Grouping MLB Hitters Using Cluster Analysis and PCA

Tyler Marshall

Introduction

In this project, I will be investigating the statistics of major league baseball hitters in the 2020 season to identify the different types of hitters in the MLB. I am looking for natural groupings of hitters based on their performance during the 2020 season. There are many different statistics used to evaluate players in baseball today with some statistics focusing on power while others demonstrate contact or speed. Since there are so many statistics, it can be hard to compare different players, but I believe cluster analysis can help solve this issue. Players often excel in different areas of the game, so I am hoping to find natural clusters of hitters based on their offensive statistics. By using cluster analysis, I will be able to identify groups of hitters based on their performance this season. After doing the cluster analysis, I will use principal components analysis (PCA) to help identify differences between the clusters of hitters. The goal is that principal components analysis will help me interpret each cluster and determine what makes each cluster unique. Since I will be using 17 quantitative variables to create the clusters, a dimension-reduction technique like principal components analysis will be helpful to find differences between clusters.

Answering this question allows me to identify how hitters are different from each other, and also to determine if certain groups of hitters are more successful than others. Currently, terms like power hitter or contact hitter are often used to describe hitters, but I want to do a better job of identifying groups of hitters than this. From my combination of cluster analysis and PCA, I was able to identify eight groups of hitters. The cluster analysis finds different groups of hitters while PCA helps describe and interpret the similarities and differences between cluster groups. I included five PCs to help me investigate the different clusters. Through the cluster analysis and PCA, I was able to identify the eight different clusters as contact-oriented all around hitters, elite hitters with speed and power, quality hitters who had limited playing time in 2020, weak hitters, hitters that excelled by working counts, speedy hitters, average hitters who had limited playing time, and power hitters.

Preliminary Analysis

```
#your read-in command goes here
hittingdata <- read_csv("~/STAT 240/MarshallTData.csv")
#Remove extra blank column, add batting average and pitches per plate appearance
hittingdata <- hittingdata %>%
  select(-X20) %>%
  mutate(pitches_per_pa = pitch_count/b_total_pa) %>%
  mutate(batting_avg = b_total_hits/b_ab)
```

Before going into preliminary analysis, I want to show that baseballsavant.com allows us to scrape data from their website.

```
#Show that baseballsavant allows us to scrape data from it
paths_allowed("https://baseballsavant.mlb.com")
```

```
 baseballsavant.mlb.com
```

```
[1] TRUE
```

My data set comes from baseballsavant.com which is a website that allows you to pick different statistics to include in a .csv file (BaseballSavant 2020). Since I am focusing on Major League Baseball hitters in my project, I found data that contained many different offensive statistics that are used to evaluate a hitter's performance. Some of these statistics are more commonly known like batting average (percentage of at bats where a hitter gets a hit) and home runs while others are more advanced statistics like average exit velocity (how hard the hitter hits the ball on average) or average launch angle (the average angle at which the hitter hits the ball). The 19 variables in my data set include information about each hitter such as first and last name, year, age, but most of the variables are different statistics (all numeric variables) as mentioned above. My data set has 414 observations which represent 414 different MLB hitters that had at least 50 plate appearances during the 2020 season. A brief description of what each variable means is below.

`last_name`: last name of the player

`first_name`: first name of the player

`player_age`: age of the player during the 2020 season

`b_ab`: number of at bats player had during the 2020 season (an at bat is any time the hitter goes to the plate and gets a hit or gets out, with a few exceptions but I won't go into that here)

`b_total_pa`: number of times a hitter goes to the plate during the 2020 season (like at bats, but includes walks, hit by pitches, sacrifices)

`b_total_hits`: number of times a hitter got a hit during the 2020 season (a hit is when the batter hits the ball and the defense does not get the batter or any runners out, with a few exceptions but I won't go into that here)

`b_double`: number of doubles a hitter had during the 2020 season (a double is when a hitter gets a hit and advances to second base on the play)

`b_triple`: number of triples a hitter had during the 2020 season (a triple is when a hitter gets a hit and advances to third base on the play)

`b_home_run`: number of home runs a hitter had during the 2020 season (a home run is when a hitter gets a hit and scores on the play)

`b_strikeout`: number of strikeouts a hitter had during the 2020 season (a strikeout is when the hitter does not hit the ball before 3 strikes occur)

`b_walk`: number of walks a hitter had during the 2020 season (a walk is when the pitcher throws four balls to the hitter and the hitter goes to first base)

`slg_percent`: slugging percentage for a hitter during the 2020 season (total bases divided by at bats where a single is one total base, double is two total bases, triple is three total bases, and home run is four total bases)

`on_base_percent`: on-base percentage for a hitter during the 2020 season (hits + walks/plate appearances)

`r_total_stolen_base`: stolen bases for the player during the 2020 season (a stolen base is when a hitter is on base and advances to the next base without the batter hitting the ball)

`wOBA`: weighted on-base average for a hitter during the 2020 season (weighted on-base average is an advanced offensive statistic that takes into account various factors to illustrate overall offensive performance)

`exit_velocity_avg`: average exit velocity for a hitter during the 2020 season (exit velocity is how hard the ball leaves the bat when the hitter hits the ball)

`launch_angle_avg`: average launch angle for a hitter during the 2020 season (launch angle is the angle at which the ball leaves the bat when the hitter hits the ball)

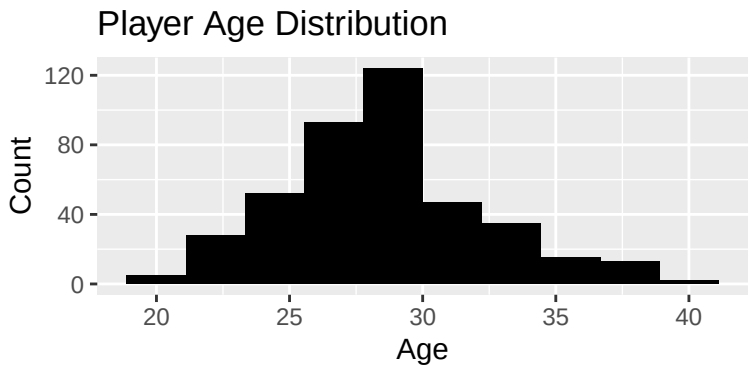
`pitches_per_pa`: pitches per plate appearance for a hitter during the 2020 season (average number of pitches a hitter sees during their plate appearance)

`batting_avg`: batting average for a hitter during the 2020 season (batting average is hits divided by at bats)

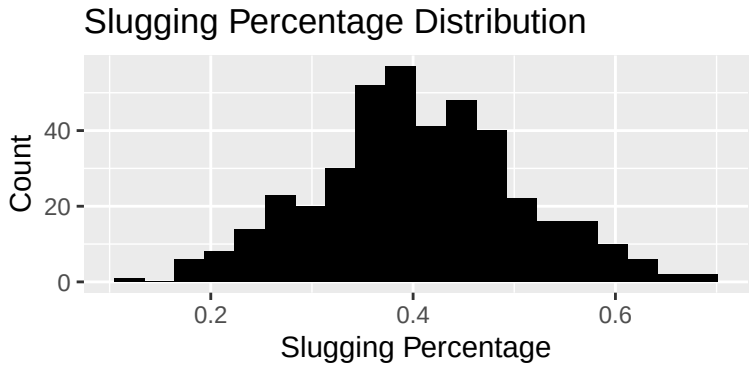
Univariate Analysis

I will start exploring the data with some univariate analysis to look at the distributions of the different variables.

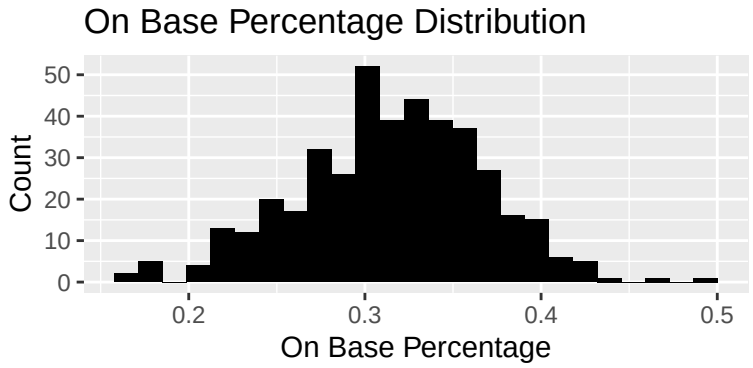
```
ggplot(data = hittingdata, aes(x = player_age)) +
  geom_histogram(bins = 10, fill = "black") +
  labs(title = "Player Age Distribution", x = "Age", y = "Count")
```



