**A Comparison of Cluster and Estimation Model Methods  
 for Evaluating MLB Player Wins-Above-Replacement (WAR) Values**

Joseph M. Lennon

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Data Science

Department of Mathematical Sciences

Central Connecticut State University

New Britain, Connecticut

October 2019

Thesis Advisor

Dr. Daniel T. Larose

Department of Mathematical Sciences

**A Comparison of Cluster and Estimation Model Methods  
 for Evaluating MLB Player Wins-Above-Replacement (WAR) Values**

Joseph M. Lennon

An Abstract of a Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Data Science

Department of Mathematical Sciences

Central Connecticut State University

New Britain, Connecticut

October 2019

Thesis Advisor

Dr. Daniel T. Larose

Department of Mathematical Sciences

Key Words: K-Means cluster, BIRCH cluster, principal components, multicollinearity, CART, Random Forests

**ABSTRACT**

Wins-Above-Replacement (WAR) is a metric that estimates overall player value in MLB. It has a proprietary formula that involves many hidden calculations and brute-force adjustments not fully published by the agencies that track it. The purpose of the study is two-fold. First, the study determines the optimal descriptive methods for profiling high vs low WAR players through cluster analysis. Further, the study develops alternative estimation models for WAR values using readily available player statistics only, as opposed to proprietary methods, and highlights the strengths and weaknesses of different methods used to profile and estimate WAR values for individual players.

Descriptive and estimation methods for analyzing WAR values are presented. Methods explored include clustering methods, both K-means and BIRCH, principal components analysis and estimation methods including CART and random forests models.

The clustering analysis illustrate that several useful cluster solutions were possible and the trade-offs of different methods in selecting the most useful number of clusters. Both K-Means and BIRCH illustrate a trade-off between cluster cohesion, in the forms of higher silhouette values, and in finding interesting additional subsets of players which are useful in profiling different player types. Principal components analysis shows the usefulness of variable reduction in profiling WAR, and in improving the ability to make inferences about drivers through eliminating multicollinearity of predictors.

CART models in all cases are with random forest models using three different subsets of variables to estimate WAR. CART models are shown to be prone to overfitting, while random forests models are shown to provide improved accuracy at the expense of some simplicity and speed of execution. Strengths and weaknesses of different variable subsets, specifically comparisons of raw variables to the principal components, highlight the increase in estimation error rates when using principal components and a reduced set of variables vs. the raw variables. Mitigating multicollinearity in the predictors is shown to improve the ability to make accurate inferences about drivers, while increasing estimation error rates in estimating WAR values. The preferred methods illustrate the trade-offs possible between explanatory inference vs. accuracy of estimation for player WAR.

The study illustrates that transparent mathematical models can be utilized to both profile and estimate MLB position player WAR values. These models have a major advantage over the proprietary methods since the models better illustrate the drivers of WAR, require little to no baseball domain knowledge to execute, and use readily available baseball statistics only available to the casual non-sabermetrics expert.

**TABLE OF CONTENTS**

[I - INTRODUCTION 6](#_Toc22823037)

[II - DATA SOURCES and ANALYTICAL DATA SET PREPARATION 11](#_Toc22823038)

[III – EXPLORATORY DATA ANALYSIS 15](#_Toc22823039)

[IV - TEST VS. TRAINING SAMPLES DEFINED, RAW VARIABLES PREPARED FOR CLUSTER ANALYSIS AND ESTIMATION MODELING 45](#_Toc22823040)

[V – CLUSTER MODELS, K-MEANS & BIRCH 46](#_Toc22823041)

[VI – PRINCIPAL COMPONENTS ANALYSIS 54](#_Toc22823042)

[VII – ADDRESSING COLLINEARITY IN THE ORIGINAL VARIABLES 61](#_Toc22823043)

[VIII – ESTIMATION MODELS - CART 64](#_Toc22823044)

[IX – ESTIMATION MODELS - RANDOM FORESTS 69](#_Toc22823045)

[X – ESTIMATION MODELS - TESTING AND COMPARISONS 72](#_Toc22823046)

[XI – CONCLUSIONS 76](#_Toc22823047)

[XII - REFERENCES 81](#_Toc22823048)

[XIII - APPENDIX - DETAILED WAR EXPLANATION FROM *BASEBALL REFERENCE* 86](#_Toc22823049)

I - INTRODUCTION

The overall purpose of the capstone project is to compare different modeling approaches to estimating the Wins-Above-Replacement statistic, an overall measure of value for major league baseball players.

Wins-Above-Replacement or “WAR” as it is commonly referred to, is a complex major league baseball statistic, that is used as a single overall measure of an individual player’s value to a baseball team. The WAR statistic was developed by the sabermetric baseball community. Sabermetrics is a term coined by Bill James in 1980 of the Society of American Baseball Research in 1971. Sabermetrics is the statistical analysis of baseball records, especially to compare individual players to one another.

WAR values, the metric estimated for the project, are published, though the formulas for calculating the WAR values are not publicly available. The exact methods to calculate WAR are proprietary. Thus, we will be using data scientific methods, such as random forests, regression trees, to *estimate* these WAR values, given the players’ other characteristics. Further, we will learn about interesting groupings of players through cluster analysis, and interesting groupings of predictors through principal components analysis.

The WAR metric is not officially recognized by Major League Baseball (MLB). However, it is a widely researched and debated metric used in sabermetrics community to value MLB players. WAR is often used to compare player’s value to one another. Player WAR estimates have been shown in several different studies to be highly correlated with total runs and total wins that an MLB team produces in a season. The ability for player WAR to accurately estimate team wins and measure player value is why WAR is gaining so much popularity.

Several different competing methodologies exist for calculating what goes into a WAR estimate. For this study the actual WAR values that are estimated with models use the *Baseball Reference* definition of WAR. Other estimates for WAR with competing methodologies are used by *Fangraphs* and *Baseball Prospectus* among others sport media outlets.

The values estimated are limited to non-pitchers, position-players only, since the components of what is behind WAR is substantially different between pitchers and position players (non-pitchers). Further, the capstone project uses only individual level metrics in estimating war value, so the models produced can theoretically estimate a player’s WAR value using only widely available information about that player. The estimation models compared will provide a “ready-reckoner”, or relatively simple estimate of actual WAR, without using all the underlying data in the actual statistic, including league average assumptions that sabermetrics practitioners use to create the actual WAR estimates.

Several different approaches to modeling estimated war values are used including, cluster membership, principal components analysis, regression trees and random forest estimation models. The common thread among all the approaches in the study is every model uses an individual player’s statistics and characteristics alone in estimating that player’s WAR value. This approach contrasts actual WAR calculation process which is both proprietary in nature and utilizes explicit comparisons of one player’s statistics to other player averages.

**Details about the Baseball Reference Definition – The WAR target used in the Study**The WAR calculation explicitly compares an individual player to an average player and replacement player performance which are difficult to obtain and calculate directly using publicly available information.

The WAR details described within the capstone are available in the Baseball Reference website, where the source describes their methods in some detail but do not provide exact calculations.

WAR is a metric meant to show how much better a player is than another player who would typically be available to replace them. To calculate WAR players are compared to the average players in various attributes related to creating wins, then compared to a theoretical “replacement” player. The final WAR statistic is intended to represent the number of wins (converted from runs) a player is above another player that could replace them. Positive WAR players are good, zero to negative WAR players, should be replaced by someone better.

There is no agreed upon framework for the one way to calculate WAR. Several different news and research agencies have their own permutations that they recommend. *Baseball Reference*, the WAR source for this study admits as much, saying“There are hundreds of steps to make this calculation, and dozens of places where reasonable people can disagree on the best way to implement a particular part of the framework (Baseball Reference, 2019, Baseball Reference.com WAR Explained, para. 2).”

WAR relies on the concept of replacement-value players. Since replacement players are relatively easy to obtain compared to average players, they are a more useful comparison point than the average player. Even average players are relatively expensive in major league baseball, whereas replacement players are typically the more cost-effective alternative when making MLB roster decisions.

**Actual WAR Is Complex and the Calculation Details Are Not Completely Published**

The actual WAR calculation is quite complex and is based on six underlying components. The six components are batting runs, base-running runs, runs added or lost due to grounding into double plays in DP situations, fielding runs, positional adjustment runs and replacement level runs (based on playing time). WAR’s six different components are estimated by *Baseball Reference.* No exact calculation for these components is provided as their methods is proprietary. *Baseball Reference* does not and will not provide a fully worked example of the actual WAR components.

The first five components above are all compared against league average, so a value of zero will equate to a league average player, while the sixth factor above is the second half of the equation below (Baseball Reference, 2019, Position Player WAR Calculations and Details, para. 2).

**Equation 1.1 - Baseball Reference WAR**

***bWAR = (Player\_runs - AvgPlayer\_runs) + (AvgPlayer\_runs - ReplPlayer\_runs)***

A brief conceptual summary with some of what each of these six components for one player represents is provided in the appendix to provide some sense of the complexity and content of each.

Clearly, one challenge of the actual WAR statistic is the lack of complete transparency in how any of these components are calculated. Further, as equation 1.1 shows, actual WAR explicitly compares individual players to other players requiring several underlying averages to calculate an actual WAR values for an individual player.

**Modeled Estimates for WAR Are Comparatively Simple and Leverage Variables that Are Far Easier to Observe and Track Directly**

By contrast to actual WAR, the estimation models built for the study use easily trackable, player performance statistics (i.e., hits, runs, batting average etc.) to estimate WAR, simplifying the process of estimating WAR therefore, not requiring 100s of steps, or explicit comparisons to other player averages, which are largely unknown to the public. The models will not require detailed understanding of the actual WAR calculation used by sabermetrics experts.

The capstone project focuses on different estimation models for WAR, comparing and contrasting strengths and weaknesses of each model type and will use publicly available individual statistics as the predictors for each player’s estimated WAR value. The estimation models below use individual statistics for players only, while the actual WAR values use difficult to obtain variables (like league average runs as an example), which are only known and available to sabermetrics experts, in this case *Baseball Reference*. As noted above different tracking agencies, such as *Fangraphs,* have their own competing proprietary methodology to track WAR actuals.

The project uses WAR values from *Baseball Reference* and individual player data as of June 20, 2019. The ability of the different model types to accurately estimate player WAR values, and the outlining the strengths and weaknesses of each model approach will be the project focus.

**Questions of interest in the study include the following:**

1. Can cluster models be used to produce useful descriptions of various player groups? How many groups of players (clusters) are preferred? Which clustering algorithm produces the most useful clusters? Models evaluated include K-Means clustering and BIRCH clustering.
2. How can principal components analysis be used in reducing the dimensionality of the data, finding groups of useful predictors and alleviating multicollinearity? How many different principal components are useful in the data? Many of the predictors in the study are highly correlated with one another. Multicollinearity among predictors has been seen in other studies to create unstable model estimates, and limits the ability to make accurate inferences about predictor’s impact on a target variable (Larose & Larose, 2015). CART estimation models, which are some of the models created in the study, have been shown to be particularly susceptible to multicollinearity issues (Hastie, T & Tibshirani, R & Friedman, J, 2008).
3. Which estimation model approaches yield the best model to estimate player WAR for each player a given year? What are the underlying model strengths and weaknesses when comparing methods? Models evaluated include CART and Random Forests.

II - DATA SOURCES and ANALYTICAL DATA SET PREPARATION

Baseball statistics by player by season were combined using R programming language to create a dataset for analysis. Two main sources used for the project which are both keyed by player id and year are described below.

**Source 1 - Sean Lahman Baseball database – Player Records since 1871**

This database is an updated version of sports journalist Sean Lahman’s baseball history database. The database contains complete batting and pitching statistics from 1871 to 2018, plus fielding statistics, standings, team stats, managerial records, post-season data, and more.”

Specifically this data located at <http://www.seanlahman.com/baseball-archive/statistics/>, which is publicly available was used as one main data input. The database itself is copyright by Sean Lanham. Terms of use of this data for research purposes appear to be broad if attribution is provided. See this site <https://creativecommons.org/licenses/by-sa/3.0/> for more specific details

This data is available for download in both a SQL database version and CSV tables. The CSV tables updated as of 2018 were utilized for the study.

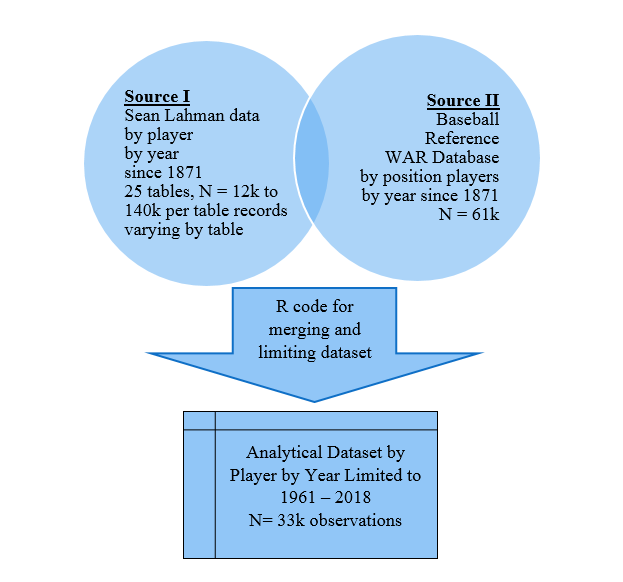
**Source 2 - Baseball-reference.com database -** [www.baseball-reference.com/data/](http://www.baseball-reference.com/data/).

This data was used to augment Sean Lanham’s data specifically to obtain player-level data on WAR (wins above replacement), and some of the components that go into WAR calculation. These statistics have garnered more mainstream popularity in more recent years. For purposes of the study the final WAR values provided from this data source are estimated by the mathematical models, without using the components, but rather from using the data in the Sean Lehman baseball database data (source 1), which is individual player statistics, excluding any WAR components.

**Combination of Data Sources and Data Cleansing**

The two data sources were combined by player, by year, by team, to yield a single analytical dataset for analysis. This combined data source has the same player and year keys that made the merging of the two data sets for analysis possible. Further, some cleansing of the team variable was executed as part of the data cleansing process to ensure all combined data records accurately reflect the sources and all usable records can be utilized for analysis. A conceptual visual of the data cleansing process which resulted in the final dataset is presented in figure 2.1 below.

**Figure 2.1 – Combination of Two Publicly Available Data Sources  
Using R Code Creates the Dataset for Analysis**



The resulting dataset used for the analysis provides a source of ~33k records by player by team by year since 1960 and more than 100 variables by player to analyze which variables and which models are most effective in estimating player WAR using mathematical models as opposed to *Baseball Reference’s* proprietary methodology for actual player WAR. A sample data record for Mookie Betts 2015 season is illustrated below to provide a detailed view of the single record of analysis for the project. For purposes of simplicity, the dataset for analysis was limited to 1960 forward. Data prior to 1960 is available, however, not as many dimensions different are clearly populated in older data, like fielding metrics as an example.

Additionally, league structure and team structures differed among the two datasets. Some of the team naming conventions between the two sources differed particularly among the older data. To limit the amount of recoding needed the dataset to avoid losing player observations, the data from 1961 forward was used for the remainder of the project. The more recent years since 1961 are used for analytical purposes out of convenience in merging the datasets in to one analytical dataset for analysis. Upon initial dataset construction, some observations of the by team, by player, by year framework were dropping out of the dataset as they were associated with teams that are no longer part of the major leagues. This data was recoded to ensure a match on team naming convention between the databases to ensure all observations were included in the dataset.

**Table 2.2 – Merged Record - Mookie Betts  
Example Record After Merge Process – Sample Fields, 90 fields total are available**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | example | Variable | example | Variable | example |
| year | 2015 | triple | 8 | sf.x | 6 |
| player\_id | bettsmo01 | hr.x | 18 | g\_idp.x | 2 |
| team\_id.x | BOS | rbi | 77 | batavg | 0.291 |
| stint.x | 1 | sb | 21 | OPS\_plus | 0.342 |
| league\_id.x | AL | cs | 6 | MVP Points Won | 4 |
| g.x | 145 | bb.x | 46 | MVP votes\_first | 0 |
| ab | 597 | so.x | 82 | age.x | 22 |
| r.x | 92 | ibb.x | 1 | PA | 654 |
| h.x | 174 | hbp.x | 2 | Inn | 1263 |
| double | 42 | sh.x | 3 | **WAR** | **5.86** |

III – EXPLORATORY DATA ANALYSIS

Following the data cleansing and merging operation a full exploratory data analysis was performed with a focus on how the potential predictors relate to WAR, the target of interest for estimation. Opportunities to transform various predictors to improve predictive power for estimating WAR values were explored as part of the exploratory data analysis. The exploratory data analysis was performed using SPSS/IBM modeler and R.

**The units of measure – The Players’ Annual Statistics, 33k observations since 1961**

Each record in the dataset is a player, by team, by year record with a WAR value for that player for that year and team. Position-player WAR is sourced from *Baseball Reference* Data as of May 20, 2019. Applying the logic to the data sources as outlined in section two, results in the distribution below. As can be seen below in Chart 3.1, WAR for position players is on average slightly above zero, at an average value of .86. This can be interpreted as the average player in the dataset has a WAR value of slightly less than one win above replacement value. The range for WAR has a long tail to the right with the best players ever having a WAR value of greater than nine. The typical player based on the average is a slightly less than a one WAR player, as most players are between zero and one wins above replacement.

Some pitchers did have both position player WAR estimates available and hitting statistics available. Most of these players were from National League teams where pitchers hit. For purposes of the study only position players were considered for analysis and all pitchers were excluded.

All players that had any plate appearances during a year are included in the analysis. WAR is correlated with playing time, as players that don’t contribute to winning see less action on the field and fewer at bats. The negative WAR players from the distribution below, tend to have lower plate appearances since they are not performing, they lose playing time to better players.

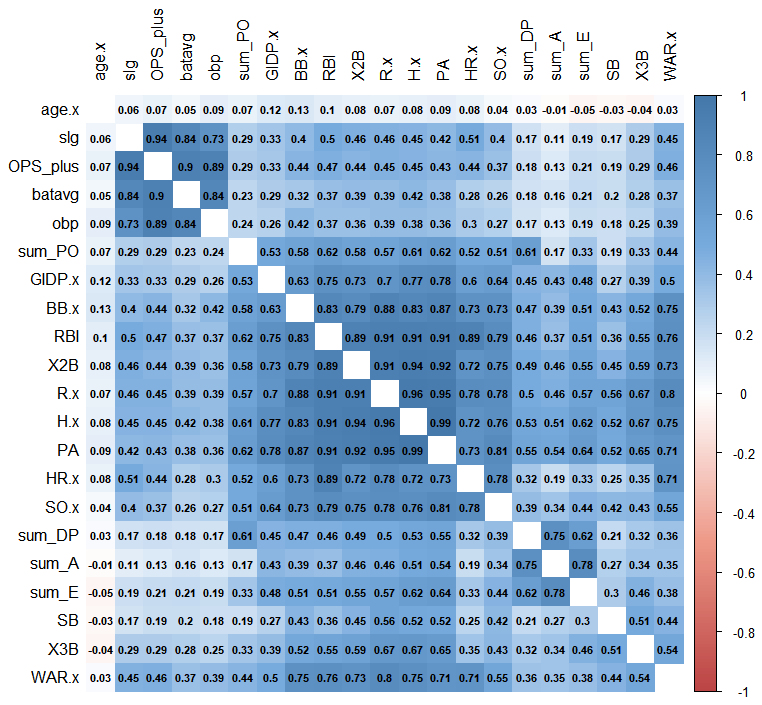
**Chart 3.1 - The Target for Estimation – WAR, Position Players 1961 to 2018**

The Pearson correlation matrix in Figure 3.2 is used as an exploratory tool which summarizes many of the available predictors’ relationship with WAR. Preliminary examination of initial Pearson correlation coefficients below indicates many potential predictors are highly correlated with WAR. Further extensive EDA of the predictors that follows illustrates opportunities for potential strong predictors and transformations of the predictors that can be used in estimating WAR. The correlation matrix below, which is a subset of the variables available, contains 20 different statistics that are all readily trackable by the casual baseball fan. These variables and transformations of them are used in the models that follow to estimate WAR values.

