# Baseball Player Performance and Its Factors

Vail Dorchester and Jackson Mediavilla

## Abstract

At the start of our research, we sought to examine the correlations between numerous factors and player performance, as well as player performance in high stake situations. We hoped that data mining game events would reveal correlations that are not expected or easily identifiable. We wanted to investigate which factors contribute most to player performance.

We wanted to focus on "outside" factors such as weather, stadium attendance, and field condition. The results show correlations of varying significance between player performance and other factors. We intended to evaluate a few player performance metrics, but as the project progressed, we realized that we needed to narrow our scope. Thus, the player performance metric we decided to focus on was batting average. The outside factors we examined, ordered from highest correlation to batting average to lowest correlation, are temperature, field condition, precipitation, wind direction, attendance, day vs. night, wind speed, and sky conditions. Based on the correlation results, we may be able to infer causation on player performance. We also analyzed the relationship between various late game 'clutch' situations and their effect on batting average.

## Introduction

In our literature survey, we found that there has been a rather extensive amount of research on evaluating individual players, which makes sense. Producing that data and selling it to recruiters or scouts proves to be quite valuable, and in fact, most professional teams employ a professional statistician to do reports and research in this exact field. However, less research has been conducted regarding general player performance (performance of all players, not just one), and the factors that affect it. For this reason, we wanted to investigate factors that have not been extensively researched.

Because this study does not focus on individual players, the information gained would most likely be of interest to baseball enthusiasts. However, the knowledge that outside factors are correlated to batting average could be used in sabermetric applications including player recruitment, predicting individual player performance given situational information, and forecasting game outcomes given the states of outside factors.

## Related Work

In his paper, The Statistical Mirage of Clutch Hitting, Harold Brooks discusses the statistics behind the phenomenon of clutch hitting, or more colloquially, when a player hits better when under pressure as opposed to normal circumstances. In this paper, Brooks defines being 'under pressure' as being tied or behind by four or fewer runs in the seventh inning, including some more detailed cases of when the bases are loaded. Brooks argues that in order for a model that predicts clutch hitting to be accurate, it must not only show strong correlations between performance and pressure, but it must be consistent throughout multiple seasons as well and not a single-season-anomaly. When following this model, Brooks argues that one can define successful models for clutch hitting in singular seasons, but that these models show little to no correlation with each other across multiple seasons and are therefore inconsistent and inaccurate. He also argues that it is necessary to first define how well the average player does in a clutch situation, and then compare if a certain player or group of players consistently plays above that average across multiple seasons. For this value to be significant, he argues that it must be greater than one standard deviation from the mean in either direction, and that certain players would fall into this statistically significant category season after season, or even that the clutch rating of players would be consistent throughout seasons. He then goes to show that neither of these cases are true; clutch rating tends to vary from season to season, and individual players tend to vary their performance and rarely bat consistently better under pressure. From this, Brooks argues that clutch hitting is irrelevant and "a mirage at best" (Brooks, H, 1988).

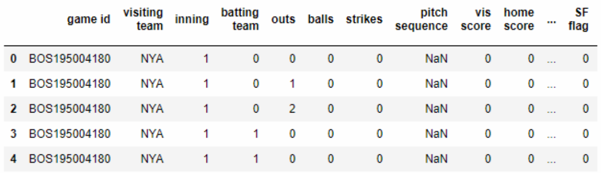
In another article, Scott Lindholm discusses the effect of temperature on batting performance. Interestingly, Lindholm notes that there is a strong positive correlation between temperature and batting performance, but he also notes that this could be due to many, many factors, and that temperature may not necessarily be a causal factor in increased batting performance, but merely a related factor. Still, however, this data would prove useful in building models to predict when a player would perform at their best (Lindholm, S, 2014).

## Data Sets

We used a total of six different datasets to complete this project. The datasets are event data, game logs, park data, city data, weather station data, and weather data. We required all of these datasets to connect game data to weather data.

