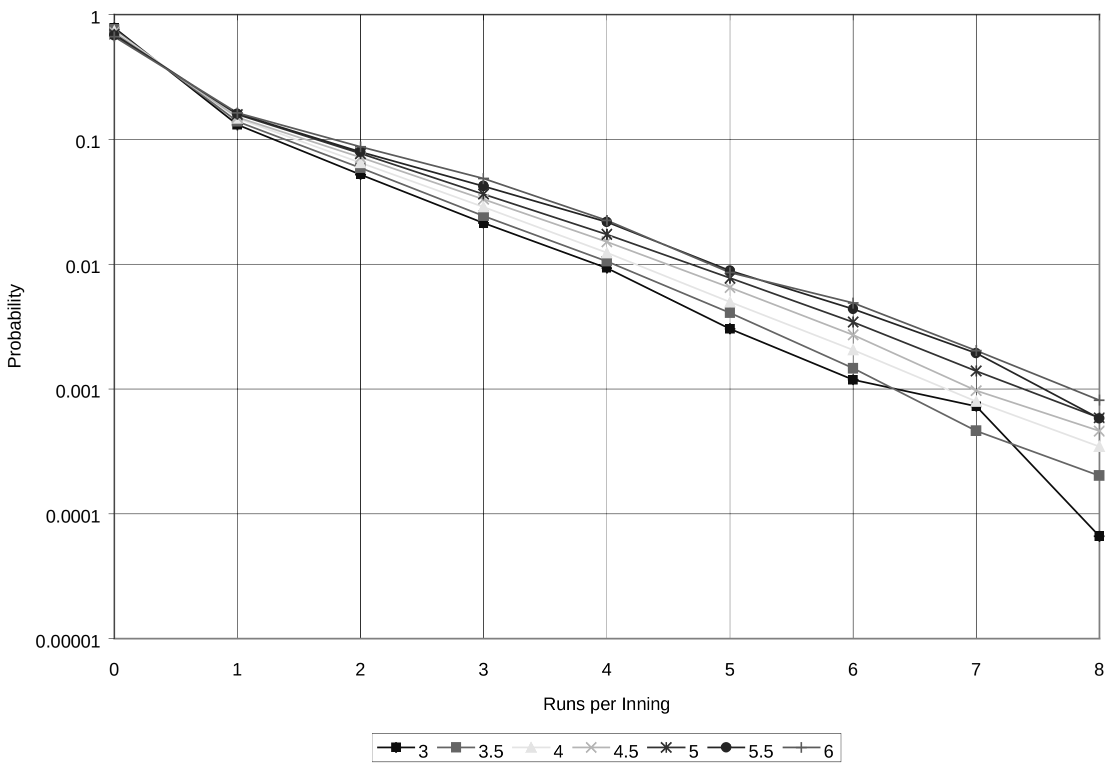


Though the run scoring distribution is a discrete distribution (it's impossible to score fractional numbers of runs per inning), there's a strong resemblance between it and an exponential distribution, which take the form Prob{team scores Y runs in an inning} = $c*e^{-kY.}$ For such a distribution, plotting the probabilities on a logarithmic graph should yield a straight line.



The results are encouraging – that's a pretty straight line (other than a slightly more negatively sloped portion between 0 and 1). An exponential approximation to this distribution will likely be a good one. Let's graph the remaining team categories similarly:

Runs/Inning for Team by Average R/G



We're on to something good. All of the team categories show basically log-linear trends in the likelihood of per-inning scoring, at least in the 1 - 6 runs/inning levels. The zero point is still slightly higher than a pure log-linear model would suggest. And at 7+ runs per inning, there's some noisiness in the results, which is reasonable to expect since the frequency of these events is so rare. In the 19 years worth of, innings with 7 or more runs scored amount to less than 0.16% of the innings played. The overall trends still appear to be linear, though with a slight bit of negative acceleration at higher run levels. A model that accurately models the 0-6 runs per inning range, and extrapolates reasonable for higher levels of scoring will still be a useful tool for our purposes.

The Runs Per Inning Scoring Formula

We've now identified a few characteristics of the RPI formula:

- Takes two variables – the average runs per game for the team, and the particular number of runs per inning for which we want to estimate the chance of the team scoring
- Basically exponential in form
- Higher probability of zero runs than pure exponential
- Slightly faster decline in probability for high runs/inning (> 7)
- Accurately matches empirical runs/inning distribution data
- Accurate predicts runs/game distribution

The following exponential formula meets the first two items of our criteria:

Probability ~= $e^{(mA+pR+n(R/A)+b)}$

where A is the average runs/game, and R is the specific Runs/Inning. The values of m, p, n, and b are constant coefficients.

Let's examine the RPI formula in depth. It's an exponential, as we desired, with the exponent dependent on the parameters A and R. The *mA +pR* portion controls most of the slope of the log-linear of the line. The *n(R/A)* is be a correction term that slightly increases the rate of decline for as R gets large. The values of the coefficients will be determined through a regression analysis that will give us the best possible fit to the actual data. The end result will do well in approximating the run scoring in the >= 1 run portion of the graph.

Which still leaves the very significant problem of determining the probability of scoring zero runs, since the formula does not do a good job of computing it if you just plug in zero for the value of R. Again, the reason is that the historical data on the log-linear graph above shows that the probability of zero is higher than we'd expect by a straight line. However, we do have a way out of the dilemma. We can simply define the probability of zero as the remaining probability after we sum the probabilities of all values of R greater than 1. As we will see, this solution is more than adequate.

The range of R deserves a little attention. While in principle R should be able to range as large as we want, the validity of calculating the probability of a team scoring, say, 20 runs in an inning is questionable. The formula and coefficients were validated against the existing history, and the incidence of scoring 9 or more runs is vanishingly small, typically less than .04% for all values of R greater than 9 combined. As a practical limit, the odds of scoring more than 8 runs per inning is virtually zero, and in any event, the discrepancies between actual and computed values in the lower run scoring range (0, 1, and 2 runs/inning) are larger than the entire effect of the R>9. Given the difficulties in validating run scoring above this level, the miniscule effect on practical strategic assessments, and the rarity of the events themselves, it makes sense to set a maximum value of R that we will explore, which we will denote as $\mathbf{R_{max}}$ in performing the calculations. Setting R_{max} in the range of 8, 9, or 10 will capture the overwhelming and significant portion of the per-inning distribution of team run scoring.

With the incorporation of the probability of zero, and the range of R, our formula now looks like:

Probability of a team that averages **A** runs per game scores **R** runs in a given inning =

$$\Pr\{A,R\} = \left[\begin{array}{ll} \text{if R > 0 then} & e^{(mA+pR+n(R/A)+b)} \\ \text{If R = 0 then} & 1-\sum_{r=1}^{R_{max}} e^{(\text{mA+pr+n(r/A)+b})} \end{array}\right]$$

There's one last step we can take to improve accuracy. We know what the per-game average scoring is supposed to be, as it's one of the parameters to the formula. We can also determine the expected runs/inning the formula predicts as follows:

$$Expected\_RPG(A) = \sum_R R * \Pr\{A,R\}$$

The coefficients used define the proper shape, but they are also best suited for the range of scoring closest to the median of actual team run scoring (A ~= 4.5 RPG). At the extremes, the approximation gives the proper shape, but loses some accuracy in the expected level of scoring. We can remedy this by scaling the probabilities for positive run scoring totals by the ratio between the actual scoring (A) and the predicted scoring (Expected_RPG). Thus our final formula (including the values for the coefficients) is:

Probability of a team that averages **A** runs per game scores **R** runs in a given inning =

$$\Pr\{A,R\} = \Bigg[\quad \text{if R > 0 then } \mathrm{C} * \mathrm{e}^{(mA+pR+n(R/A)+b)}$$

$$\text{If R = 0 then } 1-\mathrm{C}*\sum_{r=1}^{R_{\max}} \mathrm{e}^{(\mathrm{mA+pr+n(r/A)+b})}$$

$$\text{Where } \mathrm{C} = \frac{A}{9\sum_{r=1}^{R_{\max}} r * \mathrm{e}^{(mr+pr+n(r/A)+b)}}$$

and
m =-0.01219
n = -1.813
p = -0.3865
b = -1.042

]

It took us awhile to get there, and it looks pretty hairy, but the real question is how well it works. We'll explore that in the next section.