The range of correlation in the 20 statistics shown below in Figure 3.2 are interestingly all positive with WAR. Some negative outcomes, such as hitting into double plays, labeled *GIDP* below, have a positive correlation with WAR of 0.36. Fielding errors, below labeled *SUM\_E*, are also correlated positively with WAR with a Pearson correlation of 0.38. These negative actions are positively correlated with WAR since WAR is so highly correlated with playing time. The players that get playing time get more opportunities to make more errors and ground into more double plays. Transforming some of these counting statistics for fielding into rate statistics is explored in the analysis and models that follow.

As mentioned above the full dataset has 25 tables and over 100 variables to look at, covering most aspects of the game. A full listing of all variables available for analysis is listed in the appendix.

Following the simple correlation matrix shown above, the most common statistics measured in baseball were explored against the target WAR to assess which variables have the most potential as predictors in the multivariate models that follow in the later sections of the project.

**Figure 3.2 - Pearson Correlation Matrix of Some Potential Predictors**

The by-decade WAR data in chart 3.3 shows some variation, with slightly lower overall WAR for position players in the last decade. A dummy variable indicating decade for the player may potentially prove useful in the full models and was created for testing based on this exploratory data analysis. The most recent time frame, the 2010s has the lowest WAR values in the sample. Additionally, there are more different players, getting playing time in the major leagues in the latest decade compared to the 1960s. The evolution of roster management with front offices acting more to manipulate major league rosters throughout the season than they did in the past may partially account for this decrease in average WAR value.

WAR has shown a slight decline in the last ten years compared to the prior decade. The WAR statistic itself has undergone methodological changes in this time frame as outlined by *Baseball Reference*, though the paper uses the baseball prospectus methodology as of June 2019.

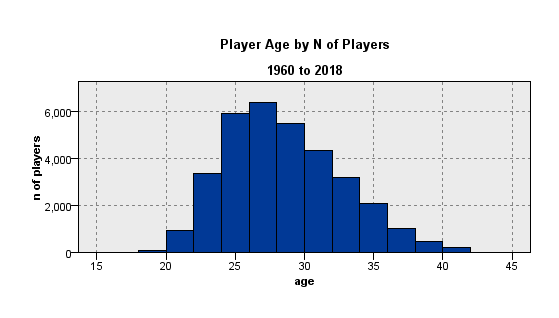
**Chart 3.3 – WAR - By Decade**

**Chart 3.4 - WAR - By Player Age**

Player age and WAR follows the conventional baseball wisdom that peak performance is typically at approximately the age 30 season and then typically steadily declining after the age of 30 for most players. Note there is some departure from that pattern for players in their later 30s. Player age has a correlation with WAR close to zero as seen above in the correlation matrix because of this humped distribution of WAR vs. age. This suggests that a transformation of the age variable may be useful as a predictor.

Most players in MLB and the dataset are under 30 as seen by the distribution in chart 3.5.

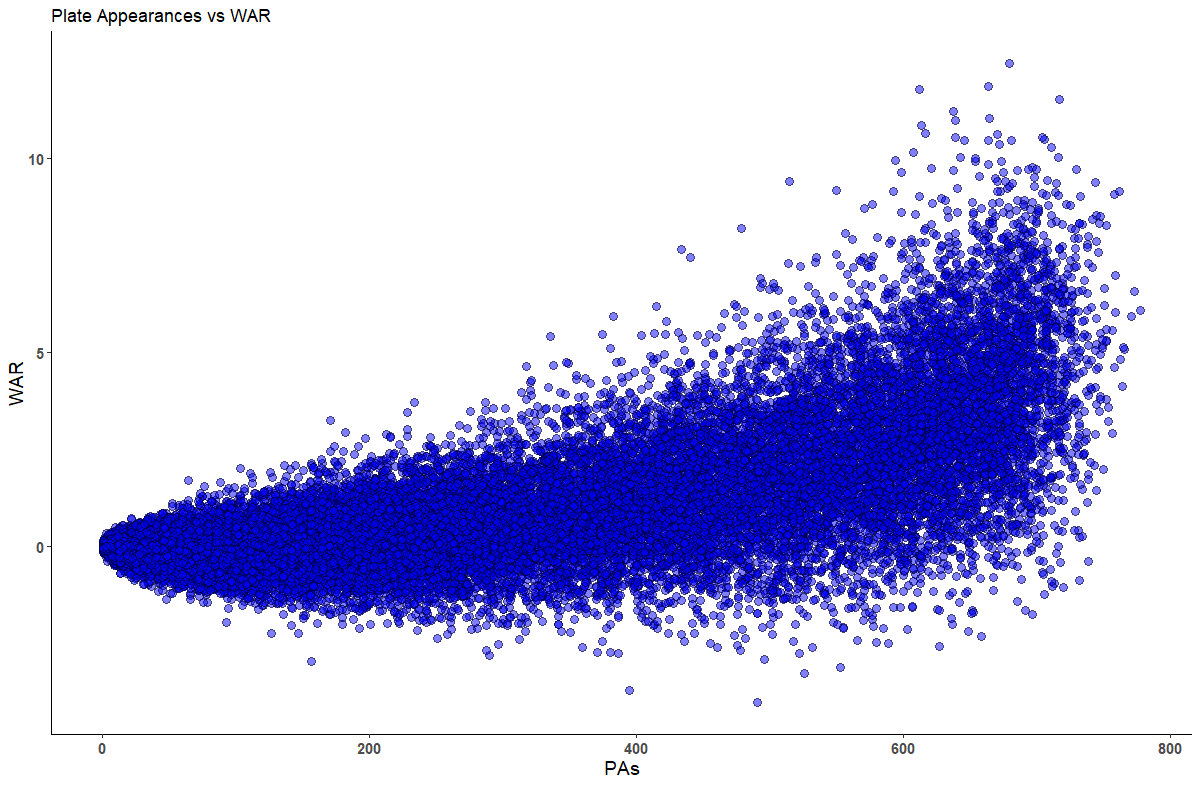
**Chart 3.5 - Player Age - Most Players Are Under 30**

****

**Plate Appearances Look to be Very Important when Measuring WAR**

Player plate appearances (chart 3.6), which unlike at-bats include walks, has a strong correlation with WAR of .71 based on the correlation table above in Figure 3.2. This also intuitively makes sense as plate appearance increase for better players as they get more playing time. Very low plate appearance players have negative WAR. This also makes some intuitive sense as positional players with low playing time or plate appearances are likely to younger, platoon players or having limited MLB playing time compared to higher WAR players. As players don’t perform their playing time is decreased.

**Chart 3.6 - Plate Appearances vs WAR**

****

Based on preliminary analysis and plate appearances high correlation with WAR, all players with any plate appearance were included in the analysis in order to more accurately capture estimating negative WAR player’s which seem to have low numbers of plate appearances.

Plate appearances also has a non-linear relationship with WAR when looking at the scatter plot above in chart 3.6. The average number of plate appearances in the data from 1960 to 2018 is 264 with some of the players with the highest WAR also having the most plate appearances at more than 600 PAs in a season.

Given the apparent non-linear increasing relationship between plate appearances and WAR in the models that follow the plate appearances squared (PA^2) variable was created to model this non-linear relationship. Plate appearances is also highly correlated with runs scored at a correlation of .96. Given this very high correlation between these two useful looking predictors an interaction term between plate appearances and runs (PA\*R) was also created for potential inclusion in the estimation models.

**Chart 3.7 - Plate Appearances vs WAR**

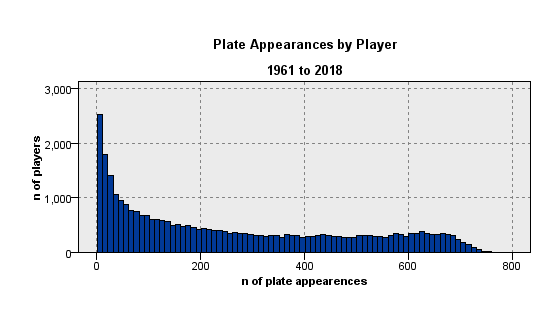
****

Chart 3.7 above shows the heavy right skew of the number of players by plate appearances. Many players in the data, approximately 2,500, have very few plate appearances in a year looking at all the years of data available since 1960. These are more likely to be replacement players who are either substituting for day to day regulars to due injury or performance reasons.

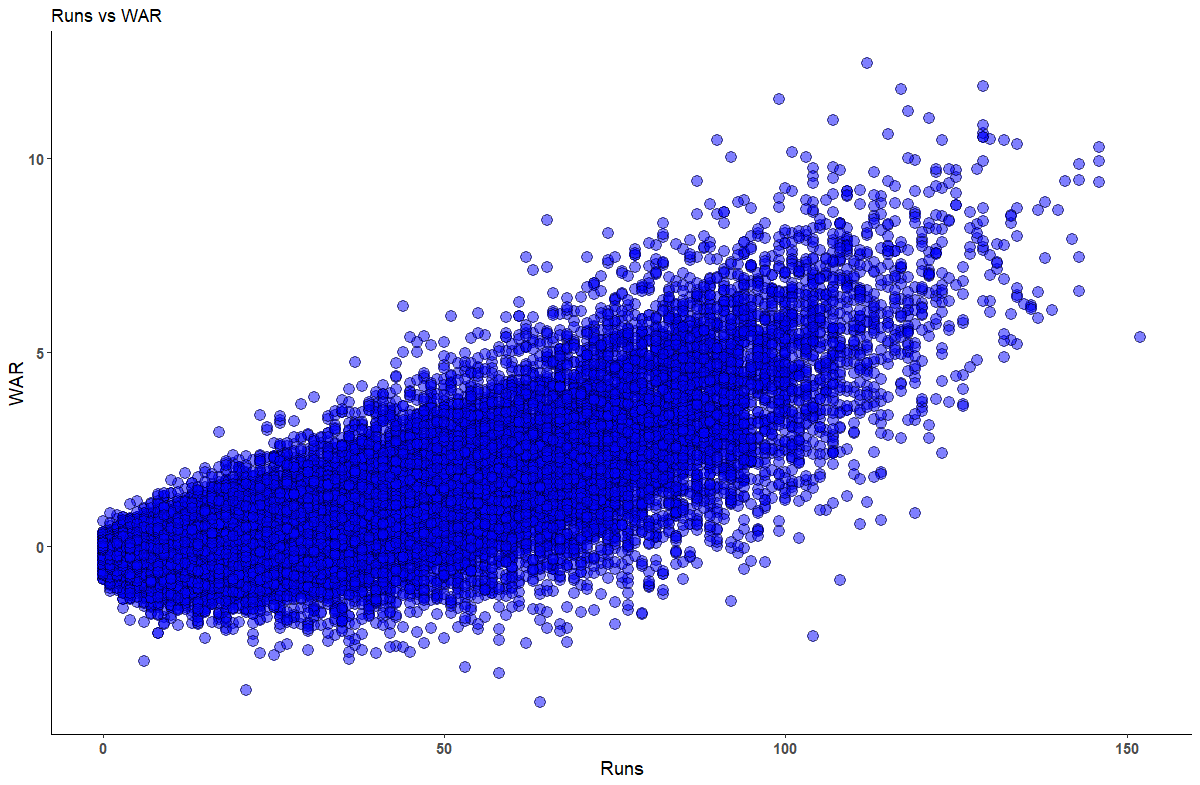
**Runs Scored by Each Player – A Main Ingredient to WAR**

Given the actual WAR statistic is primarily aimed at quantifying a player’s impact on runs, it is not surprising that actual runs scored by each player in a season is potentially a very useful predictor in estimating a player’s individual WAR value. This high correlation between runs and WAR is not unexpected as the WAR statistic itself is an estimate of run creation above replacement value for each player, as outlined in section one.

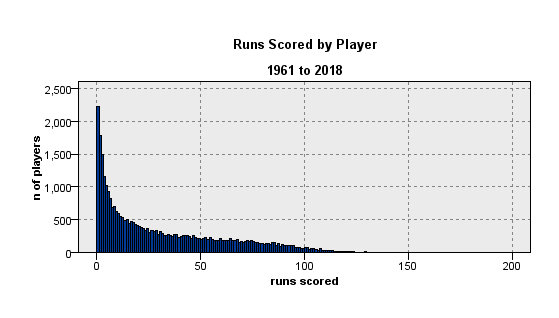
A scatterplot of runs scored by WAR in Chart 3.8 below shows a strong upward trend which curves upward as runs increase. Mean runs scored here is ~20. The most runs scored by any player in a season is 152 in one season for the game’s best players ever.

In the models that follow a runs-squared (runs^2) variable was created to model this non-linear relationship between runs and WAR.

**Chart 3.8 - Runs Scored vs. WAR**

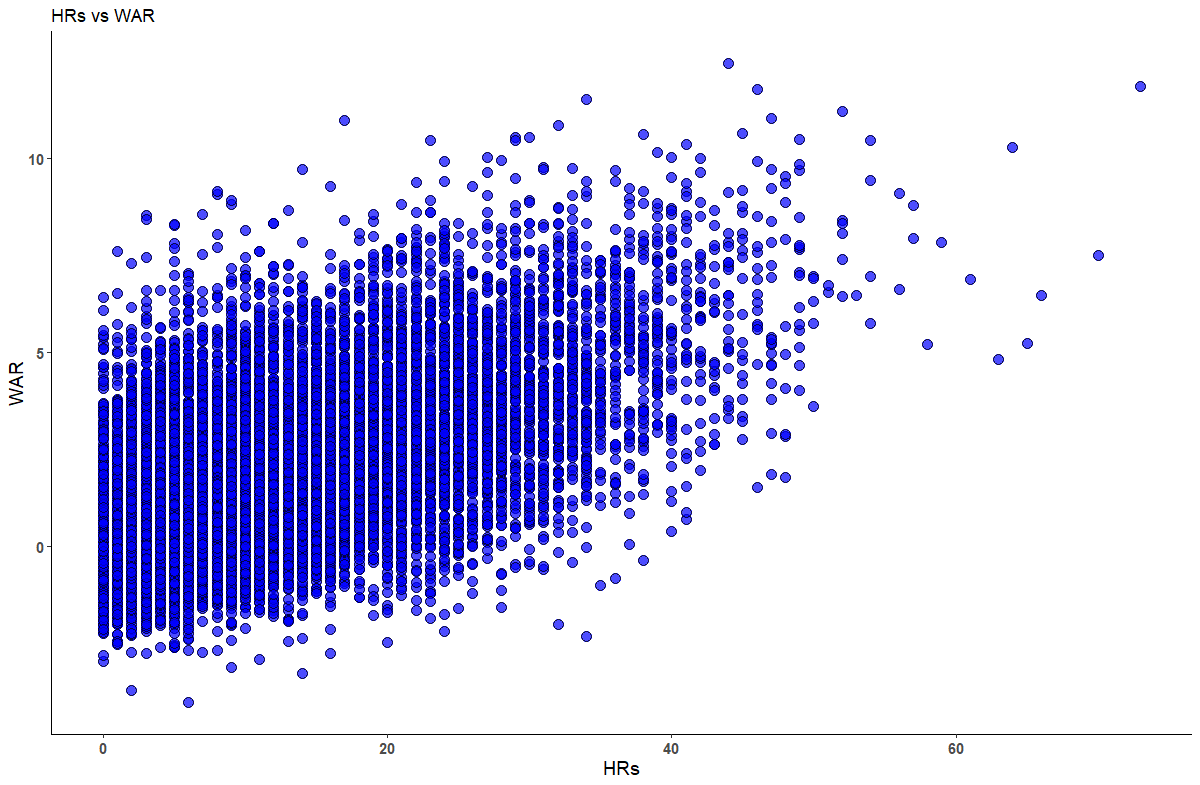
****

**Chart 3.9 - Distribution of Runs Scored – Highly Skewed Right**

****

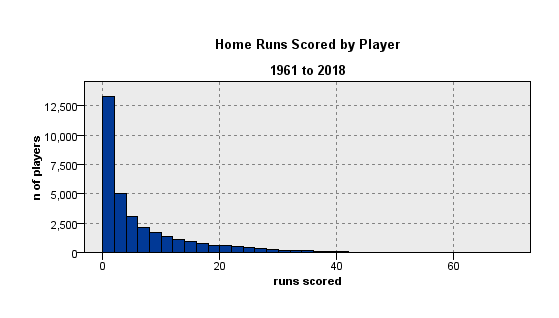
The distribution of runs scored is highly skewed right with most players scoring very few runs over the course of a season. This can be seen in chart 3.9.

**Chart 3.10 - Home Runs vs. WAR**



Home Runs by WAR is shown above in chart 3.10 and additionally is shown to be highly correlated with WAR (Pearson correlation of 0.73), indicating the home runs can also be an effective potential predictor in estimating WAR. The mean for home runs including all the players in the sample is 6.5, indicating that not all players hit home runs. The scatter above shows, fewer observations above the forty home run mark and overall linear relationship between home runs and WAR.