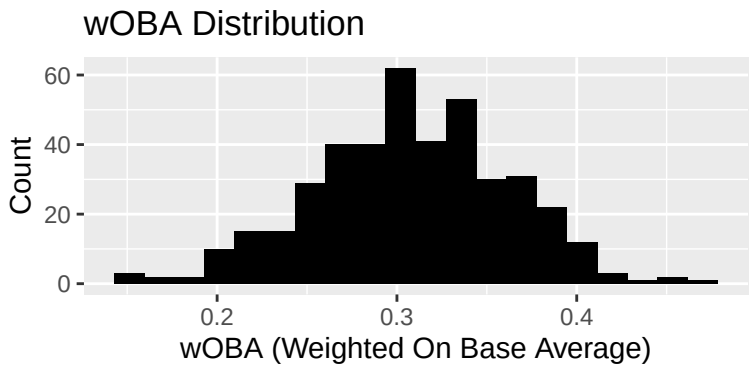

```
ggplot(data = hittingdata, aes(x = slg_percent)) +
  geom_histogram(bins = 20, fill = "black") +
  labs(title = "Slugging Percentage Distribution", x = "Slugging Percentage", y = "Count")
```



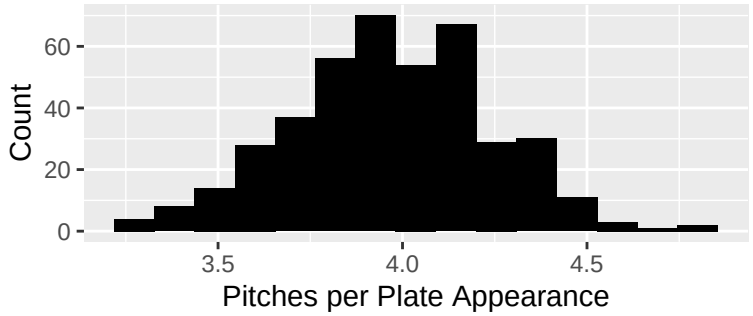
```r
ggplot(data = hittingdata, aes(x = on_base_percent)) +
  geom_histogram(bins = 25, fill = "black") +
  labs(title = "On Base Percentage Distribution", x = "On Base Percentage", y = "Count")
```




```r
ggplot(data = hittingdata, aes(x = woba)) +
  geom_histogram(bins = 20, fill = "black") +
  labs(title = "wOBA Distribution", x = "wOBA (Weighted On Base Average)", y = "Count")
```

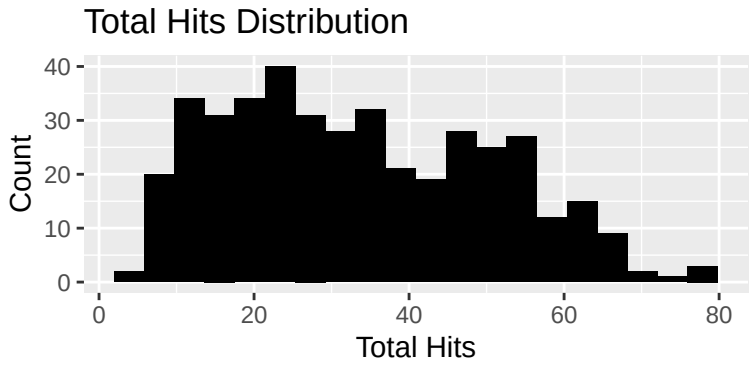



```r
ggplot(data = hittingdata, aes(x = pitches_per_pa)) +
  geom_histogram(bins = 15, fill = "black") +
  labs(title = "Pitches Per Plate Appearance Distribution",
       x = "Pitches per Plate Appearance", y = "Count")
```

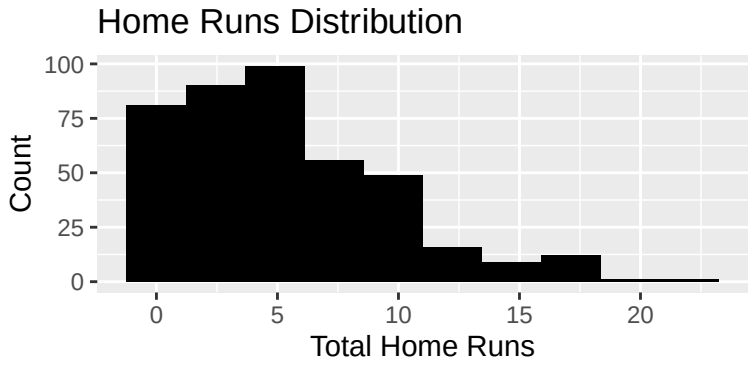


I created histograms to observe the univariate distributions for most of the numeric variables in my data set. Some of the distributions look normal while others are clearly skewed. For example, `player_age`, `slg_percent`, `on_base_percent`, `woba` (weighted on base average), and `pitches_per_pa` all seem to follow normal distributions with very little skew.

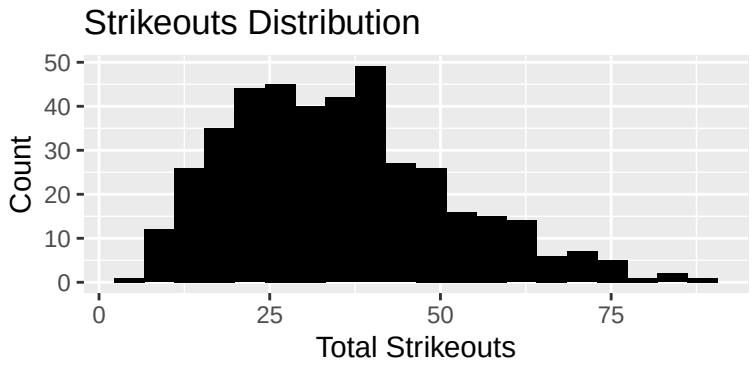
```r
ggplot(data = hittingdata, aes(x = b_total_hits)) +
  geom_histogram(bins = 20, fill = "black") +
  labs(title = "Total Hits Distribution", x = "Total Hits", y = "Count")
```



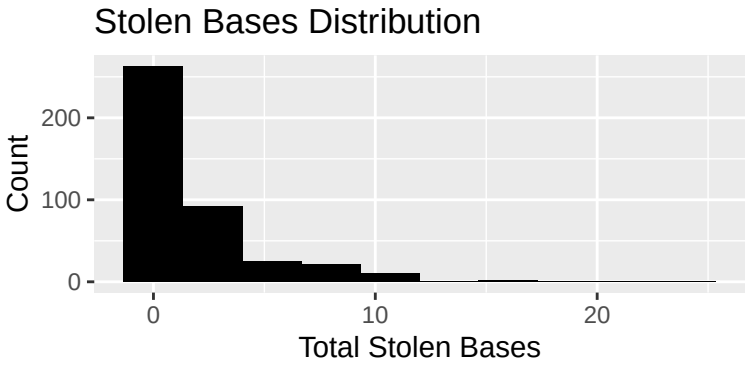

```r
ggplot(data = hittingdata, aes(x = b_home_run)) +
  geom_histogram(bins = 10, fill = "black") +
  labs(title = "Home Runs Distribution", x = "Total Home Runs", y = "Count")
```




```r
ggplot(data = hittingdata, aes(x = b_strikeout)) +
  geom_histogram(bins = 20, fill = "black") +
  labs(title = "Strikeouts Distribution", x = "Total Strikeouts", y = "Count")
```



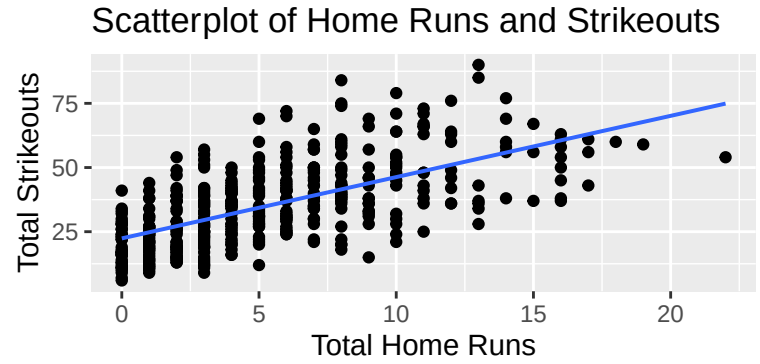

```r
ggplot(data = hittingdata, aes(x = r_total_stolen_base)) +
  geom_histogram(bins = 10, fill = "black") +
  labs(title = "Stolen Bases Distribution", x = "Total Stolen Bases", y = "Count")
```



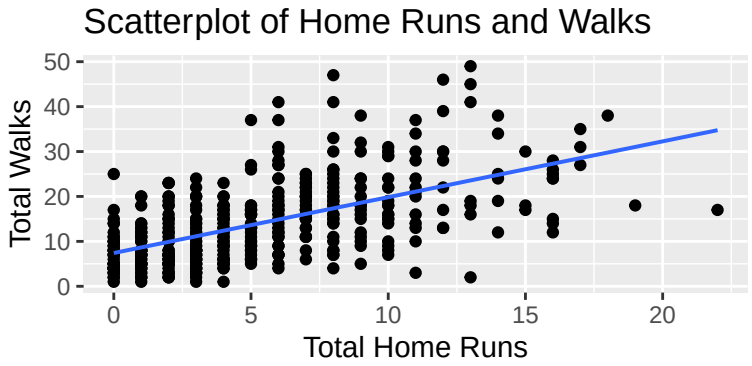
On the other hand, `b_total_hits`, `b_home_run` , `b_strikeout`, and `r_total_stolen_base` are all right skewed. Many of the statistics have a right skew which suggests some players stand out as they excel more than other players in that specific area. For example, Adalberto Mondesi stole 24 bases this season which is 8 more than any other player, and you can see this clear outlier in the stolen bases histogram. For the most part, it seems that statistics that are averages such as `woba` (weighted on base average), `on_base_percent`, `slg_percent`, and `pitches_per_pa` typically follow normal distributions while counting statistics such as `r_total_stolen_base` `b_home_run`, and `b_strikeout` are much more likely to be skewed. I did not include histograms for all of the variables that I will be using since I did not want to be too repetitive. The other variables that I did not show histograms for did not stand out, so I chose to omit them from the report. Now, that I have explored the distribution for many of my variables, I am interested to see how these variables are related to each other through bivariate analysis.