Expected vs. Actual Runs

The real test of an estimation tool is in how well is estimated what it purports to measure. In the case of the RPI formula, there are two independent parameters to consider, the team's average run scoring (strength of offense), and the specific number of runs for which we want to know the chance of the team scoring in an inning. The tables below show the results for team strengths between 3.0 and 6.0 runs per game, and for per-inning scoring between zero and eight runs. Note that while only 1 decimal place is shown, none of the values are actually zero, just less than 0.1%:

Average Scoring:		**3**
Runs	**Actual**	**Estimated**
0	78.1%	79.0%
1	13.1%	13.2%
2	5.2%	4.9%
3	2.1%	1.8%
4	0.9%	0.7%
5	0.3%	0.3%
6	0.1%	0.1%
7	0.1%	0.0%
8	0.0%	0.0%

Average Scoring:		**3.5**
Runs	**Actual**	**Estimated**
0	76.0%	76.9%
1	14.0%	13.8%
2	5.9%	5.6%

3	2.4%	2.3%
4	1.1%	0.9%
5	0.4%	0.4%
6	0.1%	0.2%
7	0.0%	0.1%
8	0.0%	0.0%

Average Scoring:		**4**
Runs	**Actual**	**Estimated**
0	73.8%	74.7%
1	14.7%	14.3%
2	6.5%	6.2%
3	2.9%	2.7%
4	1.2%	1.2%
5	0.5%	0.5%
6	0.2%	0.2%
7	0.1%	0.1%
8	0.0%	0.0%

Average Scoring:		**4.5**
Runs	**Actual**	**Estimated**
0	72.0%	72.7%
1	15.0%	14.9%
2	7.1%	6.8%
3	3.3%	3.1%
4	1.5%	1.4%
5	0.6%	0.6%
6	0.3%	0.3%
7	0.1%	0.1%
8	0.0%	0.1%

Average Scoring:		**5**
Runs	**Actual**	**Estimated**
0	69.7%	70.7%
1	15.8%	15.4%
2	7.7%	7.3%
3	3.6%	3.5%
4	1.7%	1.6%
5	0.8%	0.8%
6	0.3%	0.4%
7	0.1%	0.2%
8	0.1%	0.1%

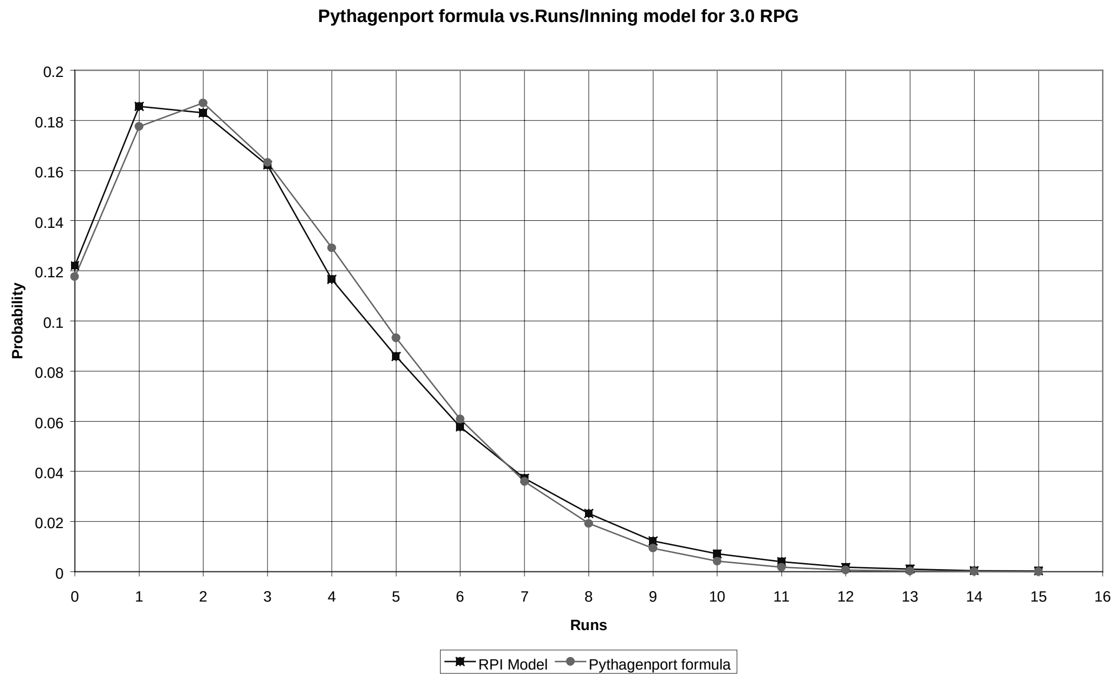
Average Scoring:		**5.5**
Runs	**Actual**	**Estimated**

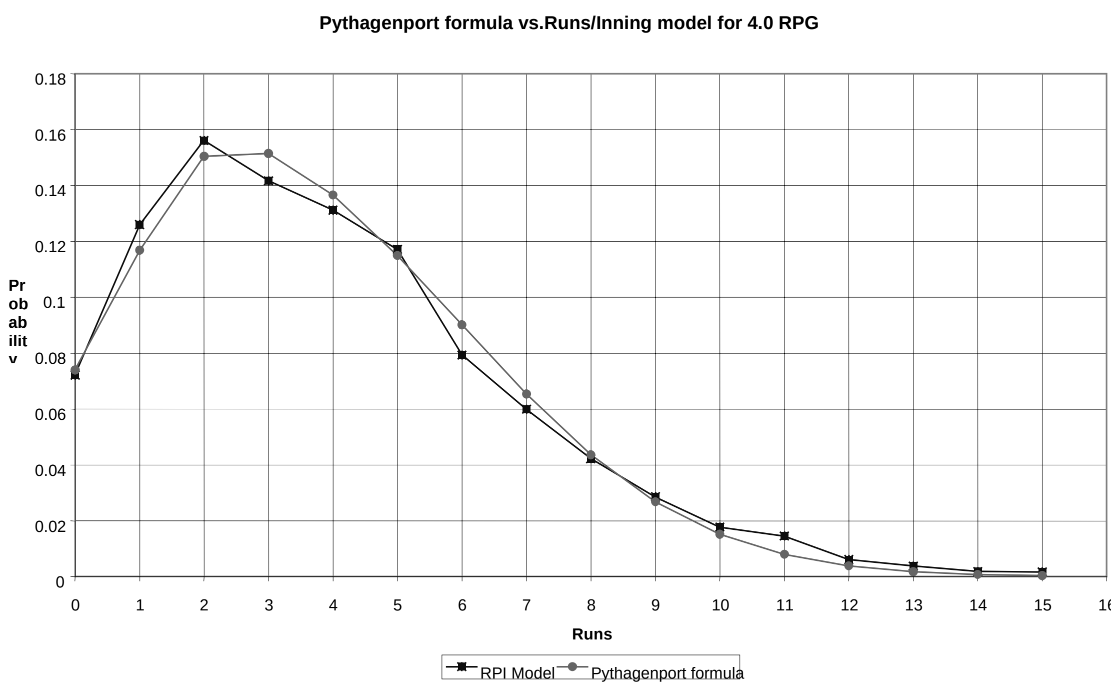
0	68.0%	68.7%
1	16.1%	16.0%
2	7.9%	7.8%
3	4.2%	3.8%
4	2.2%	1.9%
5	0.9%	0.9%
6	0.4%	0.4%
7	0.2%	0.2%
8	0.1%	0.1%