**Chart 3.11 - Distribution of Home Runs – Highly Skewed Right**

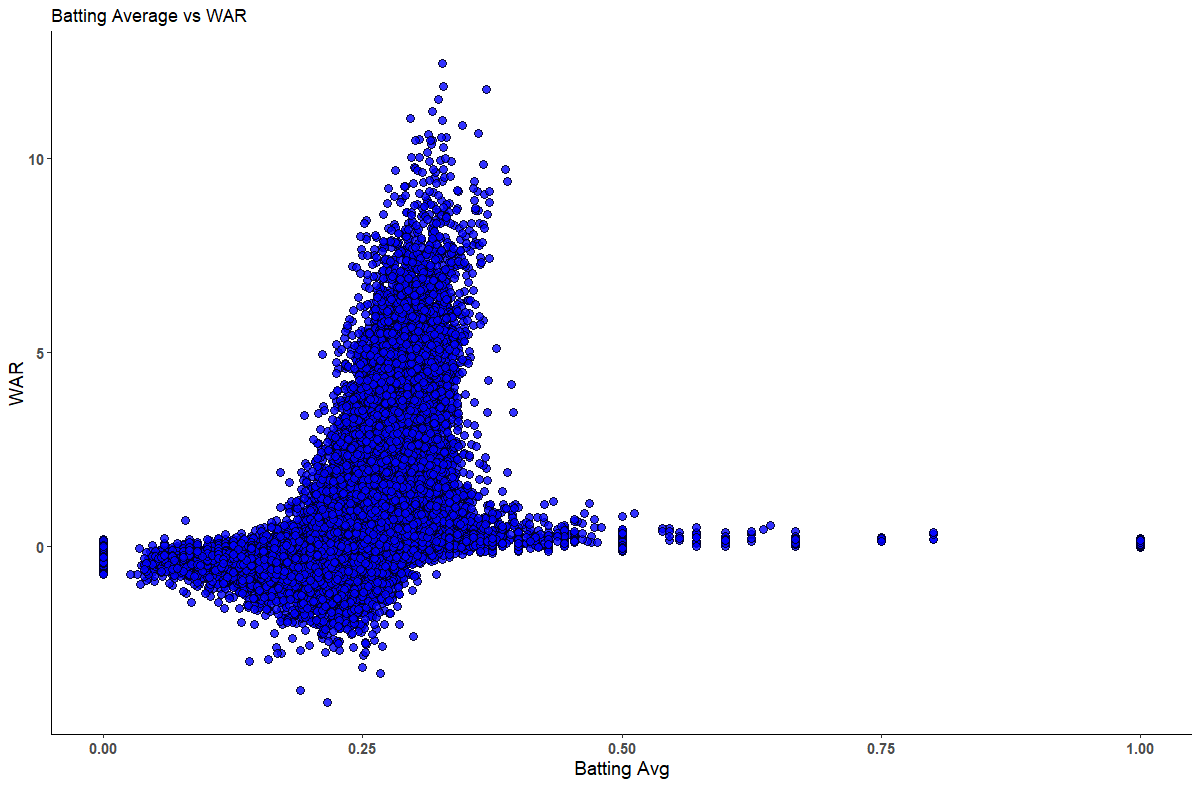
****

**Batting Average – A Complex Statistic for Estimating WAR Values**

Batting average is one of the oldest statistics tracked in MLB. Notably batting average excludes plate appearance that do not result in a hit or an out, such as walks or hit by pitch plate appearances. The formula for batting average is ***batting average = hits/at bats***.

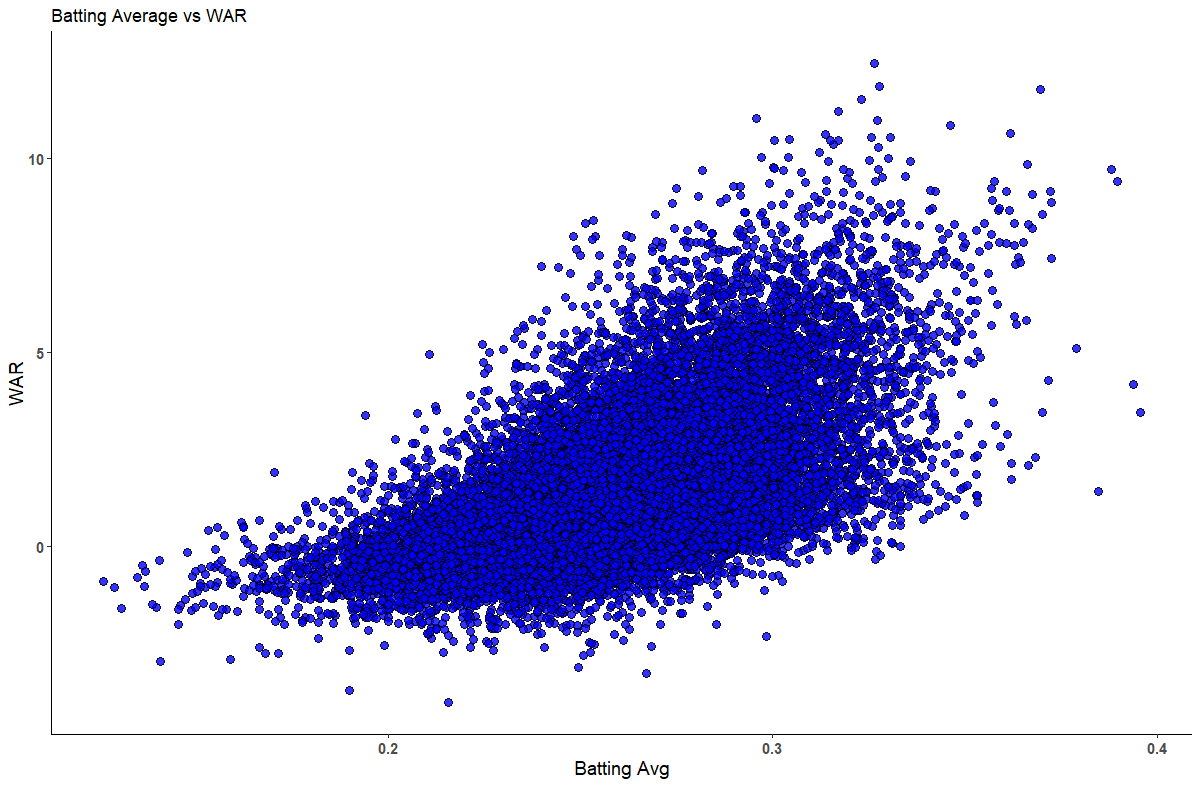
The batting average vs WAR in a scatterplot is shown in chart 3.12. As can be seen in the chart, the relationship between batting average and WAR is a complex one and not a simple linear relationship. Negative WAR players tend to have lower batting averages and the relationship between batting average and WAR appears to be positive exponentially as batting averages rise above ~.250 mark.

**Chart 3.12 - Batting Average vs WAR**

****

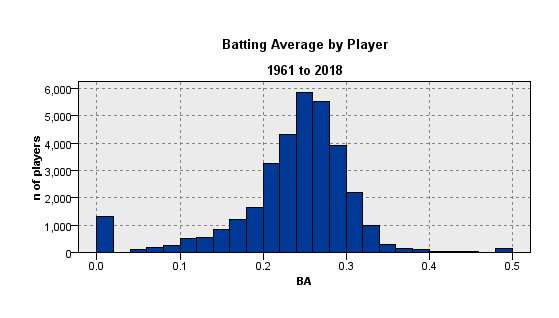
Inclusion of all players regardless of number of plate appearances further complicates the relationship as can be seen in the outliers with batting averages above .400 and around zero. These points on the above chart represent players that have only a few at bats or plate appearances. When MLB evaluates league leaders for awards, like MVP, silver slugger or statistical rankings the league often use a threshold number of at bats to limit the outliers when baseball fans look at which players are best hitters for average. The exploratory data analysis suggests including several higher order terms as potential variables. Batting average squared and batting average cubed were created to test to see if those variables are useful in the estimation models that follow. Chart 3.12 shows the large number of outliers in the unfiltered data with both batting averages of close to zero and one. These outliers on a statistic like batting average are important to include in this particular analysis of WAR, since so many low batting average players tend to have low, and in some cases negative, values for WAR. This complex pattern between WAR and rate statisitc is repeated for many of the rate statistics in the study.

**Chart 3.13 - Batting Average vs WAR With a Minimum 150 ABs**

****

Looking at batting average with a minimun number of at bats as in Chart 3.13 provides a clearer picture of the relationship between WAR and hitting for average that most causal basaeball fans talk about, exlcuding outliers. There is a positive increasing relationship between the two with the curve bending more (increasing) for batting averages in the .200 to 250 range. Note the few number of hitters in the history of baseball that have hit for average close to .400. The best hitters ever are succesful less than half the time.

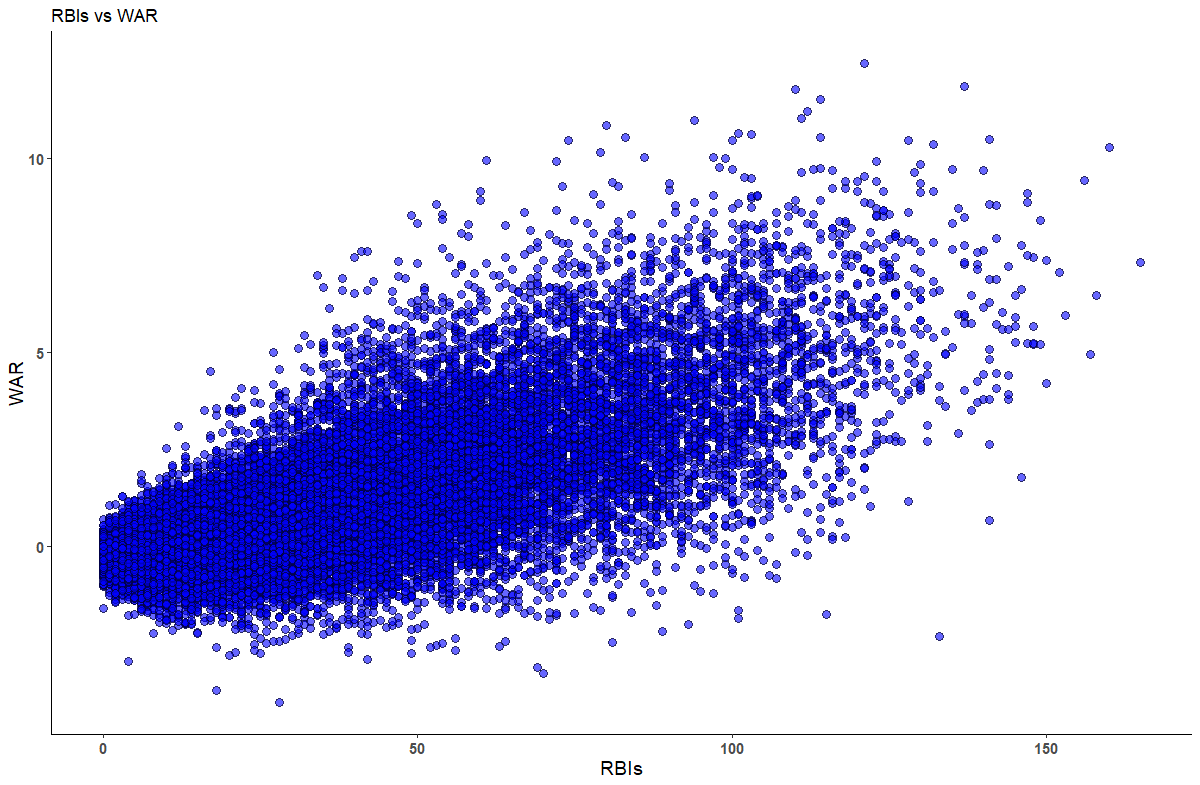
**Chart 3.14 - Batting Average by Number of Players**



**RBIs – Runs Batted In**

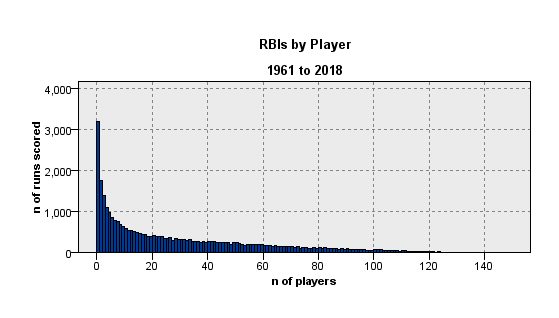
From MLB, “A batter is credited with an RBI in most cases where the result of his plate appearance is a run being scored. A player does not receive an RBI when the run scores as a result of an error or ground into double play. (MLB, 2019, Standard Stats, RBI).”

**Chart 3.15 - Runs Batted In – RBIs**

****

Runs battted in (RBIs) have a general linear relationship with WAR, partularly for players with at least ~20 RBIs in a year. High RBI totals are associacted with high run production. Interestingly, even some players very high RBI totals are still negative WAR players. The data is skewed right similar to many of the hitting counting statistics.

**Chart 3.16 - Runs Batted In by Number of Players**

****

**Slugging Percentage**

The next predictor, slugging percentage, is used to augment evaluating players by batting average alone. Slugging percentage as defined by MLB, “represents the total number of bases a player records per at-bat. Unlike on-base percentage, slugging percentage deals only with hits and does not include walks and hit-by-pitches in its equation. (MLB, 2019, Standard Stats, Slugging Percentage).”

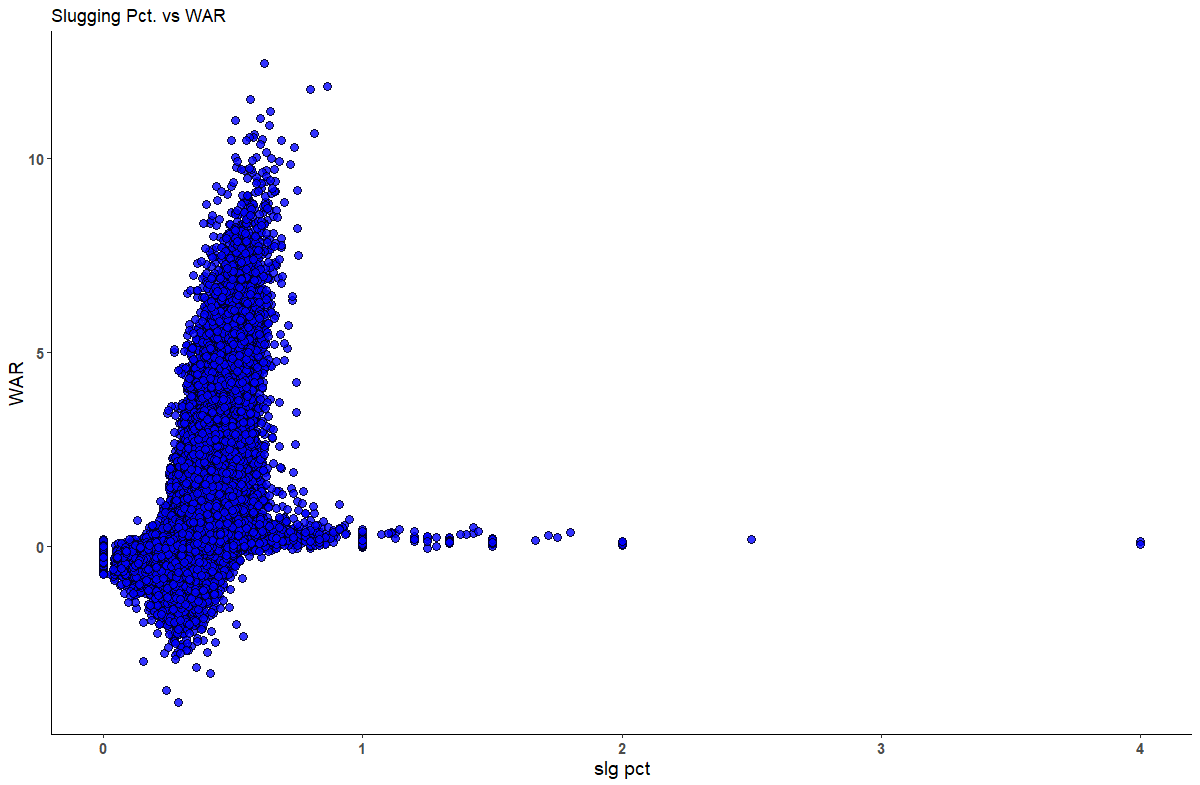
In a formula slugging percentage is expressed as follows:

***Slugging percentage = (1B + 2Bx2 + 3Bx3 + HRx4)/AB***

As can be seen in the chart 3.17, the relationship between slugging percentage is a similar pattern to batting average and WAR. The relationship is a complex one since slugging percentage is a rate statistic including all players with any plate appearances. Zero to negative WAR players tend to have lower slugging percentages and the relationship between batting average and WAR is positive exponentially as slugging percentage rises.

Like batting average, slugging percentage when including all players in the majors at all in a season has some significant outliers. These outliers are included in the study so as not to overestimate WAR.

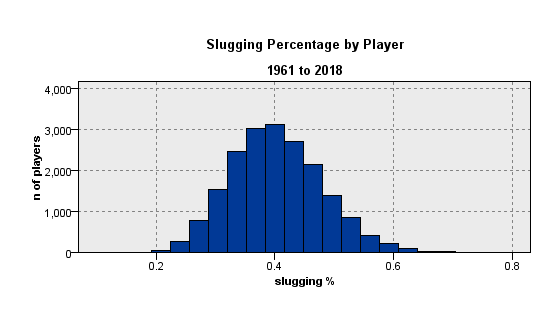
**Chart 3.17 - Slugging Percentage – Total Bases/At Bats vs. WAR**



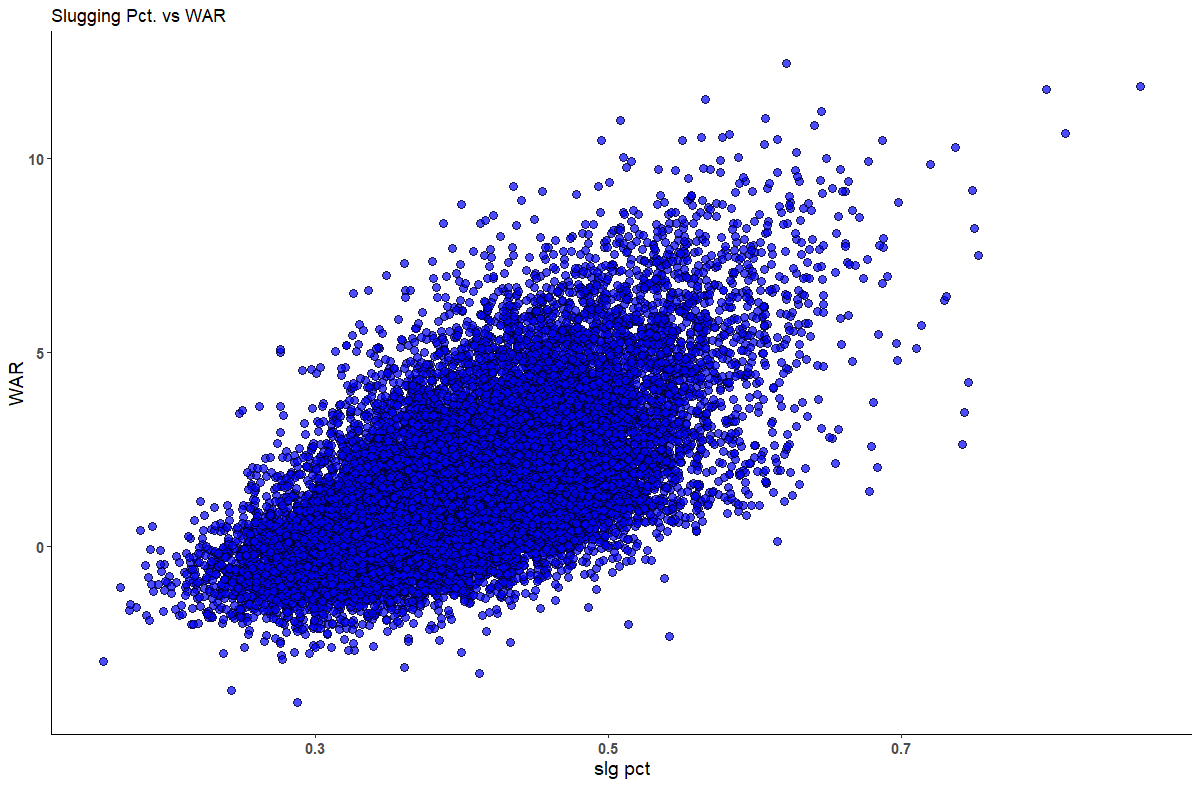
Some of the highest slugging percentage players have slugging percentages above one. Additionally, there are some extreme outliers on slugging percentage with these players likely having very few plate appearances.

