**Major League Baseball Statistics**

Texas Tech University

Multivariate Analysis ISQS 6350

Dr. Eric Brown



**Group 9**

Cynthia Iradukunda,

Tabitha Baugus,

Madison Cisowski,

Andrew Schleiss

horizontal line**Introduction**

In the realm of sports, data statistics has increasingly become a powerful tool in improving performance and driving decision making. There is no sport better known for this than baseball. This was shown in the portrayal of the Oakland Athletics baseball team’s 2002 season, as depicted in the movie "Moneyball".

Based on a true story; the film reveals how the practical application of statistical analysis, in the management of the Oakland Athletics, greatly improves the team’s performance and consequently changes the sport forever.

This paper explores the world of Major League Baseball (MLB), focusing specifically on player batting statistics, as we try to understand how various sourced variables1 impact the players, teams, and the sport.

The purpose of this paper is to perform statistical analyses and create visualizations that reveal unique patterns, outliers, and relationships in the data; all of which contribute and can potentially influence a players batting performance and ultimately how this and other factors can impact professional Major League Baseball teams and players.

All code and resulting files can be found on the paper's [GitHub](https://github.com/Slyth3/Multivariate-Analysis-Baseball-Hitting-Statistics)4 page.

horizontal line**Data Cleaning & Visualization**

Below we undergo exploratory data analysis to understand the structure and content of the data before we investigate further concepts:

From a quick analysis of the data we note a few noteworthy aspects:

* No missing values are present
* Dimensions include 461 rows and 105 variables
* 2 categorical columns and the rest being float and integer columns

**Key Variables**

Given the extensive number of variables present in our dataset, it becomes imperative to discern key variables that offer substantial insights into batting statistics. As such each section on this paper will refer to the variables used as input unless the full dataset is analyzed.

**Removing Outliers**

A popular way to determine an outlier is via z-score. The original idea when cleaning the dataset was simple: identify outliers and remove them as we specifically choose this dataset because of the completeness, meaning it was not plagued by null values.

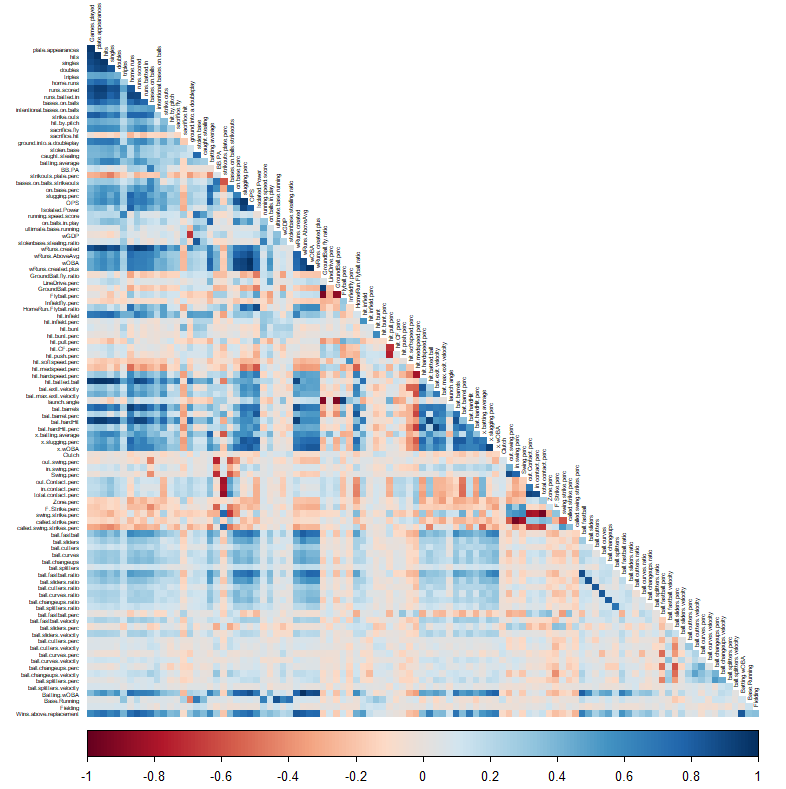
Once the data is brought to R, we isolate the categorical data and numeric data. Next, we obtain the z-scores for each variable. Scale() is a popular function used to complete this task, especially when all the data is not measured on a universal “unit”. It looks within columns, and assigns a z-score to each row which in essence makes it possible to compare previously uncentered and unscaled variables. Outliers are commonly defined by the following parameter: if the z-score is less than or equal to -3, or greater than or equal to 3.

After obtaining Z scores, we were faced with a choice: if a player is an outlier in one variable do we eliminate them completely from our analysis? It seems like the safest option but there is one issue with this idea, the sheer quantity of players that were then eliminated from the data. We started off with 461 observations but when applying this cleaning process, we dropped down to 294, ~64% of our original data. It makes sense with there being 103 variables, it wouldn't be uncommon for a player to be an outlier in at least one of these categories, especially considering the skill level of these professional athletes. So an outlier or two, per column can add up fast.

With this in mind, we pivoted our cleaning process to be a little more specific. 103 variables sounds great in theory but it is not realistic. Once we identify our “Key Variables” for each type of analyses, we will handle outliers on a case by case basis.

**Correlation Plot**

We investigate the relationships between these variables by plotting their correlations as depicted below.

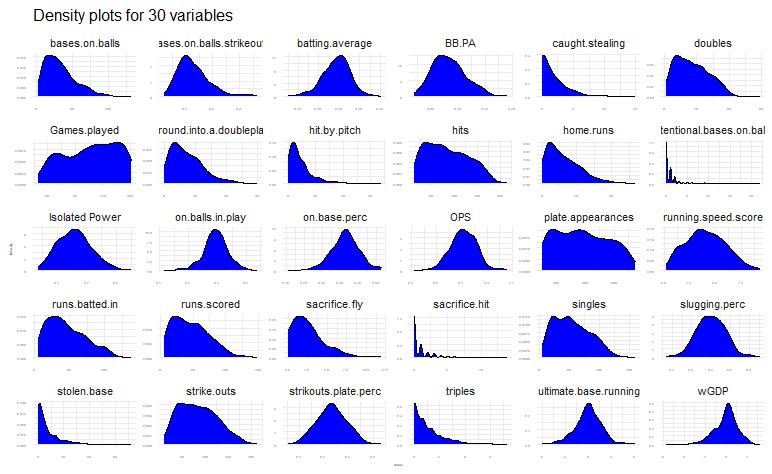


*Graph 1A: Correlation plot of all variables*

From the correlation plot, we can see that there are a large number of variables with very strong correlations. The dark red shows a strong negative correlation, whereas the navy blue depicts a strong positive correlation. It would assist interpretation and further analysis in subsequent sections, to remove one from each pair of highly correlated variables. We observe several instances of near perfect correlation, such as ‘Plate appearances’ and number of ‘at bats’. Plate appearance is when a batter goes to bat regardless of outcome, whereas at-bat is outcome driven, when a batter gets to move bases as result of a plate appearance such as a walk or hit-by-pitch, that is an at-bat. Thus, at bat is always a plate appearance but not every plate appearance is at-bat.

**Data Distribution**

Due to the number of variables, we will limit our analysis to the first 30 numerical variables.



*Graph 1B: Data density plot for first 30 numeric variables*

In Graph 1B, we note a range of distribution patterns, most of which are highly skewed distributions. Such skewness poses potential challenges in analytical approaches that focus on data variance or covariance, notably in methods like Model-Based Clustering and Principal Component Analysis (PCA). The concern arises from the possibility that PCA may not adequately capture the full variance of the data due to these skewed distributions.

In addition, these skewed distributions may be an indication of a large number of outliers which can affect certain clustering methods.

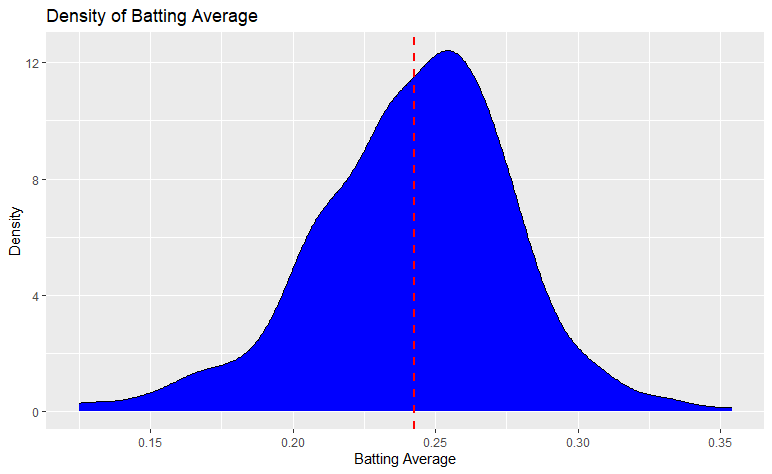
The ideal scenario for such methodologies typically ensures that the data follows a normal distribution. While there are techniques available to transform data towards normality, applying these transformations across a dataset comprising over 100 variables presents a significant analytical burden as such we will not transform the distributions.

**Batting Statistics**

This paper tries to answer questions and explore relationships around ‘batting statistics’ and those variables that may affect (positively or negatively) a players batting performance.

As such we can identify key variables in the data that can help us explore further:

The most notable variable must include “batting.average” which is the players number of ‘hits’ divided by their total ‘at bats’ (number of batter's turn batting against a pitcher)2

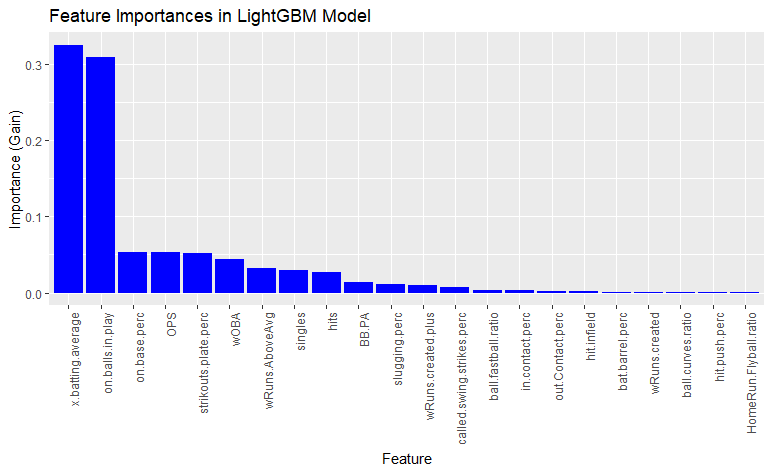


*Graph 1C: Density plot of Batting Averages*

The distribution of batting average appears to be normal, which bodes well for further analysis. The mean batting average of all players is 0.24, meaning that the majority of players for every 4 times they are “at.bats”, they will hit one time. At first glance this seems low but it is socially said that a “great” batter will have a batting average of .300 or greater. Take Baseball Hall of Famer Babe Ruth, his career batting average was .342.3

An interesting approach to identify variable relationships with batting average is to look at feature importances after modeling the data using gradient boosting decision tree models.

By applying the LightGBM algorithm to the data thus training a supervised regression model and predicted batting averages. The resulting feature importances are below.