We used the Retrosheet event dataset, which can be downloaded from retrosheet.org/game.htm. The dataset contains every play in the included games. Each entry in the dataset reports both the event and the game state after the play. There are more than 95 different fields which may apply to a single event. The fields provide situational information such as score, identity of batter, number of outs, handedness of batter and pitcher, etc. To generate the dataset, we used an executable file also found on the Retrosheet website, *BEVENT.EXE*. Upon execution, the program created a play-by-play data file designed for import into custom written programs. The file contains 10.5 million rows.

Figure 1. Event Data



| | game id | visiting team | inning | batting team | outs | balls | strikes | pitch sequence | vis score | home score | ... | SF flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BOS195004180 | NYA | 1 | 0 | 0 | 0 | 0 | NaN | 0 | 0 | ... | 0 |
| 1 | BOS195004180 | NYA | 1 | 0 | 1 | 0 | 0 | NaN | 0 | 0 | ... | 0 |
| 2 | BOS195004180 | NYA | 1 | 0 | 2 | 0 | 0 | NaN | 0 | 0 | ... | 0 |
| 3 | BOS195004180 | NYA | 1 | 1 | 0 | 0 | 0 | NaN | 0 | 0 | ... | 0 |
| 4 | BOS195004180 | NYA | 1 | 1 | 0 | 0 | 0 | NaN | 0 | 0 | ... | 0 |

This figure shows the first five rows of the event files once the data had been imported into a Pandas DataFrame within a Jupyter Notebook.

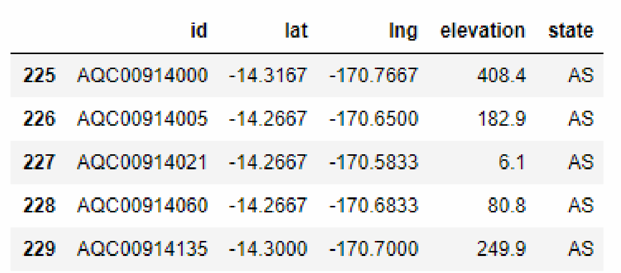
The game log dataset was also downloaded from the Retrosheet website. The executable file *BGAME.EXE* generated traditional box score data for every major league game since 1871. The fields provide information such as team statistics, attendance, scores, pitchers, etc. There are 161 fields for each record. However, the data available varies by year. The earlier years have less data, and the more recent years have more. The dataset contains 133 thousand rows.

The park data was obtained from http://www.retrosheet.org/parkcode.txt, and contains the unique baseball park ID that is associated with each event and game, as well as the city, state, and other information. This dataset contained only 252 rows.

The city data was obtained from a United States Cities database, which was downloaded from https://simplemaps.com/data/us-cities. The dataset was built from data obtained from the U.S. Geological Survey and the U.S. Census Bureau. The data is contained in one CSV file, which has one entry per city. The file size is 36 thousand rows.

Weather station data was obtained from NOAA at https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt. The fields include weather station ID, latitude, longitude, and other data unused for this project. The total size of the file is 106 thousand rows.

Figure 2. Weather Station Data



| | id | lat | lng | elevation | state |
|---|---|---|---|---|---|
| 225 | AQC00914000 | -14.3167 | -170.7667 | 408.4 | AS |
| 226 | AQC00914005 | -14.2667 | -170.6500 | 182.9 | AS |
| 227 | AQC00914021 | -14.2667 | -170.5833 | 6.1 | AS |
| 228 | AQC00914060 | -14.2667 | -170.6833 | 80.8 | AS |
| 229 | AQC00914135 | -14.3000 | -170.7000 | 249.9 | AS |

This figure shows the first five rows of the weather station file once the data had been imported into a Pandas DataFrame within a Jupyter Notebook.

The weather data itself was also obtained from NOAA. The Daily Global Historical Climatology Network dataset was downloaded from https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/all/. The dataset contains many ".dly" files. Each ".dly" file contains data for one station. Each record in a file contains one month of daily weather data. The columns contain extensive information regarding precipitation, cloud cover, wind, etc. The entire dataset contains 1.4 billion rows.