Like other average stats, slugging percentage, when limiting to players with 150 at bats in a season has a steadier positive relationship with WAR. Slugging percentage looks to be relatively normally distributed when limiting by players with at least 150 plate appearances (chart 3.18) and has a positive relationship with WAR (chart 3.19).

**Chart 3.18 - Distribution SLG% by Player w/Min 150 PAs**



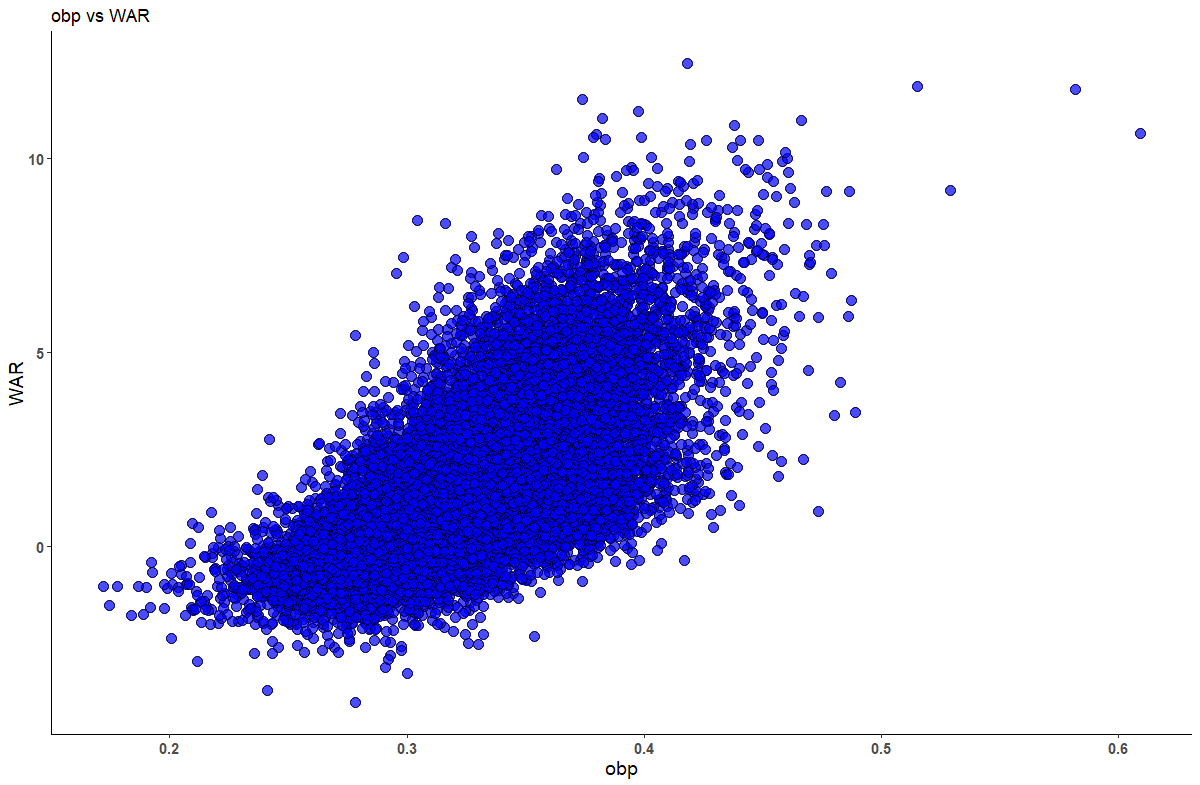
**Chart 3.19 - Slugging Percentage by Player w/Min 150 PAs**

****

**On Base Percentage, A More Complete Rate Stat than Batting Average**

On-base-percentage (OBP) is a metric that was first used in the 1940s to measure the rate at which a player reaches base. OBP gives credit for walks, hits and hit by pitches unlike batting average which only counts at bats and outs. Hitters that have high on base percentage tend to hit higher in the batting order as they can then be driven in by power hitters that tend to follow high OBP players in the batting order.

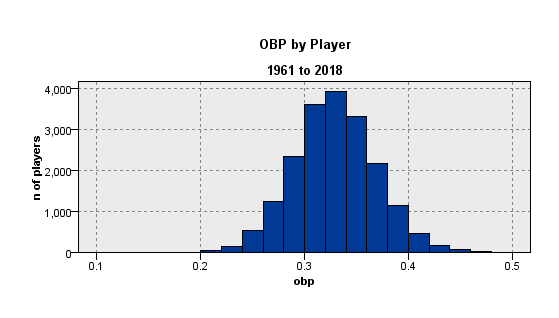
The unfiltered version of OBP (not shown), has a similar pattern in the scatter plot vs WAR as batting average as can be seen in Chart 3.20.

**Chart 3.20 - OBP vs WAR – Minimum 150 ABs/Year**

On-base-percentage and batting average are highly correlated (0.84) since they measure similar, but not identical baseball actions.

OBP has an exponential relationship with WAR when limiting to players with more than 150 at bats in a season. It has a similar relationship to batting average and WAR when not filtering out players. Chart 3.21 below shows that OBP is roughly normally distributed when looking at players with at least 150 at bats in a season.

**Chart 3.21 - OBP by Player**

****

**OPS+ – On-Base Plus Slugging, Plus**

On-base-plus-slugging plus, has in recent years been an increasingly popular metric that represents overall hitting capability. OPS plus, is another metric where different sources calculate it based on their own formula proprietary (like WAR). OPS Plus starts with on-base plus slugging or OPS, literally OBP% +SLG %. *Baseball Reference*, the source of OPS plus for this study, then adjusts OPS for home ballpark differences the player plays his home games in, increasing OPS for players who play in pitcher’s parks, and lowering OPS for those that play in hitters’ ball parks. Additionally, the number is normalized, so that the median is 100

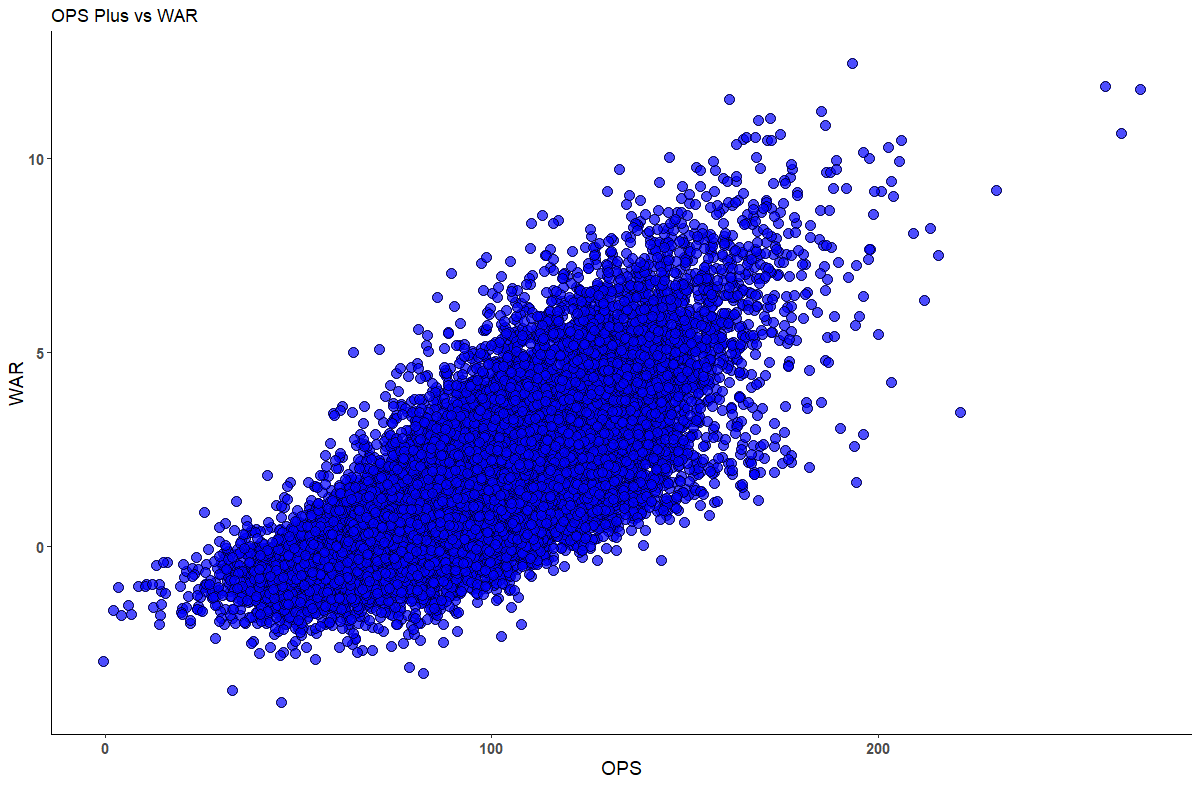
The formula for OPS+ is as follows:

***OPS+ = 100\*((OBP/lgOBP)+(SLG/lgSLG) -1),***

with lgOBP and lgSLG representing the league average for that statistic in that year (Baseball Reference, 2019, Bullpen, OPS).”

OPS plus can be thought of as OPS, normalized for the league and ballpark the player plays in.

**Chart 3.22 - OPS Plus vs WAR**

****

OPS Plus like other rate stats has a cleaner linear relationship with WAR when limiting the statistic to players with 150 at bats in a season.

**Chart 3.23 - OPS Plus by Player**

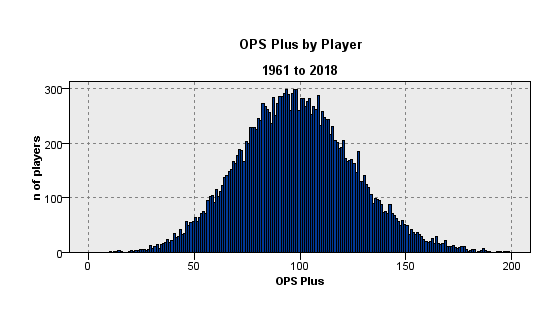


Chart 3.23 above shows that OPS+ is roughly normally distributed when looking at players with at least 150 at bats in a season.

**Playing Position**

WAR vs. playing position shows some significant variation between a player’s position on the field played most often in a season and that player’s WAR value for that season. This variable was created by classifying each player yearly by the position where that player appeared in the maximum number of games in that year. The Sean Lahman baseball database provided appearances by position for each player. By transforming that data using R code each player’s most frequent position was identified for each year. Some players could move from position to position classification each year based on their playing time. A shortstop that moves to third base later in their career would be a simple example of this reclassification by year that occurs within the data.

Of interest here there are a high number of players, more than 4,700 in the sample, or ~12% that appeared in games only as a pinch hitter or pinch runner in a given year. These are typically bench players, and/or players that are not everyday players and late game substitutions. The WAR values for this group in aggregate is generally low and not nearly as widely distributed as the other players on the field as can be seen in the chart 3.22 WAR by position shown below.

**Chart 3.24 - WAR - By Position Played Most in a Season**

****

**Fielding Percentage**

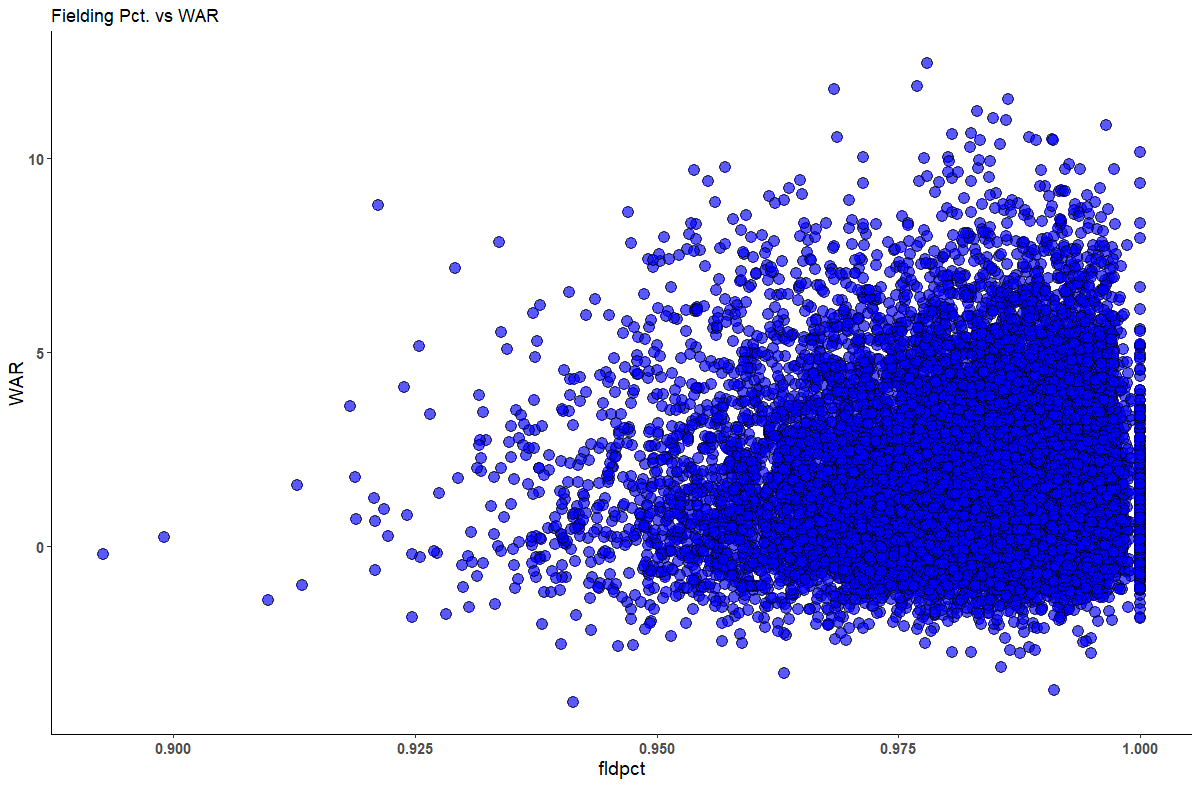
Fielding percentage is one of the only defensive rate metrics by player available from the baseball database. Fielding percentage calculation is below:

***Fielding percentage = (putouts + assists)/ (putouts + assists + errors)***

Fielding percentage was calculated for each player in the combined dataset using the raw counting statistics that were provided.

Very high fielding percentage players have higher WAR, but overall the relationship between fielding and WAR is not as strong as some of the offensive metrics. Fielding percentage here is with a minimum of 70 putouts in a year, so only shows the relationship without extreme outliers.

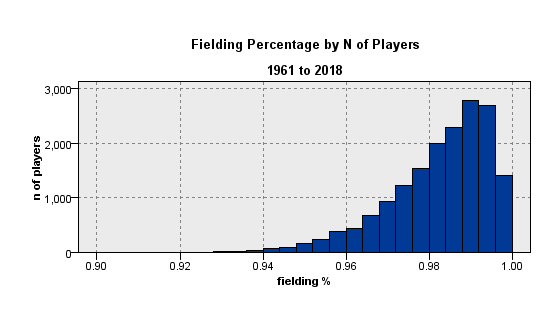
**Chart 3.25 - Fielding Pct. Vs WAR**

****

Fielding percentage has a mildly positive relationship with WAR, but not nearly as strongly correlated with WAR as the previous potential predictors that have been evaluated to this point in the project.

Chart 3.26 below shows that fielding percentage is in general left skewed. The range for fielding percentage is generally narrow as most fielders having a fielding percentage of greater than 95%, indicating that the vast majority of all major league players successfully field their position.

**Chart 3.26 - Fielding Pct. By Number of Players**

****

Fielding statistics are fairly limited in MLB compared to the number of different hitting statistics. Measuring fielding success is a growing area of research in the sabermetrics community.

**Errors**

Errors, another fielding measure also do not have an obvious a strong relationship with

WAR and in general it is highly skewed to the right as can be seen in chart 3.28.