Bivariate Analysis

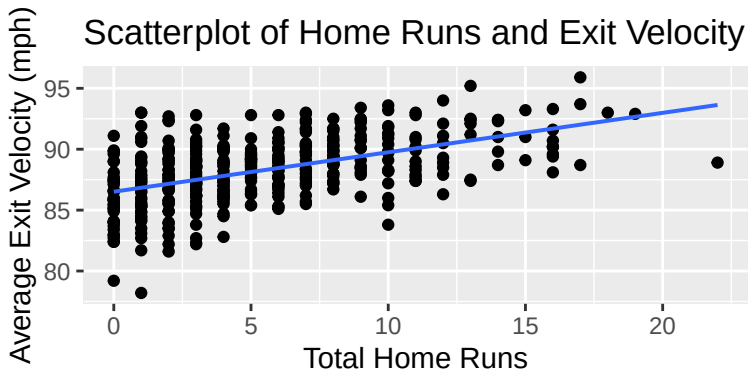
```
#Scatterplots for bivariate analysis
ggplot(data = hittingdata, aes(x = b_home_run, y = b_strikeout)) + geom_point() +
  geom_lm() +
  labs(title = "Scatterplot of Home Runs and Strikeouts",
       x = "Total Home Runs", y = "Total Strikeouts")
```




```
ggplot(data = hittingdata, aes(x = b_home_run, y = b_walk)) + geom_point() +
  geom_lm() +
  labs(title = "Scatterplot of Home Runs and Walks",
       x = "Total Home Runs", y = "Total Walks")
```

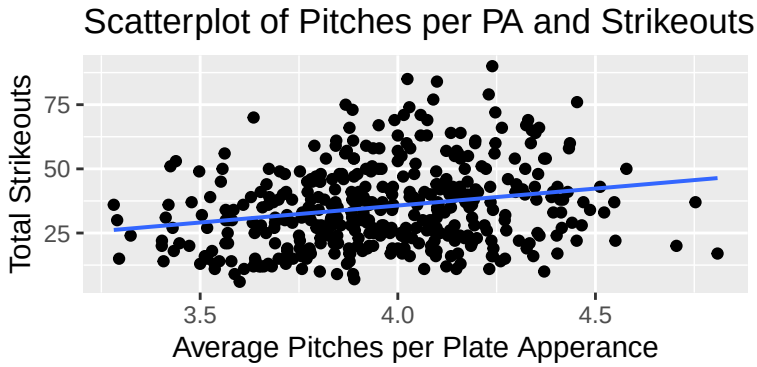



```
ggplot(data = hittingdata, aes(x = b_home_run, y = exit_velocity_avg)) + geom_point() +
  geom_lm() +
  labs(title = "Scatterplot of Home Runs and Exit Velocity",
       x = "Total Home Runs", y = "Average Exit Velocity (mph)")
```

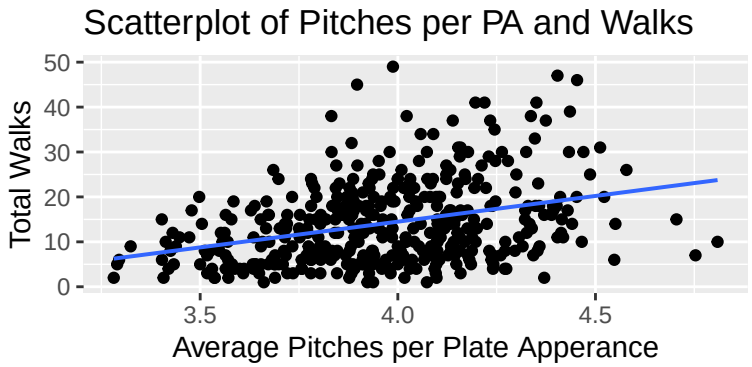


I created a variety of scatterplots to explore the relationships between quantitative variables in my data. I wanted to get a better understanding of the relationships between these variables before I perform cluster analysis and PCA. I found that home runs are positively correlated with a variety of other variables including walks, strikeouts, and average exit velocity. Walks and home runs could be positively correlated because of two different reasons. The first reason is that pitchers are more concerned about hitters that hit lots of home runs, so they also walk them more often or it could be because hitters that hit more home runs had more plate appearances, so they also walked more often. Strikeouts and home runs are also positively correlated which makes sense because people who hit more home runs typically strikeout more often which is a trade off for trying to hit for power.

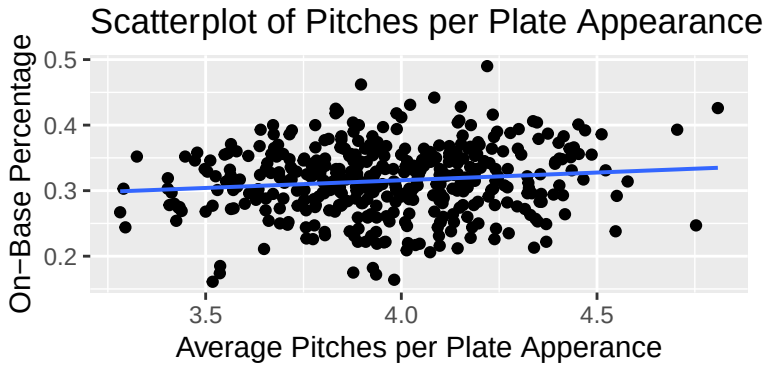
```
ggplot(data = hittingdata, aes(x = pitches_per_pa, y = b_strikeout)) + geom_point() +
  geom_lm() +
  labs(title = "Scatterplot of Pitches per PA and Strikeouts",
       x = "Average Pitches per Plate Apperance", y = "Total Strikeouts")
```



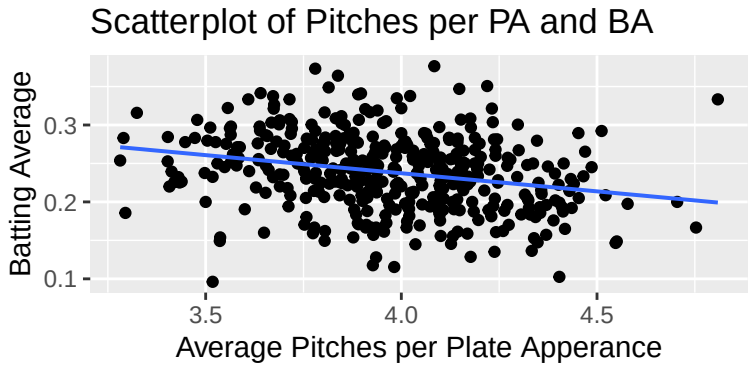

```
ggplot(data = hittingdata, aes(x = pitches_per_pa, y = b_walk)) + geom_point() +
  geom_lm() +
    labs(title = "Scatterplot of Pitches per PA and Walks",
       x = "Average Pitches per Plate Apperance", y = "Total Walks")
```




```
ggplot(data = hittingdata, aes(x = pitches_per_pa, y = on_base_percent)) + geom_point() +
  geom_lm() +
    labs(title = "Scatterplot of Pitches per Plate Appearance and OBP",
       x = "Average Pitches per Plate Apperance", y = "On-Base Percentage")
```

