###### 

###### **Pitch Crazy**

**Practicum**

**Final Report**

**Group 3**

***Nareg Artinian***

***Joshua Bradner***

***Jason Hardman***

***Roland Teruya***

***Yan Xu***

**Problem Statement**

The Baltimore Orioles collect pitch data for every pitch thrown by Orioles and opposing American League East pitchers. The pitch data collected includes velocity, horizontal and vertical movement which is utilized to determine the pitch type. In the current state, Orioles analysts manually pull and analyze each pitcher’s data to determine the number of pitches that each pitcher throws. This process is tedious and time consuming as pitchers can throw thousands of pitches throughout their careers. It is also difficult to distinguish pitches, such as two-seam and four-seam fastballs while performing visual cluster analysis. The organization is seeking a solution to automate this task.

**Executive Summary**

The group researched, applied, and tested several clustering methodologies and approaches prior to agreeing upon the final solution workflow. R code was used to run several clustering methodologies to partition, segment, and recluster the data into more manageable chunks of data to increase cluster accuracy. Techniques, fully disclosed in the remainder of the report, include partitioning pitches by single variable cluster results, adapting a set of rules to start or stop the procedure from unnecessary clustering, and exploration of pitcher handedness.

The final results of the model are promising. Though the model had a pitch count accuracy score of 55%, we expect that model accuracy can be improved with a better review of the pitcher visual clustering data set used for verifying model accuracy. We are also highly confident in the ability to increase the model accuracy with further model parameter tuning, which would require additional time and resources.

**Administration**

The team selected Nareg Artinian to be the Project Manager for Pitch Crazy. The team was initially split into two teams, one team was responsible for researching clustering packages, languages that could be used for the task in R, Python, etc. and trialing the tools discovered during the research. The other team was tasked with looking at tools that could be used to automate the work once an appropriate clustering approach could be identified. The individual teams met as needed with the entire team holding weekly meetings to discuss the project on Monday evenings and as needed beyond the regular Monday meetings.

**Data**

The data received had 6 fields and 211,545 rows. The fields were Pitcher name, Pitcher Id, Pitcher handedness, release velocity, break\_x, and break\_z. Release velocity is the velocity of the pitch captured when the ball was released from the pitcher’s hand. Break\_x is the horizontal coordinate of where the pitch was located once it reached the batter’s box, while break\_z is the vertical coordinate.

There are 98 pitchers with an average of around 1.5k pitches per pitcher at a median speed of 85mph. There was some slight data clean up that needed to be conducted around the pitcher names, but the pitcher\_id’s were used for the bulk of the procedure. There were no null values in any of the fields.

The model was validated using a validation set derived from visually clustering pitches, similar to the technique used by the Baltimore Orioles today. The validation data set was produced by all groups participating in this practicum and therefore the results of which proved to produce discrepancies between groups. The two stakeholders analyzed the draft validation set and produced a professionally updated visually clustered pitcher cluster count validation set.

The validation set consisted of three main fields pitcher\_id, cluster\_count, and confidence (H,M,L). Confidence of “H” signifies high confidence in the cluster count produced, while “M” signifies medium confidence and “L” signifies low confidence. These values are arbitrarily assigned and not based on any metrics, which may provide enough doubt in the final accuracy score to warrant a secondary check of all cluster count outputs from all models. High confidence made up 65% of the validation data set, while the medium and low confidences made up the remaining 22% and 13% respectively.

**Approach**

The approach to the problem required identifying and separating out each of the requirements stated by Sig Mejdal, and use those as the guidelines for developing a successful model. The requirements of the final model were; automation, run with limited human interaction, produce cluster count as an output, and guidelines of how to determine which pitchers’ pitches require a manual inspection.

To satisfy all aforementioned requirements, we broke into subgroups and conducted an assessment of the tools necessary to automate the work while researching various clustering algorithms suitable for these particular requirements. Tools assessment was geared towards automation and limited interaction requirements, while research of clustering algorithms were geared towards generating accurate cluster counts. The tools assessment was a high level comparison between platforms such as Alteryx, RapidMiner, and Tableau vs building custom code in R and Python. The group decided to pursue a code route that was able to maintain an agile atmosphere rather than utilize constricting platforms. The group also decided to run parallel efforts to code. Those comfortable with R would conduct analysis and build the ETL in R, while those comfortable with Python would code in Python. Whichever coding language satisfied all conditions as well as the highest accuracy score, defined by calculating the percent of pitchers with equivalent pitch counts produced by both visual and programmatic clustering methodologies, would be the final model.

While the first group was conducting research on what the vehicle for analysis would be, the second group analyzed different clustering algorithms. Analysis included manual runs comprised of a subset of the pitch data. The group’s approach to solving this problem allowed us to become more familiar with the data as well and unearth hidden insights that allowed the production of more accurate models. The group researched hypotheses such as “Handedness has 0 effect on pitch cluster identification” and “Separating out fastball from non-fastball pitches to increase the accuracy of prediction by reducing the noise a clustering algorithm must process to define cluster boundaries”. The validation of these hypotheses are discussed further below.

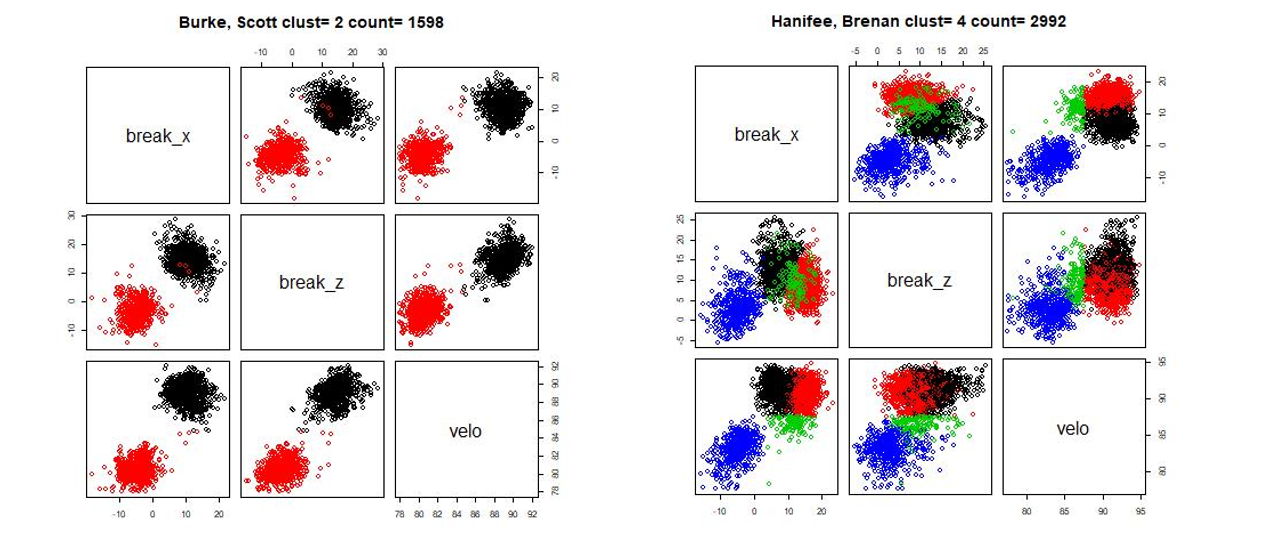
By fostering a competitive yet collaborative approach to solve this unique and challenging problem, we were able to push each other past comfort zones to achieve a worthwhile solution that challenged norms and provoked thought.

**Timeline of Findings**

We began by testing multiple clustering methods that automatically selected clustering counts such as hdbscan and mean shift. The outputs of these models, on their own, produced too high of a cluster count to be accurate. We explored clustering methods where K was an input, such as K-means, Partition Around the Medoid (PAM), Gaussian Mixture Model, and Hierarchical. Although these methods produced accurate cluster membership, it did not satisfy the requirement of automatically identifying cluster counts. We explored clustering ensemble packages such as NbClust, ClusterR, and clValid to identify optimal cluster counts, however, outputs were volatile and were unable to assign accurate cluster membership.

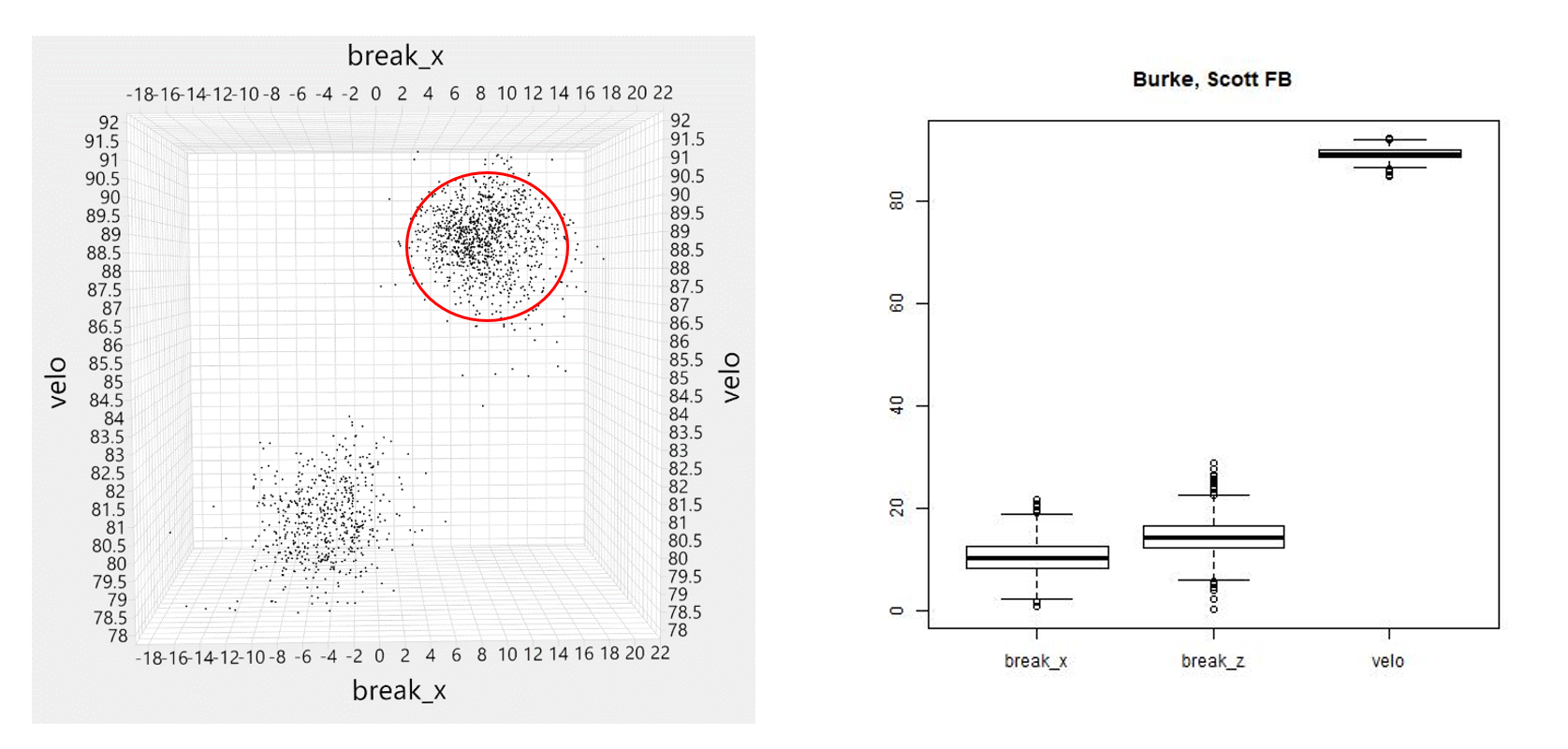
We were surprised to find that none of the clustering algorithms were able to identify clusters that were close or overlapping. We formulated a hypothesis to cluster the data univariately rather than clustering all three variables as a whole. As a result we concluded that the other two variables, regardless of what they were, generate noise that cause the cluster formations to overlap and generate noise that provided difficulties for accurately clustering the pitches. We decided to use these learnings to partition the data by the clusters generated by PAM clustering the velocity variable on its own. We found this heuristic was able to separate fastballs from non fastballs, specifically at the K=2 and K=3 values. To determine which K value to use on each pitcher, we explored internal indices such as Calinski Harabasz, Dunn, Silhouette, and c\_index. We settled on using the lowest c\_index score as the PAM selection criteria, which produced the most consistent results. By splitting the pitches, we were able to reduce the overall noise in the pitcher data, freeing us to use different approaches on the two new subsets of pitches created.

When splitting the pitches, we encountered a new problem of when to continue clustering or stop clustering. For example, Scott Burke only has two pitch types. Splitting his pitches into Fastballs and Non-Fastballs, we are left with single clusters in each category, and therefore should not be eligible for additional clustering. Another example, Brenan Hanifee has two fastballs, and therefore eligible for further clustering of his Fastballs.

****

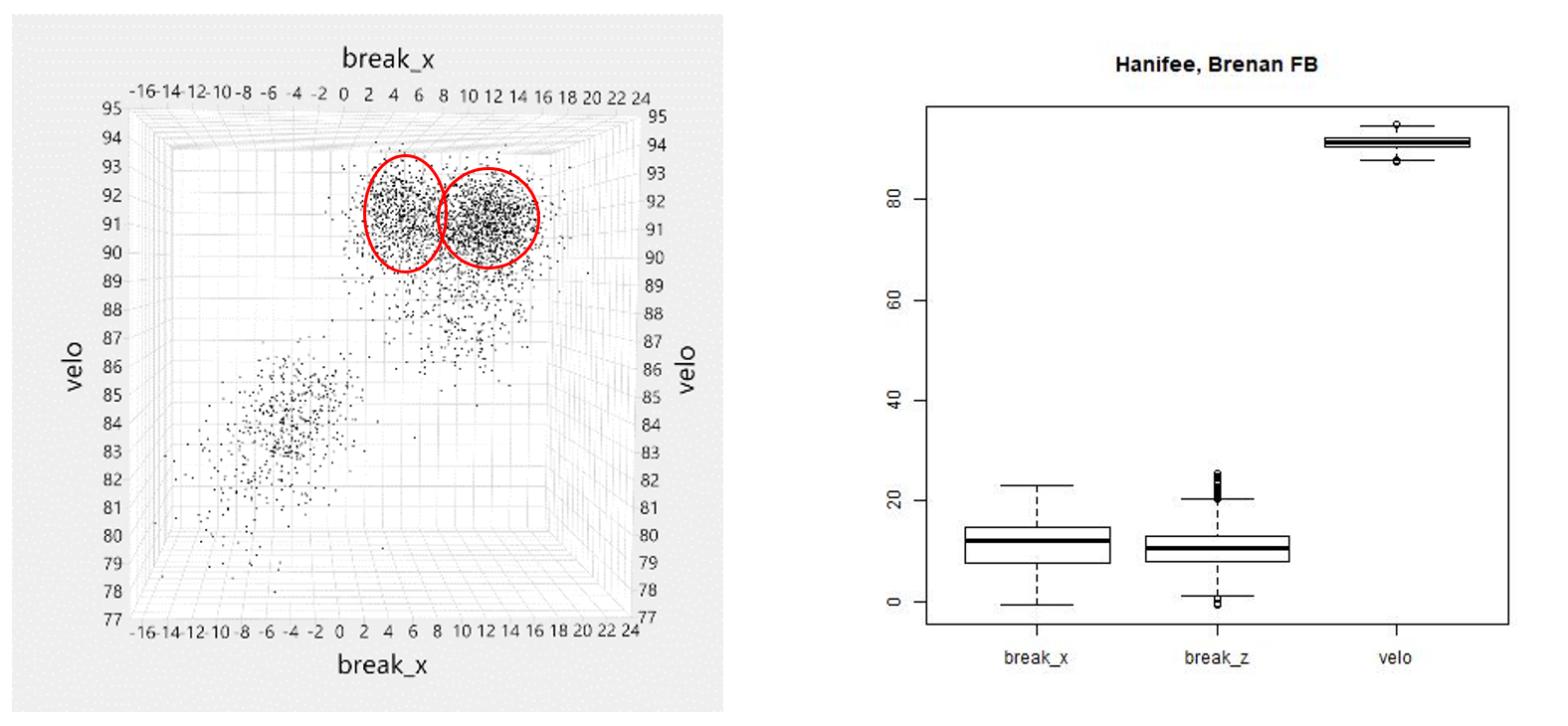
**Figure 1:** Left Plot: Black cluster represents a single fastball pitch. Right Plot: The red and black cluster identifies two different fastball pitches.

To address this phenomenon, we developed logical Stop/Go rules, detailed parameter settings listed in appendix C. The Stop/Go rules are based on the shape of the distribution and its outliers along each axis. Pitchers with a single fastball tend to have more normal break\_x and break\_z distributions where multiple fastballs tend to have multiple local maxima. In figure 2, we explore Scott Burke’s example of a single fastball. Interquartile ranges of break\_x and break\_z are normally distributed, and show minimal outliers in the whisker plot.

****

**Figure 2:** Left Plot: Scatter plot of Scott Burke representing a single fastball. Right Plot: Break\_x and break\_z both appear to be reasonably normally distributed and the distribution quartiles are reasonably similar.

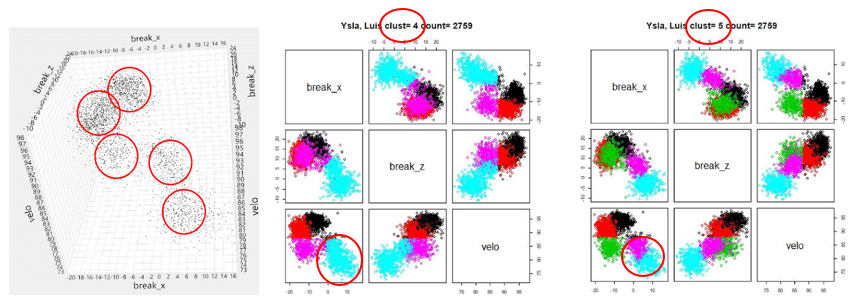
Pitchers with multiple fastballs tended to have non-normal break\_x and break\_z distributions. Figure 3, Brenan Hanifee’s pitches, is an example of pitchers with multiple fastballs. Interquartile ranges of break\_x are wider and slightly skewed, suggesting multiple local maxima or cluster medoids.

****

**Figure 3:** Left Plot: Scatter plot of Brenan Hanifee representing multiple fastballs with no clear break between the fastball clusters. Right Plot: Break\_x and break\_z interquartile range are uneven relative to each other. The distribution of break\_x is much wider than break\_z possibly suggesting multiple fastballs.

We applied tuned Stop/Go rules, found in appendix C, in combination with Hdbscan Min Pts = 10 to identify single Non-Fastball clusters and multi Non-Fastball clusters. We aggregated the cluster counts from Fastball and Non-Fastball partitions at the end of the ETL workflow, which produced an accuracy rate of 54%, After visual inspection of outputs, it was determined that a further split of Non-Fastball pitches along the break\_x axis could further improve the accuracy score beyond what Hdbscan could identify on its own.

The final version of our clustering process implemented a split of Non-Fastballs along the break\_x axis when the Hdbscan Min Pts = 10, producing a cluster count greater than 1 as shown in Stage 2 in the ETL diagram. A PAM k=2 along the break\_x axis splits the Non-Fastballs into a left and right partition. Our hypothesis, that the handedness of the pitcher did not impact model accuracy, allowed us to uniformly apply the methodology across all pitchers. Applying the same process of using Stop/Go rules in tandem with Hdbscan Min Pts = 25 to the Left and Right Non-Fastball sets, further identified clusters not detected in previous model versions. Full parameters listed in appendix C V2. V2 produced the best accuracy rate of 55% based on the same validation table, pitcher cluster counts are listed in appendix G. Figure 4 below is an example of Luis Ysla’s pitches, where additional cluster were identified in our V2 process.



**Figure 4:** Left Plot: Scatter plot of Luis Ysla who appears to have 5 clusters. Center Plot: Version 1 process only identifies 4 clusters. Right Plot: Version 2 process was able to identify that the Right side Non-Fastball contained 2 clusters for a total of 5 clusters.

As mentioned previously, the group had an internal competition to produce parallel workflows that not only satisfy the requirements of the product, as requested by Sig Mejdal, but also have a higher accuracy score. The group quickly found out that we were not able to produce a duplicate ETL flow utilizing the same clustering algorithms in Python and R.

