Baseball with Popcorn, Statistics and SAS - What A Mix

David Franklin, TheProgrammersCabin.com, Litchfield, NH

ABSTRACT

The year is almost over. At the end of September another year of baseball will be over. Then the talk will be over who was the best player for the season – a look at the statistics will those who make this decision.

Statistics in sports has been around since sports were first played, and where there is statistics, SAS® is not far behind. This paper looks at Baseball statistics, how they were developed, then look at the how some of the more common statistics are calculated. Finally, a program will be presented that was developed and used in local minor league to solve the riddle of who was the best pitcher and batter.

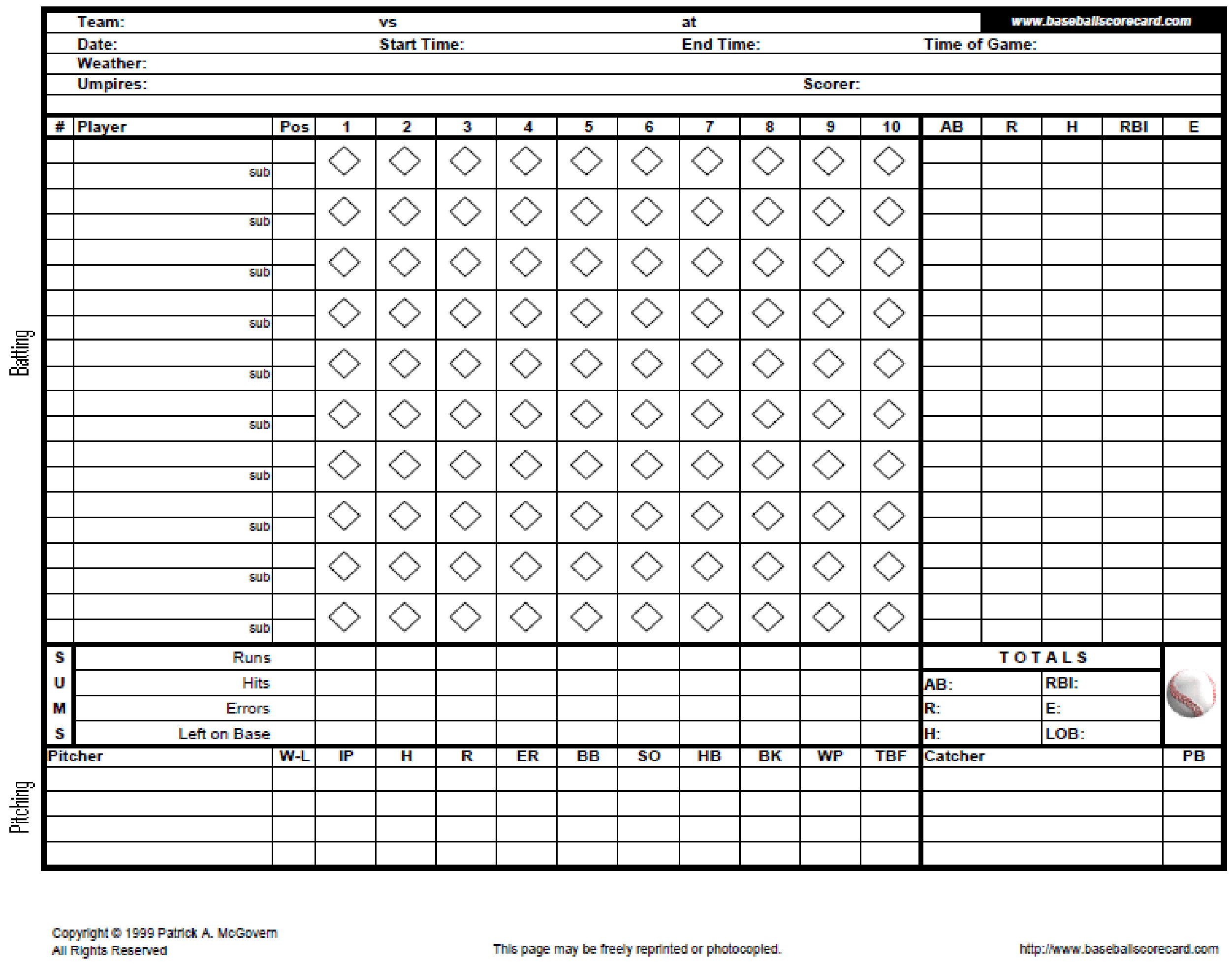
INTRODUCTION

Baseball statistics can be traced back to its roots in cricket. It was a person by the name of Henry Chadwick in the 19th century who first brought the science of numbers to the sport with his experience in cricket statistics. From the start statistics such as the batting average and earned run average have been prominent. However in recent years there have been a new slew of statistics computed that draw attention to almost every facet of baseball, whether it be batting, pitching or fielding.