**Chart 3.27 – Errors By WAR**

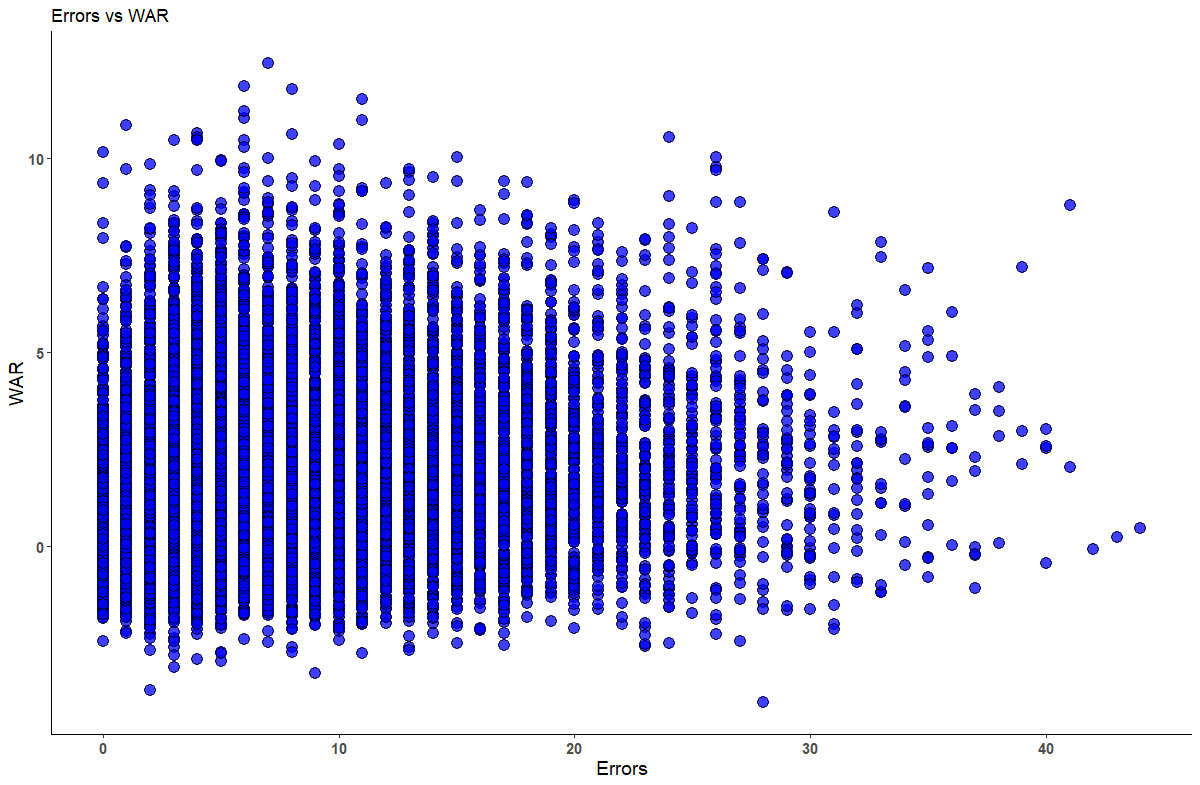
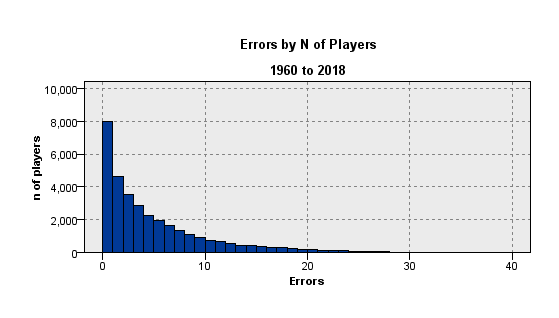
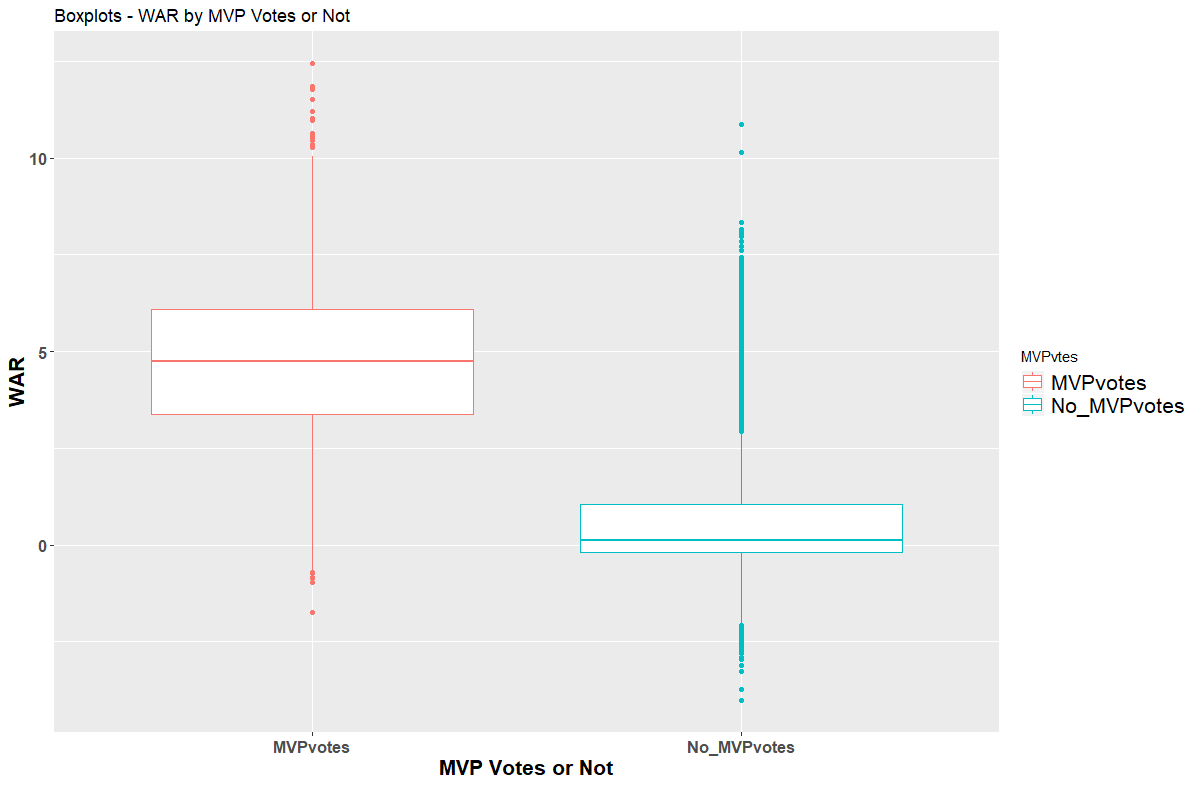
****

Figure 3.26 below shows that rate is in general right skewed.

**Chart 3.28 – Errors - By Number of Players**

****

**Chart 3.29 – MVP Winners and Vote-Getters**

****

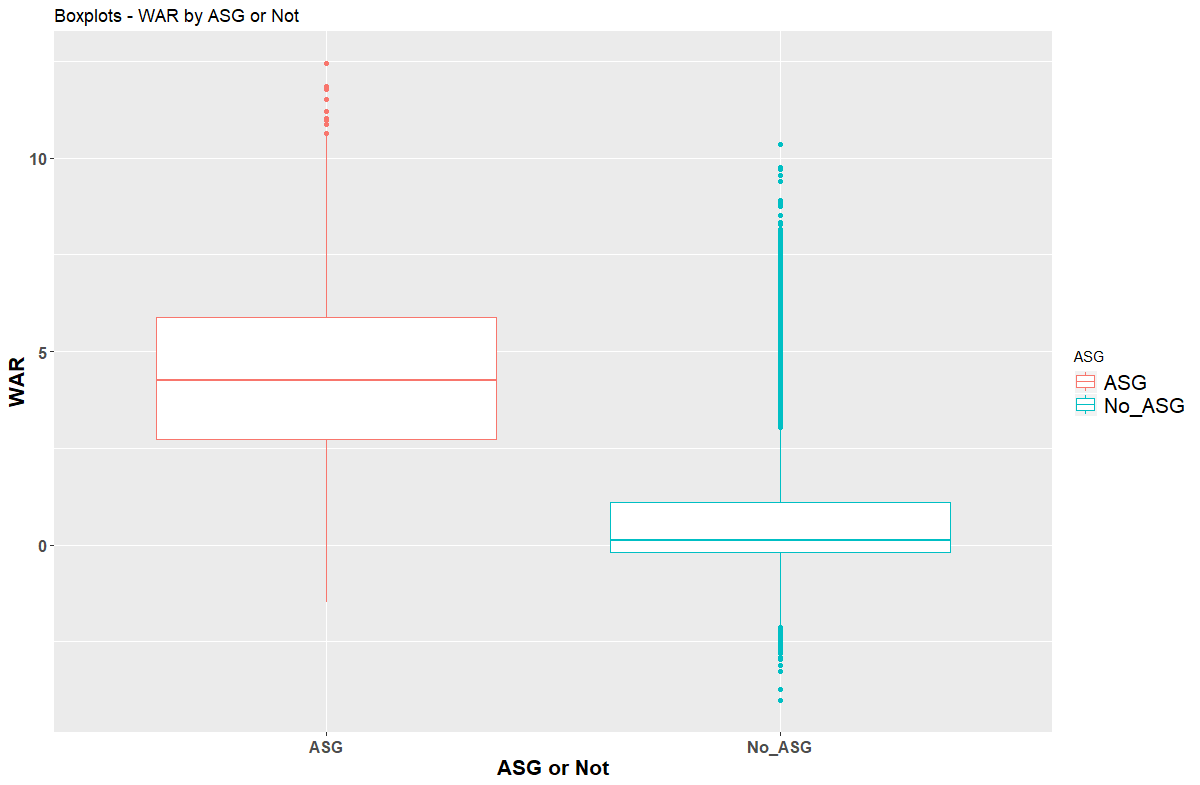
In addition to player statistics, the Sean Lahman baseball database includes player awards including, Cy Young winner, post season MVP winners, and as shown below, regular season MVP candidates. The chart above in Figure 3.29 shows that your typical MVP vote getter, has a WAR of ~5, well above the overall average of .86 for all MLB players.

**WAR by All Star Game Appearances**

The Sean Lahman database includes data on appearances in the All-Star Game, known as the midsummer classic. The All Star Game for MBL, is another potentially useful differentiator for identifying high WAR players. The All-Star game roster is voted on every year by MLB fans who pick the ASG rosters for position players.

The non all-star game players for WAR is close to zero while All-Stars, as expected, have on average WAR values greater than four wins above replacement.

**Chart 3.30 – All Star Game Roster or Not**

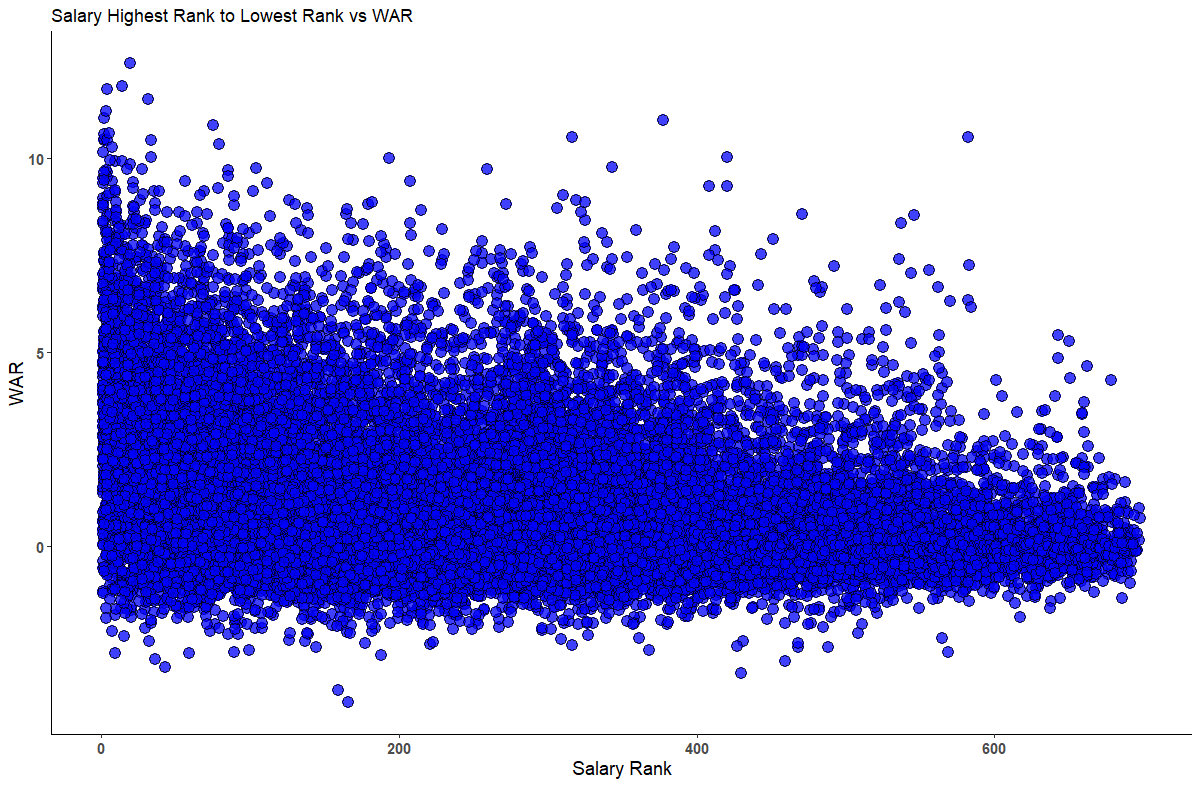
****

**Salary Rank by Year by Player**

The *Baseball Reference* database provides salary data on most records as part of their WAR database. Baseball player salaries have increased tremendously during the time span between 1961 and 2018. In 1961, Willie Mays, one of the game’s greatest players earned a league leading salary of $85,000/year. By comparison, Mike Trout, the highest paid player in 2018 earned $34M/year.

In order to use the salary data in models each player’s salary was ranked high to low within each year. Ranking salary data by year by player in this manner using R code shows that the highest paid players generally have higher WAR. The effect is particularly pronounced when players are in the top decile of salary levels for a given year. A dummy variable for top decile salary was created for testing in the estimation models based on this finding.

**Chart 3.31 – Salary Rank (Highest to Lowest Left to Right) vs. WAR values**

****

Interestingly, the range of WAR increases as salary increases, and significant numbers of players with the highest salaries have negative WAR values. This seems to indicate that some of the highest paid players, are not necessarily the most impactful in all cases. There is often a debate about free agency and whether the highest contracts are typically “worth it” as by the time players get the highest contract values their skills are at a time that is past their prime. Visually chart 3.31 suggests that there are plenty of negative WAR players, even at the highest salary levels.

IV - TEST VS. TRAINING SAMPLES DEFINED, RAW VARIABLES PREPARED FOR CLUSTER ANALYSIS AND ESTIMATION MODELING

Following the initial exploratory data analysis process the dataset was prepared for clustering and estimation modeling. First, all continuous variables tested during were transformed to z distributions having a mean of zero and a standard deviation of one. Next, the data was split at random into training and test samples, with 60% of the sample used to training models and 40% for testing as shown in chart 4.1.

Several different cluster and estimation models were created and tested to contrast and compare each model’s effectiveness. The clustering algorithms tested were K-means and BIRCH. As unsupervised techniques these models were used primarily to determine how many different clusters could be used to effectively describe the data. The estimation models tested were CART and random forest. As supervised techniques, their effectiveness was primarily evaluated on their ability to predict WAR values accurately.

**Chart 4.1 – Test vs. Training Sample**

V – CLUSTER MODELS, K-MEANS & BIRCH

Several clustering techniques were used to determine if meaningful clusters of players could be identified using the dataset to assess if naturally forming cluster of players emerged.

**K-Means Clusters**

The K-means clustering algorithm was the first unsupervised method tested. Several iterations of different numbers of clusters were attempted. The training dataset was used for all cluster analysis. The K-means cluster algorithm was used repeatedly varying the number of inputs in an iterative exploratory manner, each time assessing cluster goodness using the average silhouette value in SPSS/IBM Modeler.

The K-means cluster algorithm was then used varying the number of clusters created on the dataset to determine if descriptive useful clusters of players could be successfully identified. The variables included for the clustering model were runs, RBIs, on base percentage, plate appearances and home runs. These variables were chosen from the 54 variables created during the exploratory data analysis, as they were all shown to have impact on WAR during EDA, and they all measure different aspects of baseball performance.

Testing what K should equal in the models was done interactively varying K values from three to six leaving the variables in each model constant. Following creation of the K-means cluster models, dummy variables for the cluster membership were created to test the effectiveness of cluster membership alone in estimating WAR.

CART estimation models were then created using cluster membership alone to estimate WAR, varying only the number of clusters (K) in each model. Table 5.1 below shows the mean silhouette values from the cluster models and the mean average error rates for estimating WAR using dummy variables for cluster membership as the only predictors in the CART models.

**Table 5.1 – K-Means Cluster Model Comparisons, CART Models MAE and Mean Silhouette Values**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | K = 3 | K = 4 | K =5 | K = 6 |
| MAE | **0.768** | 0.739 | 0.731 | **0.723** |
| Mean Silhouette | **0.616** | 0.555 | 0.509 | **0.471** |
| Min cluster size % | 20.9% | 13.2% | 8.5% | 6.6% |
| Max cluster size % | 52.4% | 43.2% | 36.7% | 29.5% |

When K equals six the mean average error rate is minimized at 0.723. However, the mean silhouette value for a six cluster model is lowest among all the values of K tested at 0.471. The six cluster model also has the smallest segment 6.6% of all the number of cluster models tested.

In contrast to the six cluster model, the three cluster model (K=3) has the highest silhouette values of all the K-means models tested. What the three cluster model gains in increased separation for the clusters, it loses in accuracy in predicting WAR with a MAE of 0.768. The three cluster model the highest error rates of all K-means models tested.

The four and five cluster models, as expected, perform between the two extremes seen in the six and three cluster models. The four cluster model has a silhouette value of 0.555 which is generally considered good separation among clusters. The five cluster model, while showing improvements in mean average error (0.731), shows a relatively large decrease in mean silhouette value to 0.509 compared to the four cluster solution. This indicates inferior separation for the five cluster model relative to the three and four cluster models.

**BIRCH Clusters**

Following the K-means cluster algorithm a series or BIRCH clustering models were tested. BIRCH clustering is hierarchical clustering method. BIRCH clusters, since it is a hierarchical technique, puts observations in distinct clusters in a tree-like hierarchy. K-means tries to find the best way to divide the data into K clusters simultaneously based on how similar an observation is to other observations in the same cluster and how dissimilar it is to other clusters based on distance from cluster centers. BIRCH clusters are built incrementally splitting the data from one cluster to N, or N clusters to K, until the optimal solution is found.

BIRCH clustering is particularly sensitive to the order of the data that is used as the data is processed by the algorithm. Five different random sortings of the training dataset were used to assess the impact of the different sortings on the clusters created. The same five variables were utilized in the BIRCH clusters as in the K-means cluster. The variables included for the clustering model were runs, RBIs, on base percentage, plate appearances and home runs.

Following creation of the BIRCH cluster models, dummy variables for the clusters were created to test the effectiveness of cluster membership alone in estimating WAR in a similar manner to the K-means cluster models. CART estimation models were created to estimate WAR using dummy variables for cluster membership as the only predictors in the models.

Table 5.2 shows the various model statistics for varying sortings using the BIRCH algorithm. Sort number four creates a three cluster model, has the lowest mean average error rate at 0.762 and also have the second to lowest mean silhouette value of all the BIRCH models tested at 0.600.   
**Table 5.2 – BIRCH Cluster Models Summary Statistics, Five Different Sortings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Sort1 | Sort2 | Sort3 | Sort4 | Sort5 |
| K 🡺 | 2 | 2 | 3 | 3 | 2 |
| MAE | 0.852 | 0.839 | 0.773 | **0.762** | **0.859** |
| Mean Silhouette | 0.694 | 0.695 | **0.577** | 0.600 | **0.696** |
| Min cluster % | 37.3% | 34.8% | 20.7% | 16.6% | 35.2% |
| Max cluster % | 62.6% | 65.1% | 45.2% | 51.8% | 64.7% |

The mean silhouette value is maximized for the fifth sorting tested at 0.696, producing a two cluster model. This silhouette value is the highest among all the models tested for BIRCH of K-means. This model also has the highest mean average error rate among all the BIRCH models tested.

The three different two-cluster models produced by sortings one, two and five all have higher mean average error rates than the three cluster models produced by sortings three and four. The three two-cluster sortings all have higher mean silhouette values than the three cluster sortings. These models using different sortings show the trade-off involved between maximizing separation in the clusters by reducing the number of clusters, and in increasing the error rate in using the clusters membership alone to estimate a target. This positive relationship between MAE and mean silhouette values is shown visually summarizing all nine cluster models created for the study, including both K-means and BIRCH models. Note all the two cluster models in the upper right corner from the BIRCH cluster algorithm have the highest mean silhouette values and the highest error rates when used to predict WAR.