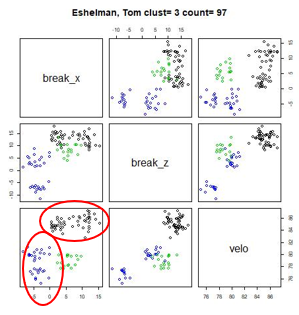
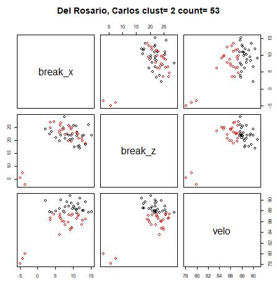
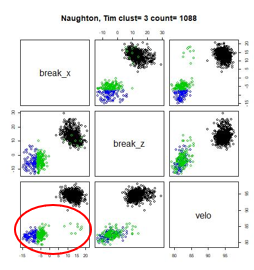
The Python group found that there is no usable DBScan package in Python, a certain bug blocks a user from being able to successfully install and call the package without needing to make edits to the source code. Because no one in the group was proficient enough in Python to quickly subdue this roadblock, the Python group substituted the use of DBScan with the use of the Mean Shift clustering algorithm. The ETL still followed the same procedures of a reiterative loop of splitting fastballs from non-fastballs and clustering both sets of pitches concurrently in isolation. The best results achieved by the Python group utilizing Mean Shift was a 41% accuracy.

The R group had less road bumps while implementing the ETL process. They were therefore able to spend more time optimizing the model outputs by tuning parameters and adding an adversarial logic to bound or force the clustering algorithm to continue or stop the clustering algorithm from attempting to partition the data further. These added capabilities as well as an accuracy score of 55% led us to choose the V2 R scripted model to be named the Pitch Crazy Model and our final product, final R script is provided in appendix G.

**Rule to Flag for Human Review**

The clustering process does have errors and unexpected clustering results at times. The following rules have been developed to identify clustering outputs with low confidence. The skew and individual cluster count information is provided in appendix E. Figure 5 provides examples of pitches flagged for review..

* skew\_x, skew\_z, skew\_velo is greater than or equal to 1.5
* Total pitch count is less than or equal to 100
* Individual cluster count is less than or equal to 50



**Figure 5:** Left plot: the green cluster has a skew\_x of 3.4, it appears that a cluster was missed. Center plot: Total pitch count was below 100, all clustering is unreliable. Right plot: Low individual cluster counts, Hdbscan is not able to consistently identify separate clusters.

A total of 37 pitchers were flagged for review based on these rules. See appendix F for the list of flagged pitcher for review and the pitchers that passed through. Based on the remaining 56 pitchers that passed through, our V2 clustering model had an estimated 84% accuracy rate based on a visual validation.

**Conclusions and Takeaways**

* Separating Fastballs from Non-Fastballs and running a parallel job to cluster subsets of fastball and non fastball pitches had the highest lift in model performance compared to other methods attempted.
* Using an ensemble process of rules and HDbscan utilizing stop/go rules produced better results than using rules or HDbscan alone.
* The validation dataset used visual clustering, which is vulnerable to interpretation and should be further scrutinized.
* Although the approach wasn’t used normalizing the handedness of the pitcher didn’t have a measurable impact model accuracy.

**Appendix A: Clustering Method Pros and Cons**

**K-Means:**

Separates samples into k groups of equal variances while minimizing the inter-clustering distance.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| One of the most popular and wide use clustering methods. It is simple to use and results are very easy to interpret. It is excellent for partitioning obvious clusters, such as separating fastball and non-fastball pitches in our model since we just need to set k = 2. | One obvious drawback for this algorithm is it will require user to define k which is our desired model output. It is not very efficient with abnormal shape or non-ellipsoid clustering. |

**HDBSCAN:**

Extension of Density-Based Spatial Clustering of Applications with Noise. Identifies clusters by separating areas of high and low densities. Central component of HDBSCAN is core sample, which is used to identify outliers.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Unlike K-Means, this algorithm is effective with abnormal shape or non-ellipsoid clustering. It clusters the data based on density and neighbors rather than predefined number of clusters. It will be useful to identify and demarcate outlier pitch in our model. | The algorithm does not work very well with low density clusters and it won’t always identify the centroids. If identifying centroids is an objective, then it is not the best choice. |

**Mean Shift:**

Centroid-based algorithm. It works by updating candidates for centroids to be the mean of points within a given region. The algorithm uses “Basins of Attraction to identify various maxima, in this case cluster centroid. It relies on a parameter called band\_width which indicated the size of the region to search through.

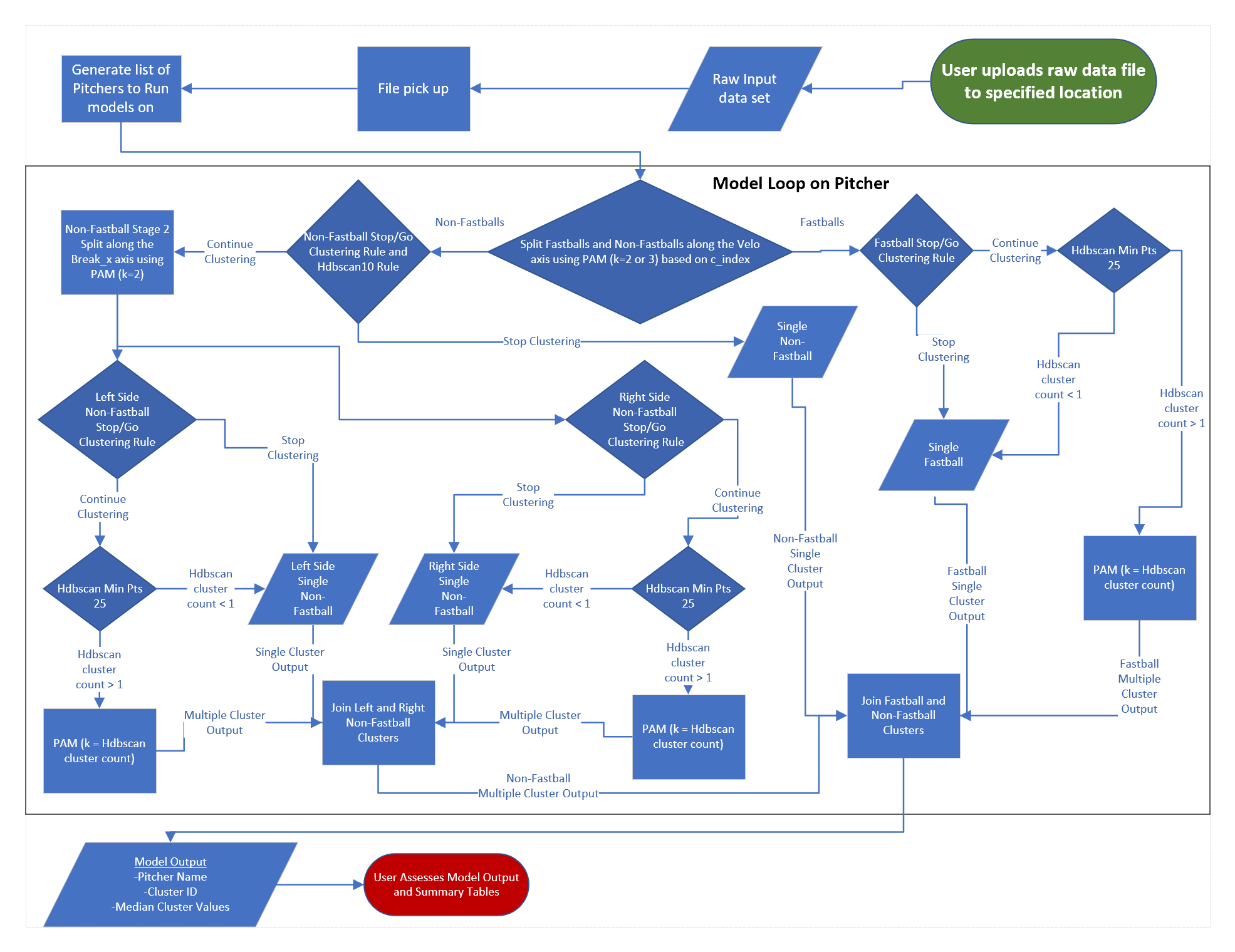
|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The algorithm will automatically set the number of clusters. It is very effective with abnormal shape or non-ellipsoid clustering. | As the process is highly iterative, it will slow down the total process time significantly. It cannot handle outlier very well and will misclassify them into incorrect clusters. |

**Spectral Clustering:**

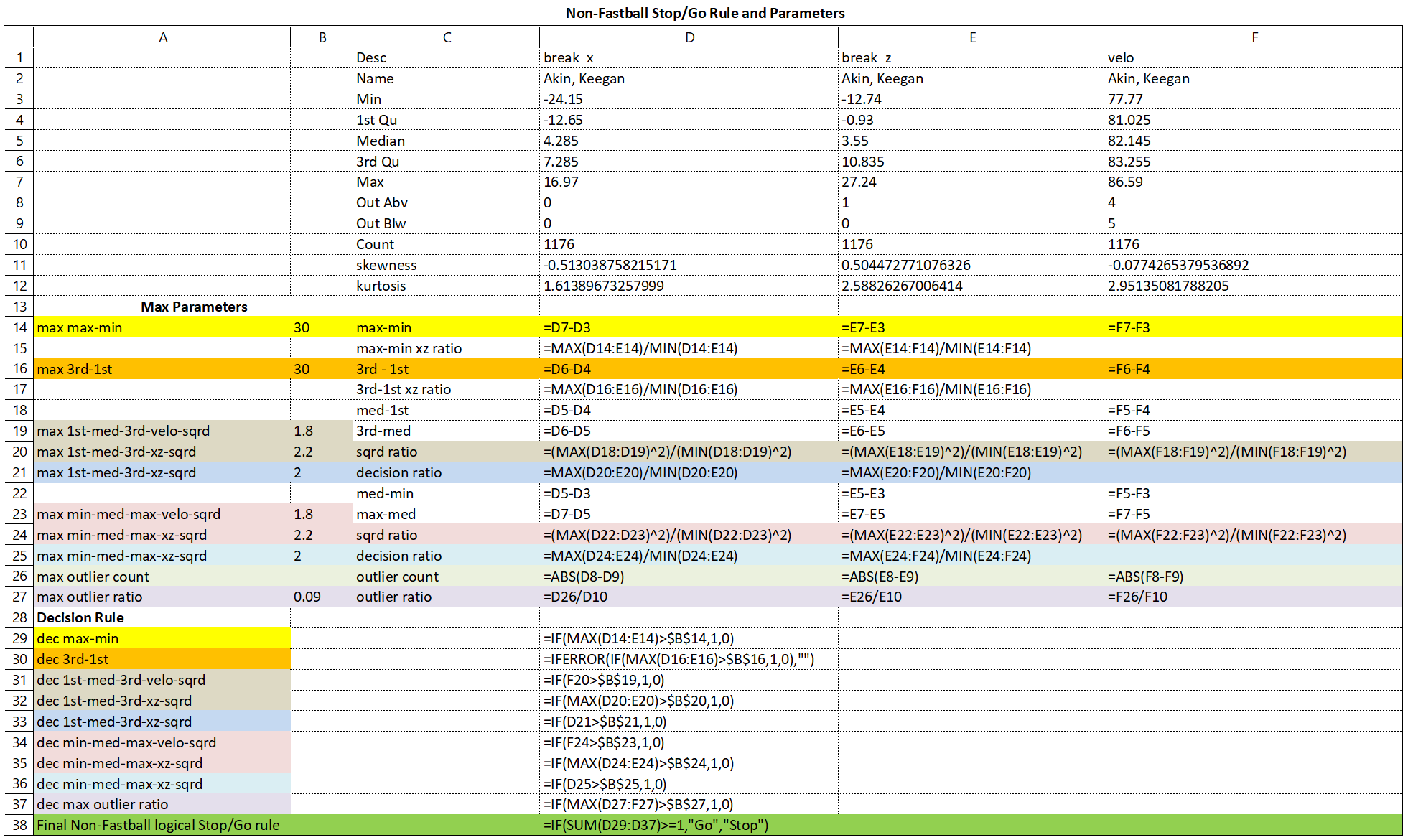
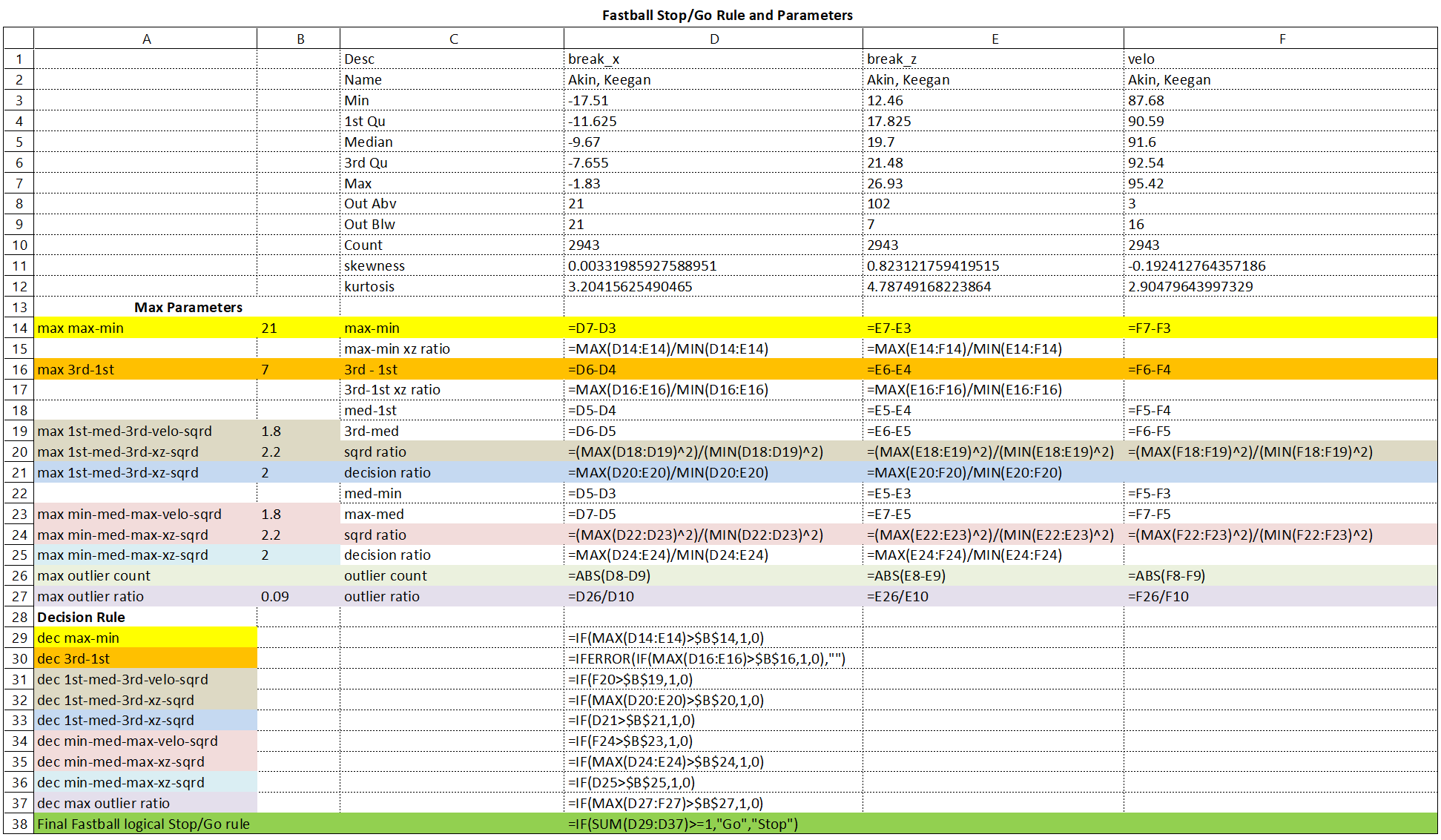
Three-stage hybrid algorithm. First, it simplifies dataset into low-dimensional matrices. Then, it will find the connectivity between data points. Last, it will score connection strength to cluster points together.

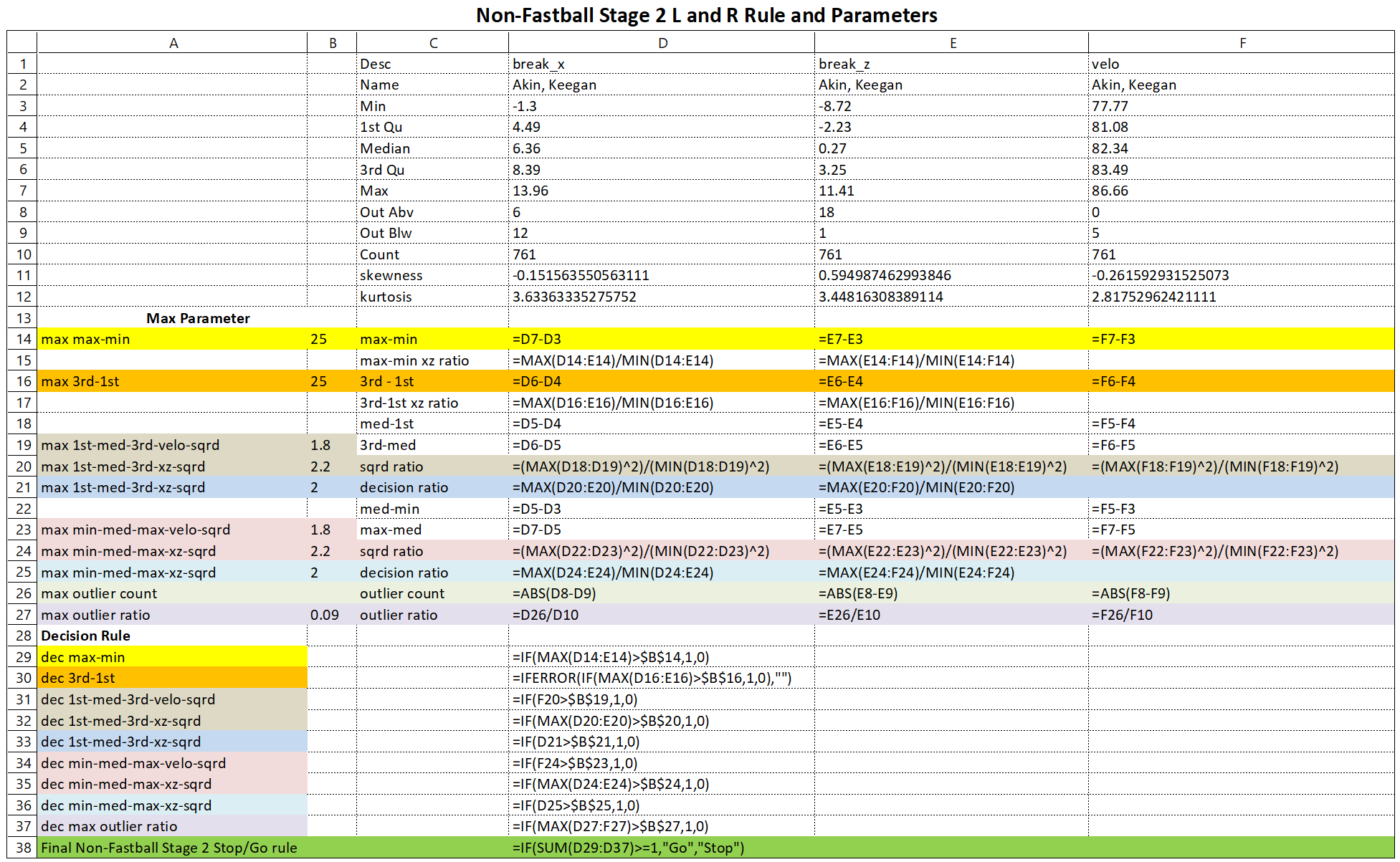
|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Works well with a small number of clusters. | Similar to K Means, the input parameter is our desired output which is not suitable for our application. It also lacks the ability to handle abnormal shape clusters. Since it is a multi-stage algorithm, it will also slow down the total process time. |