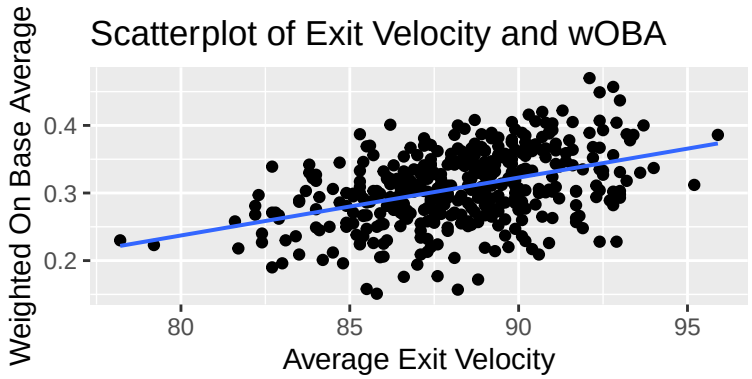



```
ggplot(data = hittingdata, aes(x = pitches_per_pa, y = batting_avg)) + geom_point() +
  geom_lm() +
    labs(title = "Scatterplot of Pitches per PA and BA",
       x = "Average Pitches per Plate Apperance", y = "Batting Average")
```

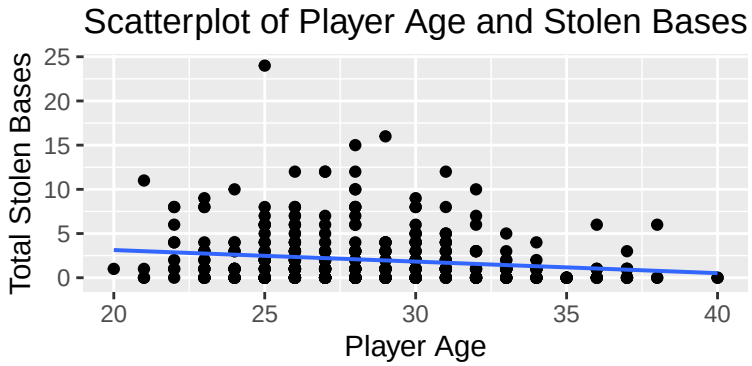


I also found that there is a positive correlation between pitchers per plate appearance and strikeouts as well as pitchers per plate appearance and walks. These both makes sense because hitters that go deeper in counts (see more pitches) are more likely to strikeout or walk compared to hitters that swing more often (don't see enough pitches to walk or strikeout as frequently). However, there is not much of a correlation between pitchers per plate appearance and on base percentage which tells us that seeing more pitches does not help you get on base more often (trade off because you are more likely to walk if you see more pitches, but are also more likely to strikeout and less likely to get a hit which can be seen in another scatterplot above).

```
ggplot(data = hittingdata, aes(x = exit_velocity_avg, y = woba)) + geom_point() +
  geom_lm() +
    labs(title = "Scatterplot of Exit Velocity and wOBA",
       x = "Average Exit Velocity", y = "Weighted On Base Average")
```




```
ggplot(data = hittingdata, aes(x = player_age, y = r_total_stolen_base)) + geom_point() +
  geom_lm() +
    labs(title = "Scatterplot of Player Age and Stolen Bases",
       x = "Player Age", y = "Total Stolen Bases")
```



There is a clear positive relationship between exit velocity average and wOBA (weighted on base average

which is a statistic to evaluate a hitter's overall performance, so a higher value is better). This tells us that better hitters typically hit the ball harder which makes sense because you are more likely to get a hit if you hit the ball harder all things else being equal. There is also a weak negative relationship between age and stolen bases which tells us that younger players typically steal more bases than older players. This makes sense because younger players are typically faster than older players as speed is an attribute that regresses as you age.

Before moving on, I want to make sure that principal components analysis is appropriate for my data set. For PCA to be appropriate, there needs to be correlation between the original variables. I have found the correlation matrix for my data below to see if this condition is met.

```
#Look at correlation matrix to see if PCA is justified
hittingnum <- hittingdata %>%
  select(-last_name, -first_name, -year, -pitch_count)
cor(hittingnum)
```

```
                   player_age       b_ab b_total_pa b_total_hits   b_double
player_age          1.00000000  0.0372493  0.0487521    0.0254457  0.0524200
b_ab                0.03724931  1.0000000  0.9930599    0.9379191  0.7398424
b_total_pa          0.04875209  0.9930599  1.0000000    0.9256431  0.7325568
b_total_hits        0.02544569  0.9379191  0.9256431    1.0000000  0.7945902
b_double            0.05241998  0.7398424  0.7325568    0.7945902  1.0000000
b_triple           -0.12906929  0.3060710  0.2996786    0.3299164  0.2245557
b_home_run          0.04723078  0.6887639  0.7022596    0.6615665  0.4760442
b_strikeout        -0.08500488  0.7330563  0.7435926    0.5707640  0.4409194
b_walk              0.09802579  0.6560271  0.7355546    0.5822397  0.4856740
slg_percent         0.00910829  0.4457538  0.4509199    0.5979634  0.5387994
on_base_percent     0.04270694  0.3796905  0.4311301    0.5644803  0.4674808
r_total_stolen_base -0.16559111  0.3108659  0.3092954    0.3200576  0.1886090
woba                0.02290094  0.4446209  0.4748384    0.6259921  0.5415529
exit_velocity_avg  -0.01334922  0.2936170  0.3062671    0.2961062  0.3210728
launch_angle_avg    0.13177821  0.0407005  0.0582473   -0.0322956  0.0692439
pitches_per_pa     -0.04262922 -0.0914051 -0.0325787   -0.1786192 -0.1084075
batting_avg        -0.03276768  0.4308730  0.4165208    0.6842394  0.5458144
                     b_triple b_home_run b_strikeout    b_walk slg_percent
player_age          -0.1290693 0.04723078  -0.0850049 0.0980258  0.00910829
b_ab                 0.3060710 0.68876386   0.7330563 0.6560271  0.44575376
b_total_pa           0.2996786 0.70225965   0.7435926 0.7355546  0.45091987
b_total_hits         0.3299164 0.66156651   0.5707640 0.5822397  0.59796342
b_double             0.2245557 0.47604415   0.4409194 0.4856740  0.53879941
b_triple             1.0000000 0.13628467   0.1987066 0.1818476  0.23290697
b_home_run           0.1362847 1.00000000   0.6355462 0.5819705  0.74860268
b_strikeout          0.1987066 0.63554616   1.0000000 0.5852958  0.28498426
b_walk               0.1818476 0.58197047   0.5852958 1.0000000  0.35848689
slg_percent          0.2329070 0.74860268   0.2849843 0.3584869  1.00000000
on_base_percent      0.1678122 0.38316571   0.1547557 0.5918411  0.66856725
r_total_stolen_base  0.3432146 0.12748363   0.2159827 0.2075327  0.08963137
woba                 0.2161007 0.62876968   0.2458894 0.5086956  0.91851590
exit_velocity_avg    0.0582697 0.51857700   0.3832662 0.3189665  0.50994934
launch_angle_avg     0.0118640 0.27294328   0.1425403 0.1214623  0.17780470
pitches_per_pa      -0.0358746 0.00847334   0.2229881 0.3396396 -0.10021920
batting_avg          0.2315127 0.32780228   0.0607836 0.2156077  0.72111409
                   on_base_percent r_total_stolen_base      woba
player_age               0.0427069          -0.1655911 0.0229009
b_ab                     0.3796905           0.3108659 0.4446209
```

```
b_total_pa               0.4311301           0.3092954 0.4748384
b_total_hits             0.5644803           0.3200576 0.6259921
b_double                 0.4674808           0.1886090 0.5415529
b_triple                 0.1678122           0.3432146 0.2161007
b_home_run               0.3831657           0.1274836 0.6287697
b_strikeout              0.1547557           0.2159827 0.2458894
b_walk                   0.5918411           0.2075327 0.5086956
slg_percent              0.6685672           0.0896314 0.9185159
on_base_percent          1.0000000           0.1611254 0.9044712
r_total_stolen_base      0.1611254           1.0000000 0.1332640
woba                     0.9044712           0.1332640 1.0000000
exit_velocity_avg        0.2587015           0.0129103 0.4227868
launch_angle_avg        -0.0374753          -0.0894841 0.0903755
pitches_per_pa           0.1207211          -0.0284550 0.0155148
batting_avg              0.7822510           0.1715151 0.8086009
                  exit_velocity_avg launch_angle_avg pitches_per_pa
player_age              -0.0133492        0.1317782    -0.04262922
b_ab                     0.2936170        0.0407005    -0.09140510
b_total_pa               0.3062671        0.0582473    -0.03257869
b_total_hits             0.2961062       -0.0322956    -0.17861921
b_double                 0.3210728        0.0692439    -0.10840750
b_triple                 0.0582697        0.0118640    -0.03587458
b_home_run               0.5185770        0.2729433     0.00847334
b_strikeout              0.3832662        0.1425403     0.22298807
b_walk                   0.3189665        0.1214623     0.33963959
slg_percent              0.5099493        0.1778047    -0.10021920
on_base_percent          0.2587015       -0.0374753     0.12072112
r_total_stolen_base      0.0129103       -0.0894841    -0.02845497
woba                     0.4227868        0.0903755     0.01551476
exit_velocity_avg        1.0000000        0.1006010     0.06881525
launch_angle_avg         0.1006010        1.0000000     0.13330489
pitches_per_pa           0.0688153        0.1333049     1.00000000
batting_avg              0.2162720       -0.1480579    -0.25986484
                  batting_avg
player_age          -0.0327677
b_ab                 0.4308730
b_total_pa           0.4165208
b_total_hits         0.6842394
b_double             0.5458144
b_triple             0.2315127
b_home_run           0.3278023
b_strikeout          0.0607836
b_walk               0.2156077
slg_percent          0.7211141
on_base_percent      0.7822510
r_total_stolen_base  0.1715151
woba                 0.8086009
exit_velocity_avg    0.2162720
launch_angle_avg    -0.1480579
pitches_per_pa      -0.2598648
batting_avg          1.0000000
```

From the correlation matrix above, we see that there is correlation between our original variables. Some of the variables are very strongly correlated with each other while others have a weak correlation, but overall

there is enough correlation between these original variables for PCA to be justified. This means we can use PCA later in the analysis. I have now explored the relationships between variables in the data through scatterplots and the correlation matrix, so I am ready to move on to explaining the methods for this project.