*Graph 1D: Feature importances for preding batting average*

Both x.batting.average and on.balls.in.play take into account batting average as expected have high importance

Pivoting to look at the next highest ; ‘on.base.perc’, ‘OPS’ and’ strikouts.plate.perc’, where we can infer that having a high batting average will result in a batter being on-base more often, therefore resulting in these variables having high importance in prediction.

horizontal line**Dimensionality Reduction**

Data in high dimensions have a plethora of insights to be discovered and patterns to be revealed. However, due to the human eye’s inability to visually comprehend data in high dimensions, it is often necessary to reduce the dimensions; a challenge often referred to as the “curse of dimensionality”. This reduction in dimensions is essential for the creation of charts and graphs that are easy for stakeholders to interpret. There are several techniques used in multivariate analysis to reduce dimensions; here we will dive deep into Principal Components Analysis (PCA) and its many visualization options.

Principal Components Analysis aims to reduce the dimensions of a dataset while capturing the optimum amount of variance. This procedure transforms the data into a new set of variables, also known as the principal components, which are linear combinations of the original variables and also uncorrelated with each other. These principal components are referred to as the eigenvectors and the amount of variance captured by each of the principal components is referred to as the eigenvalue. The principal components are often represented graphically, or used in further analyses.

**Correlation**

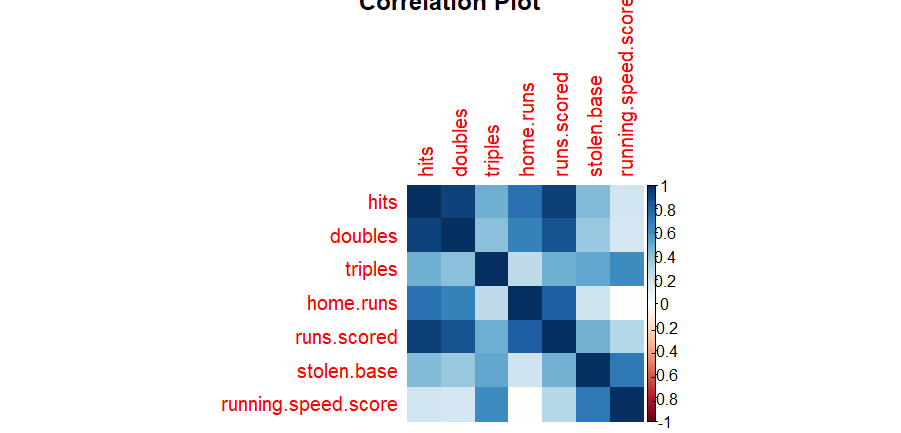
Before conducting PCA, it is wise to examine the dataset’s correlation, as the principal components are derived from the correlation matrix of the variables. Below are the correlation matrix and a correlation plot of the subset of variables representing various talents and deemed interesting for dimension reduction.

| Variable Subset | |
| --- | --- |
| hits | runs.scored |
| doubles | stolen.base |
| triples | running.speed.score |
| home.runs |  |

*Table 2A: Dimension Reduction Variable Subset*

| Correlation Matrix | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | hits | doubles | triples | home.  runs | runs.  scored | stolen.base | running.speed.score |
| hits | 1 | .922 | .480 | .740 | .936 | .436 | .197 |
| doubles | .922 | 1 | .415 | .677 | .869 | .370 | .189 |
| triples | .480 | .415 | 1 | .255 | .488 | .517 | .622 |
| home.  runs | .740 | .677 | .255 | 1 | .823 | .206 | -.006 |
| runs.  scored | .936 | .869 | .488 | .823 | 1 | .474 | .280 |
| stolen.base | .436 | .370 | .517 | .206 | .474 | 1 | .715 |
| running.speed.score | .197 | .189 | .622 | -.006 | .280 | .715 | 1 |

*Table 2B: Dimension Reduction Variable Subset Correlation Matrix*

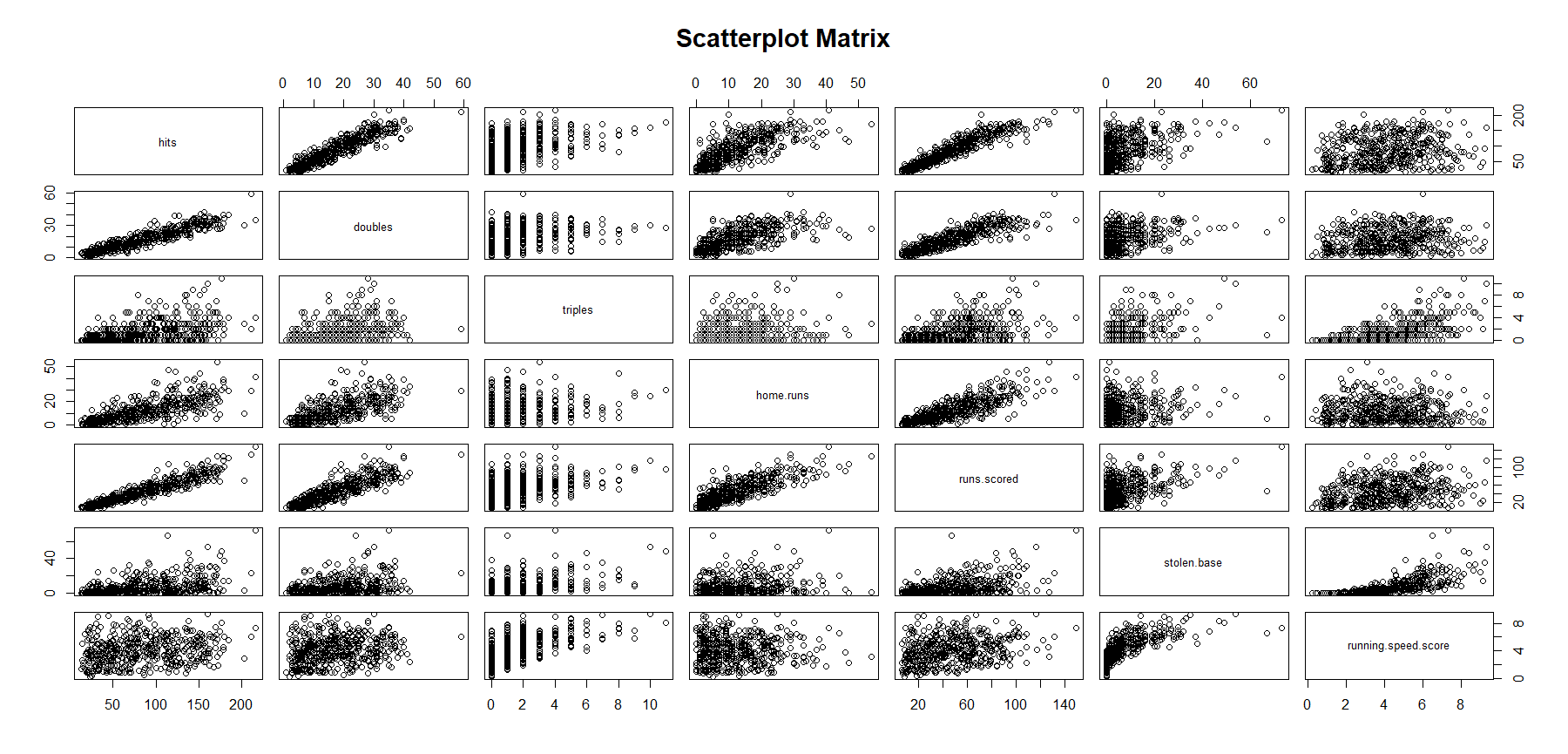


*Graph 2A: Dimension Reduction Variable Subset Correlation Plot*

From the correlation matrix and plot above we can see there are some strong correlations among variables such as runs.scored and hits and some moderate correlations such as that between home.runs and doubles. There are also some weak correlations, surprisingly, even a weak negative correlation between running.speed.score and home.runs.

**Outlier Investigation**

As with any form of data analysis, it is important to investigate possible outliers, as they can greatly skew results. A scatterplot matrix of the variables is a good starting point when attempting to detect outliers. The scatterplot matrix below plots all variables against all other variables in the dimension reduction variable subset. Strong linear relationships can immediately be identified, for example between hits and runs.scored. The majority of the relationships appear positively correlated. Although there are clearly outliers, for example in the stolen.base and doubles variables, more investigation is necessary before excluding any data points from the analysis. The triples variable scatterplots are noticeably different from the rest. Upon closer examination, that is due to the small values and discrete nature of these data points, along with an extremely small variance(3.47). Triples are a rare occurrence in baseball.

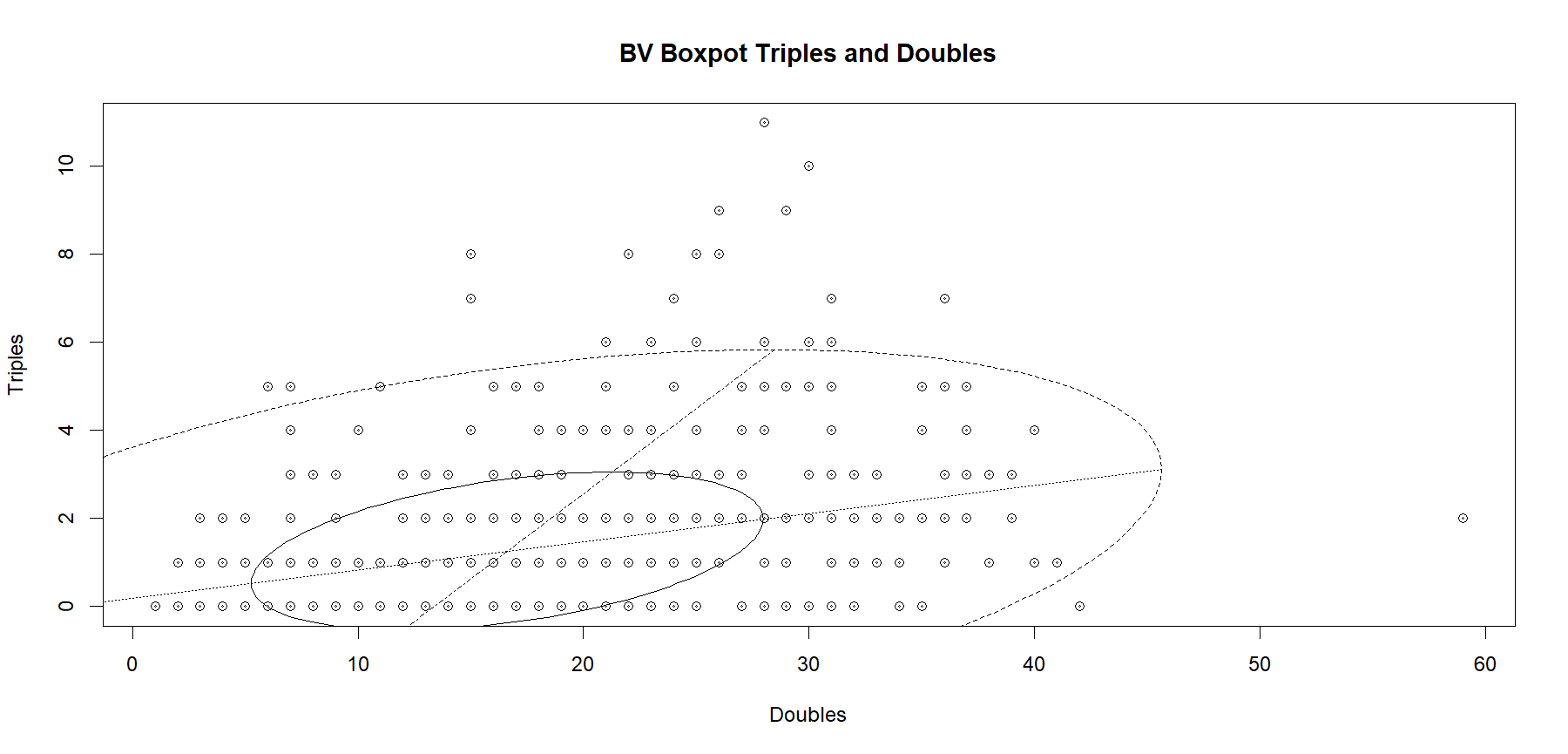


*Graph 2B: Dimension Reduction Variable Subset Scatterplot Matrix*

For a closer look into the possibility of outliers, it is necessary to construct and examine a bivariate boxplot. Bivariate boxplots provide a visualization based on “robust” measures of location, scale, and correlation. The plot includes 4 main components:

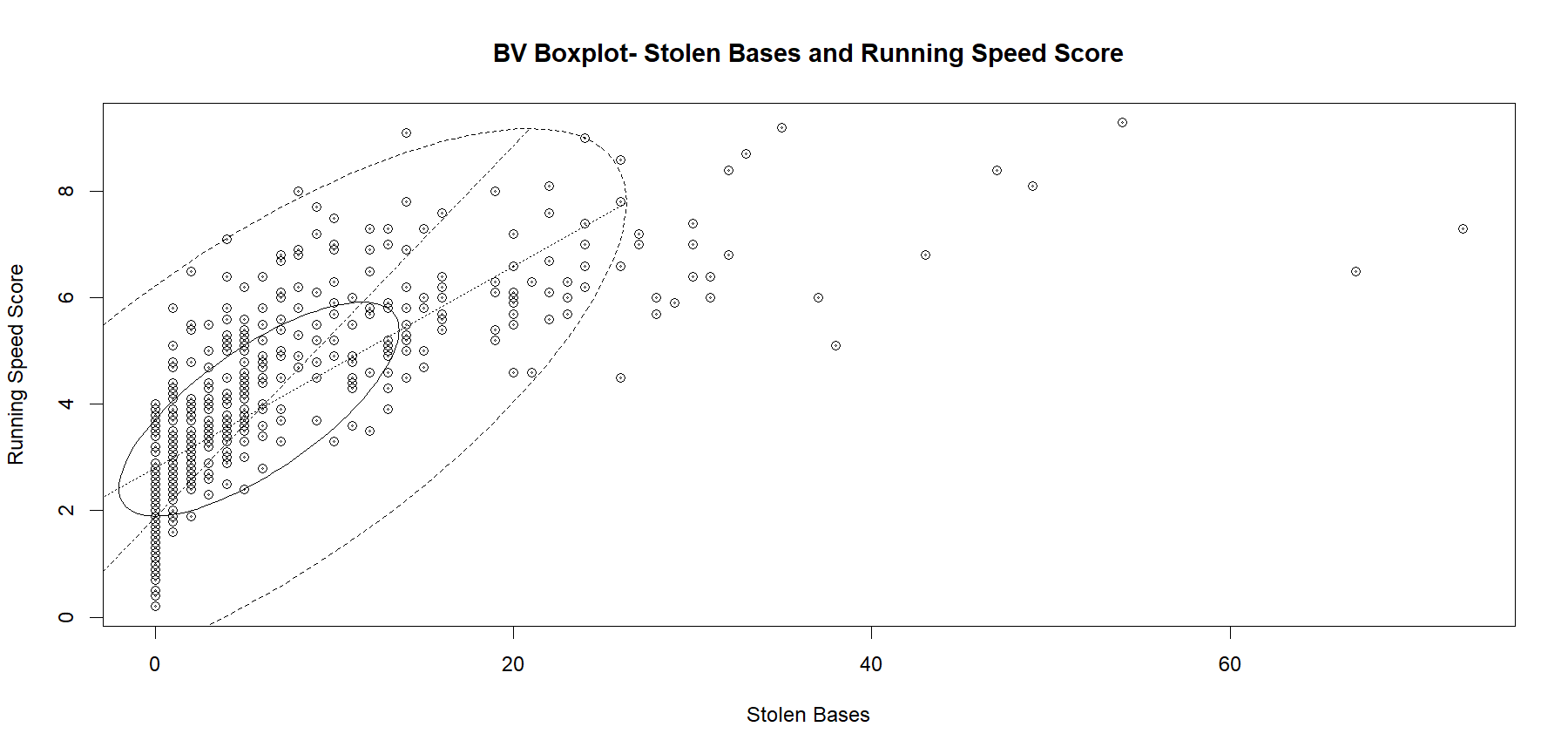
1. a hinge or bag- which contain approximately 50% of the data
2. the “fence”- which is analogous to a 99% confidence interval
3. regression lines for each variable that intersect at an angle indicating the strength of the correlation
4. datapoints- those outside the fence are generally considered outliers.5

Bivariate boxplots were built using different variables from the subset to identify potential outliers across all variables. Although points can be labeled with player names, 461 names on a plot hinders readability, so cross referencing is necessary. Boxplots with dots for points are provided here.



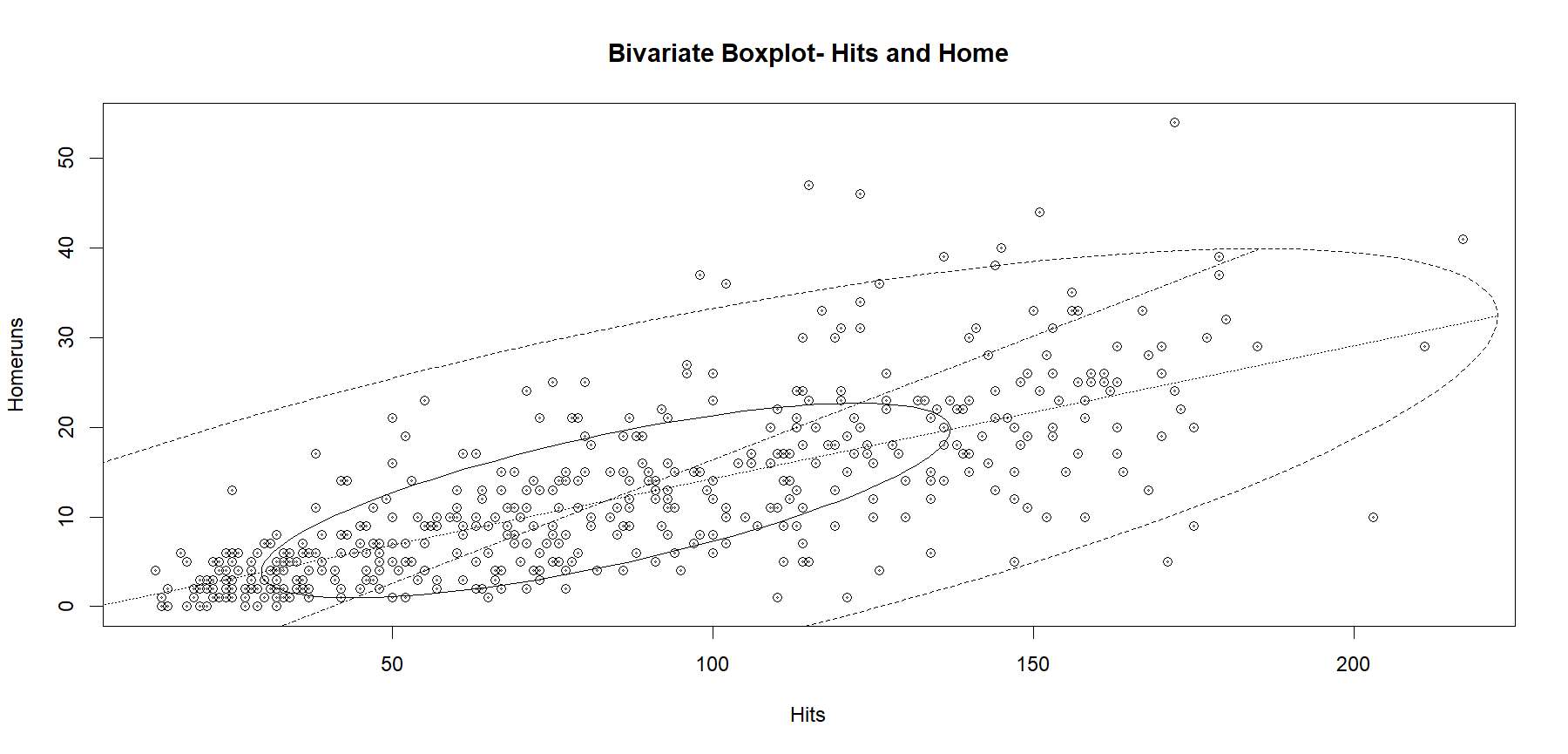
*Graph 2C: Bivariate Boxplot of Triples and Doubles*

This boxplot indicates several outliers, which is to be expected in a dataset with 461 observations. However, one player appears to be an extreme outlier. According to the outlier binary column in the clean data file, that player is Freddy Freeman with an astounding 59 doubles.



*Graph 2D: Bivariate Boxplot of Stolen Bases and Running Speed Score*

The bivariate boxplot of stolen.bases and running.speed.score also has several outliers, notably absent though, is Freddy Freeman, of doubles hitting fame. On this plot there appear to be two extreme outliers in the stolen.bases variable. According to the clean data file, these two points represent incredibly sneaky baseball players, Esteury Ruiz with 67 bases stolen and Ronald Acuna Jr., with 73 bases stolen.

**

*Graph 2E: Bivariate Boxplot of Hits and Homeruns*

This boxplot shows several outliers, but nothing too extreme. However, it is worth noting the point farthest to the right on the x-axis because it belongs to Ronald Acuna Jr., of base stealing infamy. Accordingly, the clean data file was consulted regarding this player and, since he is not an outlier in the majority of our variables of interest for dimension reduction, he will not be removed from the analysis. Since no observations consistently exhibit outlier behavior across the bivariate analyses above, they are all included in the dimension reduction analysis.

**Principal Components Analysis**

Because PCA is scale invariant, we must scale our data before conducting the analysis. After using the princomp function on our variable subset, we address the question of how many components are needed to account for the maximum amount of variance, while still reducing the dimensions. There are several schools of thought as to how many components must be kept, but a good rule of thumb, and one that is used here, is to keep enough components to account for at least 80% of the total variance. We can see from the cumulative proportion that the first two components account for 84% of the variance, so we will focus on the factor loadings of the first two components for our analysis.

Importance of components:

Comp.1 Comp.2 Comp.3 Comp.4

Standard deviation 2.0379064 1.2967988 0.69605714 0.57552453

Proportion of Variance 0.5945844 0.2407633 0.06936411 0.04742122

Cumulative Proportion 0.5945844 0.8353477 0.90471178 0.95213300

Comp.5 Comp.6 Comp.7

Standard deviation 0.46651029 0.28270132 0.19180792

Proportion of Variance 0.03115785 0.01144197 0.00526718

Cumulative Proportion 0.98329085 0.99473282 1.00000000

Additionally, we examine the factor loadings.