**Chart 5.3 - MAE vs Mean Silhouette Values for All Cluster Models**

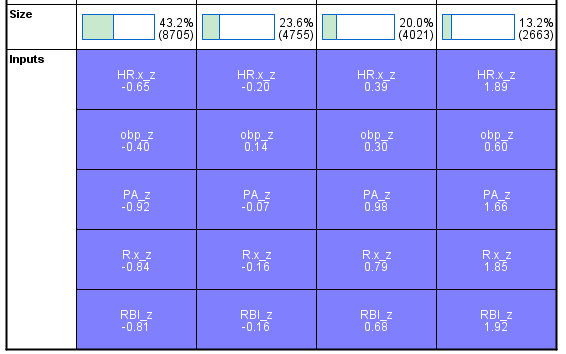
K = 2 BIRCH models

K = 6 K-Means

**Preferred Cluster Model and Cluster Descriptions**

Evaluating all nine cluster models, the four cluster K-means model was chosen as the preferred cluster model. The four-cluster K-means model is preferred, as it balances low error rate in estimating WAR, while maintaining acceptable levels of cluster separation as measured by mean silhoette values. Upon inspection of cluster means, the four cluster model solution also provides a useful framework to describe and differeniate among different clusters of players using typical baseball statistics. For descriptive purposes, the raw variables (before z-transformation) are used to analyze the clusters to ease interpretion. Chart 5.4 shows the cluster centers from the four cluster model and chart 5.5 shows a full comparison of the cluster averages across various predictors using the raw variables.

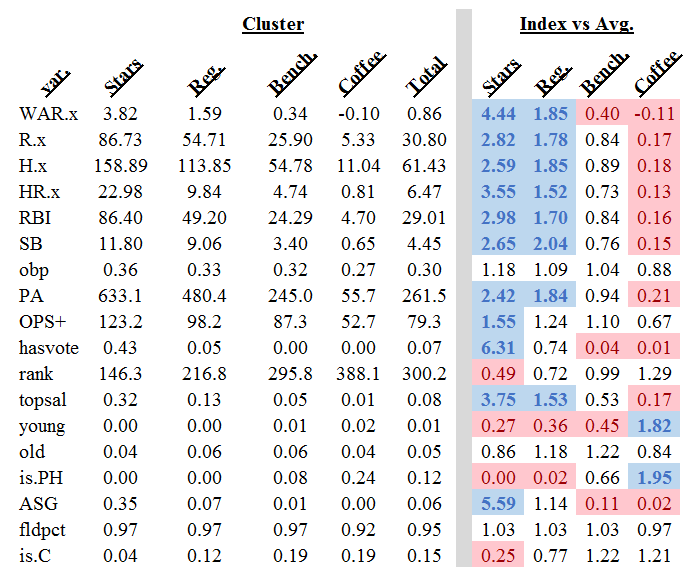
**Chart 5.4 K-Means Cluster Model Centers, Z-Transformed Variables**



***Stars of the Game - 13.2% of all players***

This cluster has the best players in major league baseball. They have the best statistics pretty much across the board. These players play every day (average 633 plate appearances) have on average 158 hits/year (2.6x the average) and average 23 home runs a year (3.5x the average). They score more the 2.8 times as many runs as the average. The best hitters are more likely to be in this cluster. They are 3.8x more likely to be in the top 10% of highest paid players annually and they are 5.6 times more likely to be selected for the all-star team compared to the average major league player. Their WAR values are on the right tail of the distribution at 4.4 times the league average. Example players in this cluster of players include David Ortiz, Carl Yastrzemski, Cal Ripken, Hank Aaron and Mike Trout. Many Hall of Famers (top 1% of all players all time) are in this cluster.

**Chart 5.5 – Mean Values for K-Means, K = 4 Solution, Training Sample**



***Every-Day Regular Players 20.0% of players***

These are quality every-day players that are usually in the starting line-up. Not as elite in performance as the stars of the game cluster, the every-day players, play most games averaging 480 plate appearances a year and have above average run scoring ability at 55 runs/player, roughly double the overall average. They are 1.5 more likely to be in the top 10% of highest paid players annually and their WAR values are 1.8 times the league average. Examples of players in the cluster, again from the Red Sox, include Jackie Bradley, Dustin Pedroia and Rafael Devers.

***Platoon Players (23.6%, cluster 1)***

These players have generally average to below production, as measured by pretty much every offensive category. They are average run producers at slightly above average runs 26 mean vs 30 overall mean runs scored in a year. They don’t play full-time, averaging roughly 245 plate appearances a year compared to an average 260 plate appearances per year. There are 19% catchers in this segment compared to 15% in all the other segments. Catchers typically play fewer games than other positions, everything else equal, due to the physical rigors of the position. Catchers require more days off throughout the season. These players’ WAR values are 0.34 on average, less than half the average WAR value for players in the sample. Eduardo Nunez, Christian Vasquez and Black Swihart are examples of players put in this cluster.

***Cup of Coffee/Back and Forth to Minor Leagues (43.2%)***

These are bench players and fill-in players. These players do not play regularly, in a given year. They may fill in for a few games and generally have very low production. This a large segment of players who play in the major leagues every year. Their run scoring production is low at 5.3 runs/year, as is most all other production as measured by home runs, stolen bases, games played on base percentage and plate appearances. These players get on the field for situational reasons only. This segment is 24% pinch hitters, roughly two times the overall average in the population of 12% pinch hitters. These players have negative WAR on average at -0.11 on average. Examples of these players include Lou Merloni, Grady Sizemore, Will Middlebrooks, and Daniel Nava. These example players from the Boston Red Sox are all players with injuries, bench players, pinch hitters and players in their first year of service time from the minor leagues.

VI – PRINCIPAL COMPONENTS ANALYSIS

The exploratory data analysis and descriptive cluster analysis have both shown that the predictors for any estimation model for WAR are highly correlated with one another. Given this generally high correlation among many predictors, and the desire to produce predictive models that could be potentially used for inference, a principle components analysis was performed on the training data to reduce the collinearity of the dataset. This process should improve the stability of estimates of the different predictors in the estimation models. For this analysis the z-transformed continuous variables were the only variables considered for the PCA, which excluded all dummy variables created during the EDA portion of the project.

Several different criteria were used to assess the number principal components to extract. Specifically, the four criteria were considered include i) the eigenvector criterion, ii) the proportion of variance explained criterion, iii) the scree plot criterion and iv) the minimum communality criterion. Each of the four criteria were examined in sequence to determine the optimal number of components to extract. Varimax rotation was applied to ease interpretation of the different principal components.

1. **The eigenvector criterion**

The eigenvalue criterion extracts each eigenvalue that is at least a value of one, with the rationale that each principle component should at least explain one variable’s worth of variability. The un-rotated solution suggests extracting seven principal components, while the rotated solution suggests up to nine variables could be extracted. Note however that principal components six through nine are all very close to one when evaluating the rotated solution in table 6.1.

**Table 6.1 – Principal Component Extraction Using Varimax Rotated and Un-rotated Components, Extracting 20 Principal Components**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Rotation Sums of | | |
|  | Initial Eigenvalues | | | Squared Loadings | | |
| Principal Component | Total | % of Var. | Cume. % | Total | % of Var. | Cume. % |
| 1 | 18.23 | 44.47 | 44.47 | 14.35 | 35.00 | 35.00 |
| 2 | 6.33 | 15.43 | 59.90 | 7.52 | 18.34 | 53.34 |
| 3 | 2.41 | 5.88 | 65.78 | 2.24 | 5.46 | 58.81 |
| 4 | 1.70 | 4.15 | 69.92 | 1.87 | 4.56 | 63.37 |
| 5 | 1.51 | 3.67 | 73.60 | 1.81 | 4.41 | 67.78 |
| 6 | 1.16 | 2.82 | 76.41 | 1.04 | 2.53 | 70.30 |
| 7 | 1.11 | 2.71 | 79.12 | 1.03 | 2.52 | 72.83 |
| 8 | 0.88 | 2.13 | 81.26 | 1.03 | 2.51 | 75.34 |
| 9 | 0.77 | 1.87 | 83.12 | 1.03 | 2.50 | 77.84 |
| 10 | 0.73 | 1.78 | 84.91 | 0.98 | 2.40 | 80.24 |
| 11 | 0.67 | 1.64 | 86.54 | 0.96 | 2.35 | 82.59 |
| 12 | 0.64 | 1.57 | 88.11 | 0.95 | 2.32 | 84.91 |
| 13 | 0.60 | 1.46 | 89.57 | 0.87 | 2.13 | 87.04 |
| 14 | 0.55 | 1.35 | 90.92 | 0.78 | 1.91 | 88.95 |
| …. |  |  |  |  |  |  |
| 41 | 0.00 | 0.00 | 100.00 |  |  |  |

**b. The Proportion of Variance Explained Criterion**

Table 6.1 can also be used to look at the heuristic for proportion of variance explained. Specifically, eight components explain 75% of the total variation, while ten components explain 80% of the variation in the data. This criterion suggests extracting a range of eight to ten principal components.

**c. The Minimum Communality Criterion**

Using the minimum communality criterion suggests that a minimum of seven principal components be extracted. This can be seen in Table 6.2 that all the variables have communalities of at least 0.5. The minimum communality rule is that each variable must share at least half of its variability with each principal component to stay in the analysis. Using this criterion, technically relaxing it a bit to 95% of all variables, suggests that at least seven principal components get extracted, as can be seen in Table 6.2.

**Table 6.3 Communality Comparisons – K1 through K4 – Communality Extraction Comparison**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | Variable | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | HBP.x\_z | *0.338* | *0.349* | *0.352* | *0.358* | *0.373* | *0.374* | *0.38* |
| 2 | ASG\_z | *0.223* | *0.227* | *0.32* | *0.374* | *0.405* | *0.41* | *0.41* |
| 3 | sum\_PO\_z | *0.371* | *0.387* | *0.404* | *0.499* | 0.507 | 0.518 | 0.518 |
| 4 | IBB.x\_z | *0.406* | *0.413* | 0.523 | 0.524 | 0.541 | 0.542 | 0.544 |
| 5 | rank\_z | *0.247* | *0.267* | *0.282* | *0.315* | *0.319* | *0.412* | 0.56 |
| 6 | SH.x\_z | *0.098* | *0.139* | 0.522 | 0.536 | 0.571 | 0.571 | 0.611 |
| 7 | SF.x\_z | 0.56 | 0.586 | 0.595 | 0.611 | 0.611 | 0.613 | 0.613 |
| 8 | X3B\_z | *0.381* | *0.404* | *0.49* | 0.589 | 0.619 | 0.621 | 0.621 |
| 9 | slg.2\_z | *0.121* | 0.591 | 0.595 | 0.595 | 0.595 | 0.596 | 0.648 |
| 10 | GIDP.x\_z | 0.614 | 0.648 | 0.65 | 0.708 | 0.71 | 0.712 | 0.712 |
| 11 | sum\_E\_z | *0.372* | *0.43* | 0.586 | 0.593 | 0.739 | 0.74 | 0.745 |
| 12 | age.x\_z | *0.011* | *0.011* | *0.05* | *0.149* | *0.155* | *0.349* | 0.751 |
| … |  |  |  |  |  |  |  |  |
|  | **% var > .5** | **37%** | **61%** | **73%** | **82%** | **87%** | **90%** | **95%** |

**d. The Scree Plot Criterion**

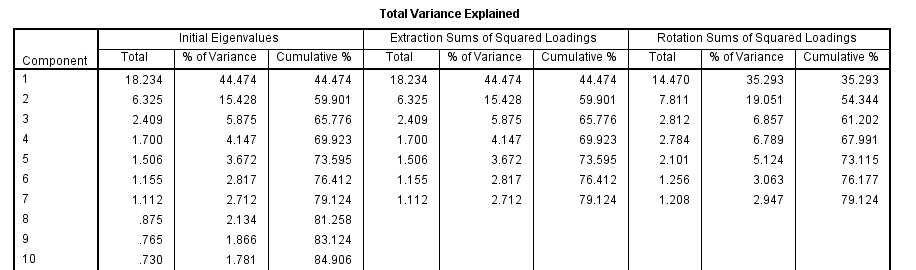
Lastly, the scree plot criterion is a visual inspection method to determine how many principal components to extract. The rule is the max number to extract is just before the plot flattens to a straight line. Using this rule would suggest extracting five principle principal components.

**Table 6.4 Scree Plot, Training Data**

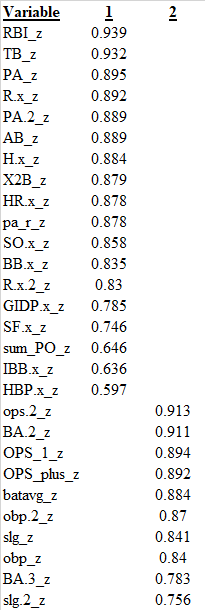
Assessing all four criteria a range of solutions can be considered anywhere from a minimum of five to ten principal components using the scree plot criterion as a floor and the proportion of variance explained as a ceiling.Approaching the decision of how many to use as range of acceptable solutions, and maximizing the interpretation clarity of the analysis, seven principal components are recommended as the number to select. This recommendation falls in the middle of the range of the four solutions, balancing the need to eliminating the collinearity in the predictors, and maximizing the explanatory capability of the model to be built using the principal components.

Table 6.5 below shows the output of PCA with seven components extracted using varimax rotation. The percentage of variance extracted in table 6.5 at 79.1% is close to of the variation in table 6.1 of 72.8%, the initial solution which extracted 20 principal components.

**Table 6.5 – Total Variance Explained from Principal Components Analysis Using Varimax Rotation; Seven Principal Components Extracted**

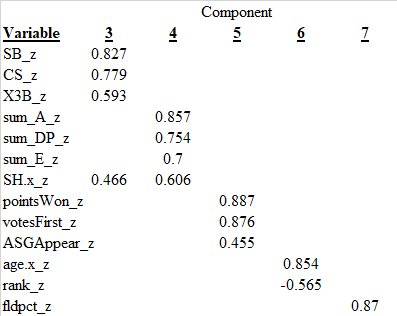
****

The varimax rotation helps considerably in easing interpretation of the PCA results. As can be seen the rotated matrix shown table 6.6, the first principal component has high loadings for many of the counting hitting statistics with loadings of more than 0.5. Component 1 (*Batting* *Counting Stats*) is correlated with RBIs and total bases among other counting stats. When suppressing all loadings under 0.5, the rotated matrix in table 6.6 shows the second component having high loadings on many of the hitting rate variables including the interaction terms added during the EDA phase of the project. Component 2 (*Hitting Rate Stats*) have high positive loadings on batting average and on base percentage, slugging percentage and other rate based statistics.

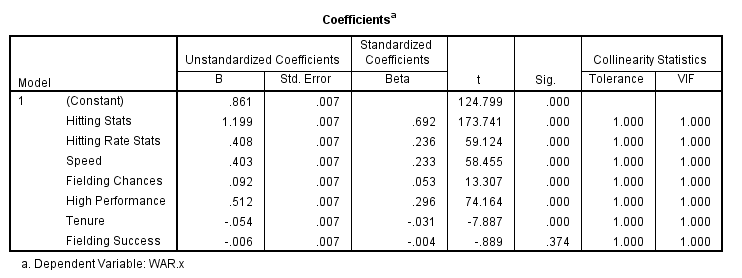
**Table 6.6 - Rotated First Two Components 🡺**

The third rotated component shown in table 6.7 has high loadings with stolen bases and triples. This third component will be called *Speed*. The fourth component of the rotated solution has high loadings on fielding counting statistics like errors, double-plays turned and assists. I will call this principal component *Fielding Chances.* The fifth component has high loadings on number of MVP votes and MVP winners. I will call this component *High Performance*. The sixth component has a high positive loading on age and a negative loading on salary rank. I will call this component *Tenure*. Finally, the last component extracted loads highly on fielding percentage, so I will call this principal component *Fielding Success.*

**Table 6.7 – Rotated Components 3 through 7**

Following identifying the components, a multiple regression model of all seven principal components on WAR was executed using forward selection. Table 6.8 shows that the most significant coefficient/principal component in estimating WAR is *hitting stats* (principal component #1). *Hitting stats* standardized coefficient is 0.692, nearly three times the standardized coefficients of principal component 3, (*speed*) and more than twelve times the influence of *fielding chances* (factor 5). Principal component number 7, *fielding success* was not significant in estimating WAR at a 95% confidence level. Table 6.8 also indicates the principal components have no collinearity issues as all variables have VIF statistic of exactly one, which indicates no presence of collinearity between predictors. Eliminating all the collinearity among predictors was the intent of the PCA and these factors will be later tested in the estimation models.

**Table 6.8 - Coefficients and Collinearity Diagnostics, Forward Selection Regression of Seven Principal Components on WAR**

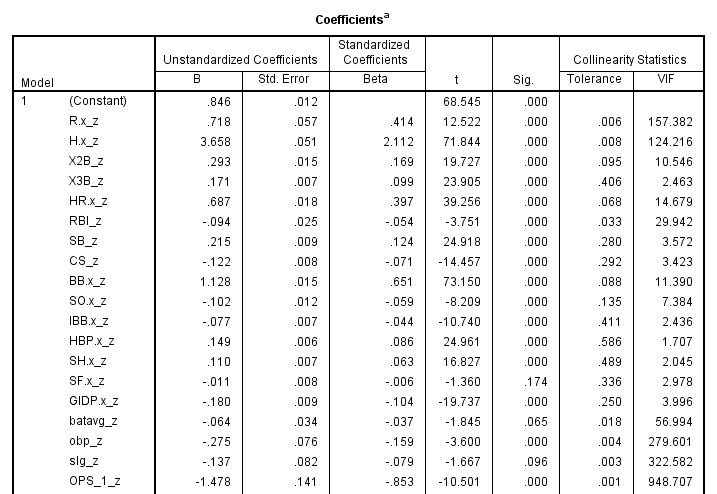


VII – ADDRESSING COLLINEARITY IN THE ORIGINAL VARIABLES

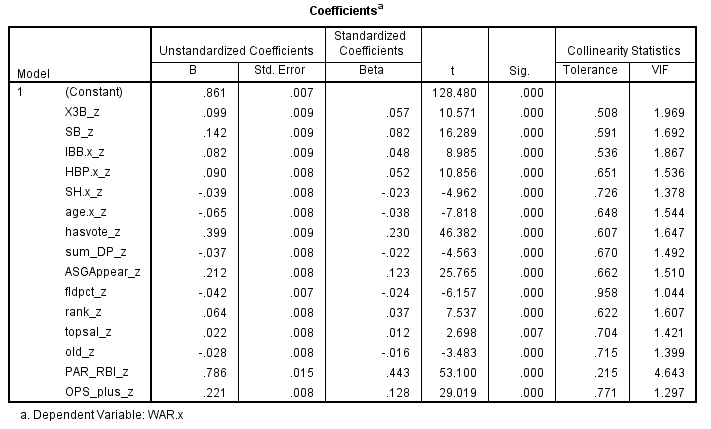
In addition to reducing dimensions of the data using PCA, collinearity in the raw variables was addressed so that inferences about the impact of individual variables on WAR could be made. These models will later be compared to the PCA approach to dimension reduction to assess how these two different approaches impact the estimation models’ error rates.