Methods

In my analysis, I will be using cluster analysis to create groups of MLB hitters, and then use principal components analysis to help interpret these groups. Cluster analysis allows us to find groups of observations that have similar characteristics which will allow us to find different types of hitters based on their offensive statistics. Once we have these clusters, PCA will also help to identify which statistics are most important for each cluster through the factor loadings.

For the cluster analysis, I will create clusters using both agglomerative hierarchical clustering and kmeans methods. I will compare the two methods to determine which does a better job, and then proceed with one method. For both of these methods, I need to choose a distance measure as well as decide whether or not to standardize the variables. I chose Euclidean distance since I am only using quantitative variables and it is the most common distance used measure used in cluster analysis. Since some of the variables in my data set are on different scales, I will standardize for the clustering solution.

Agglomerative hierarchical clustering works by starting each observation in its own cluster and merging observations into new clusters based on the distance between observations. This process repeats until all the observations are in one cluster. The distance between clusters is updated after each merge, but a linkage measure is needed to do this. I chose to use Ward's distance as my linkage measure since it does not have some of the issues that other linkage measures have. The results of hierarchical clustering are displayed in a dendrogram, but to determine the final clusters either a number of clusters needs to be specified or a height to cut the dendrogram at needs to be chosen. I chose to cut the dendrogram at a height of 45 which resulted in four clusters. I made this decision because I did not want too many clusters since I wanted to be able to interpret each of the clusters.

Kmeans clustering is a partitioning method, so no linkage measure is needed. Instead, we start with k different starting points and assign each observation to the cluster of the closest starting point. Points are then moved between clusters to attempt to minimize within group sum of squares. So, within group sum of squares (WGSS) helps us identify the number of clusters to use. We want to choose a number of clusters where the WGSS is low, and the trade off from adding another cluster is not worth the drop in WGSS. This is decided by looking for an elbow in the plot of WGSS for each number of clusters. The elbow in the plot occurred at 8, so I used 8 clusters for the kmeans solution.

To compare the two cluster solutions and decide which one to use for the final solution, I will be looking at the silhouette values. A silhouette value is calculated for each point by comparing how well the point fits in its own cluster compared to the next closest cluster. Silhouette values are bounded between -1 and 1 where a value close to 1 means the point fits very well in its own cluster while a value close to -1 means the point would fit better in the next closest cluster. The silhouette values for each observation can be averaged for the entire clustering solution to determine the overall strength of the clustering solution. In my analysis, the hierarchical clustering solution had an average silhouette value of 0.096 while the kmeans solution had an average silhouette value of 0.118.

The other method that I will be using in my analysis is principal components analysis (PCA). PCA is a dimension reduction technique that will help me interpret the clusters since there are many offensive statistics that impact the cluster groups. Since the original statistics are correlated, PCA is appropriate in this case. PCA creates new uncorrelated variables that are linear combinations of the original variables, and the weights of the original variables on each principal component (PC) are found through eigendecomposition of the correlation matrix (can use covariance matrix, but we want to use correlation matrix in this case since variables are on different scales). Once we have the PCs, we can see which statistics are most important to each PC. We can also look to see if our cluster analysis groups are recovered during the PCA which will help us interpret the clusters.

PCA creates 17 new variables (same as the original amount of variables), but I have to determine how many PCs to keep. I will be using Kaiser's rule to determine how many PCs to keep. This means that I will only keep PCs that have a square root of their eigenvalue greater than or equal to one. Biplots will be used to try to show how various statistics are weighted for the different PCs, and can also help us compare the clusters of hitters based on the PCs.

There was no missing data in my data set, so I did not have to worry about this. Now that I have explained the methods I will be using, I can proceed with the analysis.

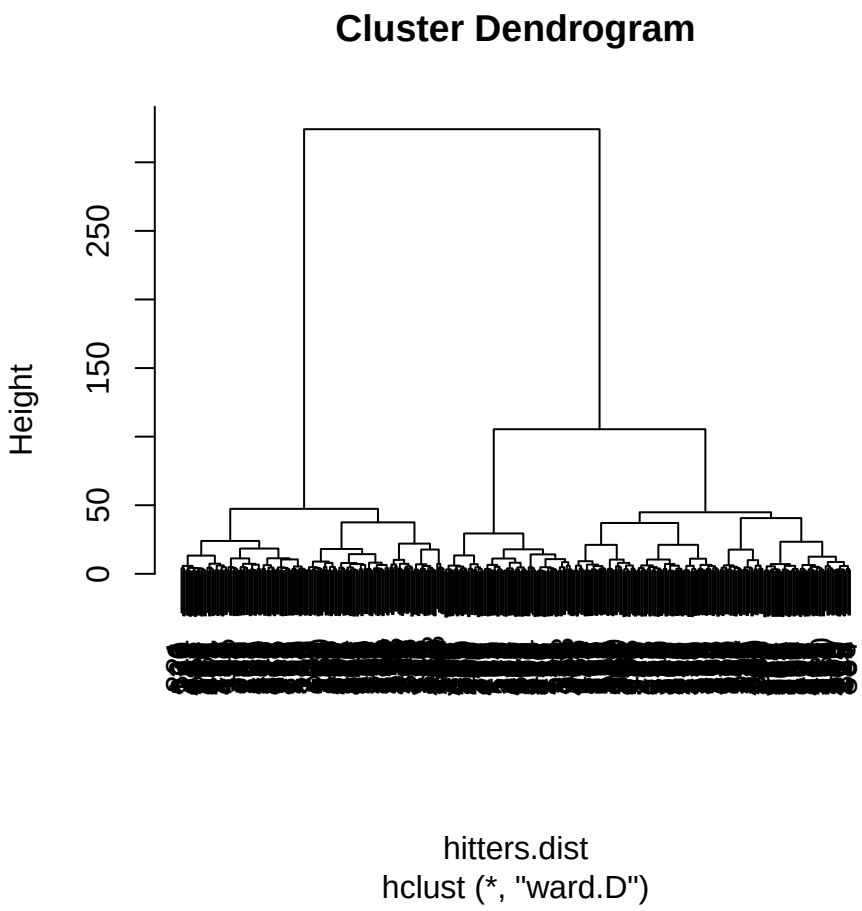
Results

First, we will run cluster analysis to find groups of hitters based on their offensive statistics. I will be trying two different cluster methods, and will compare them before choosing a final solution to proceed with. The two methods I will be using are agglomerative hierarchical clustering using Ward's Distance as the linkage measure and kmeans clustering.

Hierarchical Clustering

```
#Player names to be used later
hittingname <- hittingdata %>%
  select(last_name, first_name)
#Set up data for cluster analysis
hittingdata <- hittingdata %>%
  select(-last_name, -first_name, -year, -pitch_count)

set.seed(100)
#Hierarchical with Ward's Distance
hitters.dist <- dist(scale(hittingdata))
hittersclust <- hclust(hitters.dist, method = "ward.D")
plot(hittersclust)
```




```
#Cut the tree and look at size of clusters
wardSol <- cutree(hittersclust, h = 45)
summary(as.factor(wardSol))
```

```
  1   2   3   4
 78  80 171  85
```