## Tools Used

- Python
- Numpy
- Pandas – python data analysis library
- Matplotlib
- Sklearn
- Scipy
- Jupyter Notebook
- Excel
- Git, GitHub
- Bash

We chose specific tools for this project based on their utility; and many standard Python libraries were used, but the above list contains our main assets. We used Python within Jupyter Notebooks for our general programming; Pandas, Excel, and Bash Scripting for data wrangling; Scipy, NumPy, and SciKit-Learn for our data analysis; and GitHub for our version control.

## Main Techniques Applied

### Data Generation and Importation

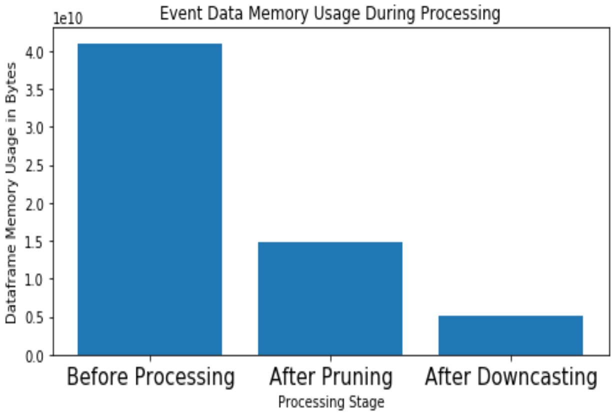
The data we used includes information from games starting in 1950 and ending in 2017. The data was downloaded as zip files from the Retrosheet website. Each zip file contained the data for one decade and was separated between roughly 350 files per decade. These files ranged from team rosters to league rosters and individual event files. The data we were interested in exists in the event files, which are marked by a ".EVX" file extension. However, the decade-folders each contained many other extraneous files. To solve this, we wrote a bash script that utilized *Regex* and *sed* commands to create a file containing the paths of every event file across all decades from 1950-2017. In conjunction with this path file, we used the executable file BEVENT.EXE (also downloaded from the Retrosheet website) to convert the files to comma separated (CSV) files for easy importation to DataFrames. Because the command can only handle one file at a time and there were hundreds of files to process, each located in one of ten separate directories, we created a Jupyter Notebook to iterate over the path file, which would then pass each path to the BEVENT program and generate a .csv for each event file. The result of executing the notebook was hundreds of CSVs, each containing one team's data from one year. These CSV files were then moved into a separate directory, and we created additional python scripts to import all of this data into hundreds of Pandas DataFrames — one for each team and year — as well as one aggregate DataFrame. Because the CSV files did not contain column information, the next step in preprocessing the data was to add field names for the 97 possible columns. This was done by reading the documentation of the *BEVENT.EXE* program and copying the attribute descriptions into a text file, which was then used in python to add the column descriptions to the DataFrames. At this point, the data was nearly ready for statistical analysis.

## Data Preprocessing

Because the datasets contain so much information and we only required a portion of it, the first step in preprocessing the data was pruning irrelevant columns. We only kept the information from the event files that is relevant to this study such as inning, hits, strikeouts, etc. The information dropped included columns like umpires' names, who fielded the ball, and fielding errors. We were also able to drop many rows from the event dataset. Because we were focused on batting average, we only needed the events that resulted in the termination of an at bat. This means that we kept all events like hits, strikeouts, and walks. Events that occurred during at bats, such as attempted pick offs, were dropped. Pruning the data resulted in reducing memory usage significantly (see Figure 3).

The pruned data was still very large and thus would have taken a long time to evaluate. We further reduced the memory usage by converting objects to categories, converting categories to integers, and down casting numeric types. In this way, we maintained the meaning of the data while representing it in the smallest possible numeric type.

Figure 3.



This figure shows the reduction in memory usage by applying data preprocessing techniques.