**Appendix B: Final ETL Flowchart**

****

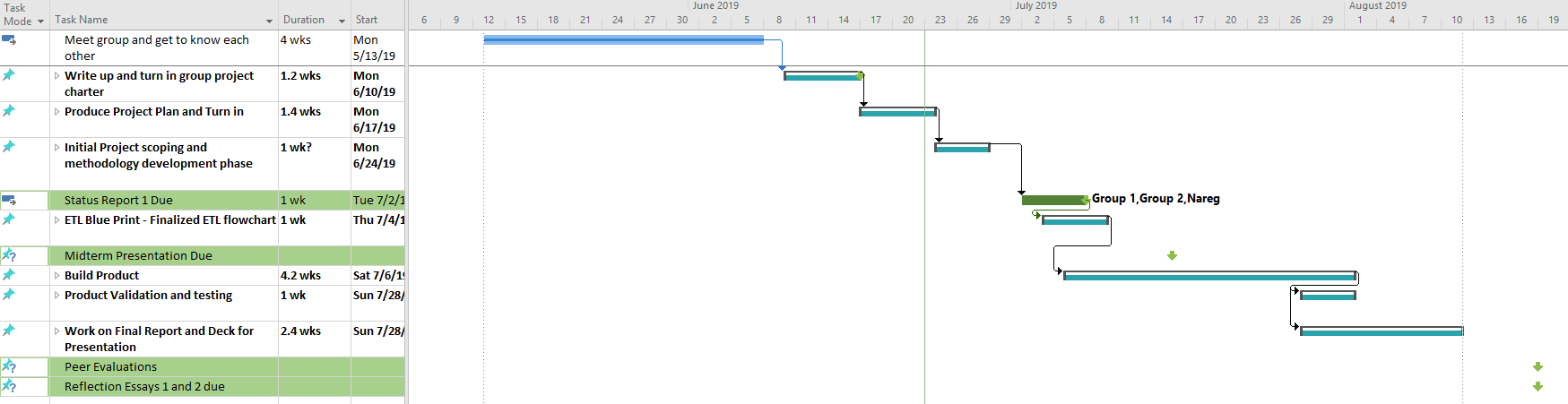
**Appendix C: Model Tuning Parameters and Logical Rule Parameters**