This paper will have a look at how a few of the more common statistics that are calculated, and then show an example based on the real world in a local minor league where the question was answered, who was the best pitcher and best batter for the season.

HOW THE DATA ARE COLLECTED

During a local minor baseball league each team will have their own scorer (in the higher leagues there will be a team of independent scorers) and they will fill out a sheet similar to that given below:



Team: vs at www.baseballscorecard.com
Date: Start Time: End Time: Time of Game:
Weather:
Umpires: Scorer:

Batting

#	Player	Pos	1	2	3	4	5	6	7	8	9	10	AB	R	H	RBI	E
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																
			◇	◇	◇	◇	◇	◇	◇	◇	◇	◇					
	sub																

S	Runs											TOTALS	
U	Hits											AB:	RBI:
M	Errors											R:	E:
S	Left on Base											H:	LOB:

Pitching

Pitcher	W-L	IP	H	R	ER	BB	SO	HB	BK	WP	TBF	Catcher	PB

Copyright © 1999 Patrick A. McGovern
All Rights Reserved

This page may be freely reprinted or photocopied.

http://www.baseballscorecard.com

The form collects numerous items of information, but the two sections that are of interest to us is the Batting and Pitching sections.

Concentrating first on the Batting part of the form, each player has recorded his/her position after hit, whether they were stuck out, caught, or any other of the ways a batter can get out. The way to record the individual information is beyond this paper but a good reference can be found at baseballscorecard.com/scoring.htm. At the end of the game the scorer will calculate the following statistics for each player:

- AB (At bat): Batting appearances, not including bases on balls, hit by pitch, sacrifices, interference, or obstruction.
- R (Runs scored): number of times a player crosses home plate
- H (Hits): times reached base because of a batted, fair ball without error by the defense
- RBI (Run batted in): number of runners who scored due to a batters' action, except when batter grounded into double play or reached on an error
- E (Error): an act where a fielder misplaying a ball in a manner that allows a batter or base runner to reach one or more additional bases

Note that other forms of the score card may contain the number of singles, doubles, triples, home runs, and many others.

The Pitching section also records the pitcher statistics for:

- IP (innings pitched)
- H (Hits): total hits allowed
- R (Runs): total runs allowed
- ER (earned runs): number of runs that did not occur as a result of errors or passed balls
- BB (walks): times pitching four balls, allowing the batter-runner to advance to first base
- SO (strikeouts)
- HB (batters hit): times hit a batter with pitch, allowing runner to advance to first base
- BK (balks): number of times pitcher commits an illegal pitching action or other illegal action while in contact with the pitching rubber, thus allowing base runners to advance
- WP (wild pitches): charged when a pitch is too high, low, or wide of home plate for the catcher to field, thereby allowing one or more runners to advance or score
- TBF (batters faced): opponent's total plate appearances

The W-L column refers to the Win/Loss percentage which is not necessarily useful.

Note that other forms of the score card may collect other forms of information including the number of Intentional Walks and number of Inherited Runs Allowed, and many others.

Some forms of the score card may also include fielding information that can record the number of Outfield Assists, Passed Balls, Pickoffs, and Putouts, but this is commonly beyond the scope of minor league statistics.

From this information collected on the form, statistics can be calculated for the game.