The game log data was pruned and downcasted in a similar manner as the event data, and preprocessing was much simpler for our other sets. For the city data, no preprocessing was required because we chose to only import the columns we would need: city, state, latitude, and longitude. The park data similarly required little preprocessing, and the station data was pruned by dropping all stations not in the United States. Finally, the weather data required relatively little preprocessing beyond specifying the import parameters and field widths, because the weather data was of _.fwf format as opposed to the more standard _.csv.

## Connecting the Data

With six different datasets, we were posed with the challenge of connecting all the information. The proposed plan was as follows: use the park data to match each park to a city and the city's coordinates. Then, use those coordinates to match each park to a weather station. Following this, we were to use the appended park data to add station data to each game and event by merging on the park id. Next, the park dataset with the appended weather station attribute was used to import the weather data only for the stations we were interested in, and in this way, we avoided dealing with the massive quantities of data that NOAA holds. Finally, we would merge the game and event files with the weather data on station id and dates in order to match each event and game with specific weather data.

The Retrosheet game logs list a city for each game, but we chose to match the park data to the correct city in the United States Cities database to get each park's coordinates so that the matching would only have to be done one time, and the park data could then be merged on the event and game data.

These coordinates were then appended to the game logs as new features. Then, we sliced the data to pull only the station id or park id, the latitude, and the longitude, and this data was appended to a temporary frame that would serve as a list of objects. We then matched the coordinates of each park to the nearest weather station using the K nearest neighbors algorithm. The usage of this algorithm was such that the first 252 rows of the temporary frame — the 252 rows containing the parks — were passed to be predicted, and for each park, it's nearest 10 neighbors were found using the Euclidean distance function. We chose to use the 10 nearest neighbors because, on rare occasion, a park's nearest neighbors would be other baseball parks, and this did not provide the desired information of matching parks to stations. The index matrix generated by the KNN algorithm was then iterated over to generate a list of index pairs in which the first index was that of the park, and the second index was that of the nearest weather station.

Finally, the data was in a state such that each game and event had a date and associated weather station, and then another DataFrame contained the weather data organized by station ID and date. At this point, connecting weather data to each event was a matter of simply merging the two sets on station ID and date. Finally, our data was matched to specific and accurate weather data.

Figure 4.

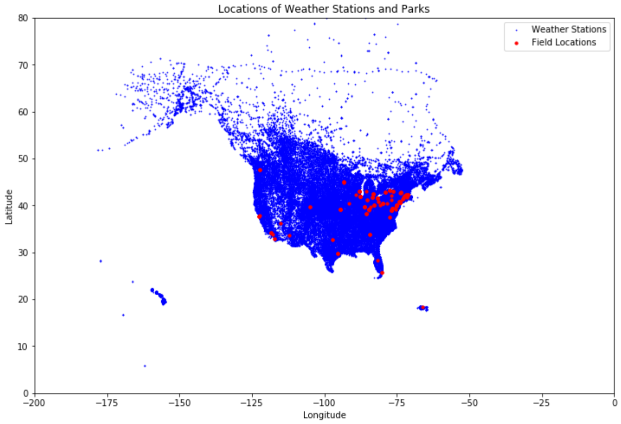


Figure 4 shows the field locations in red and the weather stations in blue. This was the information fed to the K nearest neighbors algorithm. The density of blue dots illustrates the volume of weather station data that was being analyzed.

## Data Analysis

We performed two main types of analysis on the data: linear regression and random forest regression. Linear regression is an approach to model the relationship between a dependent variable (in this case batting average) and independent variables (in this case "outside" factors). The linear regression analysis showed us whether each factor is positively or negatively correlated with batting average and to what degree; it also allowed us to see the general trends in the relationships between the features and the test statistic. Random forest regression is a method of ranking features by importance. The algorithm functions by generating many decision trees and outputting the mean regression of the individual trees. The result of this analysis shows feature importance in determining batting average.