Figure 7.1 illustrates the unstable beta estimates produced by a linear regression model for WAR regressed on all the variables in the dataset. A partial output shows very high VIF statistics indicating severe collinearity among the predictors which if not addressed could lead to erroneous estimates for sizing the impact of different variables on estimating WAR values.

**Table 7.1 - Coefficients and Collinearity Diagnostics, Forward Selection Regression of Original Z-Transformed Variables on WAR**

****

An iterative approach to testing different linear regression models VIF statistics was taken in addressing the collinearity among predictors. As a first remedy, obvious predictors that were close proxies of one another were removed from the analysis. For example, at-bats, plate appearances and games plays are all very highly correlated as they all measure playing time, albeit slightly differently. Plate appearances was chosen for inclusion among the three variables, as it was shown to have the very correlation with WAR during the EDA phase of the project and was useful in the clustering models as a framework to cluster different types of players. In addition to removing obvious close proxy variables, all interaction terms were removed from consideration as they will naturally have collinearity issues with their component terms. Next a series of multiple regression models were executed with a focus on identifying which combination of variable exhibit high VIF values. Through inspection, high VIF offenders were removed and then VIF retested without one of the high VIF variables. Finally, in order to not lose significant explanatory power of variables that proved to be highly correlated with WAR during EDA, a composite variable was created leveraging runs, plate appearances and RBIs. Specifically, a variable called PAR\_RBI was created as the weighted average of the three variables, calculated as *PAR\_RBI\_z = (PA\_z/3 + R\_z/3 + RBI\_z/3).* When this new variable is used in a linear regression, replacing the remaining three highly correlated source variables, it resulted in a VIF statistic of 4.7, reflecting collinearity below the moderate level a vast improvement to the pre-collinearity remedy model shown in table 7.1. The reduced set of variables and coefficients are shown in Figure 7.3, all having much improved VIF statistics relative to pre-collinearity adjusted model comparisons.

**Table 7.2 - Coefficients and Collinearity Diagnostics, Forward Selection Regression Reduced Set, Z-Transformed Variables on WAR, Post-Collinearity Remedies **

VIII – ESTIMATION MODELS - CART

Following the variable dimension reduction through PCA and after addressing the severe collinearity among the predictors, the same training dataset was used to estimate WAR using several CART regression models. CART is sometimes seen as useful alternative to linear models that produces more rule-based models. The rules-based structure of CART aid in interpretation and CART models can be easier to understand particularly for non-analysts. The CART model also requires fewer underlying assumptions than linear regression. CART, for example, has no requirement such as normal error terms. Additionally, CART doesn’t require as extensive data preparation and can be less sensitive to outliers than OLS regression.

The downside of CART models is they can be prone to overfitting. Additionally, CART models perform a single split at each node, so very complex splits at individual nodes in this model type are traded-off with the relative simplicity of explanation and implementation.

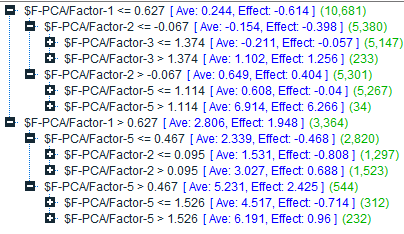
Three different models were fit using CART with different sets of predictors. The models compared used i) the principal components only (defined in section six), ii) the reduced set of z transformed variables from section seven and iii) the full set of 47 z-transformed and dummy variables created during the initial EDA, including interaction terms. In each case the CART models had a tree depth of 15 levels.

***CART Model 1 – Principal Components as Variable Drivers***

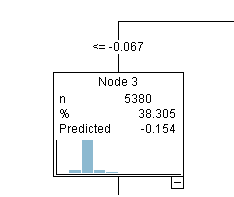
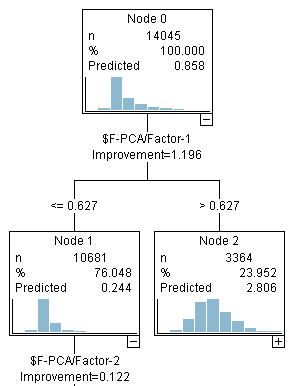
Some of the detail of the splits within the model illustrate which principal components are drivers in the PCA variable CART model. As can be seen from the split below taken from SPSS/IBM modeler (figure 8.1) PC1(batting counting stats) is the first split with players having 0.627 or lower having an average WAR of 0.24, whereas players that have more than 0.627 on PC1 in a year having an average WAR of 2.86. The individual histograms on PC1, batting counting stats, at the first split in figure 6.3 show the clear differences in WAR estimates for the overall average.

The second level split for low WAR players is on, PC2, hitting rate stats, while PC5, high performance, splits at the second level for high WAR players. The second level split for high performers have and average WAR of 5.23, well above the sample average of 0.86. The second level split to the low side on PC 2, counting hitting stats, have an average WAR value of l -0.154.

**Figure 8.1 – CART PCA Model (#1) SPSS/IBM Modeler  
 First Three Splits Training Data**



**Figure 8.2 - CART Model (#1) SPSS/IBM Modeler, First Three Nodes**

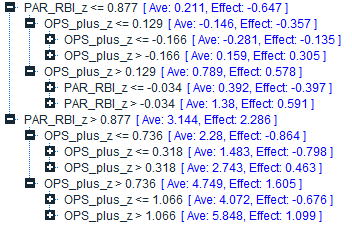


***CART Model 2 - Original Variable Z-transformed as Model Drivers***

The second CART model used the original variables as predictors, using only the reduced set of variables that were shown to not have severe collinearity among predictors. As can be seen from the split below taken from SPSS/IBM modeler (figure 8.3) the composite variable of runs, plate appearance and RBIs, called PAR\_RBI\_z, is the first split with players having 0.877 or lower having an average WAR of 0.21, whereas players have more than 0.877 PAR\_RBI\_z in a year having an average WAR of 3.86.

The second level split for low WAR players is on OPS\_plus, with those with less than 0.129 on OPS\_plus having negative WAR at -0.146. On the high side of the PAR\_RBI\_z split, the model again splits on OPS\_plus for several nodes. This makes some intuitive sense as OPS\_plus is a composite hitting statistic known to be highly correlated with WAR.

**Figure 8.3 – CART PCA Model (#2) SPSS/IBM Modeler  
 First Three Splits Training Data**

****

***CART Model 3 – All Variables Z-transformed as Model Drivers***A third CART model was fit estimating with a sole focus on minimizing WAR estimation error rates. Since the predictors are highly correlated, any inference of drivers was not performed. This model performance is included as a comparison point the first two models. This model includes all the interaction terms created during the EDA portion of the study and doesn’t include composite variable created during the collinearity diagnostics, but instead includes the source variables runs, RBIs and plat appearances.

Table 8.4 below compares and summarizes the performance for all three models, reporting number of variables, MSE and MAE. Judging by error rates alone, the full variable model has the best performance on the training data with a MAE of 0.442. The standard error of the estimate (MSE) is 0.447. This can be interpreted as the typical prediction error in the modeled for WAR is 0.447 wins above replacement value.

**Figure 8.4 – CART Models Summary Statistics, Training Data**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | 1 - CART PCA | 2 - CART Reduced | 3 - CART Full |
| MSE | 0.619 | 0.504 | 0.447 |
| MAE | 0.517 | 0.468 | 0.442 |
| N of Variables | 7 | 15 | 47 |

The PCA model has notably highest error rates of the three at an MSE of 0.619 and 0.517, so while it produces predictors that are perfectly uncorrelated, the PCA model loses some predictive power in the process. This result is expected as PCA is an unsupervised technique. PCA follows the most variation in the data only. The directions of the principal components may not relate to the target variable in as predictive a way as the original variables. The reduced CART model has performance in between the two and some mild collinearity among the predictors.

IX – ESTIMATION MODELS - RANDOM FORESTS

Following the CART models, the same training dataset was used to estimate WAR using random forest models.

A random forest model is a set or ensemble of decision trees. Each tree (model) is created from a bootstrap sample, which is a subsample of the same training data. Random forest uses bagging where multiple models are created off the boot-strap samples. Each model’s outcome or response is predicted, and the bagging process takes the average value of all predictions across all models as the ensemble’s prediction. The model parameters for each model are different, as they are selected off each sub-sample, and at each split (node) the best variables are chosen from a small subset of randomly selected variables from the set of all variables. Random forest models can be superior to individual decision tree models in that they can avoid overfitting that can occur in single decision trees since the final model is an average all the tree results. Additionally, random forest can perform better than CART as they tend to have lower variance and more stable estimates than a single tree (CART).

The random forest model is random in two major ways. First, the variables that are selected at each split are random and secondly, the observations used in each model (tree) are randomly selected.

The trees (models) are grown through an iterative process. First each model or tree is trained on roughly 2/3 of the total training dataset. Observations are chosen at random with replacement from the original dataset. Once the sample is selected the tree is built. At each node the tree is split using the best variable from a random subset of m predictors for this node. The default for the value of m is the number of predictors in the dataset divided by three.

One of the downsides random forest is they are computationally and time intensive. The random forest models’ with up to 100 trees execution times were 25 to 35 minutes in duration, compared to 4-5 minutes for each CART model execution time. This computational overhead is noticed considerably when calculating the summary model statistics as new trees are created randomly each time to calculate statistics like MSE and MAE. Stability in error rates in some cases cannot be achieved until 500 trees. For this analysis up to 150 trees were tested (with more than an hour execution time). The improvement in MAE values shown below in chart 9.2 are minimal between 100 and 150 trees, so 100 trees are shown for the final model comparisons.

Analogous to the CART model process, three different models were fit using CART with different sets of predictors. The models compared were i) the principal components only, ii) the reduced set of z-transformed variables and iii) the full set of 47 z-transformed and dummy variables created during the initial EDA including interaction terms. In each case 100 trees were built on the training data for each model.

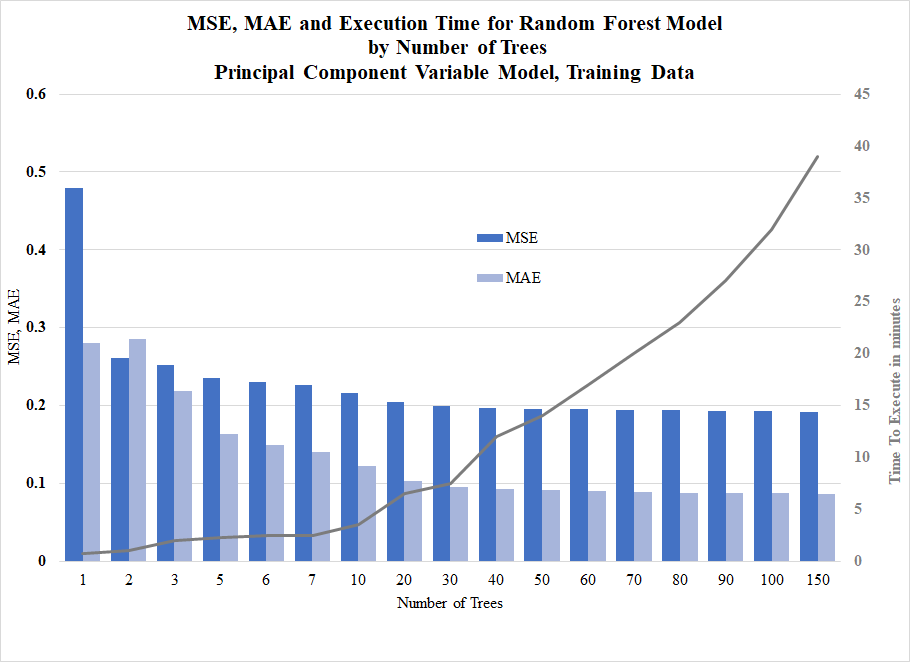
The random forest models were executed in SPSS/modeler. The code base for the SPSS/modeler implementation is based on the python implementation of random forest models.

**Table 9.1 – Random Forest Models Summary Statistics, Training Data**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | 4 – Random Forest PCA | 5 – Random Forest  Reduced | 6 – Random Forest Full |
| MSE | 0.087 | 0.102 | 0.074 |
| MAE | 0.193 | 0.204 | 0.172 |
| N of Variables | 7 | 15 | 47 |

Table 9.1 summarizes the random forest models error rates. The error rates on the training data for random forest are all much lower than the CART models. This can be attributed to the averaging of results of the random forest algorithm. In this case, one hundred trees were built. The number of trees built directly reduces the error rates. Chart 9.2 illustrates this relationship between error rates, number of trees and run time using the PCA model as an example.

**Chart 9.2 – Random Forest PCA Model MSE, MAE by Number of Trees**



The standard error of the estimate for the PCA base model is lower than either the full or reduced models when using the training data. This can be interpreted as the typical prediction error in the modeled for WAR is 0.087 wins above replacement value. The mean absolute error for the full model is lower than the other two (PCA and reduced models), using the same variables. The random forest method uses the average performance across all models built to produce the final model results, which make this algorithm less sensitive to collinearity among the predictors than other non-ensemble methods like linear regression.

X – ESTIMATION MODELS - TESTING AND COMPARISONS

After building the models on the training data each model was tested against the test dataset which had 13,445 observations. A summary of each model performance is summarized below in Table 10.1. A side by side comparison of each model’s performance using the same variables is illustrative of each model’s strengths and weaknesses.

The best model for estimating WAR using error rates as a benchmark is the full variable random forest model (#6). This model has the lowest MAE and MSE on the test data at 0.421 and 0.427 respectively. The CART PCA variable model (#1) has the worst accuracy of all models tested with a MAE of 0.840 and MSE of 0.599. The remaining models, numbers two, three, four and five, fall in between those two extremes with MAE values ranging from 0.528 to 0.685, and MSE values ranging from 0.482 to 0.527. The random forest reduced variable model (#5 in table 10.1) has the second best error rate of all estimation models tested.

**Table 10.1 – WAR Estimation Summary Statistics  
 & Strengths/Weaknesses Compared, Six Models, Test Data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model 🡺 (#) | CART PCA  (1) | RF PCA  (4) | CART Reduced  (2) | RF Reduced  (5) | CART Full  (3) | RF Full  (6) |
| MAE | 0.840 | 0.644 | 0.685 | 0.528 | 0.628 | **0.421** |
| MSE | 0.599 | 0.527 | 0.544 | 0.482 | 0.519 | **0.427** |
| Relative Error | **High** | Med | Med | Med | Med | **Low** |
| Ease of Driver Inference | Med | Med | **High** | **High** | **Low** | **Low** |
| Collinearity of Predictors | **None** | **None** | Low | Low | **High** | **High** |
| Relative Overfitting | **Higher** | **Lower** | **Higher** | **Lower** | **Higher** | **Lower** |
| Time to Execute | **Low** | **High** | **Low** | **High** | Med | **High** |
| Among Tested Best For 🡺 | Speed, driver, inference | driver inference | inference simplicity and speed | inference w/ accuracy | n/a | Best accuracy |

The two reduced variable models, both CART and random forest, are the best for overall interpretability of drivers. The predictors in those models, are in terms that can be seen and easily understood by the casual baseball fan, for example runs, plate appearances and RBIs. For inference of which variables are more influential than others, those two models have an intuitive appeal. The reduced model’s variables also have the advantage of increased interpretability, which is traded-off against some accuracy when comparing the reduced variable models to the full variable models. Since the full-variable models have high collinearity among predictors, inference on the impact of individual variables on WAR values is not appropriate, as those inferences could be inaccurate.

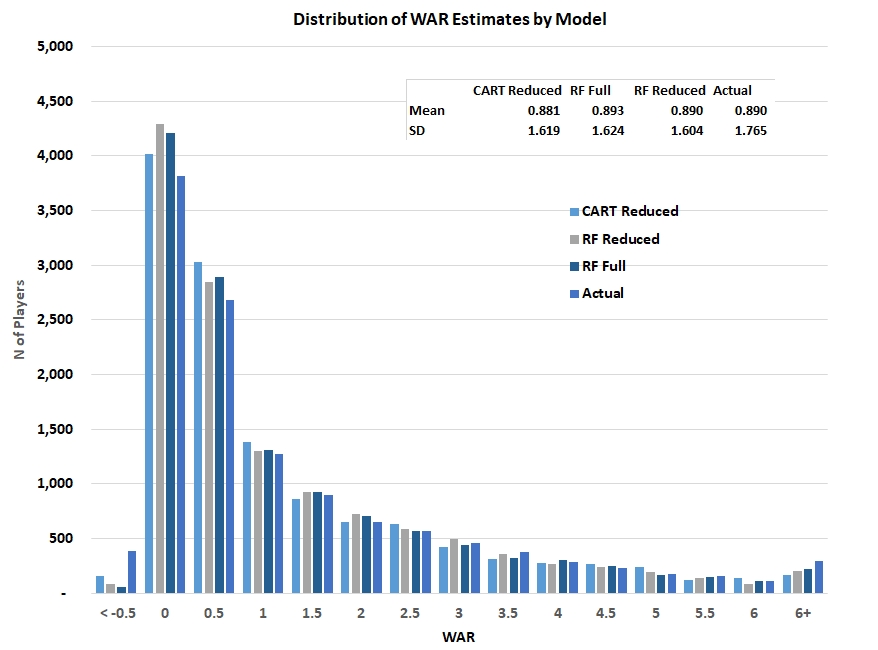
For pure stability of model variable driver inference, the principal component models (#1 and 4) are the only models tested that have zero collinearity among predictors. What is gained in purity of driver inference in both the PCA models is traded-off with higher error rates in the PCA models when estimating WAR values relative to the other models. The PCA estimation model results in a higher error rates when compared to both the CART and random forest reduced and full-variable models.

The relative overfitting of the CART models is a weakness in the higher error rates seen in every in every CART model vs. the comparable random forest model. The random, ensemble nature of the random forest algorithm mitigates the collinearity issues somewhat.

When looking at execution speed and stability the CART models radically outperform all random forest models with the random forest execution time of more than 30 minutes compared to the CART models which execute within less than 5 minutes, and can score new data within seconds, unlike random forest. The random forest models, since they are an average of N models, have long run times even when scoring new data. This intensive computation time is the downside of the random forest algorithm.

The top three models are shown in chart 10.2. Looking at the distribution comparison of the estimates for the three, each closely follows the same shape of the actual WAR distribution. Each of the three models do a reasonable job of estimating average WAR. All the models also have similar spread of the data as seen by the standard deviation of estimated WAR seen in Figure 10.2. All three models tend to slightly under-estimate the number of players and the extremes of the WAR distribution.

**Chart 10.2 - Distribution of WAR, Three Different Model Estimates Compared to Actual WAR Estimate, Test Data – 13.5k observations**

****

**Preferred Model based on Test Data - The Random Forest Method**The top three candidate models based on a balanced assessment of strengths and weakness are the full random forest model for its’ accuracy, the CART reduced model for its speed and simplicity of understanding, and the reduced random forest model for its ability to be used for inference of drivers, at the expense of some accuracy. Since the WAR values are proprietary in their calculation and estimation accuracy is a top priority the full variable random forest model is the preferred model. The random forest model provides the lower MAE value of all models tested and the lowest standard error among all four tested. The random forest model has advantages over all comparable CART regression models.