SOME USEFUL STATISTICS

There are well over 150 different statics that can be calculated, but most of these are redundant with the exception of the major leagues. Common statistics in use are:

- Batting Average: (total hits) / (times at bat)
- Earned Run Average: 9 * (earned runs) / (innings pitched)
- Base-on-balls Percentage: (total walks) / (plate appearances)
- Home Run Ratio: (at-bats) / (home runs)
- On-base Percentage: (hits + walks + hits by pitch) / (at-bats + walks + hits by pitch + sacrifice flies)
- Runs Created: [(hits + walks - caught stealing) * (total bases + (stolen bases * 0.55)] /(at-bats + walks)
- Slugging Average: (total bases) / (at-bats)
- Won-Lost Percentage: (wins) / (wins + losses)
- Fielding Average: (total putouts + assists) / (putouts + assists + errors)
- Fielder's Range Factor: (putouts + assists) / (games)
- Opponents' Batting Average: (hits allowed) / (at bats allowed)
- Strikeout Ratio: (at-bats) / (strikeouts)
- Stolen Base Percentage: (stolen bases) / (total attempts)
- Winning Percentage: (games won) / (total games played)

Some of these can be created from the form used above, others need a modified form.

These statistics above, though useful, are only calculated for the game in question. The question originally asked at the beginning was who was the best batter and who was the best pitcher for the season!

OUR REAL WORLD EXAMPLE

Our prime task at the beginning was to work who was the best pitcher and who was the best batter for the season. By tradition the best batter is the person with the highest batting average and the pitcher has the lowest earned run average.

Two datasets were used.

The first dataset which contained the Batting player summary information had the following fields (SAS variable used is given in '[ ]'s):

- Team [TEAM]
- Player Number [PLAYER]
- Date [DATE]
- Start Time (if a team had multiple games on the same day, useful if trying to get individual information) [TIME]
- Number of Times At bat [NUMBAT_B]
- Runs Scored [RUNS_B]
- Hits [HITS_B]
- RBI [RBI_B]
- Errors [ERRORS_B]

The second dataset which contained the Pitching player summary information had the following fields (SAS variable used is given in '[ ]'s):

- Team [TEAM]
- Player Number [PLAYER]
- Date [DATE]
- Start Time (if a team had multiple games on the same day, useful if trying to get individual information) [TIME]
- Innings Pitched [INPTCH_P]
- Hits [HITS_P]
- Runs [RUNS_P]
- Earned Runs [EARNRN_P]
- Walks [WALKS_P]
- Strikeouts [STRICK_P]
- Batters Hit [BATHIT_P]
- Balks [BALKS_P]
- Wild Pitches [WILD_P]
- Batters Faced [BATFCE_P]

The transfer of the information from the paper form to the dataset was carried out by the individual scorers for each team into Excel files. These files were then loaded into the two SAS datasets.

It must be noted that both datasets contain information that is collected and not used, for example the RBI and Errors count from the Batting dataset, but these may be used in the future to calculate further statistics should the league require it.

Below is a small sample of data that were collected for the Spring Season this year:

Dataset: BATTING

TEAM	PLAYER	DATE	TIME	NUMBAT_B	RUNS_B	HITS_B	RBI_B	ERRORS_B
Giants	4	4-May-11	17:00	4	2	4	0	1
Giants	6	4-May-11	17:00	5	1	2	0	1
A's	8	4-May-11	17:00	4	0	1	0	0
A's	15	4-May-11	17:00	5	4	5	1	0
Diamondbacks	3	5-May-11	17:00	4	0	1	0	1
Diamondbacks	7	5-May-11	17:00	4	1	2	0	0
Cubs	4	5-May-11	17:00	4	0	2	0	0
Cubs	5	5-May-11	17:00	4	0	1	0	0

Dataset: PITCHING

TEAM	PLAYER	DATE	TIME	INPTCH_P	HITS_P	RUNS_P	EARNRN_P	WALKS_P
Giants	7	4-May-11	17:00	3	6	4	4	2
Cubs	13	5-May-11	17:00	2	4	6	1	0

Dataset: PITCHING

TEAM	PLAYER	DATE	TIME	STRICK_P	BATHIT_P	BALKS_P	WILD_P	BATFCE_P
Giants	7	4-May-11	17:00	4	0	-	-	20
Cubs	13	5-May-11	17:00	5	0	-	-	17

The code to produce the datasets by loading each spreadsheet and concatenating the data into the relevant spreadsheets is beyond the scope of this paper. However, a good reference on how to bring in Excel data into SAS can be found by reading a paper called Reading Excel® Workbooks, presented at SAS Global Forum 2007 (paper number 119-2007).

THE CALCULATIONS

For the league that the analysis was required, only two statistics were reported on.

The calculation for the batting average is:

Batting Average: (total hits) / (times at bat)

In order to calculate this statistic the total number of hits across the season were divided by the total number of times at bat, by individual player, using the SUMMARY procedure. The RANK procedure was then used to compute the ranking of the player – due to its options an analysis by team and overall league could be computed. With the Batting Average, the higher the value the better.