The first step in determining which features affect batting average is calculating the batting average. The formula for batting average is relatively simple, and is:

$$\text{Avg} = \text{Hits} / \text{At Bats}$$

We knew that we would be calculating batting average in a variety of cases, so we found it most practical to write a function which took in a DataFrame, a feature, and a value, and returned a DataFrame holding the batting average for each player and game where the feature parameter matched the value parameter.

For our weather analysis, the information extraction process was relatively simple. Our data contained not only weather information, but the game logs also had some extra features such as field condition, sky condition, precipitation, and temperature, which could be used independently or cross referenced with the weather DataFrame. The process was as follows: we created lists of the features we wanted to analyze and the possible discrete values they could take on — many of the features were ranked 0 through 5. Then, we iterated over the list of features, and for each feature, iterated over its possible values, and passed each of those into the batting average function. Doing this, we were able to perform some preliminary analysis that showed us the batting averages as a function of these other features.

Next, we used SciKit-Learn to create a random forest regressor. This regressor took as features the weather information that we had gathered as well as some other features: temperature, precipitation, snow, field conditions, sky condition, wind speed, wind direction, attendance, and a day or night indicator. The batting averages were removed from the game data and stored as a separate array to be used for fitting and testing the model. The DataFrame holding the batting averages was used as the predictor data. The random forest regressor was fit to the predictors and batting average data, but we actually did not use it for any regression. Instead, we then extracted the features and their assigned importance from the random forest regressor. These features were then sorted based on importance, and a correlation coefficient was calculated for each feature. In this way, we were able to quantitatively answer our question of which features are most important in influencing player performance, and just how important they are.

Analyzing clutch playing proved to be more complex for a variety of reasons. First, we found that there is not an agreed upon metric of 'clutch playing', and beyond that, the entire concept of clutch playing is rather controversial. Regardless, our first step in determining the validity of clutch playing and what goes in to clutch plays was to first determine what exactly defines a clutch play.

We found that clutch playing is typically defined as performing well in 'high pressure' situations. With batting average as our metric for performing well, we needed only to define what constitutes a high-pressure situation. On a more qualitative level, high pressure situations are when the game is close and there is significant potential to sway the future of the game. Quantitatively, this means that the scores are close, typically it is late in the game, there may be multiple runners on base, and there may be multiple outs.

We decided partly for simplicity and partly for greater control to define these situations manually as opposed to creating a multivariable function that produces a pressure index as an output per play. With this in mind, we decided to first calculate the batting average with no special conditions applied.

This base batting average would be a part in nearly all of our null hypotheses during our statistical hypothesis testing. With our base batting average calculated, we then narrowed our search of clutch situations to only plays that took place in the 9th inning and set that data aside. We then did the same with plays in the 9th inning with a difference in team scores that was less than or equal to two, plays in the 9th inning with a difference of scores less than or equal to two with two or more runners on base, plays in the 9th inning with a score difference less than or equal to two with two or more runners on base and zero outs, and plays in the 9th inning with a score difference less than or equal to two with two runners on base and two outs. More colloquially, we looked at situations of increasing pressure in which a strong play was likely to influence the outcome of the game. We passed each of these data subsets into the batting average function to calculate the performance of players in these high-pressure situations.

Finally, we performed some simpler analysis on batting average versus inning, among others.

## Key Results

The first key result that we obtained involved linear regression analysis between batting average and inning. The goal here was to determine if, as the game went on, players began to perform better or worse.

Figure 5.

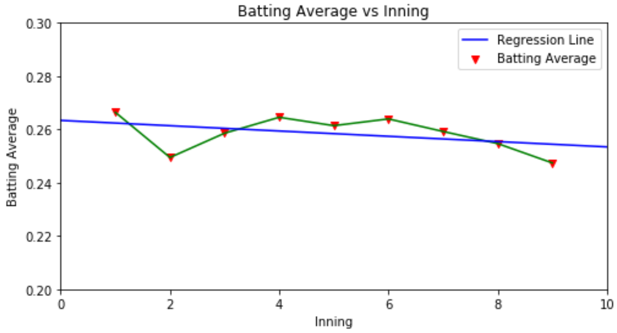


Figure 5 shows a slight negative correlation between inning and batting average. The regression equation was calculated with all available data from the Retrosheet datasets and without aggregation by player, team, or year.