Loadings:

Comp.1 Comp.2

hits 0.456 0.205

doubles 0.432 0.220

triples 0.323 -0.373

home.runs 0.367 0.375

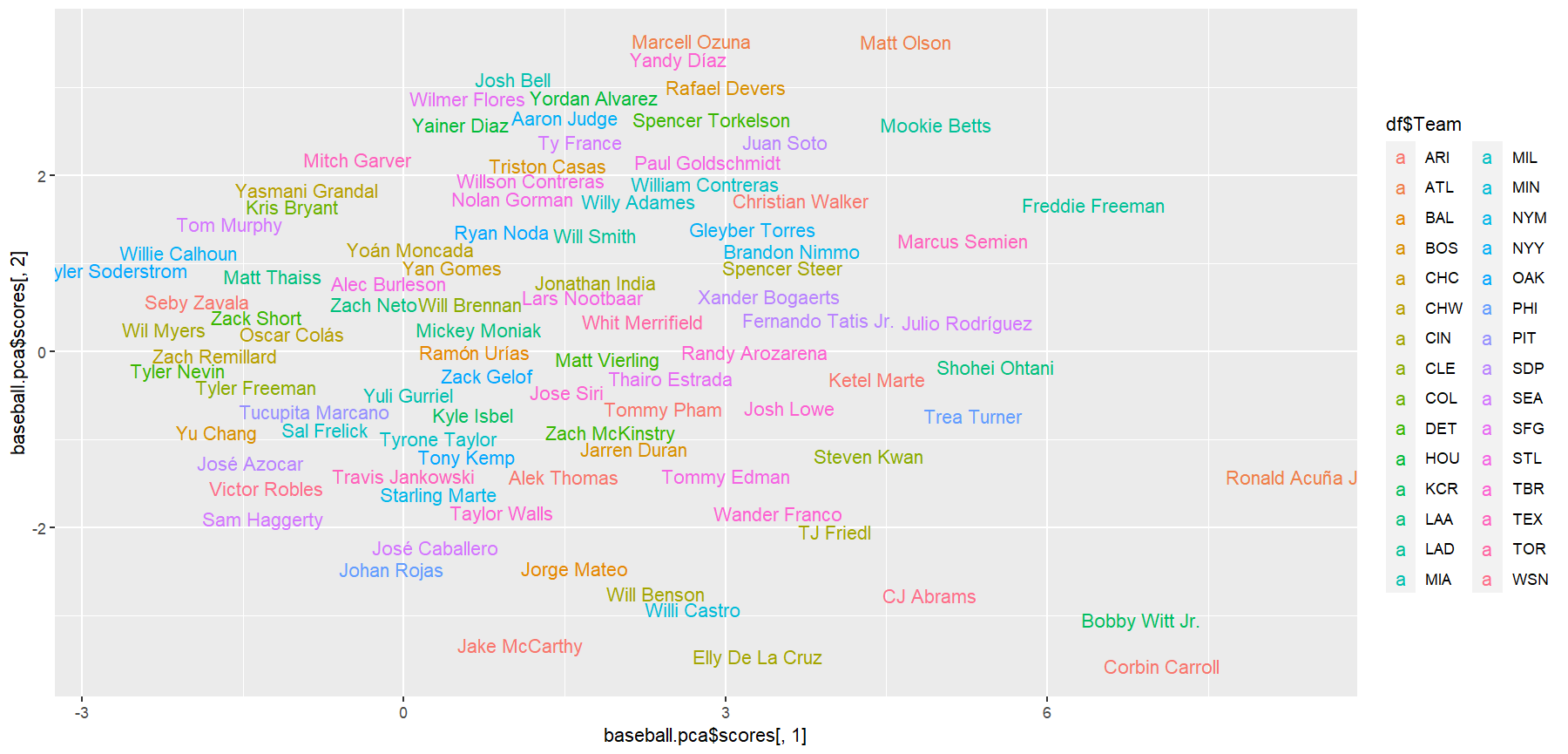
runs.scored 0.466 0.172

stolen.base 0.312 -0.455

running.speed.score 0.229 -0.627

Component 1 shows a combination, as all loadings are positive. The largest values are for the variables hits, doubles, and runs.scored. These variables are all related to scores and hitting the ball. The second component has larger loadings on the stolen.base variable and the running.speed.score variable. Both of these variables are related to the speed of a player.

Additionally, it is interesting to note that there is a fairly large negative loading on the triples variable, which is also highly tied to speed. Only the fastest players will be able to make it to 3rd base, despite not having hit the ball out of the park, also known as a homerun. Lastly, the contrast between triples and home.runs further supports this conjecture, indicating this component to be driven by players who are getting triples, but not homeruns. The players can be plotted according to their scores on the first and second components, as seen below.



*Graph 2F: Observations plotted according to component scores 1 and 2*

For interpretation of this plot, we can return to our previously stated outliers. Ronald Acuna Jr. appears to have high scores in component 1, which tracks with his hits outlier status. He also appears on the lower end of component 2 scores, which is representative of his stolen.bases outlier status, when considering the large loadings on component 2 were negative values. The plot seems to separate the players on component 1 by hitting and scoring abilities and on component 2 by speed. This is supported when cross referencing the clean data file for Corbin Carroll and Elly De La Cruz, who are among the fastest in the league.

The last plot we will reference for understanding of the principal components is the biplot. The biplot has a point for each observation and vectors representing each variable. Vectors with small angles between them indicate strong correlations, while wider angles show weak correlations among the variables. Longer vectors belong to those variables that are well represented within the first two principal components. Vectors that point to an observation indicate a large value for that observation in that variable. Although 461 observations is a large number of observations for a plot of this nature in this document, we can gain some important insights from the biplot in a coding environment, such as very strong correlations between doubles, hits, and runs.scored. We can also see a strong correlation between triples and stolen.bases. The angle between home.runs and running.speed.score vectors is almost a right angle, representing the two variables correlation of -.006. We see familiar observations such as Freddy Freeman, placed directly on the doubles vector, for which he is an extreme outlier and Elly De La Cruz situated along the running.speed.score vector, for which he is an outlier, exactly where we might expect.



*Graph 2G: Biplot of PCA scores*

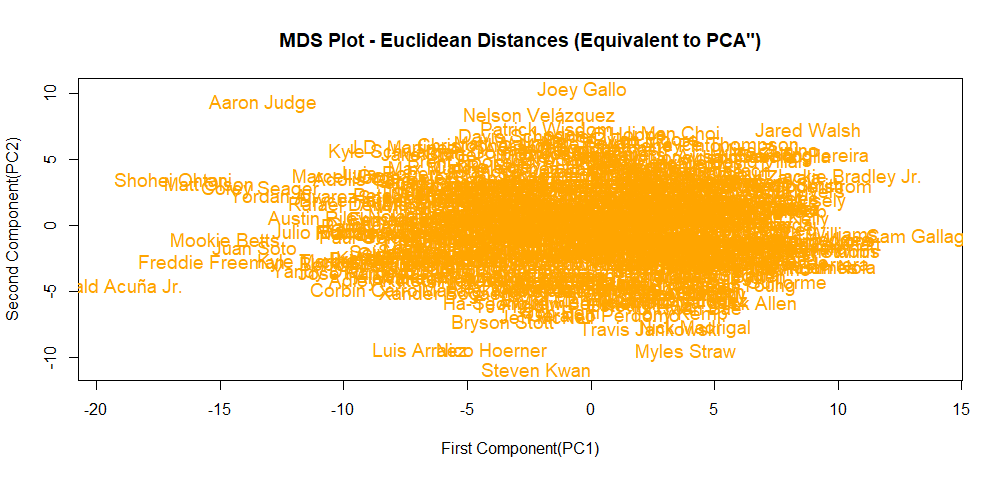
PCA may not always be the best technique for dimension reduction, but it provides a comprehensive understanding of the relationships in a high dimension space. It enables us to identify 2 variables which could be plotted to summarize the subset. It also pinpoints which variables contribute the most to the variance of the dataset and indicates their relationship. The reduced number of variables can be useful for further analysis and we were able to get an idea about player ranking across multiple attributes.

horizontal line**Cluster Analysis**

Cluster analysis is a statistical method that is used in order to group similar data points together. By creating these groups/clusters, the goal is to better explore the structure of the data as well as patterns that exist within the dataset, where objects/data points which have common traits are put in the same clusters and those with different traits separated. Likeness is measured based on distances between the data points. To perform cluster analysis, steps that need to be taken are measuring the distance, performing cluster analysis, and evaluating our results.

**Measuring Distance**

The first step in order to perform cluster analysis is measuring the distance between data points in the dataset. Whereas there are various methods to measure the distance including Euclidean or Manhattan. With our MBL dataset, we have decided to use the Euclidean distance, which is calculated by considering the length of the shortest distance between two points. This is because our data is mostly continuous and is composed of mostly numeric variables.



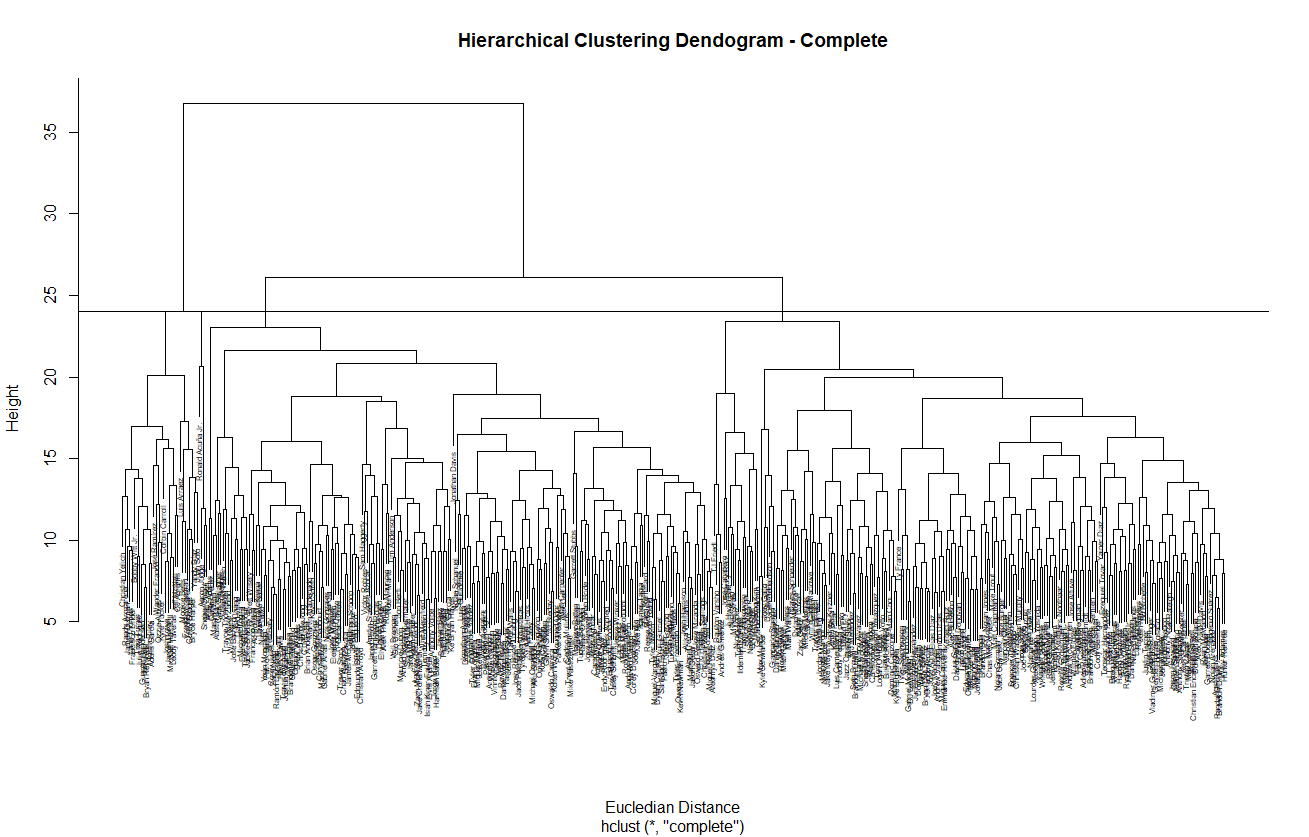
*Graph 3A: Euclidean Distance*

Based on the MDS plot in Graph 3A, it is clear that the data points are very condensed with very small distances between each other, which indicates strong relationships between different MLB players in this dataset. This result leaves us to hope that there might be some groups of batters with similar traits in the same group and different traits with players from other groups/clusters. We will explore in the following step as we perform the cluster analysis. Furthermore, it makes sense that these data points are correlated due to the fact that there is a higher likelihood of finding similar traits in batters in the MLB.