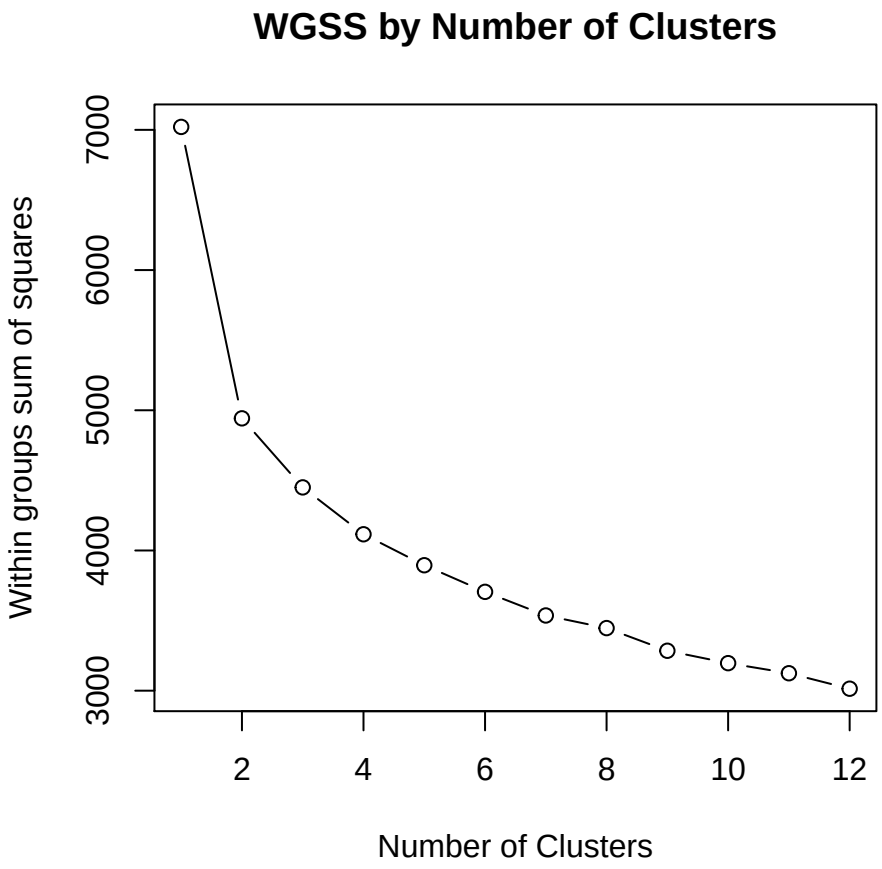
```
#Find silhouette values to assess fit
wardsil <- silhouette(wardSol, hitters.dist)
summary(wardsil)
```

```
Silhouette of 414 units in 4 clusters from silhouette.default(x = wardSol, dist = hitters.dist) :
 Cluster sizes and average silhouette widths:
       78        80       171        85
0.1561746 0.2113735 0.0610219 0.0127341
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2220  0.0172  0.0972  0.0981  0.1840  0.3840
```

From the hierarchical clustering, I chose to cut the dendrogram at a height of 45. The dendrogram is very cluttered at the bottom, so I wanted to cut it at a height where I would not end up with too many clusters. Cutting at a height of 45 gave 4 clusters of sizes 78, 80, and 171, and 85 respectively. The average silhouette value was highest for cluster 2 which had an average of 0.2114. Cluster 1 had the next highest average silhouette value at 0.1562, but the other two clusters had silhouette values less than 0.1. The overall average silhouette value was 0.098 which suggests that this clustering solution does not have strong structure.

Kmeans Clustering

```
#Set seed for reproducible analysis
set.seed(100)
wss <- rep(0, 12) #creates 12 copies of 0 to create an empty vector
for(i in 1:12){wss[i] <- sum(kmeans(scale(hittingdata[,]), centers = i)$withinss)}
plot(1:12, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares",
     main = "WGSS by Number of Clusters")
```




```
#Run kmeans clustering with 8 centers
Ksol1 <- kmeans(scale(hittingdata[,]), centers = 8)

#Find silhouette values
kmeanssil <- silhouette(Ksol1$cluster, hitters.dist)
summary(kmeanssil)
```

```
Silhouette of 414 units in 8 clusters from silhouette.default(x = Ksol1$cluster, dist = hitters.dist) :
 Cluster sizes and average silhouette widths:
       43        27        39        61        63        50        72        59
0.0860774 0.0624303 0.1126189 0.1532282 0.1096658 0.0921620 0.1344704 0.1437474
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.0537  0.0576  0.1156  0.1179  0.1776  0.3215
```

```
cluster <- Ksol1$cluster
#Add the cluster to dataset
hittingdataclust <- cbind(hittingname, cluster)
```

To determine how many clusters to include for the kmeans solution, I looked at the plot of within groups sum of squares by number of clusters. The elbow in the plot occurs at eight clusters, so I chose to use eight clusters in my kmeans solution. This gave cluster sizes of 43, 27, 39, 61, 63, 50, 72, and 59, respectively. The average cluster silhouette value was highest for cluster 4 with an average value of 0.1526, but clusters 3,5,7 and 8 also had average values greater than 0.1. The overall average silhouette value was .118 which suggests that this solution does not have great structure either, but this is a higher average silhouette value than the hierarchical solution. As a result, I will be using the kmeans solution with 8 clusters as my final solution, and will now analyze this solution further.

Before doing principal components analysis, I would like to look at the cluster centers from the kmeans solution to try to get a better understanding of what types of hitters are in each cluster.

```
#Look at cluster centers
Ksol1$centers
```

```
  player_age      b_ab b_total_pa b_total_hits  b_double  b_triple b_home_run
1  0.3687067  0.967692   0.934602     1.398203  1.470356 -0.135738  0.8440782
2 -0.0311287  1.159171   1.228137     1.272133  1.083709  2.194987  1.1474866
3 -0.3106761 -0.672595  -0.662860    -0.305218 -0.343478 -0.142431 -0.0141339
4 -0.0610784 -1.135164  -1.136610    -1.237451 -1.025622 -0.447667 -0.9785035
5  0.4866199 -0.249284  -0.225584    -0.473141 -0.228149 -0.168468 -0.0847228
6 -0.1664205  0.746710   0.682669     0.615719  0.250010  0.580474 -0.2606749
7 -0.1878993 -0.850260  -0.886409    -0.710824 -0.646472 -0.304307 -0.7878252
8 -0.1352379  1.053491   1.114186     0.730831  0.540543 -0.289247  1.1535107
  b_strikeout     b_walk slg_percent on_base_percent r_total_stolen_base
1    0.270200  0.5253634    1.135232       1.0444276          -0.2098604
2    0.638680  1.2743457    1.030393       0.9244853           1.2948127
3   -0.649370 -0.4171204    1.057650       0.9252969          -0.2567710
4   -0.787893 -0.8083920   -1.482620      -1.3233068          -0.2824591
5    0.207800 -0.0315791   -0.169351      -0.2964613          -0.3754999
6    0.306049  0.0335896   -0.282260      -0.0782766           0.9936919
7   -0.876034 -0.8009555   -0.374138      -0.2712510          -0.3058292
8    1.342449  1.1281432    0.411456       0.2861798          -0.0457672
       woba exit_velocity_avg launch_angle_avg pitches_per_pa batting_avg
1  1.173735          0.583069       -0.3488676     -0.5709984   1.3270005
2  1.056087          0.485624        0.4227376      0.1781811   0.7458382
3  1.096394          0.417120        0.0200652     -0.0252204   0.9775504
4 -1.533566         -0.870091       -0.0628967      0.2138130  -1.4363118
5 -0.231464          0.155961        0.6258993      0.7197808  -0.6027465
6 -0.207332         -0.393586       -0.4198046     -0.4679724   0.1848941
7 -0.363553         -0.365328       -0.4615366     -0.5056388   0.0417095
8  0.388608          0.589516        0.3632332      0.3752790  -0.0336077
```

Since I standardized the variables before taking the kmeans solution, the centers are standardized values and not the actual values. This means that positive values are better than average while negative values are lower than average for that particular statistic. By looking at the centers, I can make some initial interpretations of the clusters. The first cluster contains strong overall hitters that are more contact-oriented than power focused, but are very aggressive (low pitches per plate appearance). Notable hitters in this cluster include all-stars DJ Lemahieu, Freddie Freeman, Juan Soto, Anthony Rendon, Corey Seager, and Marcell Ozuna. The second cluster seems to be comprised of all-around hitters as it has high values of home runs, slugging percentage, on base percentage, but also triples and stolen bases. This suggests these hitters have a combination of speed and power that allows them to excel in different areas. Elite players in this cluster include Mike Trout, Mookie Betts, Fernando Tatis Jr., Jose Ramirez, and Trea Turner. The third cluster is comprised of hitters that did very well, but did not have as many at bats, so their counting stats (total hits, total home runs, total plate appearances) are not impressive, but their averages (slugging percentage, on-base percentage, wOBA) are very strong. A lot of these players were injured for part of the season including Aaron Judge, Bo Bichette, Ozzie Albies, and Giancarlo Stanton. The fourth cluster is comprised of hitters that had limited playing time and played poorly in that time (worse than average in almost every statistic). This cluster does not include any superstars, and instead is mainly comprised of backups like Andrew Velazquez, Eli White, Greyson Greiner, and Ehire Adrianza (players that most fans do not know much about).