After performing a two-tailed hypothesis test for each inning where the null hypothesis stated that each average was equal to the general batting average, we were able to reject the null hypothesis at the 95% confidence level for every single inning. This means that we can say that there is a measurable statistical difference between the normal batting average and the batting average when separated out by inning. We obtained extremely small p-values in our tests, and this is likely a result of the p-values being inversely proportional to the square root of the number of data points used, which, in this case, was on the order of millions. This means that, in order to be unable to reject the null hypothesis, the alternative hypothesis would need to be almost exactly equal to the null hypothesis. Another potentially more powerful solution to this problem would be to decrease the alpha value and increase the confidence interval to a much more precise level.

Another key result from the linear regression analysis involves field condition versus batting average. The results of this regression can be seen below in figure 6.

Figure 6.

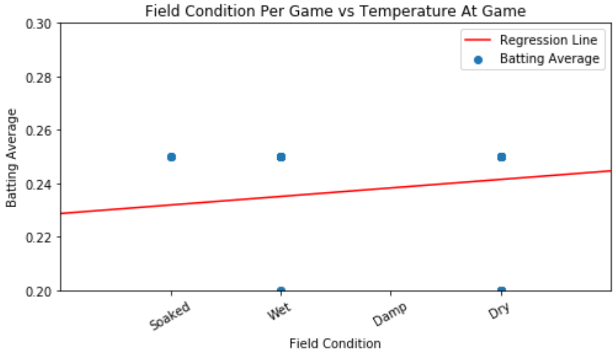


Figure 6 shows a positive correlation between field condition and batting average. In other words, batting average increases the dryer the field is. The figure above is a cropped version of a much larger graph. The rescale makes the positive correlation more readily visible.

The most significant result from the linear regression analysis was between temperature and batting average.

Figure 7.

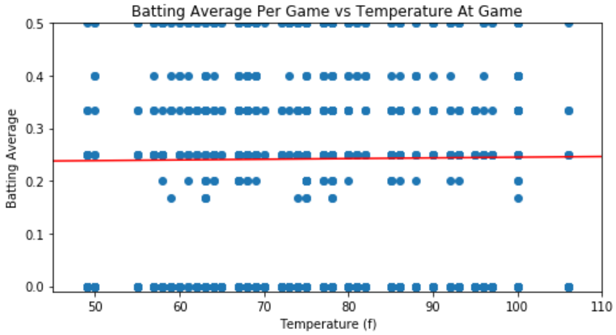


Figure 7 shows a positive correlation between temperature and batting average. Again, this figure is a cropped version of a larger graph that makes the correlation more obvious. Each point in this graph represents the batting average of a single player for one game versus its corresponding temperature.

The result of the random tree regression and feature extraction shows a comparison of how correlated each factor is to batting average.

Figure 8.

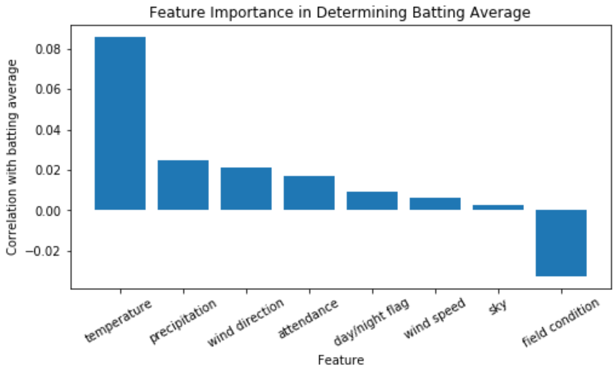


Figure 8 shows the relative correlations between all examined features and batting average. Temperature has the greatest absolute value correlation, and sky conditions have the least. Field condition shows a negative correlation because of the data's structure. In the datasets, smaller field condition values represent dryer conditions. As field conditions worsen, the numeric representation increases, and batting average decreases.