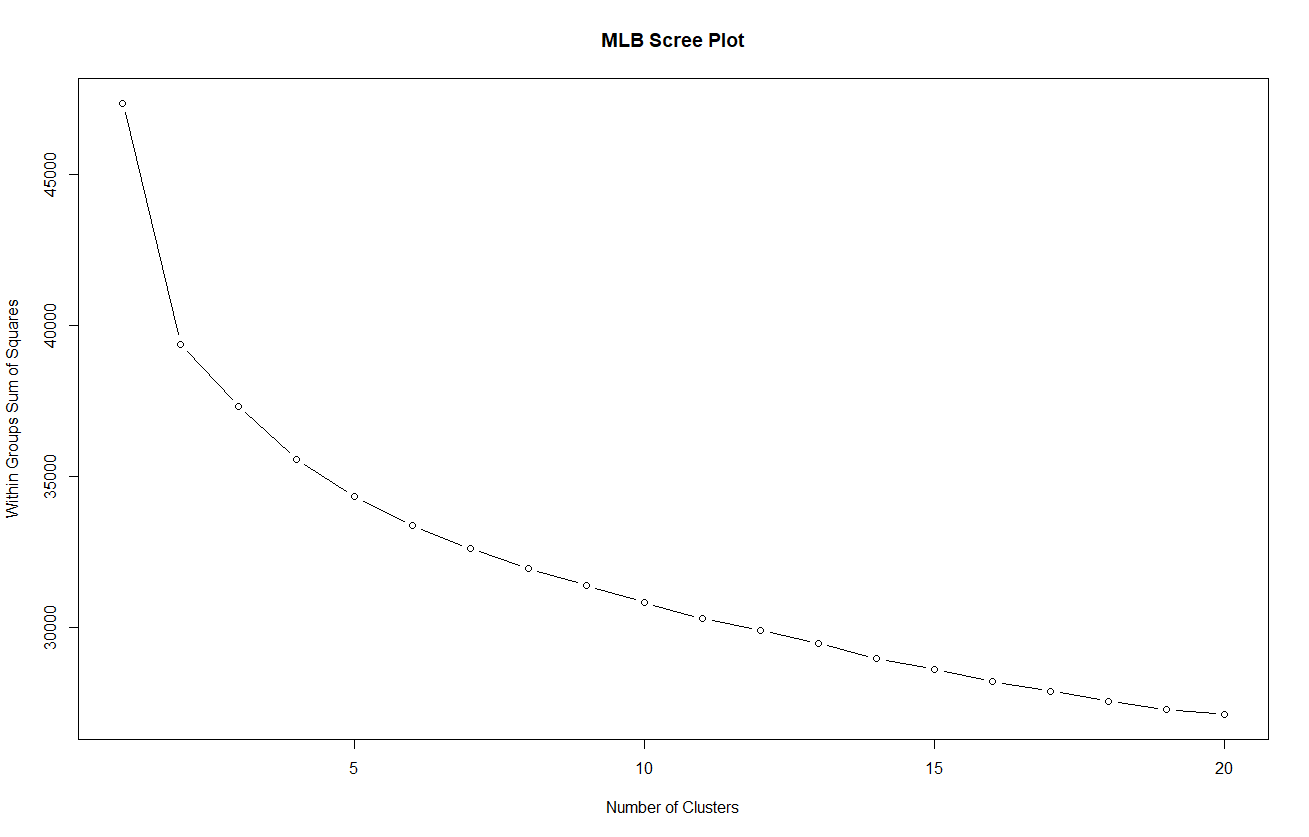
**Cluster Analysis Performance**

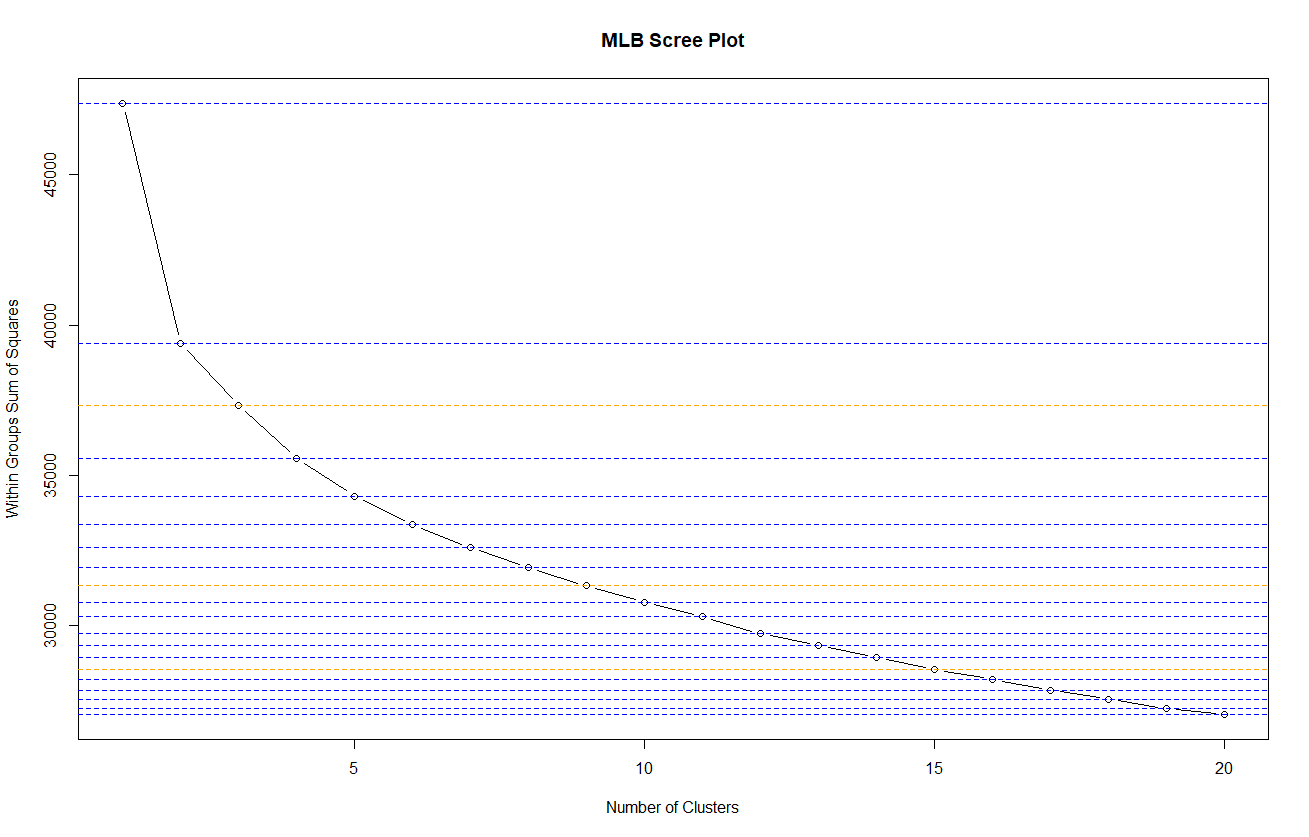
K-means cluster analysis is one of many clustering methods. This clustering method sections data points into a predefined number of clusters (k) based on the mean value of the data points. It is iterative until convergence (cluster assignment does not change significantly) and assigns data points to the cluster with the nearest centroid. Due to its fast computation time, the high number of variables, and a relatively big dataset, we decided to go with the K-means clustering method.

Through an iterative method of calculating the within groups sum of squares (WGSS). We decided to go for three clusters (k=3) which will be shown in the scree plot in Graph 3C. By using the iterative method with 20 iterations, we were able to reduce our WGSS by 69.54. It is also important to mention that we used different other clustering methods and amongst them the “complete” hierarchical clustering method. Whereas this method is not the best with larger datasets, our main goal with it was to help us decide on the number of clusters that made more sense. Using the abline() function in R, we stopped at height 24 and got three clusters, which helped us as we were deciding on our k value in K-means clustering.

**

*Graph 3B: Hierarchical Clustering (Complete)*

*Graph 3C: Scree Plot using the K-means clustering method*

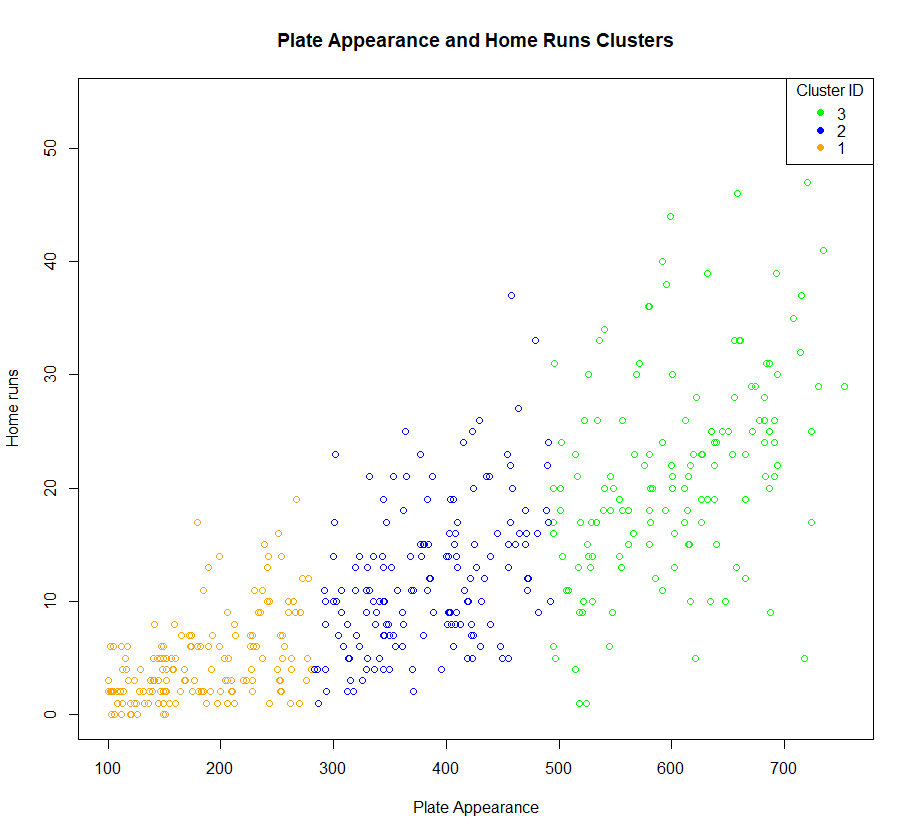
**

*Graph 3D: Scree Plot using the K-means clustering method with lines that determine clusters.*

In addition to showing us each cluster, the lines in Graph 3D are helping us clearly see where the WGSS are starting to converge. We have stopped at the first orange line and considered our k value to equal three.