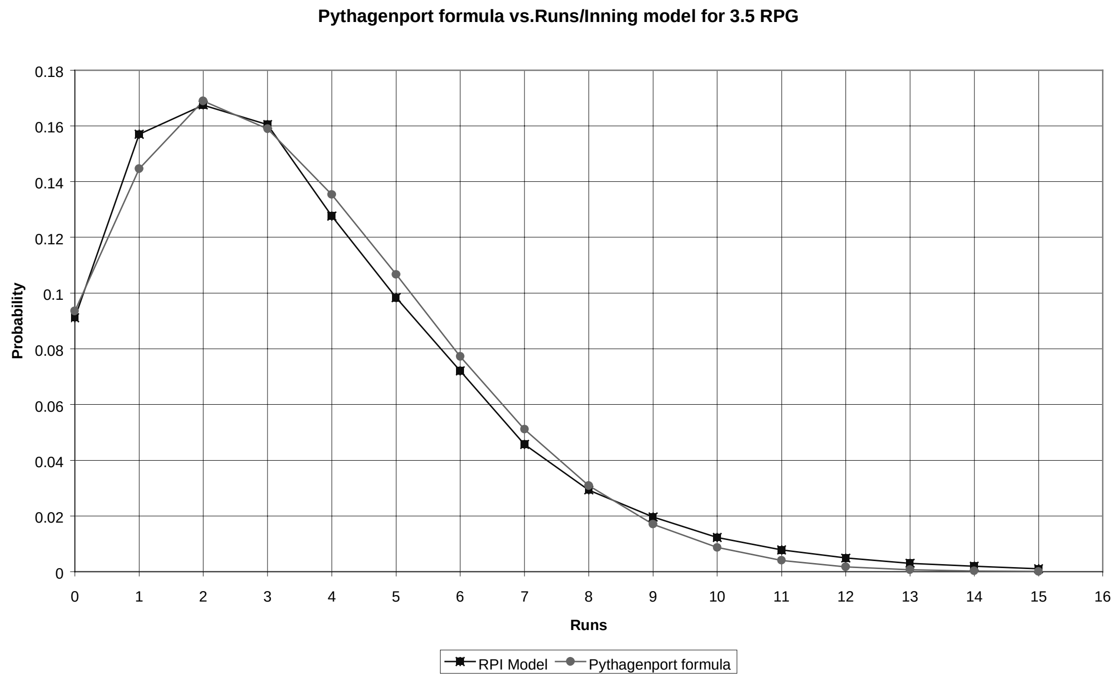
Average Scoring:		**6**
Runs	**Actual**	**Estimated**
0	66.3%	66.8%
1	16.3%	16.5%
2	8.7%	8.3%
3	4.8%	4.2%
4	2.2%	2.1%
5	0.9%	1.1%
6	0.5%	0.5%
7	0.2%	0.3%
8	0.1%	0.1%

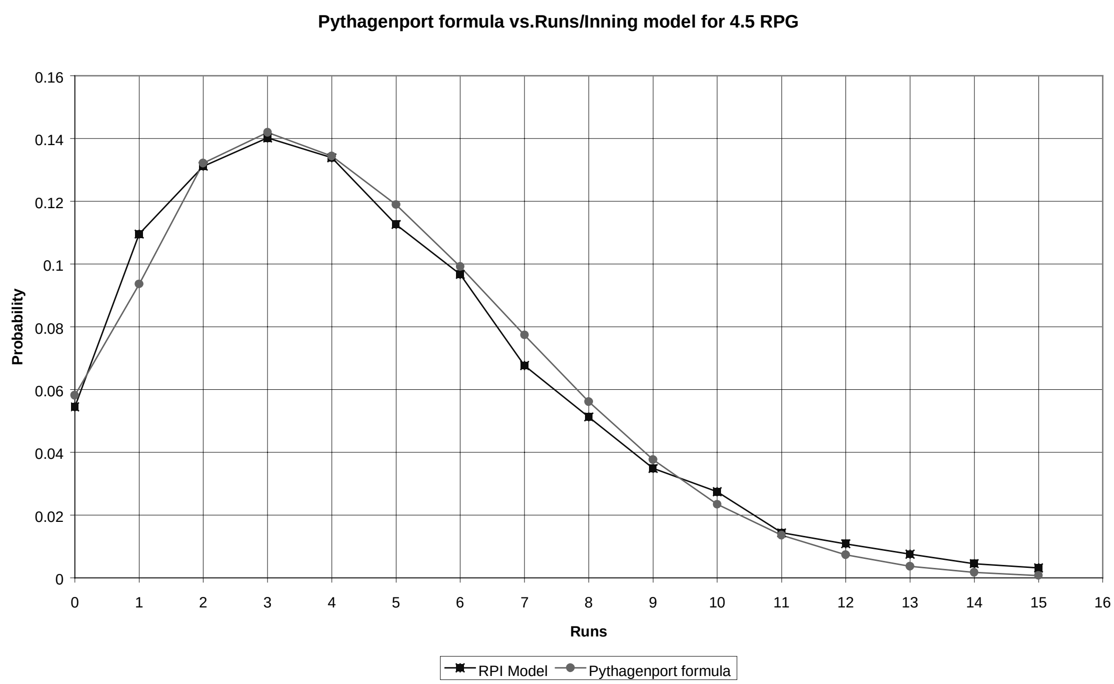
The RPI formula works very well in predicting the chances of scoring across a variety of teams. Of course, since it was the data set that the model was developed against, one would hope for a reasonably solid fit. An interesting and useful question is whether the distribution for each inning leads to an accurate prediction for the distribution of full game run scoring. Thanks to Clay Davenport's work last summer expanding on Bill James's "Pythagorean theorem" for team winning percentage, we have a framework for estimating the distribution of run scoring given a team's overall average level of offense. In lieu of an official designation for this work, I've taken to calling it the "Pythagenport" formula (if you've got a better name, send it in).

We can compare the Pythagenport formula to the RPI formula, using a probabilistic tool like Analytica, from Lumina Decision Systems. Using Analytica, we can quickly simulate a large number of games using the RPI model, and plot the frequency of each game result. Below, we've charted the RPI results against those of the Pythagenport method for various team strengths:

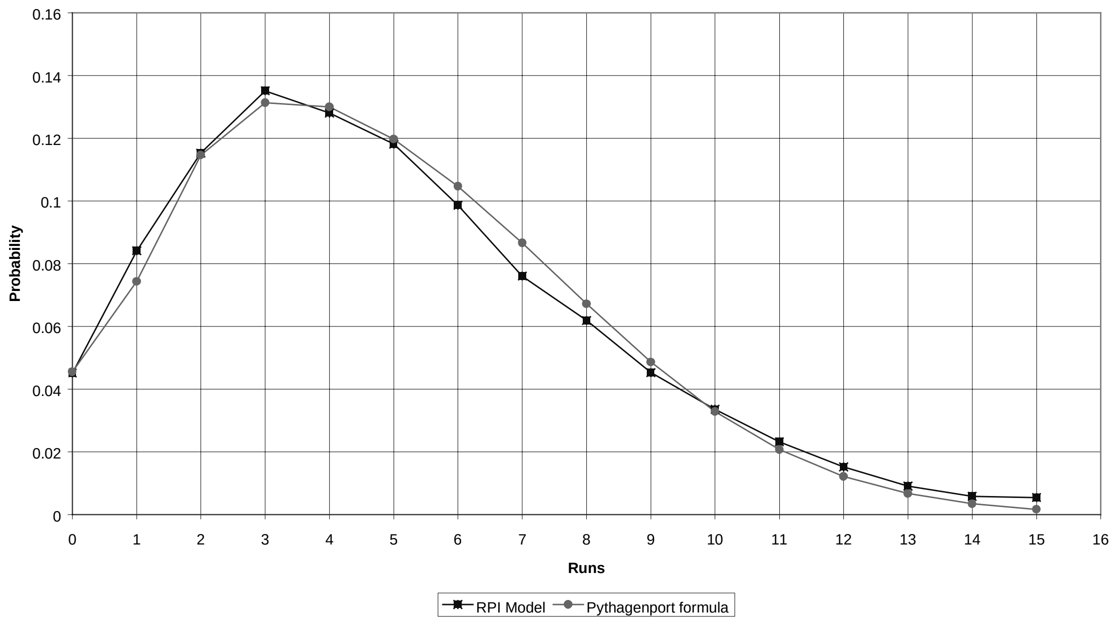

Pythagenport formula vs.Runs/Inning model for 3.0 RPG
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Probability
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Runs
RPI Model
Pythagenport formula