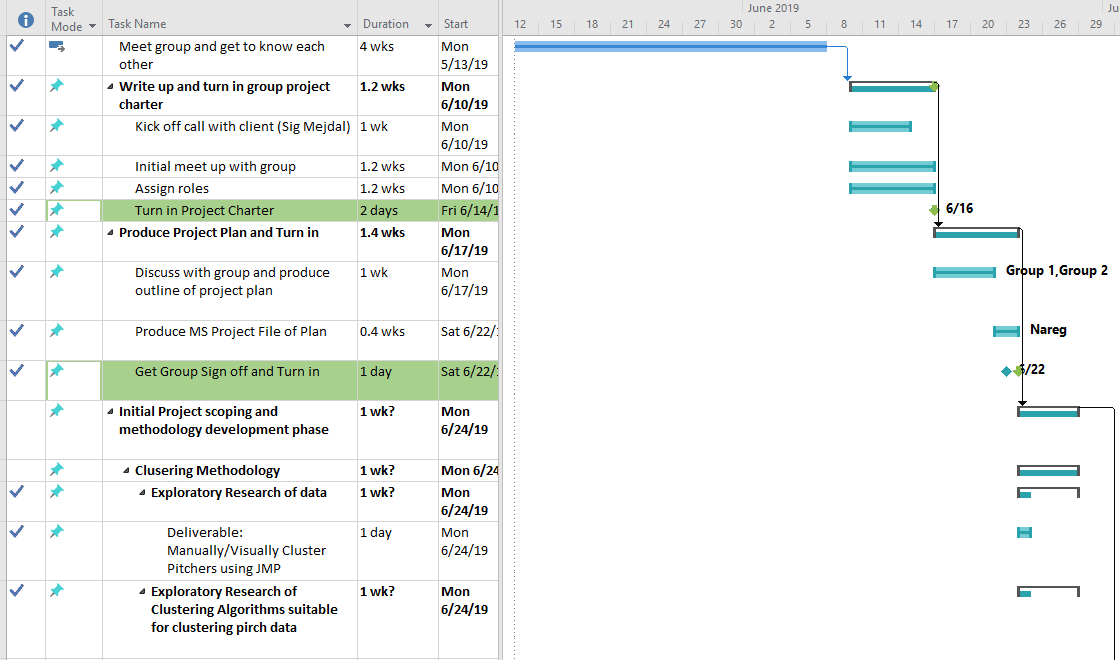


**Appendix D: Project Plan**

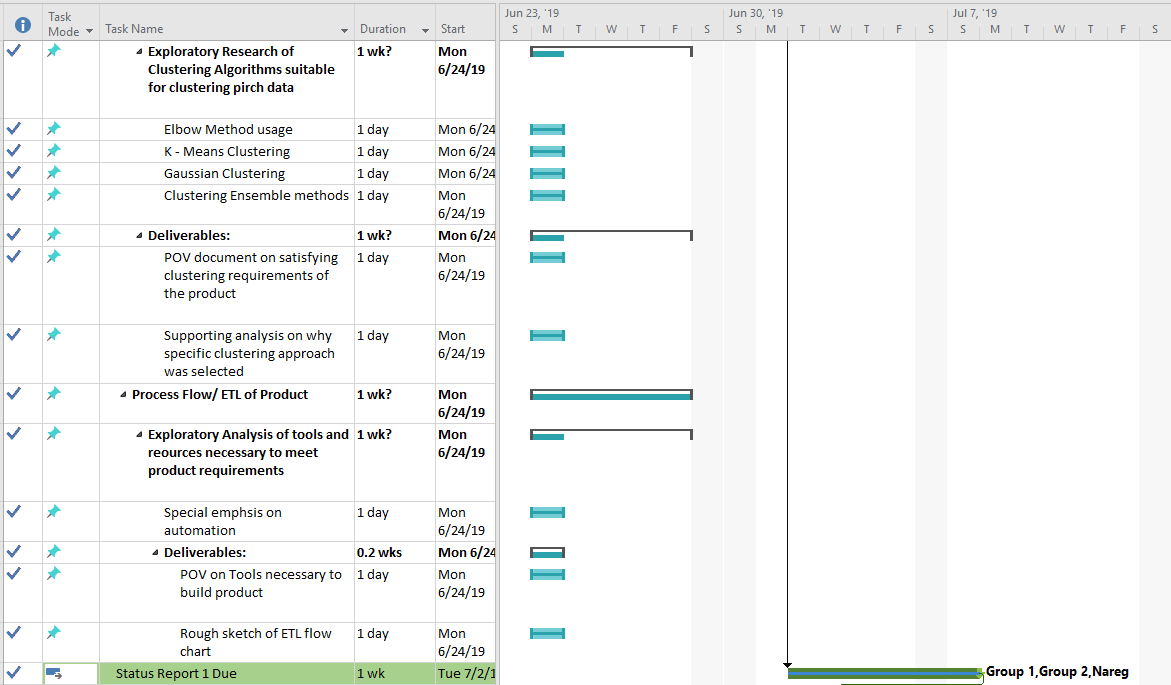
**Collapsed Project Plan**

****

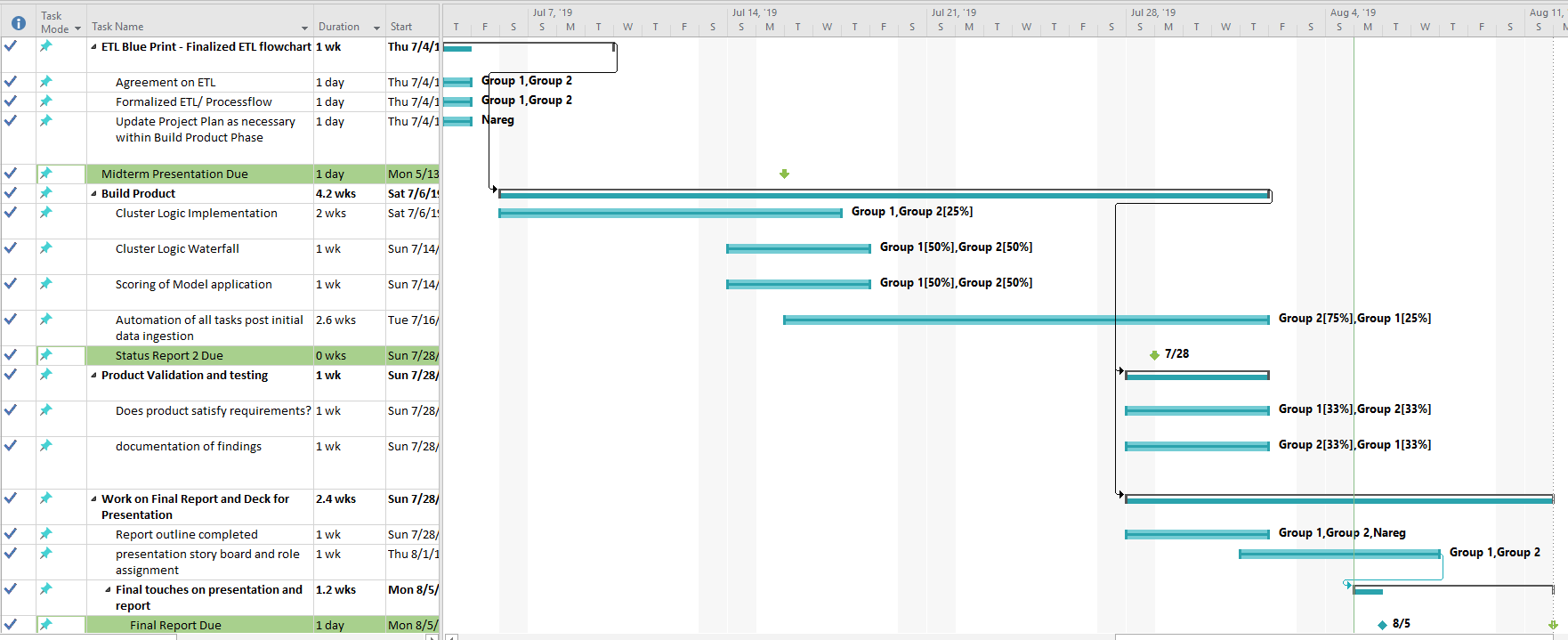
**Expanded Project Plan 1:**

****

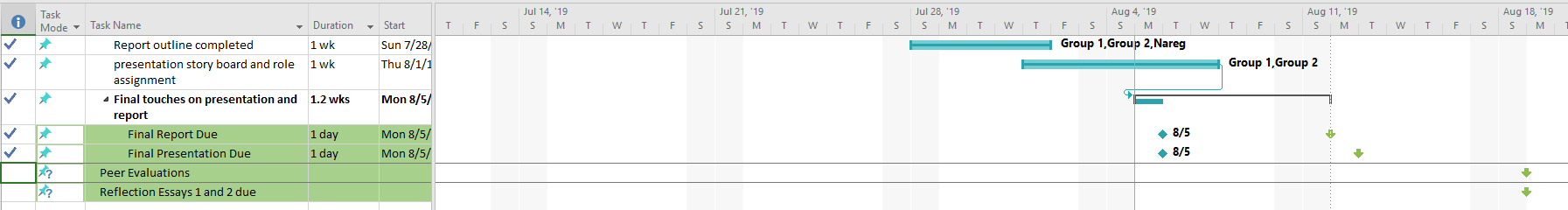
**Expanded Project Plan 2:**

****

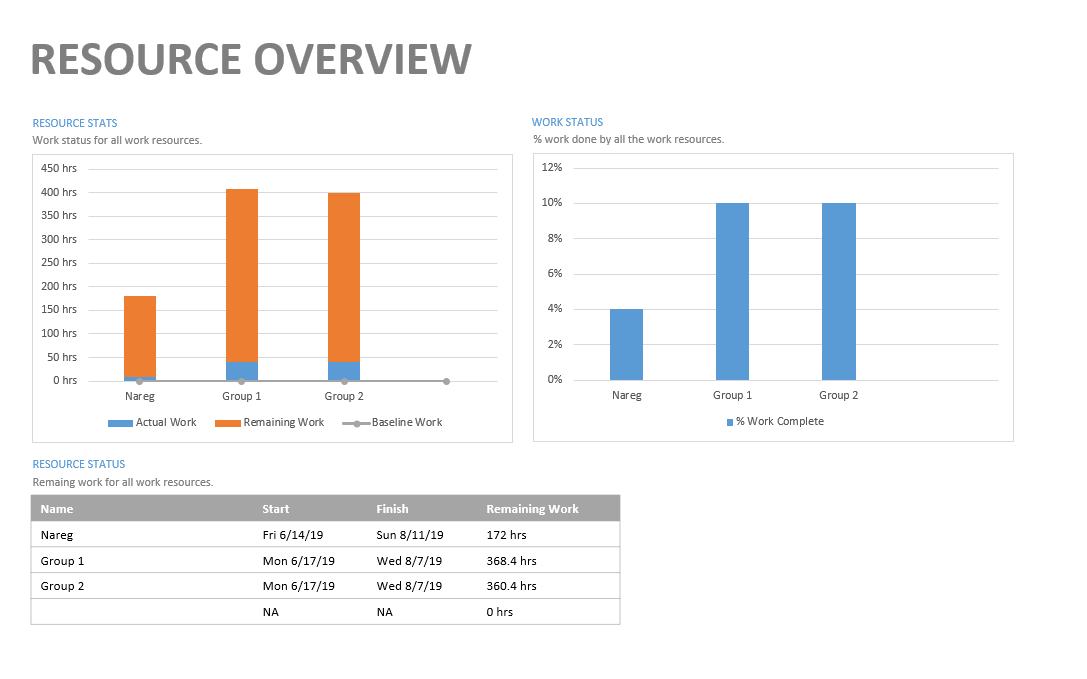
**Expanded Project Plan 3:**

****

**Expanded Project Plan 4:**

****

**Project Plan Resources Overview:**



**Appendix E: Summary Table 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Pitcher** | **Cluster** | **Median\_x** | **Median\_z** | **Median\_velo** | **Cluster\_count** | **Skew\_x** | **Skew\_z** | **Skew\_velo** |
| **Akin, Keegan** | **9** | **-9.67** | **19.7** | **91.6** | **2943** | **0.003** | **0.823** | **0.192** |
| **Akin, Keegan** | **27** | **6.36** | **0.27** | **82.34** | **761** | **0.152** | **0.595** | **0.262** |
| **Akin, Keegan** | **36** | **-14.03** | **12.33** | **81.84** | **415** | **0.421** | **0.699** | **0.268** |
| **Almengo, Diogenes** | **9** | **7.93** | **15.115** | **94.34** | **486** | **0.544** | **0.796** | **0.25** |
| **Almengo, Diogenes** | **27** | **16.155** | **7.9** | **85.725** | **156** | **0.45** | **0.336** | **0.39** |
| **Almengo, Diogenes** | **36** | **-4.185** | **1.345** | **82.64** | **158** | **0.171** | **0.392** | **0.512** |
| **Alvarado, Cristian** | **9** | **10.89** | **14.31** | **89.84** | **3277** | **0.453** | **0.623** | **0.496** |
| **Alvarado, Cristian** | **27** | **14.28** | **9.77** | **83.51** | **580** | **0.139** | **0.395** | **0.531** |
| **Alvarado, Cristian** | **36** | **-11.76** | **-6.89** | **75.72** | **427** | **0.628** | **0.043** | **0.527** |
| **Alvarado, Cristian** | **37** | **-3.36** | **3.86** | **80.71** | **903** | **0.129** | **0.883** | **0.369** |
| **Angomas, Cesar** | **9** | **-4.17** | **15.06** | **87.28** | **241** | **0.271** | **0.207** | **0.274** |
| **Angomas, Cesar** | **18** | **7.05** | **6.2** | **76.2** | **77** | **0.192** | **0.257** | **0.336** |
| **Araujo, Pedro** | **1** | **3.535** | **17.93** | **91.765** | **494** | **0.302** | **0.129** | **0.084** |
| **Araujo, Pedro** | **2** | **13** | **12.34** | **91.21** | **99** | **0.296** | **0.654** | **0.163** |
| **Araujo, Pedro** | **27** | **12.88** | **7.445** | **86.46** | **402** | **0.591** | **0.836** | **0.036** |
| **Araujo, Pedro** | **36** | **-10.43** | **3.7** | **81.24** | **249** | **0.27** | **0.525** | **0.636** |
| **Armstrong, Shawn** | **9** | **4.9** | **17.07** | **92.64** | **135** | **1.185** | **0.829** | **0.224** |
| **Armstrong, Shawn** | **27** | **-6.98** | **8.18** | **87.27** | **51** | **0.719** | **0.68** | **1.131** |
| **Armstrong, Shawn** | **36** | **-14.82** | **-3.565** | **82.33** | **48** | **0.161** | **1.923** | **0.771** |
| **Baumann, Michael** | **9** | **6.75** | **20.09** | **93.03** | **1719** | **0.067** | **0.342** | **0.575** |
| **Baumann, Michael** | **24** | **6.985** | **20.135** | **90.44** | **502** | **0.567** | **0.936** | **2.196** |
| **Baumann, Michael** | **25** | **9.255** | **14.45** | **87.21** | **250** | **0.01** | **1.644** | **0.179** |
| **Baumann, Michael** | **36** | **-2.68** | **-2.38** | **79.96** | **291** | **0.926** | **0.085** | **0.244** |
| **Baumann, Michael** | **37** | **-2.055** | **8.21** | **88.05** | **436** | **0.254** | **1.312** | **1.048** |
| **Benitez, Joel** | **9** | **10.44** | **14.25** | **90.39** | **99** | **0.183** | **0.363** | **0.853** |
| **Benitez, Joel** | **18** | **-1.54** | **0.36** | **76.64** | **35** | **1.605** | **2.53** | **0.494** |
| **Bishop, Cameron** | **9** | **-8.98** | **14.665** | **89.7** | **1694** | **0.208** | **0.127** | **0.002** |
| **Bishop, Cameron** | **27** | **3.975** | **0.73** | **80.92** | **638** | **0.147** | **0.581** | **0.317** |
| **Bishop, Cameron** | **36** | **-12.4** | **9.51** | **83.52** | **339** | **0.218** | **0.058** | **0.34** |
| **Bleier, Richard** | **1** | **-13.26** | **4.09** | **88.36** | **1187** | **0.477** | **0.535** | **0.019** |
| **Bleier, Richard** | **2** | **0.285** | **6.705** | **86.42** | **532** | **0.404** | **0.712** | **0.654** |
| **Bleier, Richard** | **24** | **1.15** | **6.31** | **83.37** | **51** | **1.062** | **0.121** | **2.613** |
| **Bleier, Richard** | **25** | **3.575** | **-1.815** | **78.34** | **130** | **0.068** | **0.459** | **0.032** |
| **Bleier, Richard** | **36** | **-15.95** | **5** | **81.03** | **143** | **0.003** | **0.907** | **0.083** |
| **Bundy, Dylan** | **9** | **7.31** | **19.74** | **92.15** | **5793** | **0.326** | **0.308** | **0.31** |
| **Bundy, Dylan** | **27** | **12.795** | **11.46** | **84.15** | **1504** | **0.081** | **0.079** | **0.1** |
| **Bundy, Dylan** | **36** | **-8.47** | **-11.13** | **75.17** | **1097** | **0.765** | **0.134** | **0.183** |
| **Bundy, Dylan** | **37** | **-4.05** | **-0.13** | **81.82** | **1843** | **0.736** | **1.425** | **0.013** |
| **Burke, Scott** | **9** | **10.32** | **14.2** | **89.06** | **1006** | **0.231** | **0.233** | **0.329** |
| **Burke, Scott** | **18** | **-4.3** | **-3.695** | **80.455** | **592** | **0.374** | **0.427** | **0.35** |
| **Carroll, Cody** | **9** | **5.8** | **17.9** | **95.96** | **353** | **0.143** | **0.026** | **0.576** |
| **Carroll, Cody** | **27** | **6.17** | **5.72** | **86.87** | **35** | **0.105** | **0.201** | **0.084** |
| **Carroll, Cody** | **36** | **-5.39** | **-4.01** | **83.43** | **130** | **0.325** | **0.934** | **0.144** |
| **Cashner, Andrew** | **1** | **8.17** | **17.36** | **93.33** | **1267** | **1.55** | **1.569** | **0.644** |
| **Cashner, Andrew** | **2** | **14.43** | **14.3** | **91.97** | **1071** | **0.122** | **0.106** | **0.125** |
| **Cashner, Andrew** | **27** | **13.08** | **8.52** | **83.96** | **683** | **0.393** | **0.292** | **0.275** |
| **Cashner, Andrew** | **36** | **-8.13** | **-8.38** | **80.85** | **516** | **0.113** | **0.384** | **0.143** |
| **Cashner, Andrew** | **37** | **-3.35** | **4.32** | **85.08** | **473** | **0.271** | **0.221** | **0.53** |
| **Castro, Miguel** | **9** | **17.24** | **9.05** | **95.72** | **2233** | **0.514** | **0.252** | **0.393** |
| **Castro, Miguel** | **27** | **17.75** | **6.725** | **87.49** | **470** | **0.365** | **0.391** | **0.314** |
| **Castro, Miguel** | **36** | **-5.26** | **0.65** | **84.66** | **976** | **0.133** | **0.038** | **0.127** |
| **Chavez, Jesus** | **9** | **-17.425** | **13.685** | **81.005** | **58** | **0.038** | **0.306** | **0.431** |
| **Chavez, Jesus** | **18** | **5.93** | **1.93** | **65.32** | **7** | **0.369** | **0.488** | **0.15** |
| **Chleborad, Tanner** | **9** | **14.53** | **13.18** | **92.19** | **1601** | **0.731** | **0.412** | **0.31** |
| **Chleborad, Tanner** | **27** | **15.465** | **7.88** | **84.315** | **150** | **0.222** | **1.411** | **0.637** |
| **Chleborad, Tanner** | **36** | **0** | **-1.4** | **81.83** | **562** | **0.938** | **0.695** | **0.145** |
| **Cobb, Alex** | **9** | **13.905** | **15.24** | **91.94** | **1402** | **0.076** | **0.377** | **0.03** |
| **Cobb, Alex** | **27** | **15.24** | **8.14** | **86.46** | **734** | **0.373** | **0.123** | **0.048** |
| **Cobb, Alex** | **36** | **-3.76** | **-15.65** | **81.64** | **591** | **0.224** | **1.818** | **0.064** |
| **Conroy, Ryan** | **9** | **8.47** | **11.17** | **90.23** | **181** | **0.395** | **0.012** | **0.121** |
| **Conroy, Ryan** | **27** | **8.425** | **2.04** | **82.895** | **60** | **0.472** | **0.683** | **0.069** |
| **Conroy, Ryan** | **36** | **-6.99** | **1.98** | **83.43** | **49** | **1.333** | **0.268** | **0.048** |
| **De La Rosa, Matt** | **9** | **8.67** | **17.86** | **91.59** | **324** | **0.161** | **0.144** | **0.033** |
| **De La Rosa, Matt** | **27** | **12.15** | **7.945** | **80.9** | **74** | **0.016** | **1.002** | **0.033** |
| **De La Rosa, Matt** | **36** | **-5.92** | **2.31** | **81.53** | **139** | **1.098** | **1.033** | **0.793** |
| **Del Rosario, Carlos** | **9** | **9.82** | **21.86** | **88.44** | **27** | **0.386** | **0.319** | **0.747** |
| **Del Rosario, Carlos** | **18** | **8.225** | **22.885** | **86.28** | **26** | **1.214** | **1.859** | **1.839** |
| **Dietz, Matthias** | **9** | **12.9** | **14.725** | **93.67** | **3368** | **0.355** | **0.111** | **0.127** |
| **Dietz, Matthias** | **27** | **13.535** | **8.615** | **87.125** | **350** | **0.072** | **0.449** | **0.63** |
| **Dietz, Matthias** | **36** | **-0.35** | **4.47** | **85.04** | **631** | **0.205** | **1.195** | **0.68** |
| **Erwin, Tyler** | **9** | **-1.01** | **10.37** | **87.57** | **1243** | **0.529** | **0.506** | **0.045** |
| **Erwin, Tyler** | **27** | **6.89** | **-0.505** | **80.83** | **512** | **0.713** | **0.677** | **0.168** |
| **Erwin, Tyler** | **36** | **-9.895** | **5.18** | **83.29** | **76** | **0.033** | **0.308** | **1.349** |
| **Erwin, Tyler** | **37** | **2.145** | **1.97** | **82.12** | **272** | **0.95** | **0.918** | **0.574** |
| **Eshelman, Tom** | **9** | **6.565** | **13.22** | **85.07** | **50** | **0.104** | **0.185** | **0.015** |
| **Eshelman, Tom** | **27** | **5.54** | **8.36** | **78.61** | **17** | **0.284** | **0.046** | **0.038** |
| **Eshelman, Tom** | **36** | **-4.27** | **1.72** | **79.285** | **30** | **0.14** | **0.141** | **0.314** |
| **Falconett, Pablo** | **9** | **6.84** | **20.265** | **87.745** | **64** | **0.115** | **0.03** | **0.136** |
| **Falconett, Pablo** | **18** | **-7.18** | **5.58** | **76.73** | **27** | **0.195** | **0.041** | **0.02** |
| **Fenter, Gray** | **9** | **6.2** | **18.98** | **92.26** | **1239** | **0.148** | **0.107** | **0.192** |
| **Fenter, Gray** | **27** | **10.3** | **14.075** | **85.18** | **124** | **0.162** | **0.054** | **0.284** |
| **Fenter, Gray** | **36** | **-12.48** | **-8.13** | **78.95** | **475** | **0.451** | **1.113** | **1.147** |
| **Flaa, Jay** | **9** | **6.76** | **20.85** | **91.065** | **1912** | **0.192** | **0.716** | **0.32** |
| **Flaa, Jay** | **27** | **1.35** | **-2.65** | **81.33** | **351** | **0.978** | **0.337** | **0.277** |
| **Flaa, Jay** | **36** | **-6.375** | **-8.065** | **78.075** | **388** | **1.39** | **0.983** | **0.762** |
| **Fry, Paul** | **9** | **-3.595** | **12.72** | **90.54** | **1800** | **0.287** | **0.603** | **0.183** |
| **Fry, Paul** | **27** | **15.88** | **-4.02** | **83.26** | **727** | **0.723** | **0.442** | **1.083** |
| **Fry, Paul** | **36** | **10.625** | **-1.715** | **83.355** | **388** | **0.791** | **0.291** | **0.575** |
| **Fry, Paul** | **37** | **-5** | **12.44** | **86.065** | **54** | **0.005** | **0.46** | **1.769** |
| **Gilmartin, Sean** | **9** | **-9.99** | **15.85** | **88.165** | **470** | **0.029** | **0.078** | **0.048** |
| **Gilmartin, Sean** | **24** | **13.43** | **-7.77** | **71.64** | **101** | **0.284** | **0.259** | **1.281** |
| **Gilmartin, Sean** | **25** | **14.14** | **-0.41** | **78.78** | **305** | **0.485** | **0.36** | **0.74** |
| **Gilmartin, Sean** | **36** | **-7.96** | **13.49** | **78.07** | **331** | **0.112** | **0.613** | **0.34** |
| **Givens, Mychal** | **9** | **8.65** | **12.42** | **95.02** | **4241** | **0.707** | **0.797** | **0.235** |
| **Givens, Mychal** | **27** | **12.03** | **-2.36** | **85.535** | **576** | **0.216** | **0.173** | **0.243** |
| **Givens, Mychal** | **36** | **-5.96** | **3.55** | **86.05** | **1359** | **0.89** | **0.071** | **0.198** |
| **Gonzalez, Brian** | **9** | **-9.55** | **16.02** | **89.68** | **3821** | **0.012** | **0.169** | **0.285** |
| **Gonzalez, Brian** | **24** | **11.2** | **-5.69** | **75.37** | **375** | **0.013** | **0.195** | **0.232** |
| **Gonzalez, Brian** | **25** | **4.76** | **1.9** | **79.9** | **394** | **0.566** | **0.418** | **0.599** |
| **Gonzalez, Brian** | **36** | **-7.15** | **12.31** | **79.88** | **955** | **0.443** | **0.466** | **0.455** |
| **Gonzalez, Luis** | **9** | **-7.47** | **19.73** | **92.78** | **2181** | **0.61** | **0.386** | **0.383** |
| **Gonzalez, Luis** | **27** | **-1.225** | **3.97** | **84.75** | **720** | **0.497** | **0.952** | **1.209** |
| **Gonzalez, Luis** | **36** | **-10.81** | **11.35** | **84.96** | **399** | **1.025** | **0.077** | **0.677** |
| **Grover, Taylor** | **9** | **5.96** | **15.7** | **93.61** | **143** | **0.331** | **0.4** | **0.208** |
| **Grover, Taylor** | **18** | **0.44** | **0.9** | **85.27** | **77** | **0.898** | **1.331** | **0.213** |
| **Gruener, Nick** | **9** | **15.66** | **10.4** | **88.8** | **491** | **0.467** | **0.144** | **0.385** |
| **Gruener, Nick** | **27** | **15.56** | **6.16** | **81.19** | **105** | **0.251** | **0.281** | **0.552** |
| **Gruener, Nick** | **36** | **-0.5** | **-2.43** | **78.72** | **314** | **0.767** | **0.087** | **0.761** |
| **Guance, Hector** | **9** | **13.07** | **10.8** | **92.495** | **1044** | **0.522** | **0.067** | **0.324** |
| **Guance, Hector** | **27** | **11.805** | **10.625** | **84.29** | **176** | **0.126** | **0.394** | **0.676** |
| **Guance, Hector** | **36** | **-2.4** | **2.075** | **81.565** | **208** | **0.157** | **0.938** | **0.187** |
| **Hall, DL** | **9** | **-7.87** | **15.41** | **93.66** | **1505** | **0.251** | **0.579** | **0.202** |
| **Hall, DL** | **27** | **13.09** | **-4.67** | **78.4** | **357** | **0.115** | **0.251** | **0.099** |
| **Hall, DL** | **36** | **-12.805** | **12.33** | **81.925** | **228** | **0.247** | **0.145** | **0.447** |
| **Hammonds, Matthew** | **9** | **-9.67** | **19.09** | **85.62** | **657** | **1.175** | **0.326** | **0.284** |
| **Hammonds, Matthew** | **27** | **6.445** | **-8.53** | **72.98** | **162** | **0.041** | **1.15** | **0.492** |
| **Hammonds, Matthew** | **36** | **-8.435** | **13.4** | **75.04** | **498** | **0.498** | **0.782** | **0.125** |
| **Hanifee, Brenan** | **1** | **7.2** | **12.87** | **91.6** | **1023** | **0.048** | **0.627** | **0.484** |
| **Hanifee, Brenan** | **2** | **14.58** | **8.73** | **91.21** | **1311** | **0.22** | **0.108** | **0.408** |
| **Hanifee, Brenan** | **27** | **11.82** | **9.38** | **86.57** | **129** | **0.315** | **0.755** | **3.394** |
| **Hanifee, Brenan** | **36** | **-4.7** | **2.61** | **83.28** | **529** | **0.366** | **0.804** | **0.646** |
| **Harvey, Hunter** | **9** | **8.795** | **18.53** | **94.82** | **1276** | **0.028** | **0.184** | **0.268** |
| **Harvey, Hunter** | **27** | **9.5** | **7.14** | **86.28** | **177** | **0.497** | **0.257** | **0.304** |
| **Harvey, Hunter** | **36** | **-8.53** | **-10.72** | **79.74** | **391** | **0.132** | **0.046** | **0.085** |
| **Herb, Tyler** | **1** | **12.26** | **7.495** | **89.225** | **334** | **0.588** | **0.242** | **0.206** |
| **Herb, Tyler** | **2** | **6.01** | **11.89** | **90.79** | **325** | **0.499** | **0.452** | **0.093** |
| **Herb, Tyler** | **27** | **-3.85** | **2.45** | **82.08** | **173** | **1.175** | **0.339** | **0.313** |
| **Herb, Tyler** | **36** | **-12.68** | **-8.02** | **77.52** | **137** | **0.602** | **1.095** | **0.202** |
| **Hess, David** | **9** | **4.99** | **18.08** | **92.35** | **6027** | **0.306** | **0.634** | **0.843** |
| **Hess, David** | **24** | **6.26** | **9** | **84.13** | **299** | **0.364** | **0.236** | **0.765** |
| **Hess, David** | **25** | **12.41** | **10.08** | **84.77** | **367** | **0.633** | **0.03** | **0.653** |
| **Hess, David** | **36** | **-12.045** | **-9.225** | **72.345** | **494** | **1.327** | **0.104** | **0.587** |
| **Hess, David** | **37** | **-6.87** | **1.845** | **80.125** | **1822** | **0.382** | **0.234** | **0.264** |
| **Jimenez, Francisco** | **1** | **7.64** | **15.86** | **91.2** | **1293** | **0.556** | **0.775** | **0.008** |
| **Jimenez, Francisco** | **2** | **13.53** | **12.43** | **91.23** | **1279** | **0.412** | **0.288** | **0.148** |
| **Jimenez, Francisco** | **27** | **8.99** | **6.42** | **86.08** | **641** | **0.608** | **0.499** | **0.454** |
| **Jimenez, Francisco** | **36** | **-3.75** | **1.04** | **82.98** | **769** | **0.118** | **0.824** | **0.01** |
| **Joyner, Tyler** | **9** | **5.255** | **17.455** | **89.95** | **470** | **0.479** | **0.011** | **0.095** |
| **Joyner, Tyler** | **27** | **6.87** | **13.07** | **81.325** | **74** | **0.075** | **1.529** | **0.69** |
| **Joyner, Tyler** | **36** | **-8.275** | **-8.625** | **78.285** | **352** | **0.084** | **2.163** | **0.207** |
| **Karns, Nate** | **9** | **6.22** | **17.44** | **91.145** | **156** | **0.745** | **0.936** | **0.359** |
| **Karns, Nate** | **27** | **14.41** | **7.32** | **85.06** | **37** | **0.464** | **0.233** | **0.571** |
| **Karns, Nate** | **36** | **-4.57** | **-9.88** | **81.13** | **55** | **0.165** | **1.684** | **0.094** |
| **Klimek, Steven** | **9** | **11.58** | **13.74** | **90.97** | **1909** | **0.302** | **0.331** | **0.393** |
| **Klimek, Steven** | **27** | **12.03** | **13.98** | **80.24** | **153** | **0.607** | **0.076** | **0.362** |
| **Klimek, Steven** | **36** | **-9.23** | **-14.89** | **76.9** | **747** | **0.363** | **1.116** | **0.464** |
| **Kline, Branden** | **9** | **7.61** | **18.31** | **95** | **1595** | **0.31** | **0.118** | **0.343** |
| **Kline, Branden** | **27** | **11.47** | **15.48** | **85.55** | **315** | **0.098** | **0.601** | **0.01** |
| **Kline, Branden** | **36** | **-2.58** | **5.13** | **85.525** | **556** | **0.49** | **0.481** | **0.598** |
| **Knight, Blaine** | **9** | **12.32** | **15.205** | **91.93** | **484** | **0.425** | **0.071** | **0.437** |
| **Knight, Blaine** | **27** | **14.86** | **9.98** | **81.48** | **105** | **0.628** | **0.016** | **0.043** |
| **Knight, Blaine** | **36** | **-4.73** | **1.97** | **82.87** | **167** | **1.251** | **0.1** | **0.216** |
| **Knight, Blaine** | **37** | **-15.225** | **-10.035** | **74.495** | **140** | **0.423** | **0.286** | **0.159** |
| **Kremer, Dean** | **9** | **8.465** | **18.865** | **91.71** | **800** | **0.02** | **0.063** | **0.109** |
| **Kremer, Dean** | **27** | **15.105** | **12.01** | **84.89** | **72** | **0.631** | **0.042** | **0.033** |
| **Kremer, Dean** | **36** | **-8.455** | **-15.4** | **74.685** | **278** | **0.105** | **0.11** | **0.162** |
| **Kremer, Dean** | **37** | **-2.95** | **6.64** | **84.825** | **106** | **0.217** | **0.485** | **0.333** |
| **Lebron, David** | **9** | **1.745** | **21.225** | **91.7** | **400** | **0.054** | **0.592** | **0.218** |
| **Lebron, David** | **27** | **8.77** | **13.485** | **82.09** | **66** | **0.217** | **1.952** | **0.434** |
| **Lebron, David** | **36** | **-4.57** | **4.02** | **83.67** | **187** | **0.146** | **0.14** | **0.822** |
| **Lee, Chris** | **9** | **-13.31** | **12.43** | **93.35** | **3329** | **0.57** | **0.029** | **0.067** |
| **Lee, Chris** | **24** | **4** | **0.375** | **84.545** | **478** | **0.358** | **0.798** | **0.268** |
| **Lee, Chris** | **25** | **0.455** | **5.7** | **87.565** | **506** | **0.12** | **1.828** | **0.595** |
| **Lee, Chris** | **36** | **-13.49** | **9.89** | **86.78** | **577** | **0.061** | **0.087** | **0.49** |
| **Long, Lucas** | **1** | **17.725** | **6.655** | **90.905** | **2046** | **0.017** | **0.248** | **0.058** |
| **Long, Lucas** | **2** | **12.13** | **12.43** | **90.64** | **1383** | **0.698** | **0.118** | **0.345** |
| **Long, Lucas** | **27** | **13.175** | **6.77** | **83.565** | **540** | **0.298** | **0.093** | **0.63** |
| **Long, Lucas** | **36** | **-5.89** | **-5.77** | **80.01** | **673** | **0.854** | **0.322** | **0.111** |
| **Long, Lucas** | **37** | **-1.55** | **5.29** | **84.4** | **287** | **0.48** | **0.701** | **0.85** |
| **Lowther, Zac** | **9** | **-12.7** | **16.38** | **89.28** | **2102** | **0.2** | **0.363** | **0.249** |
| **Lowther, Zac** | **27** | **10.595** | **-7.195** | **77.235** | **390** | **0.031** | **0.4** | **0.072** |
| **Lowther, Zac** | **36** | **-12.75** | **12.89** | **84.05** | **463** | **0.235** | **0.491** | **0.324** |
| **Lucas, Josh** | **1** | **13.75** | **13.24** | **90.6** | **111** | **0.001** | **0.574** | **0.346** |
| **Lucas, Josh** | **2** | **19.2** | **6.47** | **90.07** | **84** | **0.743** | **0.481** | **0.178** |
| **Lucas, Josh** | **18** | **-1.56** | **-0.685** | **83.495** | **188** | **2.094** | **0.547** | **0.238** |
| **Manzanillo, Yeudry** | **9** | **10.54** | **17.91** | **88.84** | **59** | **0.639** | **0.146** | **0.367** |
| **Manzanillo, Yeudry** | **18** | **-5.165** | **6.29** | **79.525** | **24** | **1.586** | **0.907** | **0.035** |
| **Marte, Christopher** | **9** | **7.19** | **17.45** | **88.2** | **55** | **0.685** | **0.232** | **2.299** |
| **Marte, Christopher** | **18** | **-5.5** | **-12.86** | **71.67** | **19** | **0.281** | **1.126** | **0.22** |
| **Martinez, Jose** | **9** | **12.71** | **18.36** | **88.17** | **89** | **0.006** | **0.301** | **1.003** |
| **Martinez, Jose** | **18** | **-1.355** | **3.895** | **76.76** | **36** | **1.661** | **1.058** | **1.458** |
| **Matson, Zach** | **9** | **-10.26** | **18.465** | **90.085** | **880** | **0.55** | **0.318** | **0.893** |
| **Matson, Zach** | **27** | **9.945** | **-16.075** | **77.27** | **206** | **0.342** | **0.8** | **0.127** |
| **Matson, Zach** | **36** | **2.85** | **-7.56** | **77.86** | **99** | **1.443** | **1.11** | **0.84** |
| **Means, John** | **9** | **-4.38** | **19.87** | **90.35** | **5418** | **0.458** | **0.617** | **0.135** |
| **Means, John** | **24** | **4.83** | **-6.95** | **75.78** | **1053** | **0.314** | **0.298** | **0.224** |
| **Means, John** | **25** | **3.215** | **4.94** | **82.39** | **810** | **0.257** | **1.796** | **0.575** |
| **Means, John** | **36** | **-10.12** | **19.41** | **82.39** | **1559** | **0.223** | **3.559** | **0.522** |
| **Meisinger, Ryan** | **9** | **7.01** | **17.93** | **91.15** | **1401** | **0.089** | **0.122** | **0.468** |
| **Meisinger, Ryan** | **27** | **13.84** | **9.39** | **83.11** | **95** | **0.398** | **0.776** | **0.494** |
| **Meisinger, Ryan** | **36** | **-4.21** | **2.695** | **82.51** | **850** | **0.158** | **0.067** | **0.194** |
| **Miley, Wade** | **1** | **-1.92** | **17.79** | **90.46** | **1180** | **0.537** | **0.485** | **0.037** |
| **Miley, Wade** | **2** | **-12.31** | **17.63** | **90.85** | **1379** | **0.505** | **0.416** | **0.185** |
| **Miley, Wade** | **24** | **6.18** | **-10.58** | **77.08** | **458** | **0.187** | **0.131** | **0.45** |
| **Miley, Wade** | **25** | **3.44** | **2.82** | **83.88** | **625** | **0.235** | **0.712** | **0.846** |
| **Miley, Wade** | **36** | **-14.63** | **13.79** | **83.27** | **523** | **0.212** | **0.5** | **0.091** |
| **Miller, Andrew** | **9** | **-4.96** | **16.43** | **94.51** | **173** | **0.005** | **0.034** | **0.134** |
| **Miller, Andrew** | **18** | **11.875** | **-2.94** | **83.705** | **166** | **0.15** | **0.114** | **0.135** |
| **Miller, Jalen** | **9** | **6.46** | **18.21** | **91.16** | **125** | **0.483** | **0.968** | **0.705** |
| **Miller, Jalen** | **27** | **12.68** | **12.87** | **79.47** | **31** | **0.234** | **0.567** | **0.094** |
| **Miller, Jalen** | **36** | **-4.83** | **-4.69** | **76.58** | **32** | **0.271** | **0.212** | **1.641** |
| **Ming, Cameron** | **9** | **-13.01** | **8.68** | **87.07** | **893** | **0.774** | **0.008** | **0.117** |
| **Ming, Cameron** | **27** | **4.65** | **3.69** | **81.07** | **713** | **0.205** | **0.006** | **0.366** |
| **Ming, Cameron** | **36** | **-11.78** | **5.105** | **83.34** | **146** | **0.285** | **0.062** | **0.484** |
| **Molina, Marcos** | **9** | **2.39** | **13.81** | **91.6** | **645** | **0.438** | **0.023** | **0.104** |
| **Molina, Marcos** | **27** | **14.24** | **10.74** | **85.97** | **174** | **1.268** | **0.012** | **0.326** |
| **Molina, Marcos** | **36** | **-12.565** | **-3.125** | **80.195** | **168** | **0.11** | **0.114** | **0.19** |
| **Molina, Marcos** | **37** | **-2.35** | **6.505** | **86.665** | **68** | **0.682** | **0.043** | **0.936** |
| **Muckenhirn, Zach** | **9** | **-4.565** | **19.27** | **90.91** | **2256** | **0.422** | **0.157** | **0.156** |
| **Muckenhirn, Zach** | **24** | **2.99** | **-9.82** | **77.54** | **273** | **0.361** | **0.301** | **0.595** |
| **Muckenhirn, Zach** | **25** | **2.34** | **3.92** | **84.26** | **513** | **0.037** | **0.991** | **0.663** |
| **Muckenhirn, Zach** | **36** | **-10.01** | **8.38** | **84.59** | **387** | **0.184** | **0.246** | **1.021** |
| **Naughton, Tim** | **9** | **13.66** | **12.45** | **94.46** | **835** | **0.499** | **0.964** | **0.107** |
| **Naughton, Tim** | **27** | **-4.88** | **-4.29** | **82.4** | **149** | **3.4** | **1.044** | **0.218** |
| **Naughton, Tim** | **36** | **-9.13** | **-5.625** | **81.995** | **104** | **0.6** | **0.647** | **0.215** |
| **Ortiz, Luis** | **9** | **12.51** | **12.23** | **93.29** | **933** | **0.445** | **0.688** | **0.271** |
| **Ortiz, Luis** | **27** | **15.04** | **5.68** | **85.67** | **173** | **0.258** | **0.573** | **0.787** |
| **Ortiz, Luis** | **36** | **-0.71** | **3.345** | **85.5** | **332** | **0.011** | **0.296** | **1.064** |
| **Ortiz, Luis** | **37** | **-8.56** | **-6.37** | **80.06** | **123** | **0.356** | **0.298** | **0.674** |
| **Peluffo, Jhon** | **9** | **10.875** | **14.73** | **91.06** | **2398** | **0.595** | **0.685** | **0.287** |
| **Peluffo, Jhon** | **24** | **12.02** | **10.93** | **82.41** | **423** | **0.377** | **1.319** | **11.118** |
| **Peluffo, Jhon** | **25** | **7.105** | **3.705** | **80.355** | **270** | **0.637** | **0.058** | **0.289** |
| **Peluffo, Jhon** | **36** | **-4.55** | **4.015** | **79.455** | **422** | **0.026** | **0.945** | **0.068** |
| **Peralta, Ofelky** | **9** | **10.08** | **16.405** | **94.37** | **3040** | **0.422** | **0.933** | **0.276** |
| **Peralta, Ofelky** | **27** | **11.11** | **11.585** | **85.205** | **830** | **0.131** | **0.669** | **0.596** |
| **Peralta, Ofelky** | **36** | **-6.68** | **-3.58** | **77.28** | **365** | **0.146** | **0.829** | **0.205** |
| **Perez, Luis** | **9** | **14.99** | **10.63** | **91.58** | **817** | **1.953** | **0.289** | **0.066** |
| **Perez, Luis** | **27** | **15.57** | **3.28** | **86.085** | **242** | **0.951** | **0.923** | **0.204** |
| **Perez, Luis** | **36** | **-12.27** | **-2.15** | **78.53** | **261** | **1.684** | **0.057** | **0.127** |
| **Perez, Luis** | **37** | **-3.17** | **1.765** | **83.7** | **116** | **0.08** | **0.673** | **0.491** |
| **Phillips, Evan** | **9** | **9.68** | **15.54** | **93.855** | **552** | **0.245** | **0.558** | **0.462** |
| **Phillips, Evan** | **27** | **11.01** | **4.705** | **86.565** | **52** | **0.052** | **0.682** | **0.248** |
| **Phillips, Evan** | **36** | **-4.755** | **-5.74** | **84.16** | **206** | **0.086** | **0.196** | **0.3** |
| **Pop, Zach** | **9** | **17.86** | **5.265** | **94.405** | **384** | **0.188** | **0.049** | **0.271** |
| **Pop, Zach** | **27** | **15.69** | **2.94** | **87.77** | **13** | **0.084** | **0.326** | **1.492** |
| **Pop, Zach** | **36** | **-3.81** | **-1.75** | **84.44** | **99** | **0.612** | **0.883** | **0.617** |
| **Ramirez, Yefry** | **9** | **9.7** | **16.52** | **92.03** | **2062** | **0.421** | **0.629** | **0.015** |
| **Ramirez, Yefry** | **27** | **14.19** | **7.81** | **85.78** | **919** | **0.298** | **0.889** | **0.15** |
| **Ramirez, Yefry** | **36** | **0.365** | **7.705** | **86.63** | **626** | **0.001** | **0.398** | **0.571** |
| **Rodriguez, Grayson** | **9** | **10.24** | **18.19** | **92.62** | **541** | **2.007** | **0.018** | **0.568** |
| **Rodriguez, Grayson** | **27** | **12.295** | **4.325** | **79.325** | **172** | **0.837** | **0.463** | **0.548** |
| **Rodriguez, Grayson** | **36** | **-10.68** | **-14.72** | **75.24** | **53** | **0.709** | **0.726** | **0.497** |
| **Rodriguez, Grayson** | **37** | **-7.465** | **7.51** | **80.91** | **130** | **0.79** | **0.092** | **0.493** |
| **Rogers, Josh** | **9** | **-11.51** | **16.49** | **88.91** | **1003** | **0.02** | **0.635** | **0.428** |
| **Rogers, Josh** | **27** | **4.55** | **-0.145** | **79.46** | **544** | **0.586** | **0.796** | **0.303** |
| **Rogers, Josh** | **36** | **-12.08** | **9.01** | **80.24** | **305** | **0.425** | **0.34** | **0.035** |
| **Rom, Drew** | **9** | **-2.45** | **14.865** | **89.165** | **464** | **0.649** | **0.275** | **0.366** |
| **Rom, Drew** | **27** | **8.56** | **-4.16** | **80.32** | **155** | **0.52** | **0.375** | **0.149** |
| **Rom, Drew** | **36** | **-1.51** | **0.08** | **82.56** | **135** | **1.178** | **0.271** | **0.516** |
| **Schultz, Bo** | **9** | **11.12** | **15.05** | **93.905** | **40** | **0.244** | **0.229** | **0.213** |
| **Schultz, Bo** | **18** | **-0.61** | **5.35** | **89.14** | **35** | **0.41** | **0.17** | **0.664** |
| **Scott, Tanner** | **9** | **-9.76** | **15** | **97.06** | **2817** | **0.464** | **0.045** | **0.188** |
| **Scott, Tanner** | **18** | **4.79** | **0.39** | **88.14** | **1052** | **1.738** | **0.473** | **0.817** |
| **Seabrooke, Travis** | **9** | **-11.24** | **13.06** | **90.82** | **1909** | **0.004** | **0.356** | **0.164** |
| **Seabrooke, Travis** | **27** | **7.04** | **-4.89** | **77.42** | **753** | **0.18** | **0.75** | **0.25** |
| **Seabrooke, Travis** | **36** | **-14.3** | **9.98** | **83.375** | **502** | **0.509** | **0.043** | **0.883** |
| **Sedlock, Cody** | **9** | **11.86** | **14.855** | **90.74** | **1436** | **0.307** | **0.596** | **0.437** |
| **Sedlock, Cody** | **27** | **9.2** | **15.09** | **82.26** | **279** | **0.418** | **0.631** | **0.647** |
| **Sedlock, Cody** | **36** | **-3.07** | **-3.525** | **79.91** | **500** | **0.56** | **0.747** | **0.351** |
| **Shepherd, Chandler** | **9** | **8.67** | **14.29** | **91.69** | **105** | **0.184** | **0.474** | **0.405** |
| **Shepherd, Chandler** | **18** | **-8.61** | **-6.11** | **81.125** | **70** | **1.545** | **0.639** | **0.405** |
| **Straily, Dan** | **9** | **8.16** | **17.135** | **90.075** | **420** | **0.193** | **0.15** | **0.074** |
| **Straily, Dan** | **27** | **12.27** | **2.81** | **84.06** | **215** | **0.125** | **0.679** | **0.165** |
| **Straily, Dan** | **36** | **-6.385** | **2.4** | **83.18** | **204** | **0.177** | **0.923** | **1.378** |
| **Tate, Dillon** | **9** | **12.065** | **9.965** | **92.695** | **692** | **0.116** | **0.165** | **0.157** |
| **Tate, Dillon** | **27** | **14.58** | **8.93** | **83.825** | **156** | **0.083** | **0.12** | **0.127** |
| **Tate, Dillon** | **36** | **-4.66** | **1.95** | **83.58** | **179** | **0.056** | **0.033** | **0.224** |
| **Vespi, Nick** | **9** | **0.395** | **7.77** | **87.87** | **1084** | **1.803** | **0.457** | **0.032** |
| **Vespi, Nick** | **24** | **11.005** | **0.03** | **81.06** | **436** | **0.033** | **0.947** | **0.409** |
| **Vespi, Nick** | **25** | **10.945** | **-7.875** | **77.75** | **460** | **0.064** | **0.563** | **0.375** |
| **Vespi, Nick** | **36** | **-5.73** | **3.94** | **82.8** | **131** | **0.566** | **0.17** | **2.064** |
| **Wells, Alex** | **1** | **-12.59** | **11.98** | **83.55** | **702** | **1.24** | **0.551** | **0.966** |
| **Wells, Alex** | **2** | **-8.98** | **19.46** | **88.24** | **3002** | **0.559** | **1.171** | **0.996** |
| **Wells, Alex** | **18** | **8.73** | **-10.455** | **72.47** | **828** | **0.496** | **1.841** | **0.955** |
| **Wilson, Ryan** | **9** | **-9.32** | **16.69** | **87.42** | **741** | **0.242** | **0.077** | **0.022** |
| **Wilson, Ryan** | **27** | **5.07** | **-4.41** | **76.625** | **296** | **0.337** | **0.019** | **0.068** |
| **Wilson, Ryan** | **36** | **-16.5** | **14.01** | **80.81** | **217** | **0.218** | **0.309** | **0.153** |
| **Wotherspoon, Matt** | **9** | **10.98** | **16.46** | **91.34** | **1637** | **0.352** | **0.287** | **0.142** |
| **Wotherspoon, Matt** | **27** | **14.67** | **9.19** | **83.53** | **203** | **0.475** | **0.002** | **0.141** |
| **Wotherspoon, Matt** | **36** | **-2.6** | **2.265** | **83.04** | **812** | **0.19** | **0.413** | **0.017** |
| **Wright Jr., Mike** | **1** | **9.96** | **14.86** | **93.01** | **6018** | **0.222** | **0.358** | **0.146** |
| **Wright Jr., Mike** | **2** | **-2.86** | **5.64** | **89.33** | **833** | **0.575** | **1.122** | **1.487** |
| **Wright Jr., Mike** | **27** | **14.64** | **10.82** | **81.69** | **900** | **0.067** | **0.013** | **0** |
| **Wright Jr., Mike** | **36** | **-6.11** | **-1.85** | **80.5** | **775** | **2.136** | **0.009** | **0.5** |
| **Wright Jr., Mike** | **37** | **-2.61** | **2.97** | **85.09** | **789** | **0.274** | **1.264** | **0.964** |
| **Wright, Dennis** | **9** | **13.195** | **8.965** | **93.145** | **26** | **0.105** | **0.048** | **0.239** |
| **Wright, Dennis** | **18** | **-3.59** | **-1.35** | **84.91** | **7** | **0.348** | **0.51** | **0.418** |
| **Yacabonis, Jimmy** | **9** | **15.77** | **10.89** | **93.89** | **4028** | **1.084** | **0.214** | **0.377** |
| **Yacabonis, Jimmy** | **27** | **15.65** | **5.32** | **86.62** | **161** | **0.731** | **0.281** | **1.009** |
| **Yacabonis, Jimmy** | **36** | **-7.09** | **-2.36** | **82.57** | **1215** | **0.303** | **0.428** | **0.084** |
| **Ynoa, Gabriel** | **9** | **14.91** | **10.36** | **93.44** | **2107** | **0.711** | **0.246** | **0.315** |
| **Ynoa, Gabriel** | **27** | **17.64** | **9.17** | **85.27** | **373** | **0.25** | **0.084** | **0.216** |
| **Ynoa, Gabriel** | **36** | **-2.16** | **1.25** | **83.9** | **1053** | **0.494** | **0.282** | **0.518** |
| **Ysla, Luis** | **1** | **-6.52** | **16.235** | **94.03** | **802** | **0.554** | **0.757** | **1.036** |
| **Ysla, Luis** | **2** | **-13.69** | **11.72** | **91.2** | **872** | **0.017** | **0.047** | **0.135** |
| **Ysla, Luis** | **24** | **6.26** | **-2.915** | **79.355** | **446** | **0.192** | **0.001** | **0.087** |
| **Ysla, Luis** | **25** | **2.325** | **6.285** | **84.75** | **318** | **0.119** | **0.763** | **0.568** |
| **Ysla, Luis** | **36** | **-11.66** | **11.64** | **84.29** | **321** | **0.269** | **0.555** | **0.363** |
| **Zimmermann, Bruce** | **9** | **-6.515** | **16.79** | **91.34** | **668** | **0.157** | **0.111** | **0.235** |
| **Zimmermann, Bruce** | **27** | **1.41** | **0.28** | **84.37** | **375** | **0.284** | **0.084** | **1.668** |
| **Zimmermann, Bruce** | **36** | **-14.38** | **11.71** | **83.79** | **273** | **0.299** | **0.127** | **0.055** |