The calculation for the earned run average is:

Earned Run Average: 9 * (earned runs) / (innings pitched)

In order to calculate this statistic the total number of runs earned across the season were divided by the total innings pitched, by individual player, using the SUMMARY procedure. The RANK procedure was then used to compute the ranking of the player – due to its options an analysis by team and overall league could be computed. With the Earned Run Average, the lower the value the better. Just as a point of interest, the calculation includes a multiplier of ‘9’ as originally a pitcher was expected to pitch for all nine innings of a game (relief pitches were not routine until after 1900).

Below is some example data along with the SAS Code and output necessary to compute the statistics.

```
*----------------------------------------------------------*;
*Batting Data;
*----------------------------------------------------------*;

*A datastep is used here to load some sample data that will be used in
 the tables generated below.  In the “live” program the datastep references
 a dataset that is created from the combined Excel files that the program
 would reference;
data batting;
   length TEAM $20
          PLAYER 8
          _DATE $7. _TIME $5.
          NUMBAT_B RUNS_B HITS_B RBI_B ERRORS_B 8;
   infile cards dlm='~';
   input TEAM $ PLAYER _DATE _TIME NUMBAT_B RUNS_B HITS_B
         RBI_B ERRORS_B;
   date=input(_date,date7.);
   time=input(_time,time5.);
   drop _date _time;
cards;
Giants~4~04May11~17:00~4~2~4~0~1
Giants~6~04May11~17:00~5~1~2~0~1
A's~8~04May11~17:00~4~0~1~0~0
A's~15~04May11~17:00~5~4~5~1~0
Diamondbacks~3~05May11~17:00~4~0~1~0~1
Diamondbacks~7~05May11~17:00~4~1~2~0~0
Cubs~4~05May11~17:00~4~0~2~0~0
Cubs~5~05May11~17:00~4~0~1~0~0
;
run;

*Summarize runs and times at bat in BATTING dataset;
proc summary data=batting nway;
   class team player;
   var HITS_B NUMBAT_B;
   output out=batsum (drop=_type_ _freq_) sum=hits NUMBAT;
run;

*Calculate the batting average;
data batsum;
   set batsum;
   if n(hits,numbat)=2 then batavg=hits/numbat;  *Only calculate if both values for the
                                                  total hits and times at bat values
                                                  are present;
run;

*Use RANK procedure to put into rank order, the higher the average, the better;
proc rank data=batsum out=batstat descending ties=low;
   var batavg;
   ranks batavgrank;
run;

*Produce output;
proc sort data=batstat;
   by descending batavg team player;
run;
proc print data=batstat noobs label;
   title1 "Batting Average for League";
   title2 "Based on Data Received by 07May2011";
   var batavgrank team player batavg hits numbat;
   label batavgrank='Rank'
         team='Team'
         player='Player'
         batavg='Batting Average'
         hits='Number of Hits'
         numbat='Number of Times at Bat';
   format batavg 5.3;
run;

*----------------------------------------------------------*;
*Pitching Data;
*----------------------------------------------------------*;

*A datastep is used here to load some sample data that will be used in
 the tables generated below.  In the “live” program the datastep references
 a dataset that is created from the combined Excel files that the program
 would reference;
data pitching;
   length TEAM $20
          PLAYER 8
          _DATE $7 _TIME $5
          INPTCH_P HITS_P RUNS_P EARNRN_P WALKS_P STRICK_P BATHIT_P BALKS_P
          WILD_P BATFCE_P 8;
   infile cards dlm='~';
   input TEAM $ PLAYER _DATE $ _TIME $ INPTCH_P HITS_P RUNS_P EARNRN_P WALKS_P
         STRICK_P BATHIT_P BALKS_P WILD_P BATFCE_P;
   date=input(_date,date7.);
   time=input(_time,time5.);
   drop _date _time;
cards;
Giants~7~04May11~17:00~3~6~4~4~2~4~0~.~.~20
Cubs~13~05May11~17:00~2~4~6~1~0~5~0~.~.~17
;
run;

*Summarize earned runs and innings pitched;
proc summary data=pitching nway;
   class team player;
   var EARNRN_P INPTCH_P;
   output out=pitchsum (drop=_type_ _freq_) sum=earnrn inptch;
run;

*Calculate the Earned Runs Average;
data pitchsum;
   set pitchsum;
   if n(earnrn,inptch)=2 then era=9*earnrn/inptch; *Only calculate if both values for the
                                                    earned runs and innings pitched values
                                                    are present;
run;

*Use RANK procedure to put into rank order – the lower the ERA value, the better;
proc rank data=pitchsum out=pitchstat ties=low;
   var era;
   ranks erarank;
run;

*Produce output;
proc sort data=pitchstat;
   by era team player;
run;
proc print data=pitchstat noobs label;
   title1 "Pitching Average for League";
   title2 "Based on Data Received by 07May2011";
   var erarank team player era earnrn inptch;
   label erarank='Rank'
         team='Team'
         player='Player'
         era='Earned Run Average'
         earnrn='Earned Runs'
         inptch='Innings Pitched';
   format era 5.3;
run;
```