Pythagenport formula vs.Runs/Inning model for 4.0 RPG
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Pr
ob
ab
ilit
y
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Runs
RPI Model
Pythagenport formula

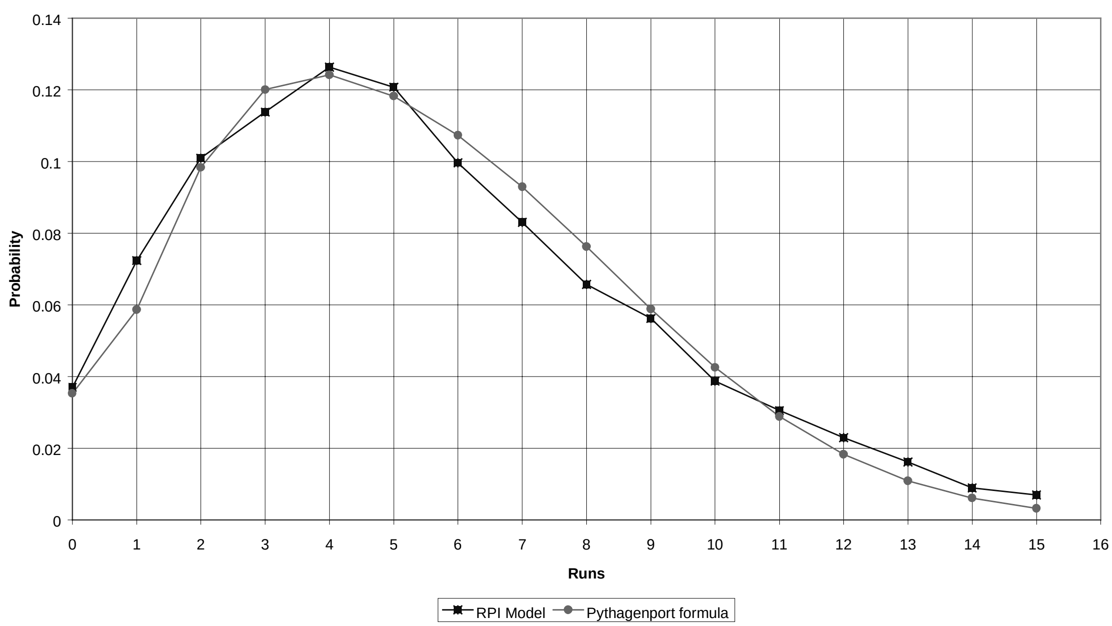

Pythagenport formula vs.Runs/Inning model for 3.5 RPG
Probability
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Runs
RPI Model
Pythagenport formula

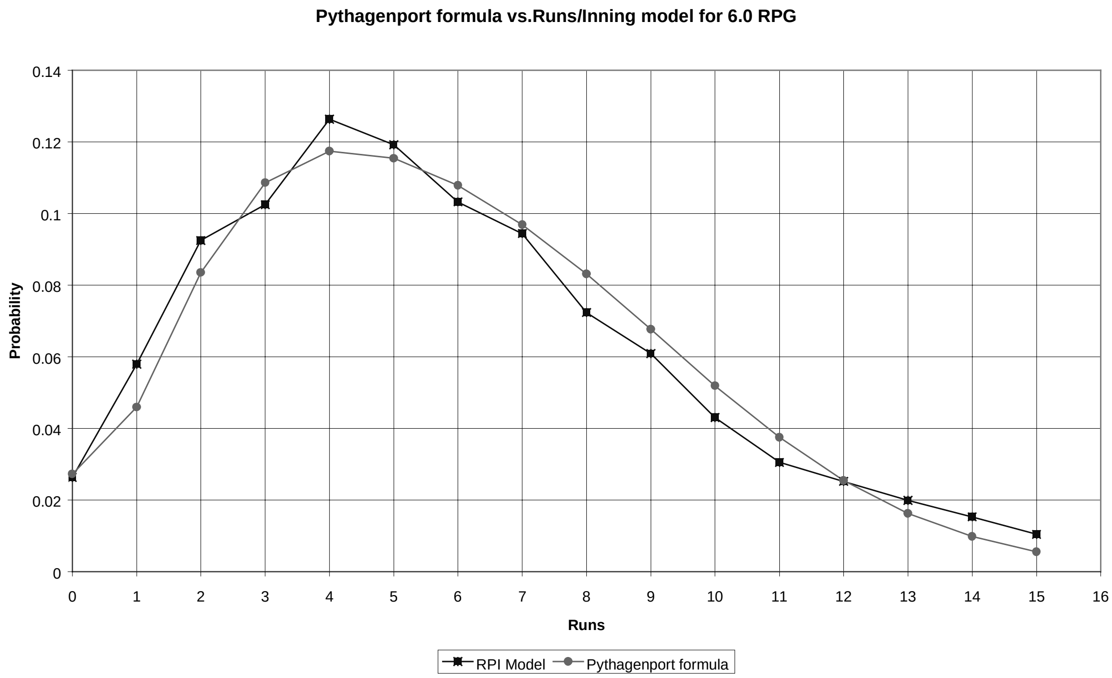

Pythagenport formula vs.Runs/Inning model for 4.5 RPG
Probability
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Runs
RPI Model
Pythagenport formula

Pythagenport formula vs.Runs/Inning model for 5.0 RPG



Pythagenport formula vs.Runs/Inning model for 5.5 RPG





The results are encouraging. The RPI model accurately estimates the overall frequency of per-game run scoring, as verified by an independently created methodology. There are some small discrepancies, mostly due to the fact that RPI models 9 innings of scoring that are independent from one another (no factors beyond the team influence the run scoring distribution), whereas actual games have a strong dependency in the form of the opposing pitcher. If you fail to score in the first inning, you are slightly less likely to score in the rest of the game, because it's slightly more likely that you're facing a Pedro Martinez or Randy Johnson than facing Jeff Fassero or Jamie Navarro. Conversely, you're more likely to score in the first against a bad pitcher, which increases your chances of scoring in subsequent innings. The net effect is that the assumption of independence used in the RPI model isn't entirely accurate, but doesn't affect the results too much, as you can see in the charts above.

The Perfect Closer

An interesting use of the RPI model is to investigate specific baseball strategies. One such example is the use of a closer like Dennis Eckersley. How much benefit does a team actually derive from such a closer? And how much of the value depends on the strength of the team's offense?