Generally, what all of our data analysis shows is generally what one would suspect; player performance tends to increase as the weather conditions improve.

Finally, through our analysis of high pressure situations, we were able to draw a few insights on clutch playing. Figure 9 shows the results of this analysis.

Figure 9.

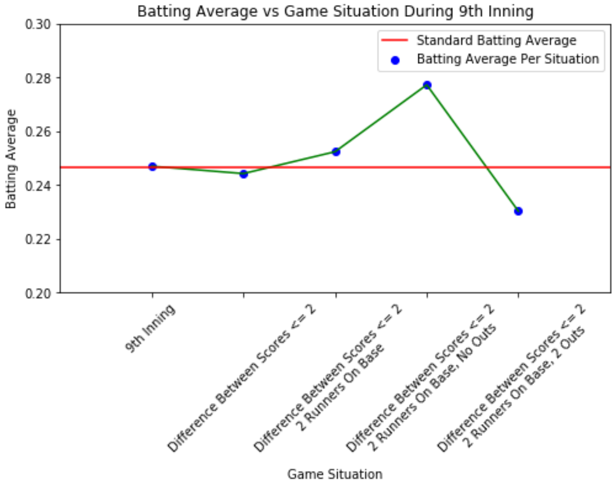


Figure 9 shows the batting average in the various high-pressure-late-game situations described previously. The red line shows the batting average under no special circumstances, and the blue points show the batting average per unique situation. The situations generally increase in pressure the further one moves along the x axis, which shows some interesting insights about clutch playing.

At first glance, one might conclude that clutch playing does exist, and measurably so, because through statistical hypothesis testing, we were able to reject the null hypothesis at the 95% confidence level and show that players performed better in those late game situations, especially when the game was in the 9th inning, the difference in scores was less than or equal to two, there were two runners on base, and there were no outs. In this specific scenario — a significant high-pressure situation — players performed noticeably better than they did otherwise. However, in the same situation but with two outs as opposed to none, an even higher-pressure situation still, player performance drops significantly.

This observation is worthy of drawing pause. Perhaps it is the case that clutch performance does not follow a linear relationship with the pressure of a situation — perhaps there is an ideal level of pressure under which players perform best. Our tests do show that players perform better under certain high-pressure situations, but, in order to gain a more holistic insight into clutch playing, further analysis needs to be done.

This future analysis should include a more precise measure of high pressure situations, perhaps taking the idea of a multivariate function to determine the pressure of a situation, or perhaps using a metric that already exists. Using this, one could perform a similar forest regression and feature extraction, among other methods.

## Applications

The knowledge gained from this project could be utilized in various ways. One application could be strategic adjustment. Individual players may want to adjust their playing strategy based on outside factors. Coaches may also want to adjust their strategies, and scouts may want to adjust their data based on the outside factors of the games observed. Perhaps coaches and fields could work together to try and make these factors work in their favor.

The correlations calculated here may have other sabermetric applications as well. Perhaps these correlations can be used in complex sabermetric equations. And baseball statistics enthusiasts may find the correlations themselves of interest.

The results from this project also open the door for more questions to be asked and for more studies to be conducted. Why is there such a variance in batting average between different high-pressure situations? Is there an ideal level of pressure under which players hit the best?

**Citations**

Brooks, H. (1988). The Statistical Mirage of Clutch Hitting. *Baseball Research Journal*. Retrieved from http://research.sabr.org/journals/the-statistical-mirage-of-clutch-hitting

Lindholm, S. (2014, March 27). Hitting and temperature. Retrieved from https://www.beyondtheboxscore.com/2014/3/27/5546952/hitting-temperature-baseball