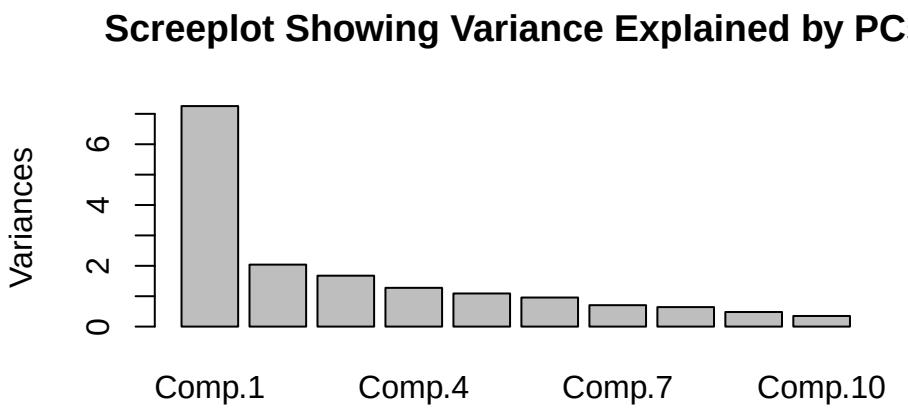
The fifth cluster contains hitters that were average in most statistics, but stood out by seeing more pitches than most (highest pitches per plate appearance). This group includes well-known players like Albert Pujols, Kris Bryant, Ryan Braun, and Matt Carpenter that did not have great seasons in 2020. The sixth cluster is comprised of speedy hitters as they stood out with a low launch angle (which means they hit more groundballs

to take advantage of their speed), lots of stolen bases and triples. This cluster includes Jose Altuve, Whit Merrifield, Jonathan Villar, and Ketel Marte who are all known for their impressive speed. Cluster 7 contains players that did not play very much but performed adequately in their limited time since batting average, on-base percentage, slugging percentage are close to average, but counting stats like plate appearances, at bats, home runs are well below average. Players in this cluster include Daniel Murphy, Yadier Molina, Tommy Pham, and Starlin Castro. Cluster 8 stands out as it has the highest values of home runs and strikeouts which suggests this cluster is comprised of power hitters that hit lots of home runs, but also strikeout frequently (common trade off of hitting a home run is increased strikeouts). This cluster includes Pete Alonso, Max Muncy, Jorge Soler, Luke Voit, and Anthony Rizzo who are known for their ability to hit for power.

Principal Components Analysis

While I do have some idea of which types of hitters are in each cluster, I am hoping the principal components analysis will give a better understanding of each cluster. I will now run the principal components analysis.

```
#Run PCA on the data
hittingpca <- princomp(hittingdata, cor = TRUE, scores = TRUE)
plot(hittingpca, main = "Screeplot Showing Variance Explained by PCs")
```




```
#Summary of PCA
summary(hittingpca)
```

```
Importance of components:
                          Comp.1   Comp.2    Comp.3    Comp.4    Comp.5
Standard deviation     2.693594 1.428037 1.2941738 1.1292549 1.0436927
Proportion of Variance 0.426791 0.119958 0.0985227 0.0750127 0.0640761
Cumulative Proportion  0.426791 0.546749 0.6452721 0.7202849 0.7843610
                          Comp.6    Comp.7    Comp.8    Comp.9   Comp.10
Standard deviation     0.9781401 0.8393446 0.8008844 0.6930595 0.5912339
Proportion of Variance 0.0562799 0.0414411 0.0377303 0.0282548 0.0205622
Cumulative Proportion  0.8406409 0.8820820 0.9198124 0.9480672 0.9686294
                         Comp.11   Comp.12    Comp.13    Comp.14     Comp.15
Standard deviation     0.497958 0.4289468 0.22271811 0.19537430 0.106987168
Proportion of Variance 0.014586 0.0108233 0.00291784 0.00224536 0.000673309
Cumulative Proportion  0.983215 0.9940386 0.99695649 0.99920185 0.999875163
                           Comp.16     Comp.17
Standard deviation     3.90225e-02 2.44842e-02
Proportion of Variance 8.95737e-05 3.52634e-05
Cumulative Proportion  9.99965e-01 1.00000e+00
```

```
#Display PCA loadings
hittingpca$loadings
```

```
Loadings:
                    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
player_age                          0.230  0.478  0.506  0.363  0.531
b_ab                 0.322 -0.238 -0.187  0.172
b_total_pa           0.328 -0.248 -0.144  0.125  0.110
b_total_hits         0.342        -0.211  0.147
b_double             0.291        -0.108  0.182                -0.169  0.105
b_triple             0.127        -0.314 -0.270 -0.223  0.519  0.251  0.638
b_home_run           0.296 -0.122  0.219  0.119 -0.245                -0.137
b_strikeout          0.237 -0.436                -0.105 -0.166
b_walk               0.268 -0.232  0.154 -0.198  0.331
slg_percent          0.290  0.294  0.203        -0.259
on_base_percent      0.264  0.335  0.121 -0.264  0.298
r_total_stolen_base  0.113        -0.404 -0.325         0.272  0.317 -0.710
woba                 0.301  0.341  0.188 -0.141
exit_velocity_avg    0.181         0.301        -0.391 -0.318  0.573
launch_angle_avg            -0.164  0.408  0.130 -0.295  0.618 -0.393 -0.175
pitches_per_pa              -0.254  0.370 -0.576  0.288                0.124
batting_avg          0.253  0.455 -0.127                       -0.106
                    Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15
player_age                  0.194
b_ab                                       -0.266  -0.162  -0.301   0.304
b_total_pa                                 -0.217  -0.131  -0.293   0.309
b_total_hits                               -0.339           0.129  -0.805
b_double            -0.662  0.152  -0.448   0.342           0.192
b_triple
b_home_run           0.458  0.120  -0.364  -0.155           0.578   0.182
b_strikeout          0.148  0.263   0.485   0.607
b_walk                     -0.605  -0.213   0.108   0.497          -0.107
slg_percent          0.168  0.258  -0.255   0.123   0.233  -0.524  -0.119
on_base_percent            -0.244   0.220   0.146  -0.492   0.200
r_total_stolen_base -0.117
woba                 0.114                  0.166  -0.262  -0.212
exit_velocity_avg   -0.411 -0.235   0.178  -0.184
launch_angle_avg    -0.227 -0.142   0.233
pitches_per_pa      -0.169  0.486          -0.313
batting_avg                 0.163   0.394  -0.195   0.570   0.250   0.282
                    Comp.16 Comp.17
player_age
b_ab                        -0.684
b_total_pa                   0.715
b_total_hits
b_double
b_triple
b_home_run
b_strikeout
b_walk
slg_percent         -0.445
on_base_percent     -0.461
r_total_stolen_base
woba                 0.754
exit_velocity_avg
launch_angle_avg
```