Since relief pitchers (even closers) are used in a variety of ways, some specificity is desirable. Let's describe an idealized, hypothetical "perfect" closer as follows:

- He comes in only in to start the 9th inning, and only to protect a 1-3 run lead
- He does not pitch in extra innings or in tie games.
- He never enters the game in the 8th inning or earlier
- He can work as often as necessary (no rest days are required)
- He works exactly 1 inning, and never surrenders a run

This simplified version of a LaRussa-style closer is easy to model, and thus we can simulate games to ascertain exactly how much benefit such a closer would be over the course of a season. We will assume in all the following examples that only the home team employs the perfect closer (the visiting team has no such advantage).

First, we need our base case, which is the performance of the home team without the benefit of the closer. The table below shows the chances of the home team winning depending on the strength of their offense and that of the opposition:

Probability of home team winning as a function of home and visitor RPG

	VIS						
HOM	3	3.5	4	4.5	5	5.5	6
3	49.9%	44.3%	39.0%	34.5%	30.3%	26.8%	23.3%
3.5	55.9%	49.9%	44.8%	40.1%	35.8%	31.9%	28.3%
4	61.1%	55.2%	49.9%	45.3%	40.8%	36.4%	32.8%
4.5	65.6%	59.8%	54.7%	49.9%	45.3%	41.2%	37.4%
5	69.5%	64.2%	59.4%	54.4%	50.0%	45.9%	41.4%
5.5	73.3%	68.5%	63.0%	58.6%	54.2%	49.7%	46.0%
6	76.6%	71.5%	67.3%	62.6%	58.6%	54.1%	50.0%

(Note that since the games are being simulated, the exact figure of 50.0% for teams of equal strength isn't perfectly touched, though 49.9% is sufficiently close for our purposes!)

Now, of all the games that are played, only a fraction will meet the criteria for the perfect closer to be used (protecting a 1-3 run lead in the 9th inning). We can have Analytica track the percentages for us:

Probability of home team using a closer, as a function of home and visitor average RPG

	HOME						
VIS	3	3.5	4	4.5	5	5.5	6
3	30.9%	31.7%	32.4%	32.7%	32.1%	31.5%	30.6%
3.5	27.8%	29.1%	30.1%	30.4%	30.4%	29.7%	29.1%
4	25.5%	26.5%	27.9%	28.2%	28.1%	28.8%	28.4%
4.5	22.5%	24.1%	25.4%	26.1%	26.3%	26.9%	26.6%
5	20.2%	21.9%	23.3%	24.3%	25.2%	25.6%	25.7%
5.5	18.0%	19.7%	21.2%	22.3%	23.1%	23.5%	24.3%
6	16.1%	17.8%	19.6%	20.8%	21.4%	22.4%	23.0%

Expected number of appearances (saves) by a perfect closer

	HOME						
VIS	3	3.5	4	4.5	5	5.5	6
3	50.06	51.35	52.49	52.97	52.00	51.03	49.57
3.5	45.04	47.14	48.76	49.25	49.25	48.11	47.14
4	41.31	42.93	45.20	45.68	45.52	46.66	46.01
4.5	36.45	39.04	41.15	42.28	42.61	43.58	43.09
5	32.72	35.48	37.75	39.37	40.82	41.47	41.63
5.5	29.16	31.91	34.34	36.13	37.42	38.07	39.37
6	26.08	28.84	31.75	33.70	34.67	36.29	37.26

Two results are clear: First, closers are used more often when teams are closely matched (this is unsurprising). Second, even among well-matched teams, closer are used less often as scoring levels increase. You can see this in the diagonal, where even teams who average 3 RPG have a closing opportunity about 31% of the time, whereas even teams averaging 6 RPG only do so 23% of the time.

Now, due to the rules we set up for the perfect closer, he will always close out the game without allowing a run. So the winning percentage for games where the closer is used is 100%, and the closer will have the same number of saves as appearances (as shown in the 2nd chart). Of course, since we're talking about a team leading in the 9th inning, there's a good chance they'd win the game regardless. The real question is how much more often does the team win than if they didn't employ a perfect closer:

Overall probability of home team winning when a perfect closer is available

HOM \ VIS	3	3.5	4	4.5	5	5.5	6
3	52.2%	46.7%	41.9%	37.4%	33.4%	29.4%	26.1%
3.5	57.8%	52.5%	47.2%	42.6%	38.5%	34.4%	30.8%
4	62.8%	57.5%	52.3%	47.3%	43.3%	39.0%	35.3%
4.5	67.4%	62.2%	57.0%	52.5%	48.1%	43.7%	40.2%
5	71.1%	66.3%	61.4%	57.0%	52.5%	48.6%	44.1%
5.5	74.7%	70.0%	65.6%	61.0%	57.0%	52.7%	48.4%
6	77.9%	73.3%	68.8%	64.7%	60.5%	56.4%	52.3%

Overall increase in home team winning percentage when a perfect closer is available

HOM \ VIS	3	3.5	4	4.5	5	5.5	6
3	2.3%	2.4%	2.9%	2.9%	3.1%	2.6%	2.8%
3.5	1.9%	2.6%	2.4%	2.5%	2.7%	2.5%	2.5%
4	1.7%	2.3%	2.4%	2.0%	2.5%	2.6%	2.5%
4.5	1.8%	2.4%	2.3%	2.6%	2.8%	2.5%	2.8%
5	1.6%	2.1%	2.0%	2.6%	2.5%	2.7%	2.7%
5.5	1.4%	1.5%	2.6%	2.4%	2.8%	3.0%	2.4%
6	1.3%	1.8%	1.5%	2.1%	1.9%	2.3%	2.3%

A typical team might expect to see about a 2.5% advantage, or about 25 points on their winning percentage. Over the course of the season, that amounts to about 4 games in the standings:

Expected increase in wins for home team employing a perfect closer (over 162 games)

HOM \ VIS	3	3.5	4	4.5	5	5.5	6
3	3.73	3.89	4.70	4.70	5.02	4.21	4.54
3.5	3.08	4.21	3.89	4.05	4.37	4.05	4.05
4	2.75	3.73	3.89	3.24	4.05	4.21	4.05
4.5	2.92	3.89	3.73	4.21	4.54	4.05	4.54
5	2.59	3.40	3.24	4.21	4.05	4.37	4.37
5.5	2.27	2.43	4.21	3.89	4.54	4.86	3.89
6	2.11	2.92	2.43	3.40	3.08	3.73	3.73

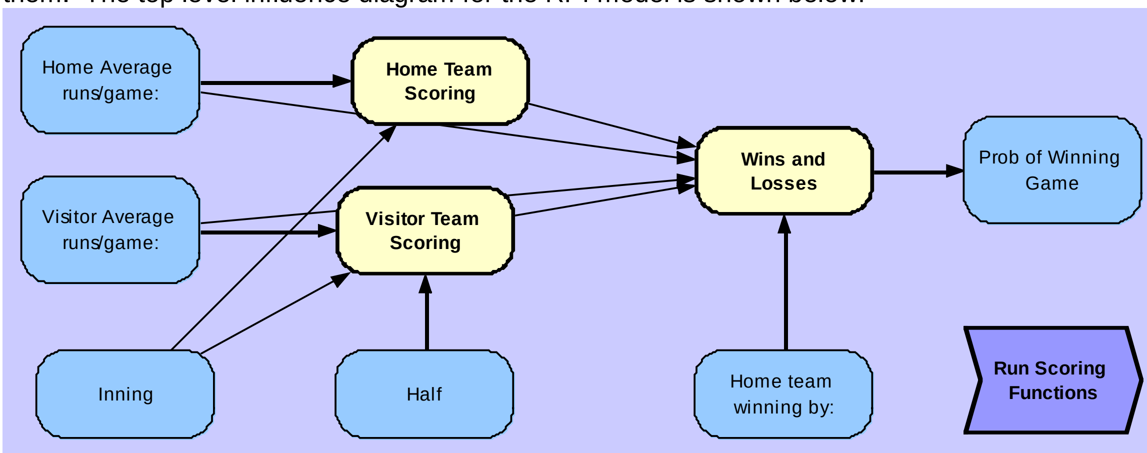
The interesting result is that even a perfect closer is only worth four to four and a half games over the course of a season, holding everything else constant. While it's always been recognized that closers work high-leverage situations, but pitch fewer innings, it's been difficult to directly assess how much help they are to a team. Using a tool like Analytica, in combination with probabilistic models like the RPI model, can start to help tackle questions like this is greater detail.