**Appendix F: List of Pitchers Flagged for Review and Pitchers that Passed Through.**

|  |  |  |
| --- | --- | --- |
| Pitcher Flagged for Review |  | Pitchers Passed Through |
| Armstrong, Shawn |  | Akin, Keegan |
| Baumann, Michael |  | Almengo, Diogenes |
| Benitez, Joel |  | Alvarado, Cristian |
| Bleier, Richard |  | Angomas, Cesar |
| Carroll, Cody |  | Araujo, Pedro |
| Cashner, Andrew |  | Bishop, Cameron |
| Chavez, Jesus |  | Bundy, Dylan |
| Cobb, Alex |  | Burke, Scott |
| Conroy, Ryan |  | Castro, Miguel |
| Del Rosario, Carlos |  | Chleborad, Tanner |
| Eshelman, Tom |  | De La Rosa, Matt |
| Falconett, Pablo |  | Dietz, Matthias |
| Fry, Paul |  | Erwin, Tyler |
| Hanifee, Brenan |  | Fenter, Gray |
| Joyner, Tyler |  | Flaa, Jay |
| Karns, Nate |  | Gilmartin, Sean |
| Lebron, David |  | Givens, Mychal |
| Lee, Chris |  | Gonzalez, Brian |
| Lucas, Josh |  | Gonzalez, Luis |
| Manzanillo, Yeudry |  | Grover, Taylor |
| Marte, Christopher |  | Gruener, Nick |
| Martinez, Jose |  | Guance, Hector |
| Means, John |  | Hall, DL |
| Miller, Jalen |  | Hammonds, Matthew |
| Naughton, Tim |  | Harvey, Hunter |
| Peluffo, Jhon |  | Herb, Tyler |
| Perez, Luis |  | Hess, David |
| Pop, Zach |  | Jimenez, Francisco |
| Rodriguez, Grayson |  | Klimek, Steven |
| Schultz, Bo |  | Kline, Branden |
| Scott, Tanner |  | Knight, Blaine |
| Shepherd, Chandler |  | Kremer, Dean |
| Vespi, Nick |  | Long, Lucas |
| Wells, Alex |  | Lowther, Zac |
| Wright Jr., Mike |  | Matson, Zach |
| Wright, Dennis |  | Meisinger, Ryan |
| Zimmermann, Bruce |  | Miley, Wade |
|  |  | Miller, Andrew |
|  |  | Ming, Cameron |
|  |  | Molina, Marcos |
|  |  | Muckenhirn, Zach |
|  |  | Ortiz, Luis |
|  |  | Peralta, Ofelky |
|  |  | Phillips, Evan |
|  |  | Ramirez, Yefry |
|  |  | Rogers, Josh |
|  |  | Rom, Drew |
|  |  | Seabrooke, Travis |
|  |  | Sedlock, Cody |
|  |  | Straily, Dan |
|  |  | Tate, Dillon |
|  |  | Wilson, Ryan |
|  |  | Wotherspoon, Matt |
|  |  | Yacabonis, Jimmy |
|  |  | Ynoa, Gabriel |
|  |  | Ysla, Luis |

.