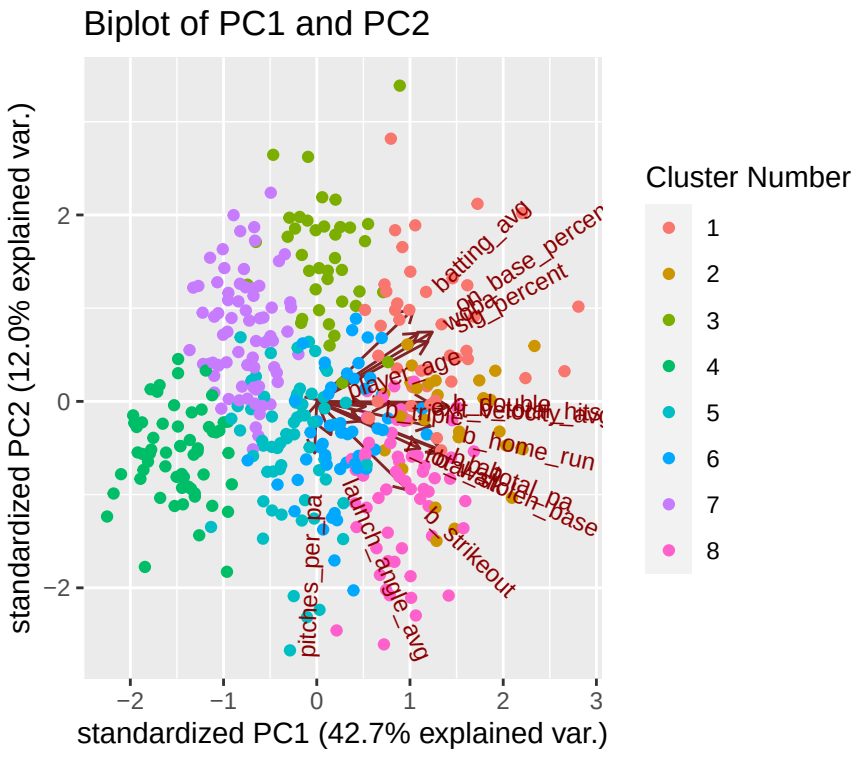
```
pitches_per_pa
batting_avg
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
SS loadings	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Proportion Var	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
Cumulative Var	0.059	0.118	0.176	0.235	0.294	0.353	0.412	0.471	0.529
	Comp.10	Comp.11	Comp.12	Comp.13	Comp.14	Comp.15	Comp.16	Comp.17	
SS loadings	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	
Proportion Var	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	
Cumulative Var	0.588	0.647	0.706	0.765	0.824	0.882	0.941	1.000	

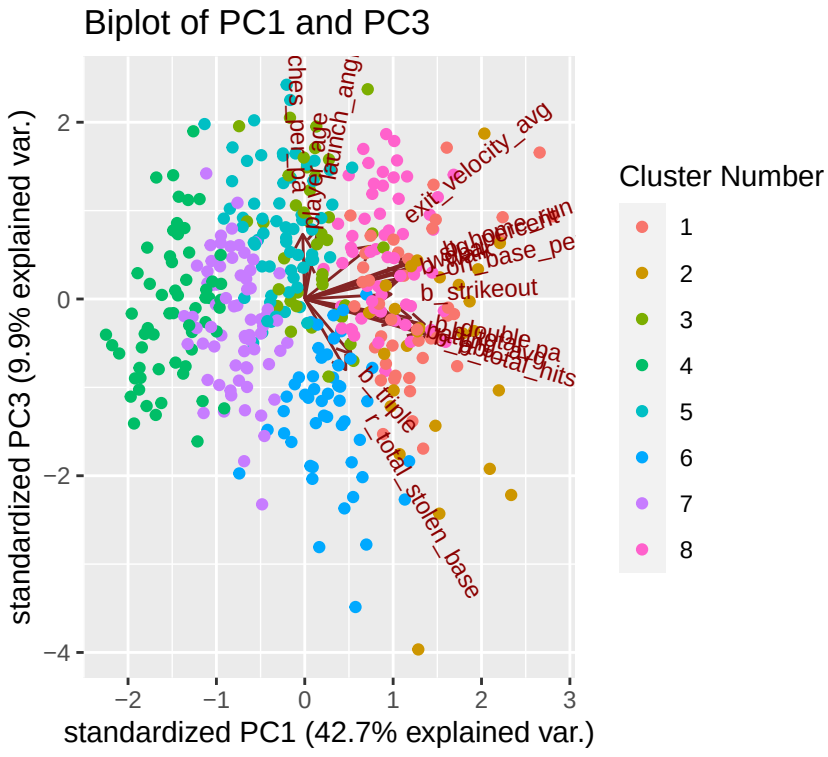
Now that I have run PCA, I need to determine how many PCs to keep. I am using Kaiser's rule to determine how many PCs to keep, so this means I will only keep PCs with a square root of the eigenvalue greater than 1. From the output above, the first 5 PC's meet this criteria, but the 6th PC does not (square root of the eigenvalue for PC6 is .978), so I will only keep the first 5 PC's. The first 5 PC's account for 78.44% of the original variation. The first PC accounts for 42.7% of the variation, the second PC accounts for 12%, the third accounts for 9.85%, the fourth accounts for 6.41% and the fifth accounts for 5.63%.

Looking at the PC loadings, we see that the first PC is a weighted average of many of the variables as they have moderate and positive loadings, but player age, average launch angle, and pitches per plate appearance do not impact PC1. PC2's highest positive loadings are on batting average, on-base percentage, and wOBA (weighted on base average) while the largest negative loading is on strikeouts. This suggests that hitters that have a high PC2 score are contact hitters as they have higher batting averages and on-base percentages than most and also do not strikeout often. PC3's largest positive loadings come from average launch angle, average exit velocity, and pitches per plate appearance while the largest negative loadings are on stolen bases and triples. This suggests that hitters that have a high PC3 score are more likely to be power hitters as they hit the ball hard and in the air, but do not have much speed. For PC4, the highest positive loading is on age while the largest negative loadings are on pitches per plate appearance, stolen bases, and triples. Hitters with high PC4 values are likely older players who are very aggressive hitters (do not see a lot of pitches), and do not have much speed at the later stage of their career. Finally, PC5 has its highest positive loadings on player age, walks, and on base percentage while it has its largest negative loadings on average exit velocity, average launch angle, and slugging percentage. This tells us that hitters with high PC5 scores are likely older players that do not hit for much power (low slugging percentage and exit velocity compared to others), but control the strike zone very well by drawing lots of walks and getting on-base at a high rate.

```
#Biplot for PC1 and PC2
ggbiplot(hittingpca, groups = as.factor(Ksol1$cluster)) %>%
  gf_labs(title = "Biplot of PC1 and PC2", color = "Cluster Number")
```




```
#Biplot for PC1 and PC3
ggbiplot(hittingpca, choices = c(1,3), groups = as.factor(Ksol1$cluster)) %>%
  gf_labs(title = "Biplot of PC1 and PC3", color = "Cluster Number")
```



By looking at these biplots, we can gain more insight into the PCs as well as our clusters. From the biplot of the first two PCs, we see that the players in clusters 1, 2, and 8 had the highest PC1 values while players

in clusters 4 and 7 had the lowest PC1 values. Since PC1 is a weighted average of many of the hitting statistics, this tells us that clusters 1,2, and 8 contain better hitters while clusters 4 and 7 contain weaker hitters. Players in clusters 3, 5, and 6 have PC1 values near zero which indicates that they are average in many areas, but likely stand out in a few particular statistics. We can also see that the players in cluster 1 excel in batting average, on base percentage, and slugging percentage which suggests these are all around hitters that have slightly more emphasis on contact than power. Players in cluster 2 have high values of home runs, average exit velocity, doubles, triples, and wOBA (weighted on base average) which further suggests these are all-around hitters that have a combination of power and speed. Cluster 3 stands out as having high values of batting average, on base percentage and slugging percentage, but not very impressive levels of home runs, hits, plate appearances, so these hitters did well in limited playing time. Cluster 4 stands out as having low values of batting average, on base percentage, and slugging percentage which tells us these hitters did not have good seasons as they struggled offensively. From the biplot of PC1 and PC3, we see that cluster 5 stands out as having high values of pitches per plate appearance, so these hitters are known for the ability to work the count and have long at bats. Cluster 6 also stands out in this biplot as the players in this cluster have high values of triples and stolen bases, so these are speedy hitters. Cluster 7 has low or average values of many statistics which shows that these players did not play that much, but performed close to average in their limited playing time. Finally, from the first biplot, we see that Cluster 8 stands out with its high values of home runs and strikeouts, so these hitters are known for their power. The other biplots were looked at, but do not provide much value about the clusters so they were not included in the final output. The first two biplots were most helpful in interpreting the clusters that were created during the cluster analysis, so these were the only two included in the report.

Conclusion

After trying two different clustering methods, my final result was a kmeans clustering solution that included 8 clusters. The clusters sizes were 43, 27, 39, 61, 63, 50, 72, and 59, respectively. I was able to interpret the eight clusters as eight different types of hitters. The first cluster is comprised of all-around hitters that focus on contact more than power. The second cluster is comprised of elite hitters that excel with their speed and power. The third cluster is comprised of quality hitters that had limited playing time during the 2020 season while cluster four contained the weakest hitters during the 2020 season as they did not play well during their limited time. Cluster five is comprised of hitters that see the most pitches per plate appearances, so these hitters can be called grinders. Cluster six is comprised of fast hitters that excel with their tremendous speed. Cluster seven contains average hitters that had limited playing time during the 2020 season while cluster eight contains power hitters.

My PCA solution involved 5 PCs after using Kaiser's rule, and these 5 PCs helped me interpret the different clusters. The biplots were particularly helpful in demonstrating which clusters excelled in certain areas which allowed me to refine my interpretation of the cluster groups. The combination of cluster analysis and PCA allowed me to achieve my goal of finding and describing clusters of hitters for the 2020 season. There were not many limitations for this project since I was able to get data on 414 MLB hitters for an entire season, and only included hitters who had at least 50 plate appearances. However, the 2020 season was only 60 games due to COVID-19, which is 102 games shorter than the typical 162 game MLB season. Hitting statistics are more variable in a 60 game season compared to a 162 season due to a smaller sample size (fewer games and fewer plate appearances in shortened season), so this could have potentially impacted the clusters. I would be interested to complete a similar project after the 2021 season to see if the clusters are similar. I would also like to see if the players would be in the same cluster as this year or if there would be a lot of variation between years. Another application of this would be to look at each team and see what types of hitters they have in their lineup. It would be interesting to see what types of hitters are on the teams that have more success. There are lots of areas in baseball where cluster analysis and PCA could be very useful like this project, so I am interested to explore some of these in the future.

Citations

Willman, Daren. "Baseball Savant: Trending MLB Players, Statcast and Visualizations." Baseballsavant.com, MLB Advanced Media, 2020, baseballsavant.mlb.com/. (This source will be referred to as BaseballSavant 2020 during in-line citations)