Modeling Game With Analytica

The analysis for this article was done using Analytica, a modeling tool from Lumina Decision Systems (who also employs the author of this article). I'll briefly describe the process of building the Runs Per Inning game model, and what Analytica allows you to do with it.

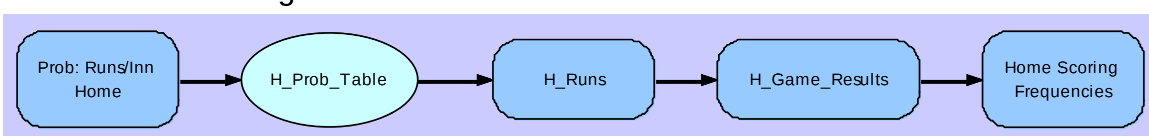
The Runs Per Inning Game Resolution Model

Models are represented with influence diagrams, which contain nodes representing key inputs and results in the model, and arrows which connect nodes to show the relationships between them. The top level influence diagram for the RPI model is shown below:

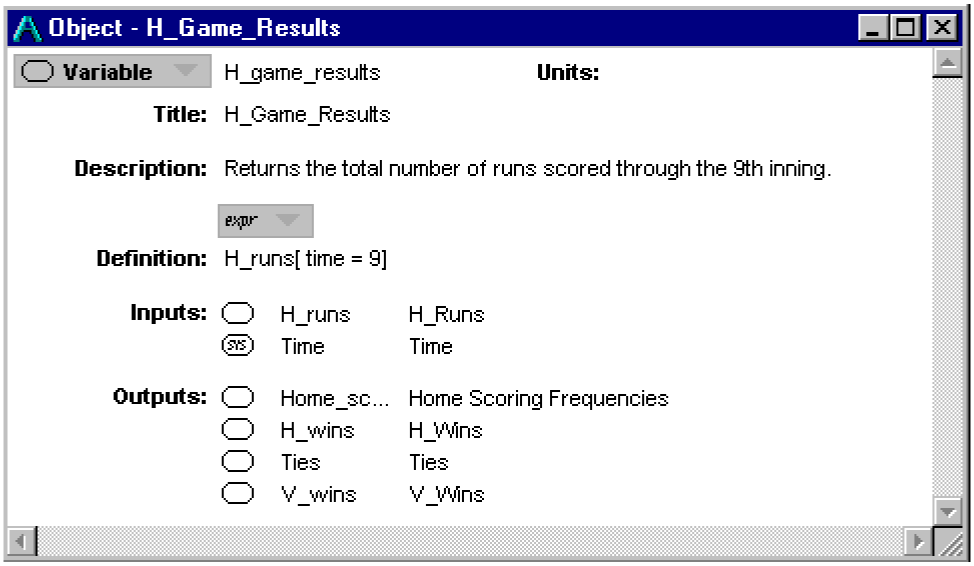


Here we see that the node "Home Team Scoring" is dependent on two inputs – the home team's average runs/game and the inning of the current situation (which determines how many more times the home team will come to bat in this game). Nodes themselves can be modules containing other nodes, allowing hierarchical models to be built. "Home Team Scoring" is such a module, and contains several other nodes used to compute the scoring distribution.

Home Team Scoring Module



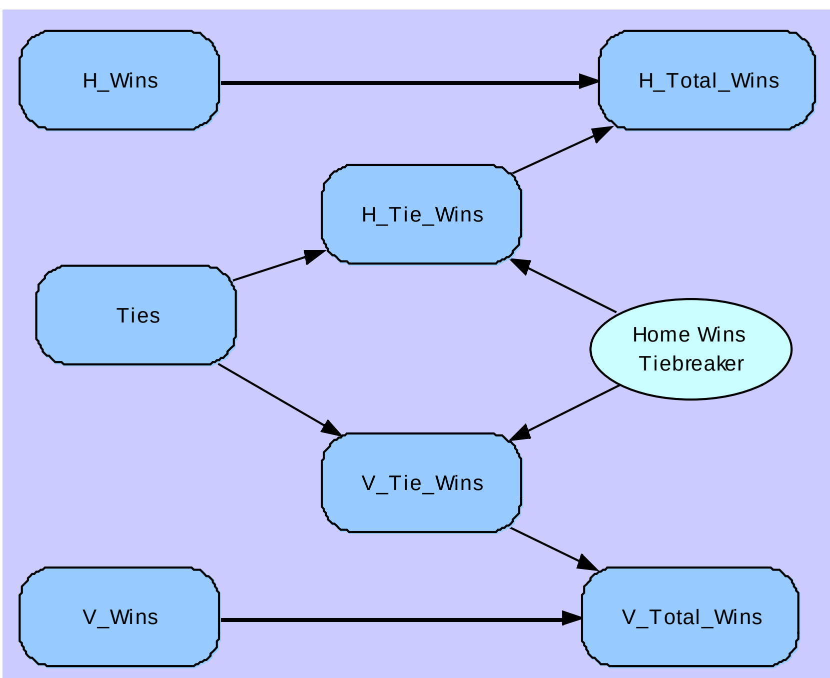
Each of these nodes contains mathematical or probabilistic definitions that relate to computing home team scoring. Double-clicking on one of the nodes, say "H_Game_Results" will show the description, definition and other information about that node:



H_Runs is an array of the team's score after each inning, and time is being used as an index for innings in the game. This node, H_Game_Results, returns the number of runs the home team has scored at the end of regular play (time equal to 9).

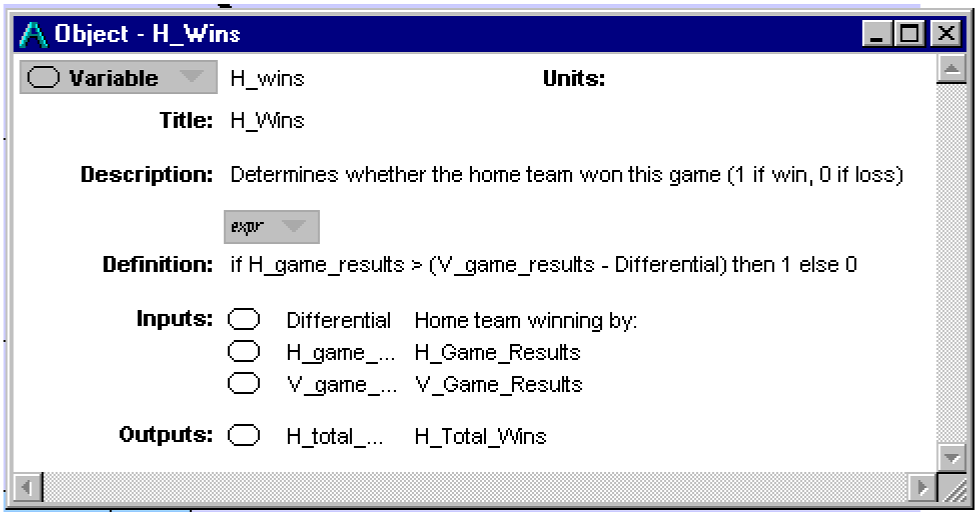
Let's look at another module:

Wins and Losses Module



At this point in the model, we've calculated the scoring for both the home and visiting teams, and are now interested in figuring the number of wins for each team. The module accounts for both games that are won outright in 9 innings, and for the odds of winning in extra innings.