**Appendix G: Summary Table 1**

|  |  |
| --- | --- |
| Pitcher | Cluster\_count |
| Akin, Keegan | 3 |
| Almengo, Diogenes | 3 |
| Alvarado, Cristian | 4 |
| Angomas, Cesar | 2 |
| Araujo, Pedro | 4 |
| Armstrong, Shawn | 3 |
| Baumann, Michael | 5 |
| Benitez, Joel | 2 |
| Bishop, Cameron | 3 |
| Bleier, Richard | 5 |
| Bundy, Dylan | 4 |
| Burke, Scott | 2 |
| Carroll, Cody | 3 |
| Cashner, Andrew | 5 |
| Castro, Miguel | 3 |
| Chavez, Jesus | 2 |
| Chleborad, Tanner | 3 |
| Cobb, Alex | 3 |
| Conroy, Ryan | 3 |
| De La Rosa, Matt | 3 |
| Del Rosario, Carlos | 2 |
| Dietz, Matthias | 3 |
| Erwin, Tyler | 4 |
| Eshelman, Tom | 3 |
| Falconett, Pablo | 2 |
| Fenter, Gray | 3 |
| Flaa, Jay | 3 |
| Fry, Paul | 4 |
| Gilmartin, Sean | 4 |
| Givens, Mychal | 3 |
| Gonzalez, Brian | 4 |
| Gonzalez, Luis | 3 |
| Grover, Taylor | 2 |
| Gruener, Nick | 3 |
| Guance, Hector | 3 |
| Hall, DL | 3 |
| Hammonds, Matthew | 3 |
| Hanifee, Brenan | 4 |
| Harvey, Hunter | 3 |
| Herb, Tyler | 4 |
| Hess, David | 5 |
| Jimenez, Francisco | 4 |
| Joyner, Tyler | 3 |
| Karns, Nate | 3 |
| Klimek, Steven | 3 |
| Kline, Branden | 3 |
| Knight, Blaine | 4 |
| Kremer, Dean | 4 |
| Lebron, David | 3 |
| Lee, Chris | 4 |
| Long, Lucas | 5 |
| Lowther, Zac | 3 |
| Lucas, Josh | 3 |
| Manzanillo, Yeudry | 2 |
| Marte, Christopher | 2 |
| Martinez, Jose | 2 |
| Matson, Zach | 3 |
| Means, John | 4 |
| Meisinger, Ryan | 3 |
| Miley, Wade | 5 |
| Miller, Andrew | 2 |
| Miller, Jalen | 3 |
| Ming, Cameron | 3 |
| Molina, Marcos | 4 |
| Muckenhirn, Zach | 4 |
| Naughton, Tim | 3 |
| Ortiz, Luis | 4 |
| Peluffo, Jhon | 4 |
| Peralta, Ofelky | 3 |
| Perez, Luis | 4 |
| Phillips, Evan | 3 |
| Pop, Zach | 3 |
| Ramirez, Yefry | 3 |
| Rodriguez, Grayson | 4 |
| Rogers, Josh | 3 |
| Rom, Drew | 3 |
| Schultz, Bo | 2 |
| Scott, Tanner | 2 |
| Seabrooke, Travis | 3 |
| Sedlock, Cody | 3 |
| Shepherd, Chandler | 2 |
| Straily, Dan | 3 |
| Tate, Dillon | 3 |
| Vespi, Nick | 4 |
| Wells, Alex | 3 |
| Wilson, Ryan | 3 |
| Wotherspoon, Matt | 3 |
| Wright Jr., Mike | 5 |
| Wright, Dennis | 2 |
| Yacabonis, Jimmy | 3 |
| Ynoa, Gabriel | 3 |
| Ysla, Luis | 5 |
| Zimmermann, Bruce | 3 |

**Appendix H: Final V2 R Script**

library(dbscan)

library(ClusterR)

library(readxl)

library(tidyverse)

library(clusterCrit)

library(scatterplot3d)

library(magrittr)

library(WriteXLS)

library(moments)

setwd("C:/Users/Rolandgrad/Documents/MSA Programs/MSA 8350 Analytics Practicum/Orioles Project/Phase 2 clean/NFB2/Cleaned Original Data")

data.org <- read\_excel("Cleaned.Clustering.Data.full.xlsx")

head(data.org)

data.org <- data.org[,c(1,4,5,6,3,2)]

head(data.org)

# re-order columns to run the 3d scatterplot function

data <- cbind(Pitcher = data.org[,1],

break\_x = data.org[,3],

break\_z = data.org[,4],

velo = data.org[,2],

throw = data.org[,5],

ID = data.org[,6])

# generate various list to use is testing

list1 <- c("Carroll, Cody")

list2 <- c("Carroll, Cody",

"Lee, Chris",

"Wilson, Ryan",

"Armstrong, Shawn",

"Rodriguez, Grayson",

"Conroy, Ryan",

"Wright Jr., Mike",

"Lucas, Josh",

"Ming, Cameron",

"Dietz, Matthias",

"Wells, Alex",

"Del Rosario, Carlos")

# full player list

list3 <- sort(unique(data.org$Pitcher))

list3

## Warning: Ensure to set working directory folder. The pitch.happy function will generate plots and a xlsx file for all pitchers.

setwd("C:/Users/Rolandgrad/Documents/MSA Programs/MSA 8350 Analytics Practicum/Orioles Project/Phase 2 clean/NFB2/outputs")

## Function to cluster pitcher data

pitch.happy <- function(player) {

data.full <- filter(data, Pitcher == c(player)) #filter player

data.full.pitch.count <- nrow(data.full) #generate pitch count

data.full.throw <- data.full[,5] #identify L or R arm

cluster.data.df <- data.full #data.frame dataset to use

cluster.data.mx.velo <- as.matrix(cluster.data.df[,c(4)]) #velocity only

# Run PAM k=2

full.pam2 <- Cluster\_Medoids(cluster.data.mx.velo, clusters = 2, distance\_metric = "euclidean", swap\_phase = TRUE)

# Generate PAM k=2 c\_index

full.pam2.int <- data.frame(intCriteria(cluster.data.mx.velo, as.integer(full.pam2$clusters),c("c\_index")))

#######################################

## temp to generate name only (need to review if this function needs to stay)

# int.name.stop <- names(data.frame(intCriteria(cluster.data.mx.velo, as.integer(full.pam2$clusters),c("all"))))

# Run PAM k=3

full.pam3 <- Cluster\_Medoids(cluster.data.mx.velo, clusters = 3, distance\_metric = "euclidean", swap\_phase = TRUE)

# Generate PAM k=3 c\_index

full.pam3.int <- data.frame(intCriteria(cluster.data.mx.velo, as.integer(full.pam3$clusters),c("c\_index")))

# Decision step based on c\_index

pam.pick <- if(full.pam2.int > full.pam3.int){full.pam3

}else{full.pam2}

# redundent step for output only

pam.pick2 <- if(full.pam2.int > full.pam3.int){3

}else{2}

# generate clusters based on c\_index Deicision

full.data.cluster <- cbind(cluster.data.df, cluster = pam.pick$clusters)

full.medoid <- data.frame(pam.pick$medoids)

full.medoid.select <- order(-full.medoid)[1]

# filter out Fastball

fb.data.cluster.full <- filter(full.data.cluster, cluster == full.medoid.select)

head(fb.data.cluster.full)

fb.data.cluster <- fb.data.cluster.full[,c(2,3,4,7)]

head(fb.data.cluster)

# Filter out Non-Fastball

nfb.data.cluster.full <- filter(full.data.cluster, cluster != full.medoid.select)

# Add 10 to Non-Fastballs for easy identification

nfb.data.cluster.full$cluster <- (nfb.data.cluster.full$cluster + 10)

head(nfb.data.cluster.full)

nfb.data.cluster <- nfb.data.cluster.full[,c(2,3,4,7)]

head(nfb.data.cluster)

# Verify total pitch count

nrow(fb.data.cluster.full) + nrow(nfb.data.cluster.full)

data.full.pitch.count

################################################################################

#### USER DEFINED FUNCTION SECTION

#### Applied to All Sections Begin

output.col.names <- c("Pitcher", "Full\_Count", "break\_x", "break\_z", "velo", "Cluster", "Type", "Count\_by\_Type")

#### Applied to All Sections End

##############################

#### Fastball User Defined Funciton Section Begin

## Fastball hdbscan rule and cluster count

fb.hdbscan25.rule <- function(go) {

fb.hdbscan25 <- tryCatch(hdbscan(go[,-4], minPts = 25), error = function(e){0}) #run hdbscan

fb.hdbscan25.nbr <- tryCatch(max(unique(fb.hdbscan25$cluster)), error = function(e){0}) #generate cluster count

fb.cluster.nbr.to.use <- if(fb.hdbscan25.nbr > 0){fb.pam.function(fb.hdbscan25.nbr) #hdbscan rule to continue or stop clustering

} else {fb.single.function()}

fb.cluster.nbr.to.use #output sent to PAM function(multiple clusters) or to Single function (single cluster)

}

## Fastball Multiple Cluster Output PAM. PAM function is used if fb.hdbscan25.rule function generates a cluster count above zero

fb.pam.function <- function(go) {

fb.pam <- Cluster\_Medoids(fb.data.cluster[,-4], clusters = go, distance\_metric = "euclidean", swap\_phase = TRUE) #run pam, k is from the fb.hdbscan25.rule output

fb.pam.clusters <- fb.pam$clusters #pam clusters

fb.pam.output <- cbind((player), data.full.pitch.count, fb.data.cluster[,-4], fb.pam.clusters, "FB\_M2", fb.count) #combine data and clusters for output

colnames(fb.pam.output) <- output.col.names #assign uniform column names

# pitcher output

fb.pam.output

}

## Fastball Single Output. Single function is used if the stop rule or hdbscan rule generates a zero cluster.

fb.single.function <- function(stop) {

fb.single.output <- cbind((player), data.full.pitch.count, fb.data.cluster[,-4], 9, "FB\_S", fb.count) #combine data and clusters for output

colnames(fb.single.output) <- output.col.names #assign uniform column names

# pitcher output

fb.single.output

}

#### Fastball User Defined Funciton Section End

#### Non-Fastball User Defined Function Section Begin

## Non-Fastball hdbscan rule and cluster count

nfb.hdbscan10.rule <- function(go) {

nfb.hdbscan10 <- tryCatch(hdbscan(go[,-4], minPts = 10), error = function(e){0}) #run hdbscan

nfb.hdbscan10.nbr <- tryCatch(max(unique(nfb.hdbscan10$cluster)), error = function(e){0}) #generate cluster count

nfb.cluster.nbr.to.use <- if(nfb.hdbscan10.nbr > 0){nfb2.pam.function(2)} #hdbscan rule to continue or stop clustering

else {nfb.single.function()}

nfb.cluster.nbr.to.use #output sent to PAM function(break\_x split) or to Single function (single cluster)

}

## Non-Fastball Single Output. Single function is used if the stop rule or hdbscan rule generates a zero cluster.

nfb.single.function <- function(stop) {

nfb.single.output <- cbind((player), data.full.pitch.count, nfb.data.cluster[,-4], 18, "NFB\_S", nfb.count) #combine data and clusters for output

colnames(nfb.single.output) <- output.col.names #assign uniform column names

# pitcher output

nfb.single.output

}

#### Non-Fastball User Defined Function Section End

#### Non-Fastball Stage 2 User Defined Function Section Begin.

#### Stage two is for pitches with Non-Fastball hdbscan cluster counts above 0

#### Stage two will split the Non-Fastball along the break\_x axis and cluster seperately

## Non-Fastball Stage 2 split along the break\_x

nfb2.pam.function <- function(go) {

#### Non-Fastball Stage 2 User Defined Function Begin. This section has a L & R

#### User defined function NFB2 R Begin

## Non-Fastball Stage 2 Right side hdbscan decision rule

nfb2.R.hdb <- function(go){

nfb2.R.hdbscan25 <- tryCatch(hdbscan(go[,-4], minPts = 25), error = function(e){0}) #run hdbscan

nfb2.R.hdbscan25.nbr <- tryCatch(max(unique(nfb2.R.hdbscan25$cluster)), error = function(e){0}) #generate cluster count

nfb2.R.cluster.nbr.to.use <- if(nfb2.R.hdbscan25.nbr >0){nfb2.R.pam.function(nfb2.R.hdbscan25.nbr)} #hdbscan rule to continue or stop clustering

else {nfb2.R.single.function()}

nfb2.R.cluster.nbr.to.use #output sent to PAM function(multiple clusters) or to Single function (single cluster)

}

## Non-Fastball Stage 2 Right side single output

nfb2.R.single.function <- function(stop) {

nfb2.R$cluster <- (nfb2.R$cluster + 23) # assign cluster id, plus 23 used to generate a unique cluster id

nfb2.R.single.output <- cbind((player), data.full.pitch.count, nfb2.R[,-4], 27, "NFB2.R\_S", nfb2.R.count) #combine data and clusters for output

colnames(nfb2.R.single.output) <- output.col.names #assign uniform column names

# pitcher output

nfb2.R.single.output

}

## Non-Fastball Stage 2 Right side PAM. PAM function is used if nfb2.R.hdb function generates a cluster count above zero

nfb2.R.pam.function <- function(go) {

nfb2.R.pam <- Cluster\_Medoids(nfb2.R[,-4], clusters = go, distance\_metric = "euclidean", swap\_phase = TRUE) #run pam, k is from the nfb2.R.hdb output

nfb2.R.pam.clusters <- (nfb2.R.pam$clusters + 23) # assign cluster id, plus 23 used to generate a unique cluster id

nfb2.R.pam.output <- cbind((player), data.full.pitch.count, nfb2.R[,-4], nfb2.R.pam.clusters, "NFB2.R\_M", nfb2.R.count) #combine data and clusters for output

colnames(nfb2.R.pam.output) <- output.col.names #assign uniform column names

# pitcher output

nfb2.R.pam.output

}

#### User defined function NFB2 R End

#### User defined function NFB2 L Begin

## Non-Fastball Stage 2 Left side hdbscan decision rule

nfb2.L.hdb <- function(go){

nfb2.L.hdbscan25 <- tryCatch(hdbscan(go[,-4], minPts = 25), error = function(e){0}) #run hdbscan

nfb2.L.hdbscan25.nbr <- tryCatch(max(unique(nfb2.L.hdbscan25$cluster)), error = function(e){0}) #generate cluster count

nfb2.L.cluster.nbr.to.use <- if(nfb2.L.hdbscan25.nbr >0){nfb2.L.pam.function(nfb2.L.hdbscan25.nbr)} #hdbscan rule to continue or stop clustering

else {nfb2.L.single.function()}

nfb2.L.cluster.nbr.to.use #output sent to PAM function(multiple clusters) or to Single function (single cluster)

}

## Non-Fastball Stage 2 Left side single output

nfb2.L.single.function <- function(stop) {

nfb2.L$cluster <- (nfb2.L$cluster + 35) # assign cluster id, plus 35 used to generate a unique cluster id

nfb2.L.single.output <- cbind((player), data.full.pitch.count, nfb2.L[,-4], 36, "NFB2.L\_S", nfb2.L.count)#combine data and clusters for output

colnames(nfb2.L.single.output) <- output.col.names #assign uniform column names

# pitcher output

nfb2.L.single.output

}

## Non-Fastball Stage 2 Right side PAM. PAM function is used if nfb2.L.hdb function generates a cluster count above zero

nfb2.L.pam.function <- function(go) {

nfb2.L.pam <- Cluster\_Medoids(nfb2.L[,-4], clusters = go, distance\_metric = "euclidean", swap\_phase = TRUE) #run pam, k is from the nfb2.L.hdb output

nfb2.L.pam.clusters <- (nfb2.L.pam$clusters + 35) # assign cluster id, plus 35 used to generate a unique cluster id

nfb2.L.pam.output <- cbind((player), data.full.pitch.count, nfb2.L[,-4], nfb2.L.pam.clusters, "NFB2.L\_M", nfb2.L.count) #combine data and clusters for output

colnames(nfb2.L.pam.output) <- output.col.names #assign uniform column names

# pitcher output

nfb2.L.pam.output

}

#### User defined function NFB2 L End

#### Non-Fastball Stage 2 User Defined Function Continuation.

#### This section is the main section of nfb2.pam.function <- function(go)

## This section is run if Non-Fastballs pass through nfb.hdbscan10.rule.

nfb2.full.df <- nfb.data.cluster[,-4]

nfb2.full.df.count <- nrow(nfb2.full.df)

x.nfb2.mx <- matrix(nfb2.full.df[,1]) # generate a matrix of only break\_x

## Run pam on Non-Fastball along the break\_x axis. go = 2, this is a static number generated in the nfb.hdbscan10.rule function

x.nfb2.pam <- Cluster\_Medoids(x.nfb2.mx, clusters = go, distance\_metric = "euclidean", swap\_phase = TRUE)

nfb2.pam.full <- cbind(nfb2.full.df, cluster = x.nfb2.pam$clusters) #combined Non-Fastball data with clusters based on break\_x axis

## Identify Right most medoid

x.nfb2.pam.medoid <- data.frame(x.nfb2.pam$medoids)

nfb2.pam.medoid.select <- order(-x.nfb2.pam.medoid)[1]

## Filter out Non-Fastball Stage 2 Right side

nfb2.R <- filter(nfb2.pam.full, cluster == nfb2.pam.medoid.select)

## Filter out Non-Fastball Stage 2 Left side

nfb2.L <-filter(nfb2.pam.full, cluster != nfb2.pam.medoid.select)

#### Non-Fastball Stage 2 Right side decision rule Begin

# Non-Fastball Stage 2 Right side count to be used in the determination metrics

nfb2.R.count <- nrow(nfb2.R)

# Non-Fastball Stage 2 Right break\_x distribution stats and ratios used in the determination metrics

nfb2.R.x.box.stat <- boxplot.stats(nfb2.R$break\_x)

nfb2.R.x.stat <- data.frame(nfb2.R.x.box.stat$stats)

nfb2.R.x.median <- nfb2.R.x.stat[3,]

nfb2.R.x.out <- nfb2.R.x.box.stat$out

nfb2.R.x.out.above.nbr <- length(nfb2.R.x.out[nfb2.R.x.out > nfb2.R.x.median])

nfb2.R.x.out.below.nbr <- length(nfb2.R.x.out[nfb2.R.x.out < nfb2.R.x.median])

nfb2.R.x.col <- rbind(nfb2.R.x.stat, nfb2.R.x.out.above.nbr,nfb2.R.x.out.below.nbr)

nfb2.R.x.3rd.minus.1st <- (nfb2.R.x.stat[4,] - nfb2.R.x.stat[2,])

nfb2.R.x.max.minus.min <- (nfb2.R.x.stat[5,] - nfb2.R.x.stat[1,])

nfb2.R.x.med.minus.1st <- (nfb2.R.x.stat[3,] - nfb2.R.x.stat[2,])

nfb2.R.x.3rd.minus.med <- (nfb2.R.x.stat[4,] - nfb2.R.x.stat[3,])

nfb2.R.x.3rd.med.1st.ratio.sqrd <- (max(nfb2.R.x.med.minus.1st, nfb2.R.x.3rd.minus.med)^2 /

min(nfb2.R.x.med.minus.1st, nfb2.R.x.3rd.minus.med)^2)

nfb2.R.x.med.minus.min <- (nfb2.R.x.stat[3,] - nfb2.R.x.stat[1,])

nfb2.R.x.max.minus.med <- (nfb2.R.x.stat[5,] - nfb2.R.x.stat[3,])

nfb2.R.x.max.med.min.ratio.sqrd <- (max(nfb2.R.x.med.minus.min, nfb2.R.x.max.minus.med)^2 /

min(nfb2.R.x.med.minus.min, nfb2.R.x.max.minus.med)^2)

nfb2.R.x.out.count.ratio <- (abs(nfb2.R.x.out.above.nbr - nfb2.R.x.out.below.nbr) /

nfb2.R.count)

# Non-Fastball Stage 2 Right break\_z distribution stats and ratios used in the determination metrics

nfb2.R.z.box.stat <- boxplot.stats(nfb2.R$break\_z)

nfb2.R.z.stat <- data.frame(nfb2.R.z.box.stat$stats)

nfb2.R.z.median <- nfb2.R.z.stat[3,]

nfb2.R.z.out <- nfb2.R.z.box.stat$out

nfb2.R.z.out.above.nbr <- length(nfb2.R.z.out[nfb2.R.z.out > nfb2.R.z.median])

nfb2.R.z.out.below.nbr <- length(nfb2.R.z.out[nfb2.R.z.out < nfb2.R.z.median])

nfb2.R.z.col <- rbind(nfb2.R.z.stat, nfb2.R.z.out.above.nbr, nfb2.R.z.out.below.nbr)

nfb2.R.z.3rd.minus.1st <- (nfb2.R.z.stat[4,] - nfb2.R.z.stat[2,])

nfb2.R.z.max.minus.min <- (nfb2.R.z.stat[5,] - nfb2.R.z.stat[1,])

nfb2.R.z.med.minus.1st <- (nfb2.R.z.stat[3,] - nfb2.R.z.stat[2,])

nfb2.R.z.3rd.minus.med <- (nfb2.R.z.stat[4,] - nfb2.R.z.stat[3,])

nfb2.R.z.3rd.med.1st.ratio.sqrd <- (max(nfb2.R.z.med.minus.1st, nfb2.R.z.3rd.minus.med)^2 /

min(nfb2.R.z.med.minus.1st, nfb2.R.z.3rd.minus.med)^2)

nfb2.R.z.med.minus.min <- (nfb2.R.z.stat[3,] - nfb2.R.z.stat[1,])

nfb2.R.z.max.minus.med <- (nfb2.R.z.stat[5,] - nfb2.R.z.stat[3,])

nfb2.R.z.max.med.min.ratio.sqrd <- (max(nfb2.R.z.med.minus.min, nfb2.R.z.max.minus.med)^2 /

min(nfb2.R.z.med.minus.min, nfb2.R.z.max.minus.med)^2)

nfb2.R.z.out.count.ratio <- (abs(nfb2.R.z.out.above.nbr - nfb2.R.z.out.below.nbr) /

nfb2.R.count)