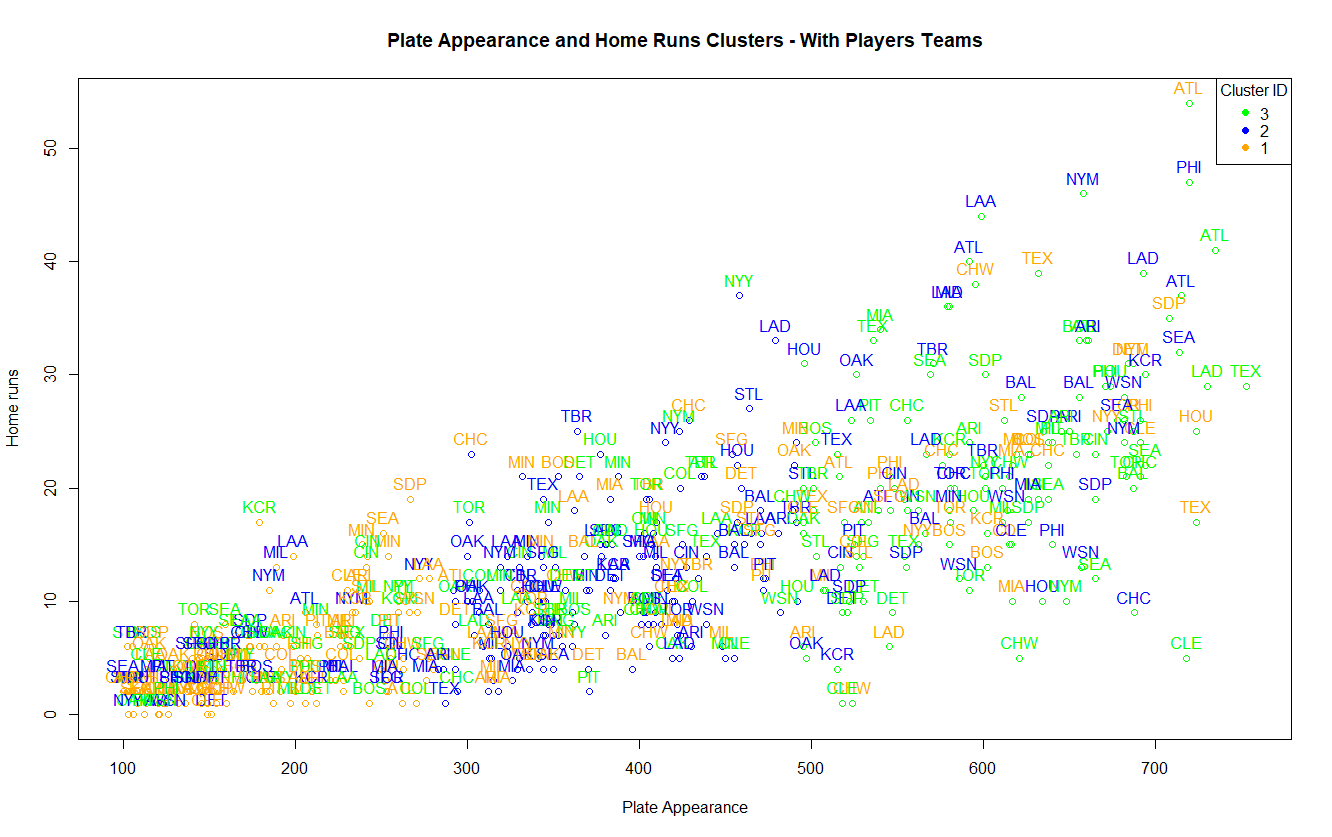
**Evaluation of Generated Clusters**

We chose one pair of variables in our dataset and performed the K-means clustering on them. These variables are plates.appearance (refers to a batter's turn at bat and includes other factors like walks, hit by pitch, sacrifice hits, and sacrifice flies) and home runs (a type of hit where the batter successfully hits the ball out of the playing field, allowing the batter and any runners on base to score).

*Graph 3E: Plate Appearances v Home Runs Cluster* 

These two variables are highly correlated as many others are in the MLB dataset, however, we were more interested in the players grouping criterias in individual clusters. First of all, the three clusters were composed of sizes 160, 142, and 159 respectively.

As the graph showcases, the three clusters are represented by colors; orange, blue, and green. The clusters were mostly formed based on the number of plate appearances and home runs each player achieved. This will be more discussed when we talk about the z-scores.



*Graph 3F: K-means Clusters for Plate Appearance and Home Runs with Players’ Teams as Labels*

Another interest that we had as we were creating clusters is exploring whether the players in clusters were coming from the same teams or at least many players in one cluster were coming from the same teams. As shown in Graph 3F, teams that players come from are all over the place in all clusters. This means that players from different teams can have similar abilities and can appear in different clusters when it comes to plate appearance and home runs.

**Z-score Tables per Cluster**

The following tables represent z-scores of home runs and plate appearances in different clusters. Only the first and last 20 records are displayed.

**Cluster #1**

| **First 20 records with teams - Cluster 1** | **Last 20 records with teams - Cluster 1** |
| --- | --- |
|  |  |

*Table 3A:Z-scores players in Cluster 1*

Although the first cluster is the one on the lower left of our graph in terms of raw values, it has the highest z-scores due to the normalization process involved in calculating z-scores. This high z-score is because this cluster’s values are relatively higher compared to the overall average especially if we look at plate appearances. For example, Zach McKinstry has 518 plate appearances and 9 home runs. Yandy Diaz has 600 plate appearances and 22 home runs. Lastly, we have Alex Verdugo with 602 plate appearances and 13 home runs. It is crucial to mention that as the number of plate appearances increase, the number of home runs increase and vice versa, this is why these two variables are strongly correlated.

**Cluster #2**

| **First 20 records - Cluster 2** | **Last 20 records - Cluster 2** |
| --- | --- |
|  |  |

*Table 3B:Z-scores players in Cluster 2*

Looking at the z-scores for the second cluster, most are lower or close to the mean. This is because most of the values for home runs and plate appearances for this cluster are around the three hundreds (300) for the plate appearances. This aligns with the average plate appearances being ~381.

**Cluster #3**

| **First 20 records - Cluster 3** | **Last 20 records - Cluster 3** |
| --- | --- |
|  |  |

*Table 3C:Z-scores players in Cluster 3*

The lowest z-scores in the third cluster supports the fact that players like Billy McKinney, Austin Wynns, Anthony Rendon, and Victor Robles among a few have the lowest plate appearances of which are 147, 145, 183, 126 respectively.

horizontal line**Confirmatory Factor Analysis**

**EFA v CFA**

In order to identify further relationships and underlying dimensions that affect batting statistics, we can undergo Exploratory Factor Analysis (EFA). This will identify potential underlying latent variable groupings that might not be obvious through our initial investigation. We can then use these groupings to assist us in forming baseball centric factors that underlie and explain our data in baseball terms.We will then model and assess these new factors through Confirmatory Factor Analysis (CFA) to confirm if they represent our data.

In short, we will run EFA and analyze the resulting factor loadings to assist in the identification of new factors that help explain our data in baseball terminologies. The process was completed in R and is described below:

**Rotational Issues**

We initially ran EFA using the factanal function. However, the process was not able to complete and resulted in the error:

Error in solve.default(cv) :

system is computationally singular: reciprocal condition number = 1.6762e-18

This can be due to the high autocorrelation across the variables and its skewed dimensionality (461 observations and 105 variables), as shown in our data analysis section.

This was confirmed when investigating the determinant of the covariance of the numerical values in our data, represented as A below.

det(cov(A))

This gives a value close enough to be considered zero, indicating that we are unable to invert the covariance matrix and as such we are unable to rotate our covariance matrix during factor analysis.

**Variable Selection**

Due to the high number of variables that have strong correlation within our data, we underwent an interactive process using EFA with different variables selected. Through this process we identify three fields which when removed allow for EFA to complete as the covariance matrix of the data is now invertible.

These fields included:

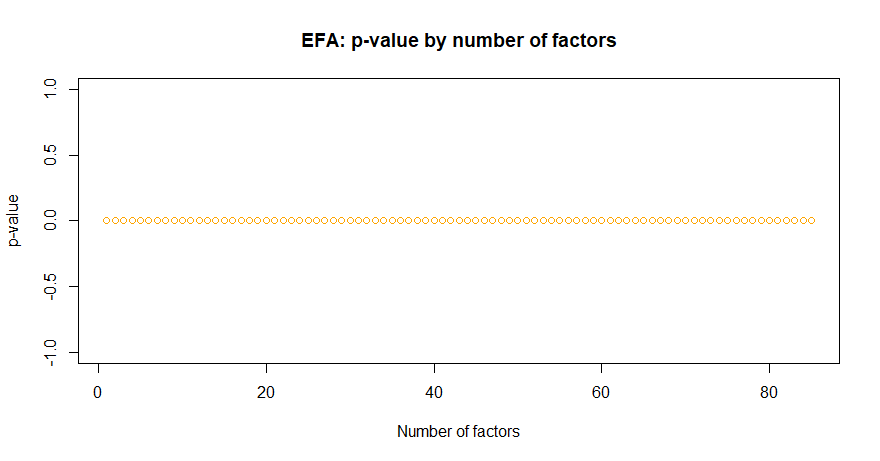
***'home.runs', 'hit.batted.ball'*** and ***'Base.Running'***

**Identifying Number of Factors**

To assess the optimal number of factors, we employ an approach which involves two methodologies:

1. Repeatedly executing the factanal function with varying factor counts and scrutinizing the resulting p-values.
2. Assess the cumulative variance of the factors below then set a threshold of 80%.

Our first approach observed that p-values for all possible factor counts (up to 85) were at or near zero, signifying their statistical significance. This finding suggests that no possible number of factors would be able to significantly represent our data



*Graphs 4A: EFA p-value results by number of factors*

Our second approach resulted in only a slightly more practical number of factors, as these factors would represent approximately 80% of the data variance and could be considered a good approximation of the data. However the below table shows that the first 30 factors would result in a cumulative variance of under 80% of the data:





*Table 4B: First 30 factor loadings*

However, this number of factors is not feasible as it does not lend to an easy interpretation of our baseball data. As such, knowing that our data may not be fully represented by our data, we can only hope to limit the number of new factors to 4 by finding groupings of variables with regards to baseball terminologies.

**EFA Run**

Rerunning the EFA for 4 factors, we analyze the factor loadings for values greater than 0.5 or less than -0.5 as this indicates a strong positive or negative correlation of the factor with the corresponding variable, respectively.

The table below shows the factor loadings with values greater than 0.5 or less than -0.5, labeled by their variable name. It is noted that there are some variables that are present across multiple factors:



*Table 4C: Factors with significant variables*

As we can note from Table 4C, these factors have significant relationships with only 56 variables; with the remaining variables not having any significant loading values. We therefore see that our new modeled factors may not represent all the data variables as we aren't able to interpret any groupings from these missing variables.