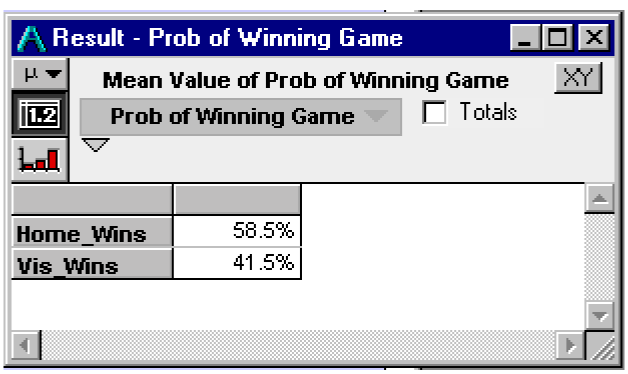
The node definitions needed to determine the winner is quite straightforward. Let's take a look at the H_Wins, which determines whether the home team wins after 9 innings.



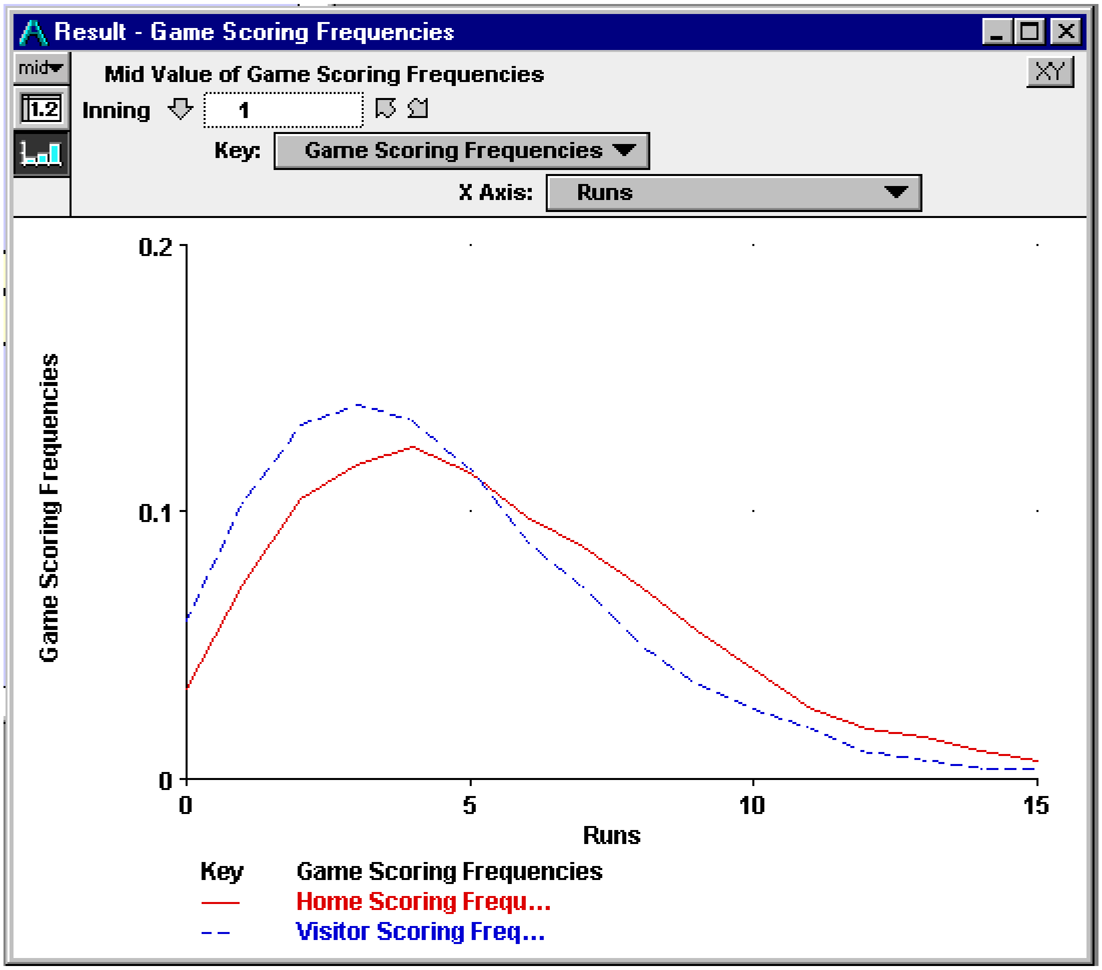
Simply put, the node compares the number of runs each team scored from the initial game situation (which is calculated by the H_game_results and V_game_results nodes) through the end of the game. If the home team outscored the visitors by more than the starting difference (value of the Differential node) between them, then the home team gets the win.

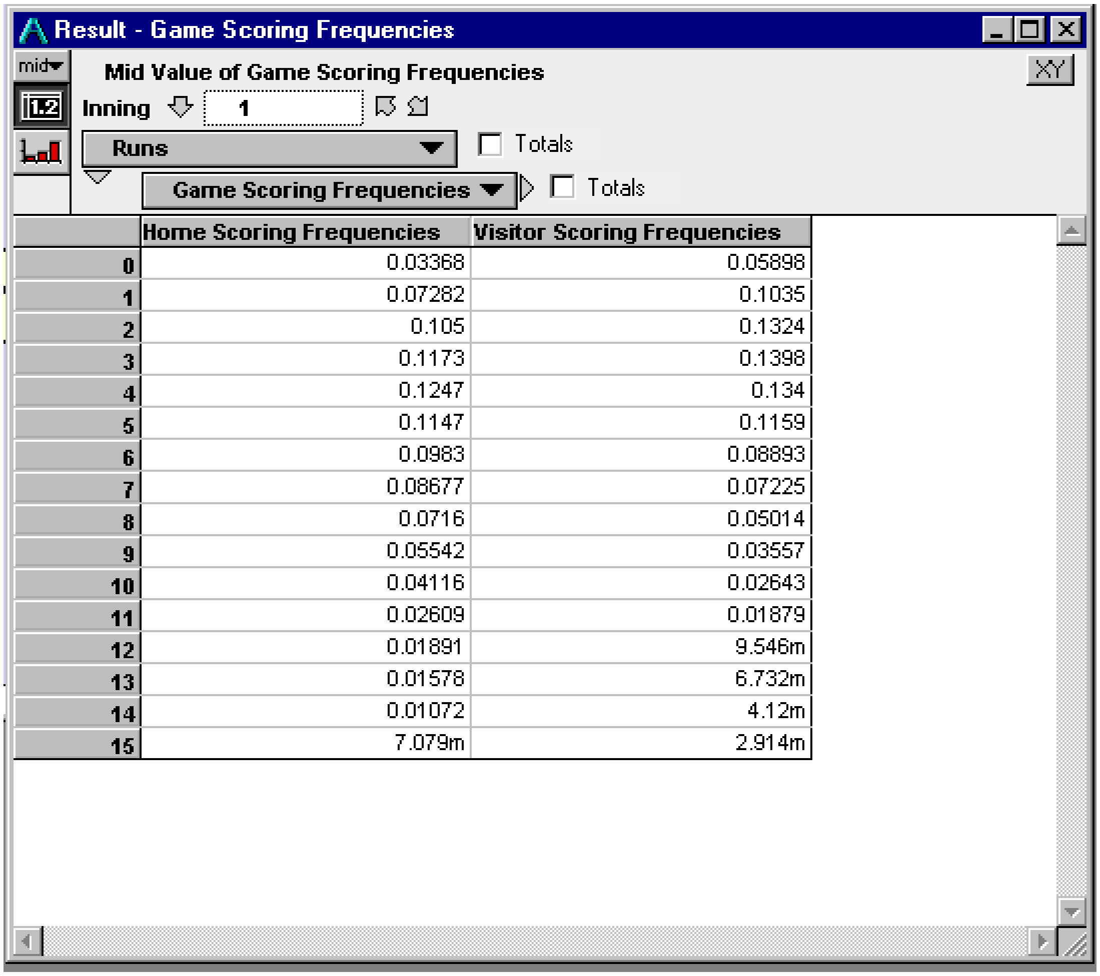
Displaying Results in Analytica

Analytica can evaluate the value any node in the model, and display both single-value and probabilistic results in a variety of tabular and graphical representations. The simplest results might be simply the likelihoods of each team winning the game (which is the “Prob of winning game” node in the top level of the model):