# Non-Fastball Stage 2 Right velo distribution stats and ratios used in the determination metrics

nfb2.R.v.box.stat <- boxplot.stats(nfb2.R$velo)

nfb2.R.v.stat <- data.frame(nfb2.R.v.box.stat$stats)

nfb2.R.v.median <- nfb2.R.v.stat[3,]

nfb2.R.v.out <- nfb2.R.v.box.stat$out

nfb2.R.v.out.above.nbr <- length(nfb2.R.v.out[nfb2.R.v.out > nfb2.R.v.median])

nfb2.R.v.out.below.nbr <- length(nfb2.R.v.out[nfb2.R.v.out < nfb2.R.v.median])

nfb2.R.v.col <- rbind(nfb2.R.v.stat, nfb2.R.v.out.above.nbr, nfb2.R.v.out.below.nbr)

nfb2.R.v.med.minus.1st <- (nfb2.R.v.stat[3,] - nfb2.R.v.stat[2,])

nfb2.R.v.3rd.minus.med <- (nfb2.R.v.stat[4,] - nfb2.R.v.stat[3,])

nfb2.R.v.3rd.med.1st.ratio.sqrd <- (max(nfb2.R.v.med.minus.1st, nfb2.R.v.3rd.minus.med)^2 /

min(nfb2.R.v.med.minus.1st, nfb2.R.v.3rd.minus.med)^2)

nfb2.R.v.med.minus.min <- (nfb2.R.v.stat[3,] - nfb2.R.v.stat[1,])

nfb2.R.v.max.minus.med <- (nfb2.R.v.stat[5,] - nfb2.R.v.stat[3,])

nfb2.R.v.max.med.min.ratio.sqrd <- (max(nfb2.R.v.med.minus.min, nfb2.R.v.max.minus.med)^2 /

min(nfb2.R.v.med.minus.min, nfb2.R.v.max.minus.med)^2)

nfb2.R.v.out.count.ratio <- (abs(nfb2.R.v.out.above.nbr - nfb2.R.v.out.below.nbr) /

nfb2.R.count)

##### Non-Fastball Stage 2 Right Determination metrics

nfb2.R.x.z.max.minus.min <- max(nfb2.R.z.max.minus.min, nfb2.R.x.max.minus.min)

nfb2.R.x.z.3rd.minus.1st <- max(nfb2.R.z.3rd.minus.1st, nfb2.R.x.3rd.minus.1st)

nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd <- max(nfb2.R.x.3rd.med.1st.ratio.sqrd, nfb2.R.z.3rd.med.1st.ratio.sqrd)

nfb2.R.v.max.3rd.med.1st.ratio.sqrd <- nfb2.R.v.3rd.med.1st.ratio.sqrd

nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- (max(nfb2.R.x.3rd.med.1st.ratio.sqrd, nfb2.R.z.3rd.med.1st.ratio.sqrd) /

min(nfb2.R.x.3rd.med.1st.ratio.sqrd, nfb2.R.z.3rd.med.1st.ratio.sqrd))

nfb2.R.x.z.max.max.med.min.ratio.sqrd <- max(nfb2.R.x.max.med.min.ratio.sqrd, nfb2.R.z.max.med.min.ratio.sqrd)

nfb2.R.v.max.max.med.min.ratio.sqrd <- nfb2.R.v.max.med.min.ratio.sqrd

nfb2.R.x.z.max.max.med.min.ratio.sqrd.ratio <- (max(nfb2.R.x.max.med.min.ratio.sqrd, nfb2.R.z.max.med.min.ratio.sqrd) /

min(nfb2.R.x.max.med.min.ratio.sqrd, nfb2.R.z.max.med.min.ratio.sqrd))

nfb2.R.x.z.v.max.out.ratio <- max(nfb2.R.x.out.count.ratio, nfb2.R.z.out.count.ratio, nfb2.R.v.out.count.ratio)

##### Non-Fastball Stage 2 Right Tuning Determination Parameters

maximum.nfb2.R.x.z.max.minus.min <- 25

maximum.nfb2.R.x.z.3rd.minus.1st <- 25

maximum.nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd <- 2.2

maximum.nfb2.R.v.max.3rd.med.1st.ratio.sqrd <- 1.8

maximum.nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- 2

maximum.nfb2.R.x.z.max.max.med.min.ratio.sqrd <- 2.2

maximum.nfb2.R.v.max.max.med.min.ratio.sqrd <- 1.8

maximum.nfb2.R.x.z.max.max.med.min.ratio.sqrd.ratio <- 2

maximum.nfb2.R.x.z.v.max.out.ratio <- 0.09

##### Decision to Stop or Continue clustering Non-Fastball Stage 2 Right

nfb2.R.d1 <- if(nfb2.R.x.z.max.minus.min > maximum.nfb2.R.x.z.max.minus.min){1} else {0}

nfb2.R.d2 <- if(nfb2.R.x.z.3rd.minus.1st > maximum.nfb2.R.x.z.3rd.minus.1st){1} else {0}

nfb2.R.d3 <- if(nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd > maximum.nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb2.R.d4 <- if(nfb2.R.v.max.3rd.med.1st.ratio.sqrd > maximum.nfb2.R.v.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb2.R.d5 <- if(nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd.ratio > maximum.nfb2.R.x.z.max.3rd.med.1st.ratio.sqrd.ratio){1} else {0}

nfb2.R.d6 <- if(nfb2.R.x.z.max.max.med.min.ratio.sqrd > maximum.nfb2.R.x.z.max.max.med.min.ratio.sqrd){1} else {0}

nfb2.R.d7 <- if(nfb2.R.v.max.max.med.min.ratio.sqrd > maximum.nfb2.R.v.max.max.med.min.ratio.sqrd){1} else {0}

nfb2.R.d8 <- if(nfb2.R.x.z.max.max.med.min.ratio.sqrd.ratio > maximum.nfb2.R.x.z.max.max.med.min.ratio.sqrd.ratio){1} else {0}

nfb2.R.d9 <- if(nfb2.R.x.z.v.max.out.ratio > maximum.nfb2.R.x.z.v.max.out.ratio){1} else {0}

nfb2.R.stop.go.rule <- if(sum(nfb2.R.d1,nfb2.R.d2,nfb2.R.d3,nfb2.R.d4,nfb2.R.d5,nfb2.R.d6,nfb2.R.d7,nfb2.R.d8,nfb2.R.d9) == 0){nfb2.R.single.function()} else {nfb2.R.hdb(nfb2.R)}

nfb2.R.stop.go.rule # these rules will feed into the Stage 2 Right side user defined section above

#### Non-Fastball Stage 2 Right side decision rule End

#### Non-Fastball Stage 2 Left side decision rule Begin

# Non-Fastball Stage 2 Left side count to be used in the determination metrics

nfb2.L.count <- nrow(nfb2.L)

# Non-Fastball Stage 2 Left break\_x distribution stats and ratios used in the determination metrics

nfb2.L.x.box.stat <- boxplot.stats(nfb2.L$break\_x)

nfb2.L.x.stat <- data.frame(nfb2.L.x.box.stat$stats)

nfb2.L.x.median <- nfb2.L.x.stat[3,]

nfb2.L.x.out <- nfb2.L.x.box.stat$out

nfb2.L.x.out.above.nbr <- length(nfb2.L.x.out[nfb2.L.x.out > nfb2.L.x.median])

nfb2.L.x.out.below.nbr <- length(nfb2.L.x.out[nfb2.L.x.out < nfb2.L.x.median])

nfb2.L.x.col <- rbind(nfb2.L.x.stat, nfb2.L.x.out.above.nbr,nfb2.L.x.out.below.nbr)

nfb2.L.x.3rd.minus.1st <- (nfb2.L.x.stat[4,] - nfb2.L.x.stat[2,])

nfb2.L.x.max.minus.min <- (nfb2.L.x.stat[5,] - nfb2.L.x.stat[1,])

nfb2.L.x.med.minus.1st <- (nfb2.L.x.stat[3,] - nfb2.L.x.stat[2,])

nfb2.L.x.3rd.minus.med <- (nfb2.L.x.stat[4,] - nfb2.L.x.stat[3,])

nfb2.L.x.3rd.med.1st.ratio.sqrd <- (max(nfb2.L.x.med.minus.1st, nfb2.L.x.3rd.minus.med)^2 /

min(nfb2.L.x.med.minus.1st, nfb2.L.x.3rd.minus.med)^2)

nfb2.L.x.med.minus.min <- (nfb2.L.x.stat[3,] - nfb2.L.x.stat[1,])

nfb2.L.x.max.minus.med <- (nfb2.L.x.stat[5,] - nfb2.L.x.stat[3,])

nfb2.L.x.max.med.min.ratio.sqrd <- (max(nfb2.L.x.med.minus.min, nfb2.L.x.max.minus.med)^2 /

min(nfb2.L.x.med.minus.min, nfb2.L.x.max.minus.med)^2)

nfb2.L.x.out.count.ratio <- (abs(nfb2.L.x.out.above.nbr - nfb2.L.x.out.below.nbr) /

nfb2.L.count)

# Non-Fastball Stage 2 Left break\_z distribution stats and ratios used in the determination metrics

nfb2.L.z.box.stat <- boxplot.stats(nfb2.L$break\_z)

nfb2.L.z.stat <- data.frame(nfb2.L.z.box.stat$stats)

nfb2.L.z.median <- nfb2.L.z.stat[3,]

nfb2.L.z.out <- nfb2.L.z.box.stat$out

nfb2.L.z.out.above.nbr <- length(nfb2.L.z.out[nfb2.L.z.out > nfb2.L.z.median])

nfb2.L.z.out.below.nbr <- length(nfb2.L.z.out[nfb2.L.z.out < nfb2.L.z.median])

nfb2.L.z.col <- rbind(nfb2.L.z.stat, nfb2.L.z.out.above.nbr, nfb2.L.z.out.below.nbr)

nfb2.L.z.3rd.minus.1st <- (nfb2.L.z.stat[4,] - nfb2.L.z.stat[2,])

nfb2.L.z.max.minus.min <- (nfb2.L.z.stat[5,] - nfb2.L.z.stat[1,])

nfb2.L.z.med.minus.1st <- (nfb2.L.z.stat[3,] - nfb2.L.z.stat[2,])

nfb2.L.z.3rd.minus.med <- (nfb2.L.z.stat[4,] - nfb2.L.z.stat[3,])

nfb2.L.z.3rd.med.1st.ratio.sqrd <- (max(nfb2.L.z.med.minus.1st, nfb2.L.z.3rd.minus.med)^2 /

min(nfb2.L.z.med.minus.1st, nfb2.L.z.3rd.minus.med)^2)

nfb2.L.z.med.minus.min <- (nfb2.L.z.stat[3,] - nfb2.L.z.stat[1,])

nfb2.L.z.max.minus.med <- (nfb2.L.z.stat[5,] - nfb2.L.z.stat[3,])

nfb2.L.z.max.med.min.ratio.sqrd <- (max(nfb2.L.z.med.minus.min, nfb2.L.z.max.minus.med)^2 /

min(nfb2.L.z.med.minus.min, nfb2.L.z.max.minus.med)^2)

nfb2.L.z.out.count.ratio <- (abs(nfb2.L.z.out.above.nbr - nfb2.L.z.out.below.nbr) /

nfb2.L.count)

# Non-Fastball Stage 2 Left velo distribution stats and ratios used in the determination metrics

nfb2.L.v.box.stat <- boxplot.stats(nfb2.L$velo)

nfb2.L.v.stat <- data.frame(nfb2.L.v.box.stat$stats)

nfb2.L.v.median <- nfb2.L.v.stat[3,]

nfb2.L.v.out <- nfb2.L.v.box.stat$out

nfb2.L.v.out.above.nbr <- length(nfb2.L.v.out[nfb2.L.v.out > nfb2.L.v.median])

nfb2.L.v.out.below.nbr <- length(nfb2.L.v.out[nfb2.L.v.out < nfb2.L.v.median])

nfb2.L.v.col <- rbind(nfb2.L.v.stat, nfb2.L.v.out.above.nbr, nfb2.L.v.out.below.nbr)

nfb2.L.v.med.minus.1st <- (nfb2.L.v.stat[3,] - nfb2.L.v.stat[2,])

nfb2.L.v.3rd.minus.med <- (nfb2.L.v.stat[4,] - nfb2.L.v.stat[3,])

nfb2.L.v.3rd.med.1st.ratio.sqrd <- (max(nfb2.L.v.med.minus.1st, nfb2.L.v.3rd.minus.med)^2 /

min(nfb2.L.v.med.minus.1st, nfb2.L.v.3rd.minus.med)^2)

nfb2.L.v.med.minus.min <- (nfb2.L.v.stat[3,] - nfb2.L.v.stat[1,])

nfb2.L.v.max.minus.med <- (nfb2.L.v.stat[5,] - nfb2.L.v.stat[3,])

nfb2.L.v.max.med.min.ratio.sqrd <- (max(nfb2.L.v.med.minus.min, nfb2.L.v.max.minus.med)^2 /

min(nfb2.L.v.med.minus.min, nfb2.L.v.max.minus.med)^2)

nfb2.L.v.out.count.ratio <- (abs(nfb2.L.v.out.above.nbr - nfb2.L.v.out.below.nbr) /

nfb2.L.count)

##### Non-Fastball Stage 2 Left Determination metrics

nfb2.L.x.z.max.minus.min <- max(nfb2.L.z.max.minus.min, nfb2.L.x.max.minus.min)

nfb2.L.x.z.3rd.minus.1st <- max(nfb2.L.z.3rd.minus.1st, nfb2.L.x.3rd.minus.1st)

nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd <- max(nfb2.L.x.3rd.med.1st.ratio.sqrd, nfb2.L.z.3rd.med.1st.ratio.sqrd)

nfb2.L.v.max.3rd.med.1st.ratio.sqrd <- nfb2.L.v.3rd.med.1st.ratio.sqrd

nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- (max(nfb2.L.x.3rd.med.1st.ratio.sqrd, nfb2.L.z.3rd.med.1st.ratio.sqrd) /

min(nfb2.L.x.3rd.med.1st.ratio.sqrd, nfb2.L.z.3rd.med.1st.ratio.sqrd))

nfb2.L.x.z.max.max.med.min.ratio.sqrd <- max(nfb2.L.x.max.med.min.ratio.sqrd, nfb2.L.z.max.med.min.ratio.sqrd)

nfb2.L.v.max.max.med.min.ratio.sqrd <- nfb2.L.v.max.med.min.ratio.sqrd

nfb2.L.x.z.max.max.med.min.ratio.sqrd.ratio <- (max(nfb2.L.x.max.med.min.ratio.sqrd, nfb2.L.z.max.med.min.ratio.sqrd) /

min(nfb2.L.x.max.med.min.ratio.sqrd, nfb2.L.z.max.med.min.ratio.sqrd))

nfb2.L.x.z.v.max.out.ratio <- max(nfb2.L.x.out.count.ratio, nfb2.L.z.out.count.ratio, nfb2.L.v.out.count.ratio)

##### Non-Fastball Stage 2 Left Tuning Determination Parameters

maximum.nfb2.L.x.z.max.minus.min <- 25

maximum.nfb2.L.x.z.3rd.minus.1st <- 25

maximum.nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd <- 2.2

maximum.nfb2.L.v.max.3rd.med.1st.ratio.sqrd <- 1.8

maximum.nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- 2

maximum.nfb2.L.x.z.max.max.med.min.ratio.sqrd <- 2.2

maximum.nfb2.L.v.max.max.med.min.ratio.sqrd <- 1.8

maximum.nfb2.L.x.z.max.max.med.min.ratio.sqrd.ratio <- 2

maximum.nfb2.L.x.z.v.max.out.ratio <- 0.09

##### Decision to Stop or Continue clustering Non-Fastball Stage 2 Left

nfb2.L.d1 <- if(nfb2.L.x.z.max.minus.min > maximum.nfb2.L.x.z.max.minus.min){1} else {0}

nfb2.L.d2 <- if(nfb2.L.x.z.3rd.minus.1st > maximum.nfb2.L.x.z.3rd.minus.1st){1} else {0}

nfb2.L.d3 <- if(nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd > maximum.nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb2.L.d4 <- if(nfb2.L.v.max.3rd.med.1st.ratio.sqrd > maximum.nfb2.L.v.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb2.L.d5 <- if(nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd.ratio > maximum.nfb2.L.x.z.max.3rd.med.1st.ratio.sqrd.ratio){1} else {0}

nfb2.L.d6 <- if(nfb2.L.x.z.max.max.med.min.ratio.sqrd > maximum.nfb2.L.x.z.max.max.med.min.ratio.sqrd){1} else {0}

nfb2.L.d7 <- if(nfb2.L.v.max.max.med.min.ratio.sqrd > maximum.nfb2.L.v.max.max.med.min.ratio.sqrd){1} else {0}

nfb2.L.d8 <- if(nfb2.L.x.z.max.max.med.min.ratio.sqrd.ratio > maximum.nfb2.L.x.z.max.max.med.min.ratio.sqrd.ratio){1} else {0}

nfb2.L.d9 <- if(nfb2.L.x.z.v.max.out.ratio > maximum.nfb2.L.x.z.v.max.out.ratio){1} else {0}

nfb2.L.stop.go.rule <- if(sum(nfb2.L.d1,nfb2.L.d2,nfb2.L.d3,nfb2.L.d4,nfb2.L.d5,nfb2.L.d6,nfb2.L.d7,nfb2.L.d8,nfb2.L.d9) == 0){nfb2.L.single.function()} else {nfb2.L.hdb(nfb2.L)}

nfb2.L.stop.go.rule # these rules will feed into the Stage 2 Left side user defined section above

#### Non-Fastball Stage 2 Left side decision rule End

## Combine NFB2.L output and NFB2.R output

nfb2.L.and.nfb2.R.output <- rbind(nfb2.L.stop.go.rule, nfb2.R.stop.go.rule)

nfb2.L.and.nfb2.R.row.count <- nrow(nfb2.L.and.nfb2.R.output)

nfb2.L.and.nfb2.R.cluster.count <- length(unique(nfb2.L.and.nfb2.R.output$Cluster))

## Non-Fastball Stage 2 output

nfb2.L.and.nfb2.R.output

}

#### Fastball and Non-Fastball Decsion Rule seciton

##### Fastball Stop Decision Rule Begin

# Fastbll count to be used in the determination metrics

fb.count <- nrow(fb.data.cluster)

# Fastball break\_x distribution stats and ratios used in the determination metrics

fb.x.box.stat <- boxplot.stats(fb.data.cluster$break\_x)

fb.x.stat <- data.frame(fb.x.box.stat$stats)

fb.x.median <- fb.x.stat[3,]

fb.x.out <- fb.x.box.stat$out

fb.x.out.above.nbr <- length(fb.x.out[fb.x.out > fb.x.median])

fb.x.out.below.nbr <- length(fb.x.out[fb.x.out < fb.x.median])

fb.x.col <- rbind(fb.x.stat, fb.x.out.above.nbr, fb.x.out.below.nbr)

fb.x.3rd.minus.1st <- (fb.x.stat[4,] - fb.x.stat[2,])

fb.x.max.minus.min <- (fb.x.stat[5,] - fb.x.stat[1,])

fb.x.med.minus.1st <- (fb.x.stat[3,] - fb.x.stat[2,])

fb.x.3rd.minus.med <- (fb.x.stat[4,] - fb.x.stat[3,])

fb.x.3rd.med.1st.ratio.sqrd <- (max(fb.x.med.minus.1st, fb.x.3rd.minus.med)^2 /

min(fb.x.med.minus.1st, fb.x.3rd.minus.med)^2)

fb.x.med.minus.min <- (fb.x.stat[3,] - fb.x.stat[1,])

fb.x.max.minus.med <- (fb.x.stat[5,] - fb.x.stat[3,])

fb.x.max.med.min.ratio.sqrd <- (max(fb.x.med.minus.min, fb.x.max.minus.med)^2 /

min(fb.x.med.minus.min, fb.x.max.minus.med)^2)

fb.x.out.count.ratio <- (abs(fb.x.out.above.nbr - fb.x.out.below.nbr) /

fb.count)