Random forest models have two main parameters, the number trees in the “forest” and number of variables to try at each split at the tree, so they are relatively simple to tune. They also require little data preparation. Single decision trees like the CART model may be prone to overfitting whereas random forest which uses bagging and combines results of many models tend to help overfitting.

On the downside, random forests are not good at generalizing to data that have observation values outside the range of the training data. Another weakness is random forests is they are biased towards categorical variables with multiple categories. That weakness is mitigated in this case as all categories for this data, like player position, are modeled using dummy variables. Finally, random forest models can be computationally intensive and because of their complexity may be less intuitive to understand though, there are importance measures for variables. Given the primary focus of the models is predictive accuracy using available to the public statistics above all else, the random forest’s model ability to accurately estimate WAR is the preferred choice for estimating WAR.

XI – CONCLUSIONS

The WAR statistic itself is becoming the one overall number that baseball fans and sabermetrics experts alike are using to track player value. Though the exact calculation for WAR is proprietary and not standardized across sources, this paper has shown different mathematical modeling algorithms can be utilized to understand WAR value, including unsupervised methods such as clustering, and estimation modeling methods, including dummy variable regression, CART and random forest models.

K-means and BIRCH clustering were both successful at describing meaningful clusters of players using those players’ standard baseball statistics. The cluster approach, that is using cluster membership alone to estimate WAR, provided a rough guide for WAR estimation though with much less overall accuracy than any of the other estimation model types. The cluster analysis was able to provide meaningful descriptive clusters of players, roughly based on plate appearances. The various cluster models tested between BIRCH and K-Means clusters illustrated the trade-off between creating clusters that maximize separation statistics such as silhouette values, and can be used for estimating a target and describing additional groups of players uniquely. Fewer clusters, while resulting in more cohesive clusters, had more error in estimating WAR values and lost value for descriptive purposes relative to higher K cluster solutions. Ultimately a K equal to four cluster solution provided a nice balance of descriptive usefulness, reasonable silhouette value and predictive power in estimating WAR.

WAR itself is highly correlated with many different hitting related statistics and many of those statistics highly correlated with one another. The nature of this relationship in the predictors necessitated data dimension reduction techniques including principal component analysis and remedies for collinearity. For collinearity issues limiting the number of predictors, repeatedly testing VIF statistics in linear models for adequacy and creating a composite variable for plate appearances, runs and RBIs greatly improved the collinearity statistics among the remaining variables. These three statistics were highly correlated with one another and very useful in estimating player WAR values.

Principal components analysis on the transformed continuous variables also proved a useful dimension reduction technique. For PCA, the ideal number of components to extract was explored testing four different rules from the eigenvector criterion to the scree plot criterion. Ultimately the number of components to extract was based on a similar balance as the cluster analysis decision for K. Specifically, a seven component model balanced clarity of explanation and fits within the guidelines of the PCA criteria.

The estimation models leveraged three sets of variables for CART and random forest. The two sets of three models ranging in complexity and number of variables used resulted in a useful comparison of the strengths and weaknesses of those models. These two sets of three models illustrate the trade-offs of more dimension reduction vs less dimension reduction, and the differences in the CART and random forest algorithms. Different models can all estimate WAR with reasonable accuracy, but each have their pros and cons depending on the priorities of the modeling exercise.

The CART approach is intuitively appealing for its ease of interpretation and simple rule-based approach. The CART model is most useful in providing a rules-based understanding to the user to which critical points provide the best splits for WAR.

While all models provide a useful estimate, the random forest full variable model is the preferred model for this task of providing the most accurate estimation of WAR. When compared to the other models, the full variable random forest model minimizes MAE and MSE on the test data. Random forest’s ability to have few necessary assumptions makes it a good choice to handle the many highly correlated predictors and requires limited transformation of variables to produce a useful model of WAR. One downside, of the random forest method, is the lack of ability to use the full model for inference purposes. The random forest method, and models are computationally significantly more time-intensive than the other model types to execute. However, these shortcomings of the random forest generated WAR model are more than offset by the random forest algorithms ability produce an accurate WAR estimate using only player statistics.

All the models in the study including the cluster models, the dimension reduction analysis and the estimation models illustrate the trade-offs involved in choosing between different techniques and options to solve the same problem. In this case, those trade-offs are between speed/interpretability and accuracy. This trade-off is seen in all the models tested. This illustrates the importance of prioritizing the goal in any modeling exercise. For this application, since the secret formula is proprietary and complex, and has a high payoff/cost for identifying the target, minimizing error rates at the expense of inference are the priority, leading to the complex, random forest full model as the first choice foe estimating WAR.

The WAR metric itself shows a strong correlation with many hitting statistics which is part of the reason it is so predictable with relative accuracy and it is highly predictive of team wins. However, fewer fielding statistics were particularly predictive in the estimation models or useful in the descriptive cluster part of the analysis identifying groups of players. Measuring player fielding skill accurately is an area of future development in baseball sabermetrics in general. Additional fielding performance data would be welcome addition to understanding baseball performance as accurate and predictive fielding statistics remains an area of active research. Given the number of different choices, approaches and interest in measuring player value through WAR additional player-level granular impactful fielding capability will continue to be an area of development in the future.

Wins-above-replacement measurement appears here to stay in baseball circles. There is much debate and various proprietary methods around published WAR estimates and not all sources agree on how to calculate WAR. There is little argument, however, around why WAR is worth estimating. WAR continues to be studied and modified, as baseball evaluators continue to refine the metric calculation methodology, to predict team wins. The intent is of these estimation models is for them to by utilized as a “ready reckoner” that will allow estimating WAR relatively simply, using the player stats alone when the actual underlying WAR values are not available. This paper has shown that mathematical techniques including clustering, principal component analysis, dimension reduction and estimation models such as CART and random forest can be used, instead of proprietary methods, to estimate WAR values for MLB baseball players.

XII - REFERENCES

Baseball Reference.com WAR Explained (2019). *Baseball Reference.*    
Retrieved June 20th, 2019, from <https://www.baseball-reference.com/about/war_explained.shtml>

Position Player WAR Calculations and Details (2019). *Baseball Reference.*  Retrieved June 20th, 2019, from https://www.baseball reference.com/about/  
war\_explained.shtml

Benefits of decision trees in solving predictive analytics problems (2019). *Prognoz.* Retrieved June 30th, 2019, from http://www.prognoz.com/blog/platform/benefits-of-decision-trees-in-solving-predictive-analytics-problems/

Hastie, T & Tibshirani, R & Friedman, J (2008). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer Series and Business Media, New York, NY.

Modern Machine Learning Algorithms: Strengths and Weaknesses. (2019). *Elite Data Science.* Retrieved June 30th, 2019, from https://elitedatascience.com/machine-learning-algorithms

MAE and RMSE — Which Metric is Better? (2016). *Human in a Machine World.* Retrieved September 15, 2019, from <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>

How to select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics. (2018). *Towards Data Science.* Retrieved September 15, 2019, from <https://towardsdatascience.com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0>

Wins Above Replacement. (2019). *Wikipedia.* Retrieved September 15, 2019, from <https://en.wikipedia.org/wiki/Wins_Above_Replacement>

Catania, Jason. (2019). *Making the Case for WAR as Baseball's Most Perfect Statistic*. Bleacher Report*.* Retrieved September 15, 2019, from <https://bleacherreport.com/articles/1642919-making-the-case-for-war-as-baseballs-most-perfect-statistic>

Foreman, Sean, Van Slooten, Hans (2018). Wins Above Replacement or How I Learned to Stop Worrying and Love deGrom. *Sports Reference LLC.* Retrieved September 15, 2019, from https://www.dropbox.com/s/ukkz4g4y1rorxi5/2018-03-WAR-Explainer.pdf?dl=0

Lahman, Sean (2018). The Lahman Baseball Database. *The Baseball Archive.* Retrieved June 1, 2019, from http://www.seanlahman.com/files/database /readme2017.txt

Friendly, Michael & Dalzell, Chris & Monkman, Martin & Murphy, Dennis &, Foot, Vanessa & Zaki-Azat, Justeena (2019). Package ‘Lahman’. *Cran.r-project.org.* Retrieved June 1, 2019, from https://cran.r- project.org/ web/packages/Lahman/ Lahman.pdf

GGplot2 Box Plot: Quick start guide - R software and data visualization (2019). *STHDA*. Retrieved September 15, 2019, from http: //www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization

Alison, Paul (2012). When Can You Safely Ignore Multicollinearity? *Statistical Horizons*. Retrieved September 14, 2019, from https: //statisticalhorizons.com/ multicollinearity

McClelland, Gary H. & Irwin, Julie R & Disatnik, David & Sivan, Liron (2016). Multicollinearity is a red herring in the search for moderator variables: A guide to interpreting moderated multiple regression models and a critique of Iacobucci, Schneider, Popovich, and Bakamitsos (2016). *Behavior Research Methods*. Retrieved September 14, 2019, from https://link.springer.com/ article/10.3758/ s13428-016-0785-2

Dziuda, Darius, PhD (2018). Unit 4 Course Notes. *Predictive Modeling for Continuous Response Variables STAT 529*. Central Connecticut State University.

Bhalla, Deepanshu (2014). A complete guide to Random Forest in R. *Listen Data.*  Retrieved September 14, 2019, from https://www.listendata.com/2014 /11/random-forest-with-r.html

Beri, Satvik (2013). Ensemble Learning: What are some shortcomings of random forests? *Quora*. Retrieved September 14, 2019, from https://www.quora.com/Ensemble-Learning-What-are-some-shortcomings-of-random-forests

IBM SPSS Modeler 18.1.1 Modeling Nodes (2018). Chapter 17 - Python Nodes. *IBM.* Retrieved September 14, 2019, from ftp://public.dhe.ibm.com/software/ analytics/spss/ documentation/modeler/18.1.1/en/ ModelerModelingNodes.pdf

Lewinson, Eryk (2019). Explaining Feature Importance by Example of a Random Forest. *Towards Data Science*. Retrieved September 14, 2019, from https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e

Breiman, Leo & Cutler, Adele (2001). Random Forests. *Machine Learning Journal*. Retrieved September 14, 2019, from https://www.stat.berkeley.edu/ ~breiman/RandomForests/ cc\_home.htm#papers

Larose, D. (2017). *STAT 523* course notes. Central Connecticut State University.

Standard Stats. (2019). *MLB.com.* Retrieved June 20th, 2019, from http://m.mlb.com/ glossary/standard-statshttp://m.mlb.com/glossary/standard-stats/batting-average

Haupert, Michael (2018). MLB's annual salary leaders since 1874. Society for American Baseball Research. Retrieved July 7, 2019 from https://sabr.org/research/mlbs-annual-salary-leaders-1874-2012

K-Means Cluster Algorithm Differences - Modeler & SPSS Statistics (2018). IBM Support. *IBM*. Retrieved July 15, 2019 from  
<https://www-01.ibm.com/support/docview.wss?uid=swg21566488>

IBM SPSS Modeler 18.2 Algorithms Guide (2018). K-Means Algorithm. IBM. Retrieved September 14, 2019, from ftp://public.dhe.ibm.com/software/analytics/ spss/documentation/modeler/ 18.2/en/AlgorithmsGuide.pdf

Larose, D. (2016). *STAT 522* course notes. Central Connecticut State University.

Saslow, Elliot (2018). Collinearity - What it means, Why its bad, and How does it affect other models? *Future Vision*. Retrieved September 14, 2019, from <https://medium.com/future-vision/collinearity-what-it-means-why-its-bad-and-how-does-it-affect-other-models-94e1db984168>

Brownlee, Jason (2014). Feature Selection with The Caret R Package. *Machine Learning Mastery,* Retrieved August 2, 2019, from https://machinelearningmastery.com/ feature-selection-with-the-caret-r-package/

Larose, D. (2015). *STAT 521* course notes. Central Connecticut State University.

Feature Selection – Ten Effective Techniques with Examples (2019). *Machine Learning Plus.* Retrieved September 5, 2019, from https://www.machinelearningplus.com/ machine-learning/feature-selection/

Larose, D. &, Larose, C (2015). *Data Mining and Predictive Analytics*. Hoboken, NJ: John Wiley & Sons, Inc.

Kuhn, M. &, Johnson, K (2013). *Applied Predictive Modeling*. New York, NY, Springer Science & Business Media.

XIII - APPENDIX -   
DETAILED WAR EXPLANATION FROM *BASEBALL REFERENCE*

***Batting Runs***   
These are runs created by the player for all hitting actions compared to the league average. In *Baseball Reference’s* language they describe batting runs as follows:

“For batting runs we use a linear weights system based on Tom Tango's wOBA (weighted on-base average) framework, but we add a number of improvements to our calculation of weighted runs above average (Baseball Reference, 2019, WAR for Position Players, Rbat, Batting Runs section, para. 1).”

*Baseball Reference’s* website has extensive conceptual detail on the improvements they add to adjust weighted runs above average. They estimate caught stealing totals when data is not available and, additionally in some years they differentiate infield vs outfield singles and runs on errors when that data is available, which is not for all years. The make many adjustments year to year as they change the methodology within the actual WAR calculation.

***Base-running Runs***   
These are runs associated with stealing bases, caught stealing and creating runs by items like going first to third base on a single and getting thrown out on the bases.

Some examples of the 30-plus examples *Baseball Reference* tracks are below.   
“For infield ground balls with less than two out that are not an infield hit and with no runner on first, the number of times the runner advances to third.  
For infield ground balls with less than two out that are not an infield hit and with no runner on first, the number of times the runner stays at second.  
For infield ground balls with less than two out that are not an infield hit and with no runner on first, the number of times the runner is out (Baseball Reference, 2019, WAR for Position Players, Rbr, Baserunning Runs section, para. 1). ”

***Runs added or lost due to Grounding into Double Plays in DP situations***Avoiding a double play results in incremental runs. Players are tracked by how many double plays they hit into and avoid compared to the average and this results in incremental runs created. Slow Power hitters may have negative runs created on this component, while players that beat out more than the average number of double plays may be positive on this attribute.

***Fielding Runs***  
Fielding runs are calculated using the system *Baseball Reference’s* Defensive Runs Saved (DRS) system described as, “the most sophisticated public system available. It includes eight factors:

1. Fielding Range Plus/Minus Runs Saved based on BIS-trained scorer observations and batted ball timing to determine the velocity of each batted ball.
2. Outfield arm runs saved based on exact counts of baserunner advancements and kills and the velocity of the hit ball.
3. Infielder double plays based on opportunities and rates they were turned based also on batted ball velocity.
4. Good play-bad play values which include 28 positive play types. For example: HR-saving catches, backing up a play, blocking a pitch in the dirt, and 54 misplays like missing the cutoff man, failing to anticipate the wall and allowing extra bases, not covering a base, pulling a foot off the bag, etc...
5. Bunt Fielding
6. Catcher SB/CS data (which is tweaked by the pitchers caught)
7. Pitcher SB/CS data (which is tweaked by the catchers behind the plate)
8. Catcher handling of the pitching staff via things like pitch framing and pitch calling.

(Baseball Reference, 2019, WAR for Position Players, Rdef. Fielding Runs section,   
para. 1).”

***Positional Adjustment Runs***This WAR component is an adjustment made to a player’s WAR for their position played on the field. In general defensive-focused positions like middle infielders and catcher (which sacrifice offense for defense) have positive runs above replacement for this component, while offensive positions, like designated hitter and corner outfielders are penalized with negative runs above replacement values. Specifically, positional adjustment runs are calculated as follows:

“To compute a player's Positional adjustment Runs, we add together for each non-pitching position: Position multiplier (from above) × innings played at position / 1,350 Innings (Baseball Reference, 2019, WAR for Position Players, Rpos, Positional Adjustment Runs section, para. 1). ”

To force the total positional adjustment to be overall zero, the total league positional runs are calculated and the difference between the sum of the player’s positional adjustment runs and league total is allocated out to players to ensure all the players positional runs add to zero. Note, the position multiplier in the above equation changes every year.

***Replacement Level Runs***The last part of the components, the sixth component, is particularly somewhat challenging to understand.

According to *Baseball Reference*

“Replacement level is something of a touchy subject with non-sabermetricians, and probably the least understood of the ideas here. Currently, we set replacement level at .294 winning percentage (changed from .320 in March 2013) for the major leagues, which means there are 30\*162\*(.500-.294) = 1,000 Wins above replacement in the major leagues as a whole (Baseball Reference, 2019, WAR for Position Players, Replacement Level section, para. 1).”

Position players get 59% of the wins allocated to them, based on their split of free agent salaries, over the last four seasons. This split for position players equates to 20.5 runs for a player with 600 plate appearances above replacement value. This means that if a team replaced a league average starter with a replacement, we would expect a 20.5 run difference to the downside in run differential. *Baseball Reference* calls this 20.5 runs the *replacement level multiplier*.

Essentially, the last component of WAR is the amount runs above replacement value for the average player adjusted for an induvial player’s playing time as measured by number of plate appearances.

***Conversion of Runs to Wins to get to Actual WAR Values***After the run components are estimated, each player’s run total above replacement value is converted into an estimated number of wins. The actual conversion of runs to wins is significantly more complex than the estimate shown below. The actual formula is not published by *Baseball Reference* and uses the team winning percentage leveraging a formula developed by sabermetrician PythagenPat for win/loss percentage. The method has gone through several iterations since 2001.

Research shows that 10 runs per game as a rough approximation of the conversion rate that is generally accurate in converting runs to wins (Baseball Reference, 2019, Baseball-Reference.com WAR Explained, Converting Runs to Wins section, para. 1). The assumed true conversion rate has changed multiple times over the past ten years, but we will use ten for simplicity of understanding and illustrating the conversion process.

Applying the estimates of the six components and the conversion to wins using 10 runs/win provides the following actual WAR details for Jim Rice (HOF) 1986 using the *Baseball Reference* WAR database.

**Jim Rice, 1986 Actual WAR with components**

|  |  |
| --- | --- |
| Variable | Runs |
| runs\_bat | 32.01 |
| runs\_br | -0.79 |
| runs\_dp | -2.11 |
| runs\_defense | 10.1 |
| runs\_position | -5.55 |
| runs\_replacement | 22.85 |
| Total Runs above replacement | 56.51 |
| Assume conversion  to wins @10 runs/win, WAR🡺 | 5.651 |

(Source: *Baseball Reference* WAR database)

As can be seen from the above, Jim Rice’s actual WAR for 1986 season is 5.65 according to *Baseball Reference.* The bulk of his WAR value comes from the batting component vs. the average player and the last component based on playing time (plate appearances) vs. a typical replacement player.

Note, Jim Rice was a rather slow power hitter, who hit into more than his fair share of double plays. This is reflected in his negative values for *runs\_br and runs\_dp.* Additionally, Rice played left field, typically an offensive-focused position which is also consistent with his -5 value for *runs\_position* above.

Clearly, one challenge of the actual WAR statistic illustrated above is the lack of complete transparency in how these components are calculated, the general high complexity level of overall calculations, and number of underlying assumptions and other player averages needed to calculate actual WAR values for an individual player.