Building and Applying Statistical Modeling Tools for an MLB Dataset

*Michael Cowan*

*10/30/2017*

In 2003, Michael Lewis released book titled *Moneyball*.1 The piece discussed Billy Beane, the general manager for the Oakland A’s, and his new and interesting approach on running a major league baseball (MLB) team. Mr. Beane was utilizing an analytics software known as *sabermetrics* to predict player’s talent based on their previous stats. In fact, the data-driven method was adopted by the Boston Red Sox who credited the technique as being a main reason for their World Series victory in 2004; a win which broke the long standing curse of the Great Bambino. The success of the Red Sox helped propel data science into the main stream of the MLB as it is today. Due to my love for the game (and especially the Red Sox), I found it fitting to apply my newly learned statistical modeling skills to an MLB batting dataset.

Baseball is truly a unique sport that favors endurance and consistency. Every team plays a 162 game season (not including post-season games). The games themselves are quite long with a minimum of nine innings being played out in a 9-on-9 fashion. The core rules of the game allow for many different stats to be recorded. Since baseball is played in a discretized pitch-by-pitch manner, it is easy to collect quantitative data on each player’s performance. In fact, the dataset chosen for this work only incorporates batting statistics. This excludes fielding, pitching, and even overall team stats that could also be analyzed. With so many stats being generated by the game, it is easy to see why the MLB is an excellent source to gather and study data.

The overall dataset chosen for this work consists of batting statistics ranging from 1955 to 2016 for both the American and National Leagues in the MLB. The data was refined such that only batting championship-qualifying seasons were looked at. The criteria to qualify for a batting award is having an average plate appearance (PA) no lower than 3.1.2 PA can be described as:

. (1)

Once the given criterion was applied, the dataset was found have 6863 rows and 30 columns. The specifications of the dataset have been summarized in Table 1 below.

Table 1: Summary of batting dataset

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Data Type** |
| Player | Unique player ID | Identifier |
| Year | year of the season | Integer |
| Team | player's team | String |
| League | American or National League | Boolean |
| G | games played | Integer |
| AB | at bats | Integer |
| R | runs scored | Integer |
| H | hits | Integer |
| 2B | doubles | Integer |
| 3B | triples | Integer |
| HR | homeruns | Integer |
| RBI | runs batted in | Integer |
| SB | stolen bases | Integer |
| CS | caught stealing | Integer |
| BB | walks | Integer |
| SO | strike outs | Integer |
| IBB | intentional walks | Integer |
| HBP | hit by pitch | Integer |
| SH | sacrifice hits | Integer |
| SF | sacrifice flies | Integer |
| GIDP | grounded into double plays | Integer |
| Avg. | batting average | Float |
| PA | plate appearances | Float |
| Age | age of player | Integer |
| Height | height of player | Integer |
| Weight | weight of player | Integer |
| Bats | left, right, or switch | Ternary |
| All Star | if made/played | Ternary |
| All Star Start | starting position | Integer |

With Table 1 alone, one can learn a lot about the dataset described. For instance, the majority of the parameters are integers which results in the data being highly discontinuous. This can cause complications for certain statistical modeling approaches. The only two continuous parameters are plate appearances and batting average. However, both PA and Avg. are not independent parameters. This will be important when fitting multivariate models since it is imperative that there are no strong dependencies between inputs. Besides integers, the dataset also contains some binary and ternary values. This allows for some interesting investigations into defining “cut-off scores”. For instance, one can ask what the minimum hits, batting average, and RBI’s are required to make the All Star team? One last important characteristic of this dataset is that there are no defined outputs. Since virtually any parameter can be considered an output, the possibilities are almost endless when attempting to fit models. As great as it sounds in theory, one must be aware of the curse of dimensionality. For this reason, most of the models applied to the dataset are based on logical assumptions about the parameters and the game of baseball itself.

As I have grown to enjoy statistical modeling, I have also developed a strong interest in coding. Learning how to use a “black box” modeling package can absolutely be done. However, I instead decided to spend the time to code my own modeling tools and apply them to the chosen dataset. With my Python package, *statmod* 3, I have created tools to conduct single and multivariate ordinary least squares, regularized regression which include LASSO, Ridge, and Elastic Net, and **(Kriging or Neural Network)**. Writing the package myself has allowed for a more fundamental understanding of not only how statistical modeling techniques work, but also why they are effective.

1. Lewis, Michael. Moneyball. *W.W. Norton*. **2003**.
2. Major League Baseball 2017 Official Rules. MLB.com. *MLB Advanced Media, LP*. **2017**. p 137.
3. http://www.Github.com/Michael-Cowan/statmod.

**Outline**

* Introduction
  + Reasons for choosing dataset
    - Billy Beane of Oakland A’s using sabermetrics
    - Boston Red Sox winning the World Series in 2004 by applying similar approach
    - There is a relatively large amount of data compared to other sports
      * 162 games per year
      * 9 batters per team
      * Rules allow for many different stats to be recorded
  + About the dataset
    - Mostly discontinuous data
    - Not all data is independent
    - No defined output
    - Batting stats of all MLB players (AL & NL) ranging from 1955 to 2016
    - Only included MVP-eligible players
      * PA >= 3.1
    - Different ways to view the data
      * Player’s stats over their career
      * One large dataset of stats
  + Building a stat modeling toolset
    - Incorporate my strong interest in coding into the project
    - Achieve fundamental understanding of the modeling approaches
* Statistical modeling approaches used
  + OLS
    - Searching for high correlation coefficients with multivariate OLS
      * Iterate between all combinations of both two and three inputs with one output
  + Regularized regression
    - Use Elastic net to further analyze best fits from OLS
    - Attempt to make educated guesses on some non-linear equations
  + **Kriging (or maybe neural network?)**
    - Define a cumulative rate of failure based on player’s season stats
    - Predict players cumulative rate of failure
* Results
  + OLS
    - No strong fits were found
    - Not surprising due to highly discontinuous data