# Fastball break\_z distribution stats and ratios used in the determination metrics

fb.z.box.stat <- boxplot.stats(fb.data.cluster$break\_z)

fb.z.stat <- data.frame(fb.z.box.stat$stats)

fb.z.median <- fb.z.stat[3,]

fb.z.out <- fb.z.box.stat$out

fb.z.out.above.nbr <- length(fb.z.out[fb.z.out > fb.z.median])

fb.z.out.below.nbr <- length(fb.z.out[fb.z.out < fb.z.median])

fb.z.col <- rbind(fb.z.stat, fb.z.out.above.nbr, fb.z.out.below.nbr)

fb.z.3rd.minus.1st <- (fb.z.stat[4,] - fb.z.stat[2,])

fb.z.max.minus.min <- (fb.z.stat[5,] - fb.z.stat[1,])

fb.z.med.minus.1st <- (fb.z.stat[3,] - fb.z.stat[2,])

fb.z.3rd.minus.med <- (fb.z.stat[4,] - fb.z.stat[3,])

fb.z.3rd.med.1st.ratio.sqrd <- (max(fb.z.med.minus.1st, fb.z.3rd.minus.med)^2 /

min(fb.z.med.minus.1st, fb.z.3rd.minus.med)^2)

fb.z.med.minus.min <- (fb.z.stat[3,] - fb.z.stat[1,])

fb.z.max.minus.med <- (fb.z.stat[5,] - fb.z.stat[3,])

fb.z.max.med.min.ratio.sqrd <- (max(fb.z.med.minus.min, fb.z.max.minus.med)^2 /

min(fb.z.med.minus.min, fb.z.max.minus.med)^2)

fb.z.out.count.ratio <- (abs(fb.z.out.above.nbr - fb.z.out.below.nbr) /

fb.count)

# Fastball velo distribution stats and ratios used in the determination metrics

fb.v.box.stat <- boxplot.stats(fb.data.cluster$velo)

fb.v.stat <- data.frame(fb.v.box.stat$stats)

fb.v.median <- fb.v.stat[3,]

fb.v.out <- fb.v.box.stat$out

fb.v.out.above.nbr <- length(fb.v.out[fb.v.out > fb.v.median])

fb.v.out.below.nbr <- length(fb.v.out[fb.v.out < fb.v.median])

fb.v.col <- rbind(fb.v.stat, fb.v.out.above.nbr, fb.v.out.below.nbr)

fb.v.3rd.minus.1st <- (fb.v.stat[4,] - fb.v.stat[2,])

fb.v.max.minus.min <- (fb.v.stat[5,] - fb.v.stat[1,])

fb.v.med.minus.1st <- (fb.v.stat[3,] - fb.v.stat[2,])

fb.v.3rd.minus.med <- (fb.v.stat[4,] - fb.v.stat[3,])

fb.v.3rd.med.1st.ratio.sqrd <- (max(fb.v.med.minus.1st, fb.v.3rd.minus.med)^2 /

min(fb.v.med.minus.1st, fb.v.3rd.minus.med)^2)

fb.v.med.minus.min <- (fb.v.stat[3,] - fb.v.stat[1,])

fb.v.max.minus.med <- (fb.v.stat[5,] - fb.v.stat[3,])

fb.v.max.med.min.ratio.sqrd <- (max(fb.v.med.minus.min, fb.v.max.minus.med)^2 /

min(fb.v.med.minus.min, fb.v.max.minus.med)^2)

fb.v.out.count.ratio <- (abs(fb.v.out.above.nbr - fb.v.out.below.nbr) /

fb.count)

##### Fastball Determination metrics

fb.x.z.max.minus.min <- max(fb.z.max.minus.min, fb.x.max.minus.min)

fb.x.z.3rd.minus.1st <- max(fb.z.3rd.minus.1st, fb.x.3rd.minus.1st)

fb.x.z.max.3rd.med.1st.ratio.sqrd <- max(fb.x.3rd.med.1st.ratio.sqrd, fb.z.3rd.med.1st.ratio.sqrd)

fb.v.max.3rd.med.1st.ratio.sqrd <- fb.v.3rd.med.1st.ratio.sqrd

fb.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- (max(fb.x.3rd.med.1st.ratio.sqrd, fb.z.3rd.med.1st.ratio.sqrd) /

min(fb.x.3rd.med.1st.ratio.sqrd, fb.z.3rd.med.1st.ratio.sqrd))

fb.x.z.max.max.med.min.ratio.sqrd <- max(fb.x.max.med.min.ratio.sqrd, fb.z.max.med.min.ratio.sqrd)

fb.v.max.max.med.min.ratio.sqrd <- fb.v.max.med.min.ratio.sqrd

fb.x.z.max.max.med.min.ratio.sqrd.ratio <- (max(fb.x.max.med.min.ratio.sqrd, fb.z.max.med.min.ratio.sqrd) /

min(fb.x.max.med.min.ratio.sqrd, fb.z.max.med.min.ratio.sqrd))

fb.x.z.v.max.out.ratio <- max(fb.x.out.count.ratio, fb.z.out.count.ratio, fb.v.out.count.ratio)

##### Fastball Tuning Determination Parameters

maximum.fb.x.z.max.minus.min <- 21

maximum.fb.x.z.3rd.minus.1st <- 7

maximum.fb.x.z.max.3rd.med.1st.ratio.sqrd <- 2.2

maximum.fb.v.max.3rd.med.1st.ratio.sqrd <- 1.8

maximum.fb.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- 2

maximum.fb.x.z.max.max.med.min.ratio.sqrd <- 2.2

maximum.fb.v.max.max.med.min.ratio.sqrd <- 1.8

maximum.fb.x.z.max.max.med.min.ratio.sqrd.ratio <- 2

maximum.fb.x.z.v.max.out.ratio <- 0.09

##### Fastball Decision to Stop or Continue clustering

fb.d1 <- if(fb.x.z.max.minus.min > maximum.fb.x.z.max.minus.min){1} else {0}

fb.d2 <- if(fb.x.z.3rd.minus.1st > maximum.fb.x.z.3rd.minus.1st){1} else {0}

fb.d3 <- if(fb.x.z.max.3rd.med.1st.ratio.sqrd > maximum.fb.x.z.max.3rd.med.1st.ratio.sqrd){1} else {0}

fb.d4 <- if(fb.v.max.3rd.med.1st.ratio.sqrd > maximum.fb.v.max.3rd.med.1st.ratio.sqrd){1} else {0}

fb.d5 <- if(fb.x.z.max.3rd.med.1st.ratio.sqrd.ratio > maximum.fb.x.z.max.3rd.med.1st.ratio.sqrd.ratio){1} else {0}

fb.d6 <- if(fb.x.z.max.max.med.min.ratio.sqrd > maximum.fb.x.z.max.max.med.min.ratio.sqrd){1} else {0}

fb.d7 <- if(fb.v.max.max.med.min.ratio.sqrd > maximum.fb.v.max.max.med.min.ratio.sqrd){1} else {0}

fb.d8 <- if(fb.x.z.max.max.med.min.ratio.sqrd.ratio > maximum.fb.x.z.max.max.med.min.ratio.sqrd.ratio){1} else {0}

fb.d9 <- if(fb.x.z.v.max.out.ratio > maximum.fb.x.z.v.max.out.ratio){1} else {0}

# these rules will feed into the Fastball user defined section above

fb.stop.go.rule <- if(sum(fb.d1,fb.d2,fb.d3,fb.d4,fb.d5,fb.d6,fb.d7,fb.d8,fb.d9) == 0){fb.single.function()} else {fb.hdbscan25.rule(fb.data.cluster)}

#### Fastball Stop Decision Rule End

#### Non-Fastball stop decision rule Begin

# Non-Fastball count to be used in the determination metrics

nfb.count <- nrow(nfb.data.cluster)

# Non-Fastball break\_x distribution stats and ratios used in the determination metrics

nfb.x.box.stat <- boxplot.stats(nfb.data.cluster$break\_x)

nfb.x.stat <- data.frame(nfb.x.box.stat$stats)

nfb.x.median <- nfb.x.stat[3,]

nfb.x.out <- nfb.x.box.stat$out

nfb.x.out.above.nbr <- length(nfb.x.out[nfb.x.out > nfb.x.median])

nfb.x.out.below.nbr <- length(nfb.x.out[nfb.x.out < nfb.x.median])

nfb.x.col <- rbind(nfb.x.stat, nfb.x.out.above.nbr, nfb.x.out.below.nbr)

nfb.x.3rd.minus.1st <- (nfb.x.stat[4,] - nfb.x.stat[2,])

nfb.x.max.minus.min <- (nfb.x.stat[5,] - nfb.x.stat[1,])

nfb.x.med.minus.1st <- (nfb.x.stat[3,] - nfb.x.stat[2,])

nfb.x.3rd.minus.med <- (nfb.x.stat[4,] - nfb.x.stat[3,])

nfb.x.3rd.med.1st.ratio.sqrd <- (max(nfb.x.med.minus.1st, nfb.x.3rd.minus.med)^2 /

min(nfb.x.med.minus.1st, nfb.x.3rd.minus.med)^2)

nfb.x.med.minus.min <- (nfb.x.stat[3,] - nfb.x.stat[1,])

nfb.x.max.minus.med <- (nfb.x.stat[5,] - nfb.x.stat[3,])

nfb.x.max.med.min.ratio.sqrd <- (max(nfb.x.med.minus.min, nfb.x.max.minus.med)^2 /

min(nfb.x.med.minus.min, nfb.x.max.minus.med)^2)

nfb.x.out.count.ratio <- (abs(nfb.x.out.above.nbr - nfb.x.out.below.nbr) /

nfb.count)

# Non-Fastball break\_z distribution stats and ratios used in the determination metrics

nfb.z.box.stat <- boxplot.stats(nfb.data.cluster$break\_z)

nfb.z.stat <- data.frame(nfb.z.box.stat$stats)

nfb.z.median <- nfb.z.stat[3,]

nfb.z.out <- nfb.z.box.stat$out

nfb.z.out.above.nbr <- length(nfb.z.out[nfb.z.out > nfb.z.median])

nfb.z.out.below.nbr <- length(nfb.z.out[nfb.z.out < nfb.z.median])

nfb.z.col <- rbind(nfb.z.stat, nfb.z.out.above.nbr, nfb.z.out.below.nbr)

nfb.z.3rd.minus.1st <- (nfb.z.stat[4,] - nfb.z.stat[2,])

nfb.z.max.minus.min <- (nfb.z.stat[5,] - nfb.z.stat[1,])

nfb.z.med.minus.1st <- (nfb.z.stat[3,] - nfb.z.stat[2,])

nfb.z.3rd.minus.med <- (nfb.z.stat[4,] - nfb.z.stat[3,])

nfb.z.3rd.med.1st.ratio.sqrd <- (max(nfb.z.med.minus.1st, nfb.z.3rd.minus.med)^2 /

min(nfb.z.med.minus.1st, nfb.z.3rd.minus.med)^2)

nfb.z.med.minus.min <- (nfb.z.stat[3,] - nfb.z.stat[1,])

nfb.z.max.minus.med <- (nfb.z.stat[5,] - nfb.z.stat[3,])

nfb.z.max.med.min.ratio.sqrd <- (max(nfb.z.med.minus.min, nfb.z.max.minus.med)^2 /

min(nfb.z.med.minus.min, nfb.z.max.minus.med)^2)

nfb.z.out.count.ratio <- (abs(nfb.z.out.above.nbr - nfb.z.out.below.nbr) /

nfb.count)

# Non-Fastball velo distribution stats and ratios used in the determination metrics

nfb.v.box.stat <- boxplot.stats(nfb.data.cluster$velo)

nfb.v.stat <- data.frame(nfb.v.box.stat$stats)

nfb.v.median <- nfb.v.stat[3,]

nfb.v.out <- nfb.v.box.stat$out

nfb.v.out.above.nbr <- length(nfb.v.out[nfb.v.out > nfb.v.median])

nfb.v.out.below.nbr <- length(nfb.v.out[nfb.v.out < nfb.v.median])

nfb.v.col <- rbind(nfb.v.stat, nfb.v.out.above.nbr, nfb.v.out.below.nbr)

nfb.v.med.minus.1st <- (nfb.v.stat[3,] - nfb.v.stat[2,])

nfb.v.3rd.minus.med <- (nfb.v.stat[4,] - nfb.v.stat[3,])

nfb.v.3rd.med.1st.ratio.sqrd <- (max(nfb.v.med.minus.1st, nfb.v.3rd.minus.med)^2 /

min(nfb.v.med.minus.1st, nfb.v.3rd.minus.med)^2)

nfb.v.med.minus.min <- (nfb.v.stat[3,] - nfb.v.stat[1,])

nfb.v.max.minus.med <- (nfb.v.stat[5,] - nfb.v.stat[3,])

nfb.v.max.med.min.ratio.sqrd <- (max(nfb.v.med.minus.min, nfb.v.max.minus.med)^2 /

min(nfb.v.med.minus.min, nfb.v.max.minus.med)^2)

nfb.v.out.count.ratio <- (abs(nfb.v.out.above.nbr - nfb.v.out.below.nbr) /

nfb.count)

##### Non-Fastball Determination metrics

nfb.x.z.max.minus.min <- max(nfb.z.max.minus.min, nfb.x.max.minus.min)

nfb.x.z.3rd.minus.1st <- max(nfb.z.3rd.minus.1st, nfb.x.3rd.minus.1st)

nfb.x.z.max.3rd.med.1st.ratio.sqrd <- max(nfb.x.3rd.med.1st.ratio.sqrd, nfb.z.3rd.med.1st.ratio.sqrd)

nfb.v.max.3rd.med.1st.ratio.sqrd <- nfb.v.3rd.med.1st.ratio.sqrd

nfb.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- (max(nfb.x.3rd.med.1st.ratio.sqrd, nfb.z.3rd.med.1st.ratio.sqrd) /

min(nfb.x.3rd.med.1st.ratio.sqrd, nfb.z.3rd.med.1st.ratio.sqrd))

nfb.x.z.max.max.med.min.ratio.sqrd <- max(nfb.x.max.med.min.ratio.sqrd, nfb.z.max.med.min.ratio.sqrd)

nfb.v.max.max.med.min.ratio.sqrd <- nfb.v.max.med.min.ratio.sqrd

nfb.x.z.max.max.med.min.ratio.sqrd.ratio <- (max(nfb.x.max.med.min.ratio.sqrd, nfb.z.max.med.min.ratio.sqrd) /

min(nfb.x.max.med.min.ratio.sqrd, nfb.z.max.med.min.ratio.sqrd))

nfb.x.z.v.max.out.ratio <- max(nfb.x.out.count.ratio, nfb.z.out.count.ratio, nfb.v.out.count.ratio)

##### Non-Fastball Tuning Determination Parameters

maximum.nfb.x.z.max.minus.min <- 30

maximum.nfb.x.z.3rd.minus.1st <- 30

maximum.nfb.x.z.max.3rd.med.1st.ratio.sqrd <- 2.2

maximum.nfb.v.max.3rd.med.1st.ratio.sqrd <- 1.8

maximum.nfb.x.z.max.3rd.med.1st.ratio.sqrd.ratio <- 2

maximum.nfb.x.z.max.max.med.min.ratio.sqrd <- 2.2

maximum.nfb.v.max.max.med.min.ratio.sqrd <- 1.8

maximum.nfb.x.z.max.max.med.min.ratio.sqrd.ratio <- 2

maximum.nfb.x.z.v.max.out.ratio <- 0.09

##### Decision to Stop or Continue clustering NFB

nfb.d1 <- if(nfb.x.z.max.minus.min > maximum.nfb.x.z.max.minus.min){1} else {0}

nfb.d2 <- if(nfb.x.z.3rd.minus.1st > maximum.nfb.x.z.3rd.minus.1st){1} else {0}

nfb.d3 <- if(nfb.x.z.max.3rd.med.1st.ratio.sqrd > maximum.nfb.x.z.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb.d4 <- if(nfb.v.max.3rd.med.1st.ratio.sqrd > maximum.nfb.v.max.3rd.med.1st.ratio.sqrd){1} else {0}

nfb.d5 <- if(nfb.x.z.max.3rd.med.1st.ratio.sqrd.ratio > maximum.nfb.x.z.max.3rd.med.1st.ratio.sqrd.ratio){1} else {0}

nfb.d6 <- if(nfb.x.z.max.max.med.min.ratio.sqrd > maximum.nfb.x.z.max.max.med.min.ratio.sqrd){1} else {0}

nfb.d7 <- if(nfb.v.max.max.med.min.ratio.sqrd > maximum.nfb.v.max.max.med.min.ratio.sqrd){1} else {0}

nfb.d8 <- if(nfb.x.z.max.max.med.min.ratio.sqrd.ratio > maximum.nfb.x.z.max.max.med.min.ratio.sqrd.ratio){1} else {0}

nfb.d9 <- if(nfb.x.z.v.max.out.ratio > maximum.nfb.x.z.v.max.out.ratio){1} else {0}

# these rules will feed into the Non-Fastball user defined section above

nfb.stop.go.rule <- if(sum(nfb.d1,nfb.d2,nfb.d3,nfb.d4,nfb.d5,nfb.d6,nfb.d7,nfb.d8,nfb.d9) == 0){nfb.single.function()} else {nfb.hdbscan10.rule(nfb.data.cluster)}

## Combine Fastball and (Non-Fasball or Non-Fastball Stage 2) pitches

fb.nfb.nfb2.output <- rbind(fb.stop.go.rule, nfb.stop.go.rule)

fb.nfb.nfb2.row.count <- nrow(fb.nfb.nfb2.output)

fb.nfb.nfb2.cluster.count <- length(unique(fb.nfb.nfb2.output$Cluster))

# generate plots

remove.list <- list(" ",",")

remove.string <- paste(unlist(remove.list), collapse = "|")

jpeg(file= gsub(remove.string, "",paste(player,"FB.and.NFB.output.jpeg")))

plot(fb.nfb.nfb2.output[,c(3,4,5)], col = fb.nfb.nfb2.output[,6], main = paste((player),"clust=",(fb.nfb.nfb2.cluster.count), "count=",(fb.nfb.nfb2.row.count)))

dev.off()

# generate 3D plots

remove.list <- list(" ",",")

remove.string <- paste(unlist(remove.list), collapse = "|")

jpeg(file= gsub(remove.string, "",paste("3D",player,"FB.and.NFB.output.jpeg")))

scatterplot3d(fb.nfb.nfb2.output[,c(3,4,5)], color = fb.nfb.nfb2.output[,6], angle = 55, main = paste((player),"clust=",(fb.nfb.nfb2.cluster.count), "count=",(fb.nfb.nfb2.row.count)))

dev.off()

## FINAL OUTPUT

fb.nfb.nfb2.output

}

# run function loop

player.output <- lapply(list3, pitch.happy)

# generate a combined table of all pitchers in the loop function player.output

table <- do.call(rbind, player.output)

# export table

WriteXLS(table, ExcelFileName = "FB.and.NFB.cluster.output.full.xlsx")

## summary tables

summary.table.1 <- table %>%

group\_by(Pitcher) %>%

summarise(Cluster\_count = length(unique(Cluster)))

summary.table.1.df <- data.frame(summary.table.1)

summary.table.1.df

WriteXLS(summary.table.1.df, ExcelFileName = "Summary.table.1.xlsx")

summary.table.2 <- table %>%

group\_by(Pitcher, Cluster) %>%

summarise(Median\_x = median(break\_x),

Median\_z = median(break\_z),

Median\_velo = median(velo),

Cluster\_count = length(Cluster),

Skew\_x = round(abs(skewness(break\_x)), digits = 3),

Skew\_z = round(abs(skewness(break\_z)), digits = 3),

Skew\_velo = round(abs(skewness(velo)), digits = 3),

Kurt\_x = round(kurtosis(break\_x), digits = 3),

Kurt\_z = round(kurtosis(break\_z), digits = 3),

Kurt\_velo = round(kurtosis(velo), digits = 3)

)

summary.table.2.df <- data.frame(summary.table.2)

summary.table.2.df

WriteXLS(summary.table.2.df, ExcelFileName = "Summary.table.2.xlsx")