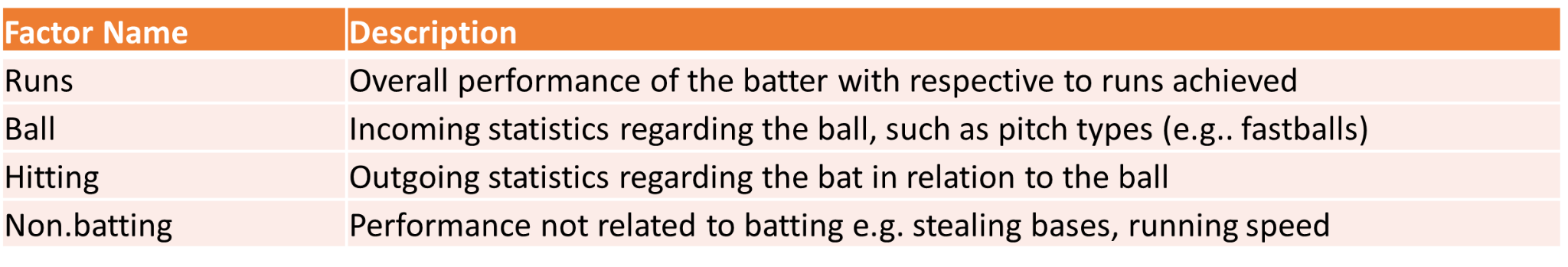
**Determining New Factors**

From Table 4C, we attempt to identify baseball terminology groupings that will encompass these variables and ultimately assist in our model creation.

We resultantly align with the EFA factors and try to identify four underlying areas that each factors encompasses:

* Factor 1 pertains more to variables that align to runs, onbase percentages (wOBA), batting averages and slugging percentage i.e. batting and score/run statistics
* Factor 2 aligns similarly to the variables in factor 1 but more focused on ‘non-batting’ statistics such as plate appearances and games played, with some score and batting metrics i.e. non-batting and score statistics
* Factors 3 and 4 have some crossover of variables however they generally relate to ball statistics, such as how the ball is pitched and where the ball is in relation to the bat or field

This defines our ‘new factors’ in four groupings:



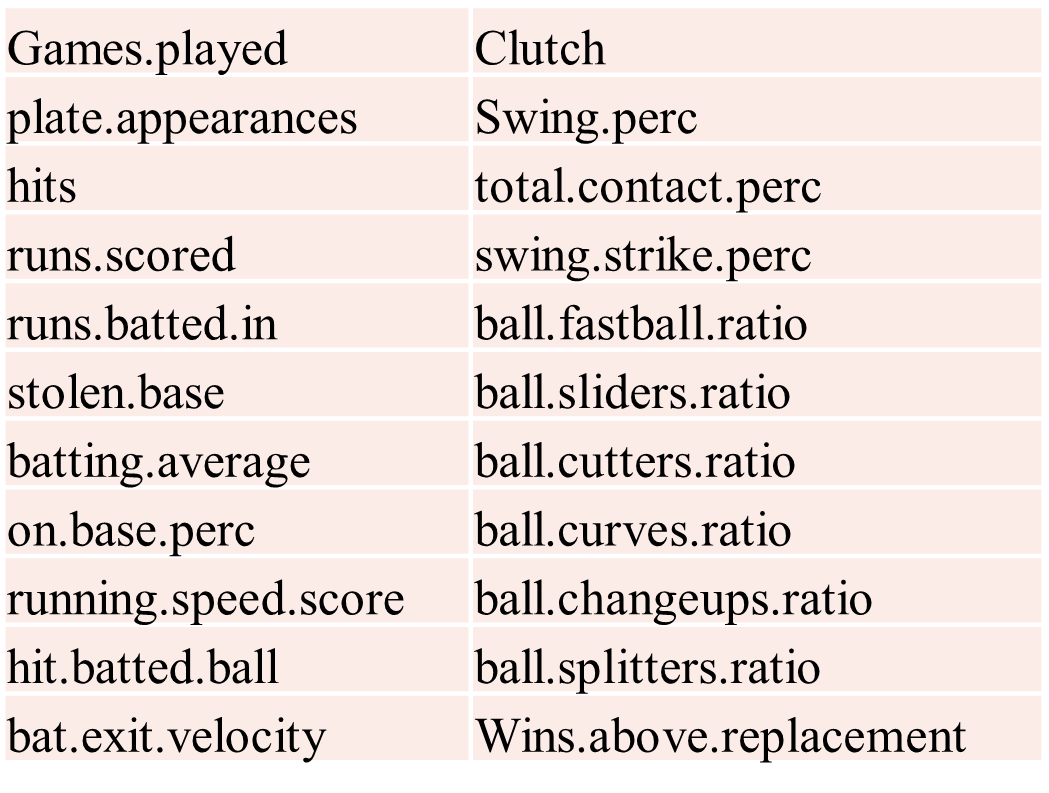
*Table 4D: New Factors*

Next, we test if these new factors represent our data by defining a model where the data variables are linear combinations of our new factors and assess this model through the CFA process.

**High Number of Variables**

In lieu of the complexity of defining a model for over 100 variables, we will limit the number of variables.These variables should provide us with a broad range that encompasses most aspects of the batting statistics present in the dataset and our newly defined factors. This should give us an indication of how our model fits to statistical batting data however we should note that this is not ideal as we are removing potentially important variables for factor analysis.

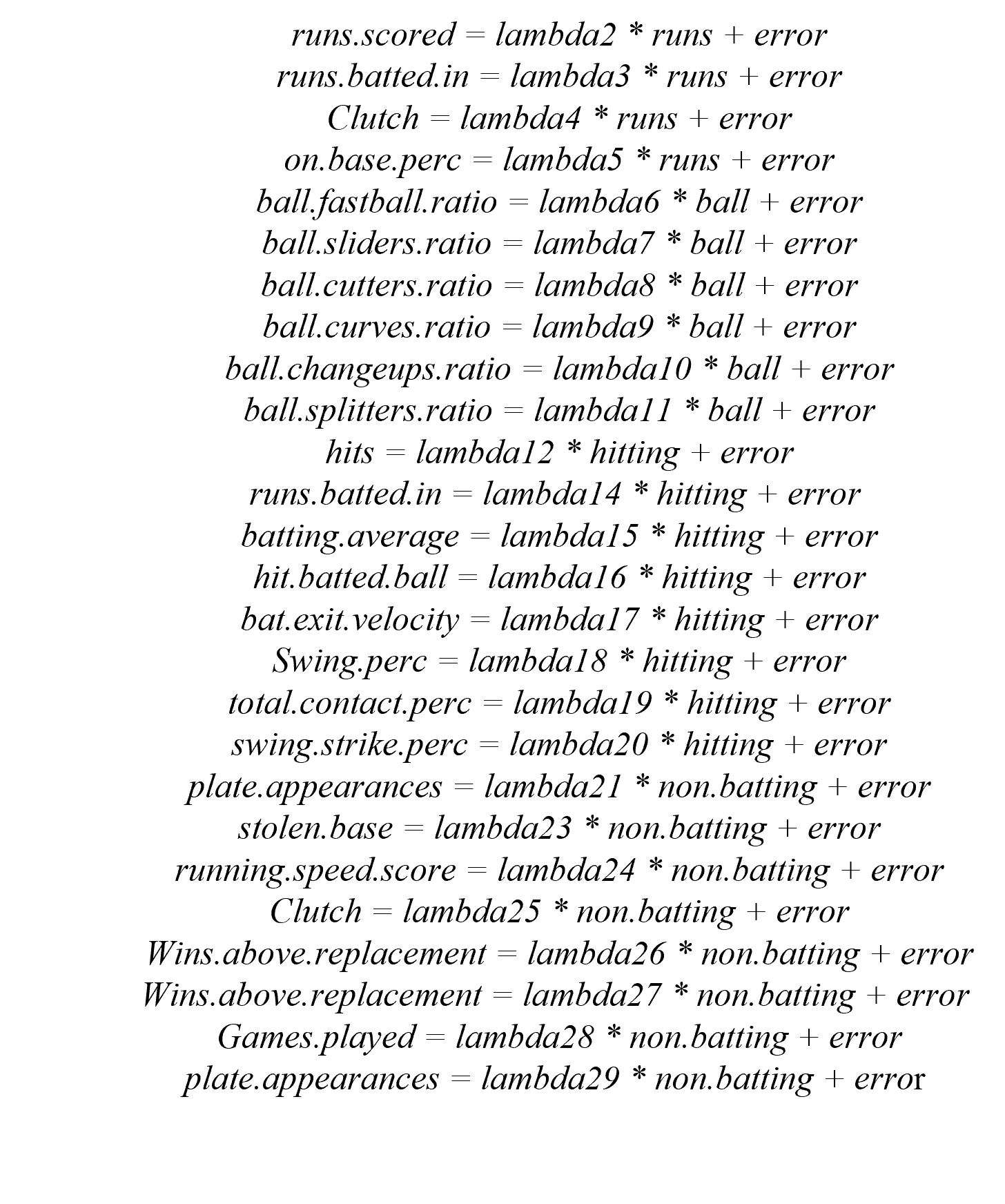
Key variables excluding the categorical variables are as follows:



*Table 4E: Key numeric variables for Model Creation*

**Model Creation**

We now define our model by writing out the variables as linear combinations of the factors:

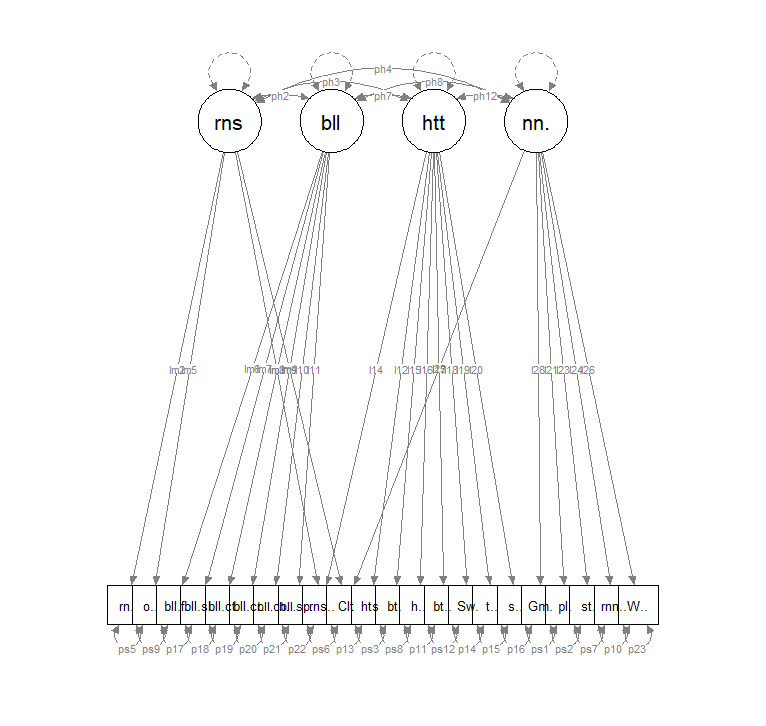
**

*Graph 4F: CFA model*

**CFA Using Sem**

Using the sem library, we pass our model as input to estimate the factor loadings, uniqueness and covariance values. The resulting output can be visualized via a path diagram; although it may be

hard to interpret due to the number of variables:



*Graph 4G: Path diagram of completed CFA model*

Although the values were determined; the model did not converge and resulted in a warning, with a Chisquare value of -776298956:

In eval(substitute(expr), data, enclos = parent.frame()) :

Could not compute QR decomposition of Hessian.

Optimization probably did not converge.