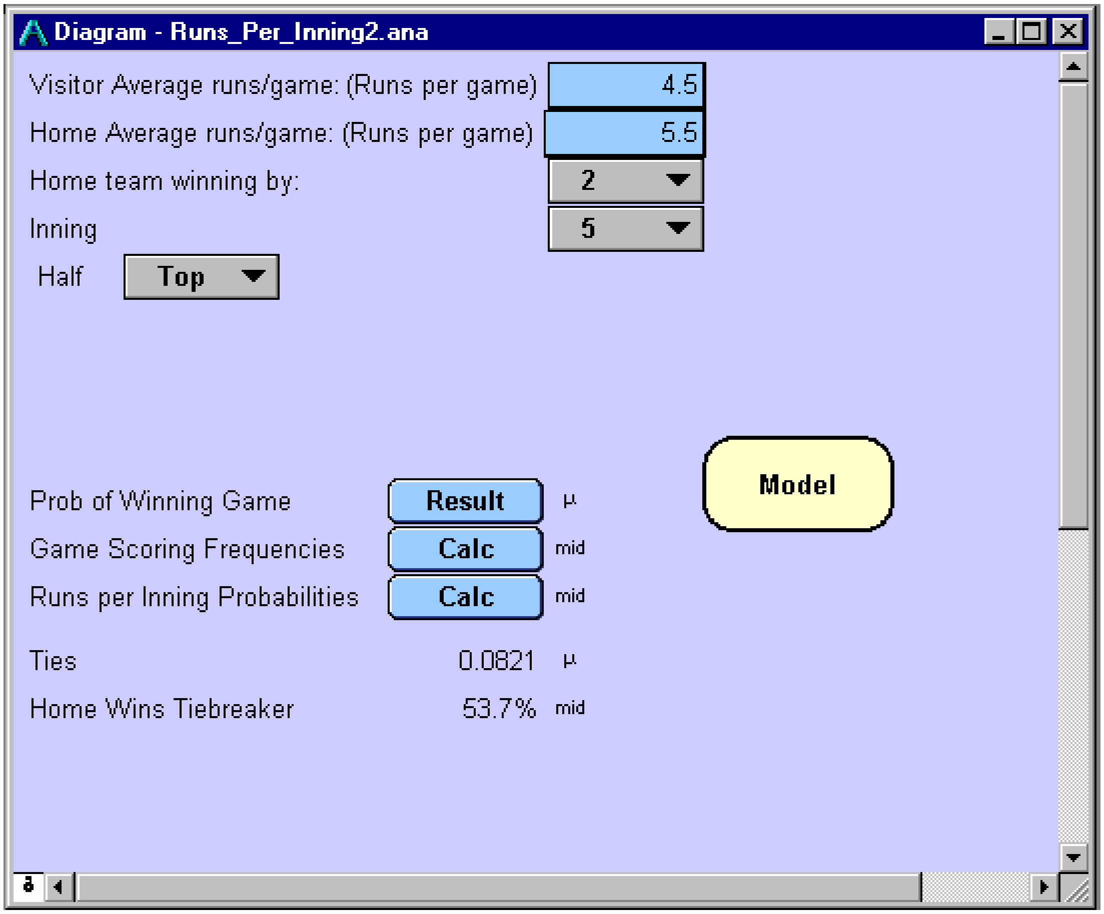
Analytica can also show probabilistic results, either in a graph, or as a spreadsheet-like table:


Result - Game Scoring Frequencies
mid
1.2
Mid Value of Game Scoring Frequencies
XY
Inning
1
Key:
Game Scoring Frequencies
X Axis:
Runs
Game Scoring Frequencies
0.2
0.1
0
0
5
10
15
Runs
Key
Game Scoring Frequencies
Home Scoring Frequ...
Visitor Scoring Freq...



	Home Scoring Frequencies	Visitor Scoring Frequencies
0	0.03368	0.05898
1	0.07282	0.1035
2	0.105	0.1324
3	0.1173	0.1398
4	0.1247	0.134
5	0.1147	0.1159
6	0.0983	0.08893
7	0.08677	0.07225
8	0.0716	0.05014
9	0.05542	0.03557
10	0.04116	0.02643
11	0.02609	0.01879
12	0.01891	9.546m
13	0.01578	6.732m
14	0.01072	4.12m
15	7.079m	2.914m

Lastly, you can built easy-to-use forms to collect all the input and output nodes into a single location:



There's lots more than Analytica can do, but this is just meant to be a quick introduction, not a marketing blurb. ☺ For more Analytica product information, visit http://www.lumina.com

The Online Probabilistic Game Predictor

We've taken the basic runs per inning scoring model and turned it into a web page at http://lux.lumina.com/baseball There, you can play with different scenarios without having to know anything about Analytica or modeling. You can enter in different team strengths, the game situation, and the number of games you want to simulate, and view the results online.

Of course, if you want to get more involved with modeling baseball probabilistically, we've got just the thing for you.

The Contest

Lumina Decision Systems and the Baseball Prospectus are sponsoring a contest for you to build your own baseball models with Analytica. Any baseball-related topic can be explored – from investigating batting averages and hitting streaks, to attendance modeling, to strategic analysis, to game simulators. We encourage you to be creative!

We'll be giving away copies of Baseball Prospectus 2000 to the top submissions, and the grand prize winner will also win a copy of Analytica 2.0 (a $1000 value), and a chance to write an article for the BP web site explaining your award-winning model.

Here's how to participate.

1) Visit http://www.lumina.com/reg/trialanalytica.htm and download the free 30-day trial version of Analytica and the on-line documentation. (Mention in the comments section that you are participating in the Baseball Prospectus contest).

2) Acquaint yourself with Analytica, using the included tutorial and examples. You can also download the Runs_Per_Inning model here.

3) Develop your own baseball-related model on your favorite topic, team, or player. If you have questions or difficulties during the contest, you can get help by emailing kwoolner@baseballprospectus.com

4) Once your model is complete, send your model in an email attachment to kwoolner@baseballprospectus.com. Remember to include your name and email address, any documentation or explanation necessary to use your model (ideally included in the model itself using Analytica's documentation features). The deadline for submitting models is March 15, 2000.

5) Multiple models from the same person can be submitted, but there's a limit of one prize per person.

6) We'll be announcing the results of the contest on the BP website, and contacting the winners via email to arrange delivery of prizes (so make sure we have your contact information correct when you send in your models!)

The decision of the Baseball Prospectus judges will be final in any and all matters regarding the contest. Employees and affiliates of Lumina and/or the Baseball Prospectus are not eligible to win. Void where prohibited. Etc, etc, etc. ☺

What more could you ask for? Fame, glory, cool technology, and free books can be yours! So get started soon!

Acknowledgements

I owe many debts of gratitude to various colleagues at Lumina for assistance I received in preparing this article. In particular, thanks to Tim Nieman for consulting on early versions of the scoring model, and for reviewing a draft of this article. Also special thanks to Hoi Fung and Lutz Mueller for his efforts in deploying the Game Predictor web application. Finally, thanks to Tom Tomasetti and Eric Shiroma, the guys who control the purse strings, for sponsoring the BP modeling contest.