The outputs produced are:

```
                             Batting Average for League
                          Based on Data Received by 07May2011

                                                                  Number of
                                             Batting     Number     Times
          Rank     Team            Player    Average    of Hits     at Bat

            1      A's               15       1.000        5          5
            1      Giants             4       1.000        4          4
            3      Cubs               4       0.500        2          4
            3      Diamondbacks       7       0.500        2          4
            5      Giants             6       0.400        2          5
            6      A's                8       0.250        1          4
            6      Cubs               5       0.250        1          4
            6      Diamondbacks       3       0.250        1          4

                            Pitching Average for League
                          Based on Data Received by 07May2011

                                            Earned
                                              Run      Earned     Innings
               Rank      Team      Player   Average     Runs      Pitched

                 1      Cubs         13      4.500        1          2
                 2      Giants        7      12.00        4          3
```

Note that the data used in this paper is only a very small subset of the data. In the actual tournament the number of games each team would play is 12.

In the code given a datastep is used here to load the sample data (this same data is used to generate the output) -- in the "live" program the datastep references a dataset that is created from the combined Excel files that the program would reference, as noted above.

The two calculations for the Batting Average and Earned Runs Average are only done if both sets of data for each is given – this is done so that NOTES and WARNING messages do not appear in the LOG due to either missing value or division of missing (this can occur if input is incomplete).

The reason for using the RANK function was to put a numerical "rank" against the data and dealing with tied values easily. The TIES= option controls the treatment of tied values. Possible values for this option are:

- LOW: assigns the smallest of the corresponding ranks to these observations (if values are 0.1, 0.2, 0.2 and 0.3 then ranks will be 1, 2, 2 and 4)
- HIGH: assigns the largest of the corresponding ranks (if values are 0.1, 0.2, 0.2 and 0.3 then ranks will be 1, 3, 3 and 4)
- MEAN: assigns the mean of the corresponding rank (if values are 0.1, 0.2, 0.2 and 0.3 then ranks will be 1, 2.5, 2.5 and 4).

In each of these cases the values are treated as distinguishable values.

In recent versions of SAS, the option TIES=DENSE has been available that computes scores and ranks by treating tied values as a single-order statistic, i.e. tied values are treated as indistinguishable with each value within a tied group is assigned the same ordinal.

A decision not to find the best batter or pitcher by team was made early in the specifications, but the program could be easily adapted to do BY TEAM processing.

CONCLUSION

The question asked in the local minor league was who is the best batter and who is the best pitcher. Using SAS, the league was able to compute and easily report the result. Because of the number of data points collected it is also possible to do more statistical computations and there is an idea for the fall season to have this report available to the league officials after half-way though the season on a week by week basis.

REFERENCES

Heaton, E. Reading Excel® Workbooks. SAS Global Forum, 119-2007, 2007.
Patrick A. McGovern. "Horizontal Scorecard" http://www.baseballscorecard.com/downloads/Scorecard-h.pdf (11Jul2011)
SAS Institute Inc. 2006. Base SAS® 9.1.3 Procedures Guide, Second Edition, Volumes 1, 2, 3, and 4. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name:	David Franklin
Company:	TheProgrammersCabin.com
Address:	16 Roberts Rd
City, State ZIP:	Litchfield, NH 03052
Work Phone:	603-275-6809
Email:	dfranklin@TheProgrammersCabin.com
Web:	TheProgrammersCabin.com