As such we cannot investigate the model through any fitness tests

Possible reasons for not converging could include:

* Underidentified Model: The model may be underidentified, meaning there are not we may be missing key baseball variables to estimate all the model parameters.
* Parameter Estimates: Our factors and their parameters may be implausible and we would need to redefine new factors.
* Model Complexity: Our data has a large number of variables but a low number of samples which can hinder convergence

**CFA Using Lavaan**

We repeat CFA using the Lavaan library, which resulted in a converged model, however this outputted a warning where the estimates are unreliable and without defined p-values for the variables.

lavaan 0.6.16 did NOT end normally after 1988 iterations

\*\* WARNING \*\* Estimates below are most likely unreliable

**Analysis of CFA and EFA**

We can only conclude that the data does not fit our hypothesized factors and defined parameters.

This is most likely as result of the high correlation present in the data, which is reaffirmed by the high number potential factors recommended in the EFA analysis and the skewed dimensionality of the data. The data is also sampled across multiple areas within baseball, therefore our batting statistics may be obscured by external influences that aren't present in the dataset and as such key relationships are missing.

Although there is rich information that is present in our data we aren't able to identify factors that represent this data holistically, as such we can conclude that factor analysis is not an optimal methodology for explaining this data.

horizontal line**Conclusion**

There's an infinite number of paths one can take when investigating a dataset. In this scenario, we utilized several statistical methods to attempt to unlock the story this data had to tell. It was not without its struggles, our dataset is a prime example that too much correlation can be a bad thing. When you take courses such as these, data is strategically picked to have the outcome of what is being taught. In the real world, that is not a luxury we often have. This gave the opportunity to work through real world issues and problem solving. Along with real world experience, we were able to collect valuable insights using Dimension Reduction, Clustering, and Confirmatory Factor Analysis.

horizontal line

**Appendix**

**Citations and References**

1. “Major League Baseball Hitting.”, Sept. 2023-Nov. 2023

<https://www.kaggle.com/datasets/m000sey/major-league-baseball-hitting-data>

1. “Fan Graphs.”, Nov. 2023

<https://library.fangraphs.com/>

1. “MLB Career Batting Leaders”, Nov. 2023

<https://www.espn.com/mlb/history/leaders>

1. Multivariate GitHub, Nov. 2023

https://github.com/Slyth3/Multivariate-Analysis-Baseball-Hitting-Statistics

1. “ISQS 6350: Principal Components Analysis, Module 4” Notes, Brown, Eric, Sept. 2023-Nov. 2023
2. “An Introduction to Applied Multivariate Analysis with R.”, Everitt, Brian and Torsten Hothorn, Sept. 2023-Nov. 2023

**Data Fields**

| **Variable Name Original** | **Variable Name ‘cleandata.csv’** | **Type** | **Description** |
| --- | --- | --- | --- |
| Name | Name | <chr> | hitter's name |
| Team | Team | <chr> | hitter's team (or last team they were on) |
| Clutch | Clutch | <dbl> | (Win Probability Added / a hitter's Leverage index for all game events) - (Win Probability Added / Leverage index), which essentially measures how much better a player does in a high leverage situation compared to a neutral situation. |
| Swing. | Swing.perc | <dbl> | % of total pitches a batter swings at |
| SwStr. | swing.strike .perc | <dbl> | Swinging strike % |
| Barrel. | bat.barrel.perc | <dbl> | % of Batted balls that are classified as barrels |
| F.Strike. | F-Strike.perc | <dbl> | First pitch strike percentage |
| fastball. | ball.fastball.perc | <dbl> | The percentage the hitter sees a fastball |
| Batting | Batting.wOBA | <dbl> | Park adjusted runs above average based on wOBA |
| AVG | batting.average | <dbl> | batting average |
| BB. | BB.PA | <dbl> | BB / PA |
| K. | strikouts.plate.perc | <dbl> | SO/PA |
| BB.K | bases.on.balls.strikeouts | <dbl> | BB/SO |
| OBP | on.base.perc | <dbl> | on base percentage |
| SLG | slugging.perc | <dbl> | slugging percentage |
| OPS | OPS | <dbl> | OBP + SLG |
| ISO | Isolated Power | <dbl> | SLG - AVG |
| Spd | running.speed.score | <dbl> | running speed score |
| BABIP | on.balls.in.play | <dbl> | AVG on balls in play |
| UBR | ultimate.base.running | <dbl> | ultimate base running in runs above average |
| velocity wGDP | wGDP | <dbl> | weighted ground into double play runs above average |
| wSB | stolenbase.stealing.ratio | <dbl> | SB and CS runs above average |
| wRAA | wRuns.AboveAvg | <dbl> | weighted runs above averaged based on wOBA |
| wOBA | wOBA | <dbl> | weight on base percentage average |
| GB.FB | GroundBall.fly.ratio | <dbl> | ground ball to fly ball ratio |
| LD. | LineDrive.perc | <dbl> | line drive % (LB / balls in play) |
| GB. | GroundBall.perc | <dbl> | ground ball % (GB/ balls in play) |
| Flyball. | Flyball.perc | <dbl> | flyball% also commonly known as FB% (Flyball/ balls in play) |
| IFFB. | Infieldfly.perc | <dbl> | infield flyball % (in field flyball / flyballs) |
| HR.FB | HomeRun.Flyball.ratio | <dbl> | home run / Flyball |
| IFH. | hit.infield.perc | <dbl> | IFH / GB |
| BUH. | hit.bunt.perc | <dbl> | BUH / bunts |
| Pull. | hit.pull.perc | <dbl> | % of balls that were pulled by hitter |
| Cent. | hit.CF.perc | <dbl> | % of balls that were pushed by hitter |
| Oppo. | hit.push.perc | <dbl> | % of balls that were hit to CF by hitter |
| Soft. | hit.softspeed.perc | <dbl> | % of balls hit in play that were classified as hit with soft speed |
| Med. | hit.medspeed.perc | <dbl> | % of balls hit in play that were classified as hit with medium speed |
| Hard. | hit.hardspeed.perc | <dbl> | % of balls hit in play that were classified as hit with hard speed |
| EV | bat.exit.velocity | <dbl> | average exit velocity of Batted ball |
| maxEV | bat.max.exit.velocity | <dbl> | maximum exit velocity of Batted ball |
| LA | launch.angle | <dbl> | Launch angle |
| HardHit. | bat.hardHit.perc | <dbl> | % of Battled balls with an EV of 95 or higher |
| xBA | x.batting.average | <dbl> | expected batting average |
| xSLG | x.slugging.perc | <dbl> | expected slugging percentage |
| xwOBA | x.wOBA | <dbl> | expected weighted on base average |
| O.Swing. | out.swing.perc | <dbl> | % of pitches a batter swings at outside of the strike zone |
| Z.Swing. | in.swing.perc | <dbl> | % of pitches a batter swings at inside of the strike zone |
| O.Contact. | out.Contact.perc | <dbl> | % of times a batter makes contact with the ball when swinging at pitches thrown outside of the zone |
| Z.Contact. | in.contact.perc | <dbl> | % of times a batter makes contact with the ball when swining at pitches thrown inside of the zone |
| Contact. | total.contact.perc | <dbl> | total percentage of contact made when swinging at all pitches |
| Zone. | Zone.perc | <dbl> | % of pitches seen inside the strike zone |
| CStr. | called.strike.perc | <dbl> | Called strike % |
| CSW. | called.swing.strikes.perc | <dbl> | SwStr% + CStr% |
| wFB | ball.fastball | <dbl> | How well does the batter do vs fastballs? Using pitch types linear weights |
| wSL | ball.sliders | <dbl> | How well does the batter do vs sliders? Using pitch types linear weights |
| wCT | ball.cutters.perc | <dbl> | How well does the batter do vs cutters? Using pitch types linear weights |
| wCB | ball.curves | <dbl> | How well does the batter do vs curves? Using pitch types linear weights |
| wCH | ball.changeups | <dbl> | How well does the batter do vs change-ups? Using pitch types linear weights |
| wSF | ball.splitters | <dbl> | How well does the batter do vs splitters? Using pitch types linear weights |
| wFB.C | ball.fastball.ratio | <dbl> | How well does the batter do vs fastballs per 100 pitches? |
| wSL.C | ball.sliders.ratio | <dbl> | How well does the batter do vs sliders per 100 pitches? |
| wCT.C | ball.cutters.ratio | <dbl> | How well does the batter do vs cutters per 100 pitches? |
| wCB.C | ball.curves.ratio | <dbl> | How well does the batter do vs curves per 100 pitches? |
| wCH.C | ball.changeups.ratio | <dbl> | How well does the batter do vs change-ups per 100 pitches? |
| wSF.C | ball.splitters.ratio | <dbl> | How well does the batter do vs splitters per 100 pitches? |
| FBv | ball.fastball.velocity | <dbl> | the velocity of incoming fastballs seen |
| SL. | ball.sliders.perc | <dbl> | The percentage that the hitter sees a slider |
| SLv | ball.sliders.velocity | <dbl> | the velocity of the incoming slider seen |
| CT. | ball.cutters.perc | <dbl> | The percentage that the hitter sees a cutter |
| CTv | ball.cutters.velocity | <dbl> | the velocity of the incoming cutter seen |
| CB. | ball.curves.perc | <dbl> | The percentage that the hitter sees a curve |
| CBv | ball.curves.velocity | <dbl> | the velocity of the incoming curve seen |
| CH. | ball.changeups.perc | <dbl> | The percentage that the hitter sees a change-up |
| CHv | ball.changeups.velocity | <dbl> | the velocity of incoming change-up seen |
| SF. | ball.splitters.perc | <dbl> | The percentage that the hitter sees a splitter |
| SFv | ball.splitters.velocity | <dbl> | the velocity of the incoming splitter seen |
| Base.Running | Base.Running | <dbl> | Baserunning runs above average including SB and CS |
| Fielding | Fielding | <dbl> | Fielding runs above average based on UZR |
| WAR | Wins.above.replacement | <dbl> | Wins above replacement |
| HardHit | bat.hardHit | <int> | # of Batted balls with an EV of 95 or higher |
| Barrels | bat.barrels | <int> | a batted ball with an exit velocity of at least 98mph and LA between 26-30 degrees. For EV mph over 98 degrees, the LA range gets higher by 1 degree |
| G | Games.played | <int> | games played |
| AB | at.bats | <int> | # of at bats |
| PA | plate.appearances | <int> | plate appearances |
| H | hits | <int> | hits |
| X1B | singles | <int> | singles |
| X2B | doubles | <int> | doubles |
| X3B | triples | <int> | triples |
| HR | home.runs | <int> | home runs |
| R | runs.scored | <int> | runs scored |
| RBI | runs.batted.in | <int> | runs batted in |
| BB | bases.on.balls | <int> | bases on balls |
| IBB | intentional.bases.on.balls | <int> | intentional bases on balls |
| SO | strike.outs | <int> | strike outs |
| HBP | hit.by.pitch | <int> | hit by pitch |
| SF | sacrifice.fly | <int> | sacrifice fly |
| SH | sacrifice.hit | <int> | sacrifice hit |
| GDP | ground.into.a.doubleplay | <int> | ground into a double play |
| SB | stolen.base | <int> | stolen base |
| CS | caught.stealing | <int> | caught stealing |
| wRC | wRuns.created | <int> | weighted runs created based on wOBA |
| wRC. | wRuns.created.plus | <int> | rwRC plus, whereby additional factors are taken into consideration like ball park or era |
| IFH | hit.infield | <int> | infield hits |
| BUH | hit.bunt | <int> | bunt hits |
| batted.ball | hit.batted.ball | <int> | PA - SO - BB - HBP |

*TABLE X1: MLB dataset variables description table*