Statcast 2023 Principal Component Analysis

Lance Brady

Introduction

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of a dataset. It does this by creating new variables, called principal components, that are linear combinations of the original variables. These principal components capture the underlying patterns in the data, allowing us to simplify the dataset and identify the most important variables. In this analysis, we will use PCA to analyze Statcast data from qualified hitters in 2023. We will start with 19 variables (listed below), and end up with 3 components that capture most of the variation in the data. We will identify the most important components in the data and create a Principal Component Score Plot to visualize the data.

Variables Used

The variables used in this analysis are:

1. player_age: Player Age
2. pa: Plate Appearances
3. k_percent: Strikeout percentage
4. bb_percent: Walk percentage
5. babip: Batting Average on Balls in Play
6. b_intent_walk: Intentional Walks
7. xba: Expected Batting Average 8: xslg: Expected Slugging Percentage
8. xwoba: Expected Weighted On-Base Average
9. xobp: Expected On-Base Percentage
10. xiso: Expected Isolated Power
11. xbacon: Expected Batting Average on Contact
12. sweet_spot_percent: Sweet Spot Percentage
13. barrel_batted_rate: Barrel Batted Rate
14. hard_hit_percent: Hard Hit Percentage
15. avg_best_speed: EV50
16. avg_hyper_speed: Adjusted EV
17. whiff_percent: Whiff Percentage
18. swing_percent: Swing Percentage

Load in Libraries

```
library(car)
library(corrplot)
```

Load in Data

```
## Load the data
statcast2023_1 <- read.csv("statcastbatting2023.csv")
```

```r
## Remove the first three columns (player_id, player_name, and year) as they are
## not needed
statcast2023 <- statcast2023_1[, -c(1, 2, 3)]
```

Check Dimensions and Variable Names

```r
dim(statcast2023)
```

```
## [1] 134  19
```

```r
names(statcast2023)
```

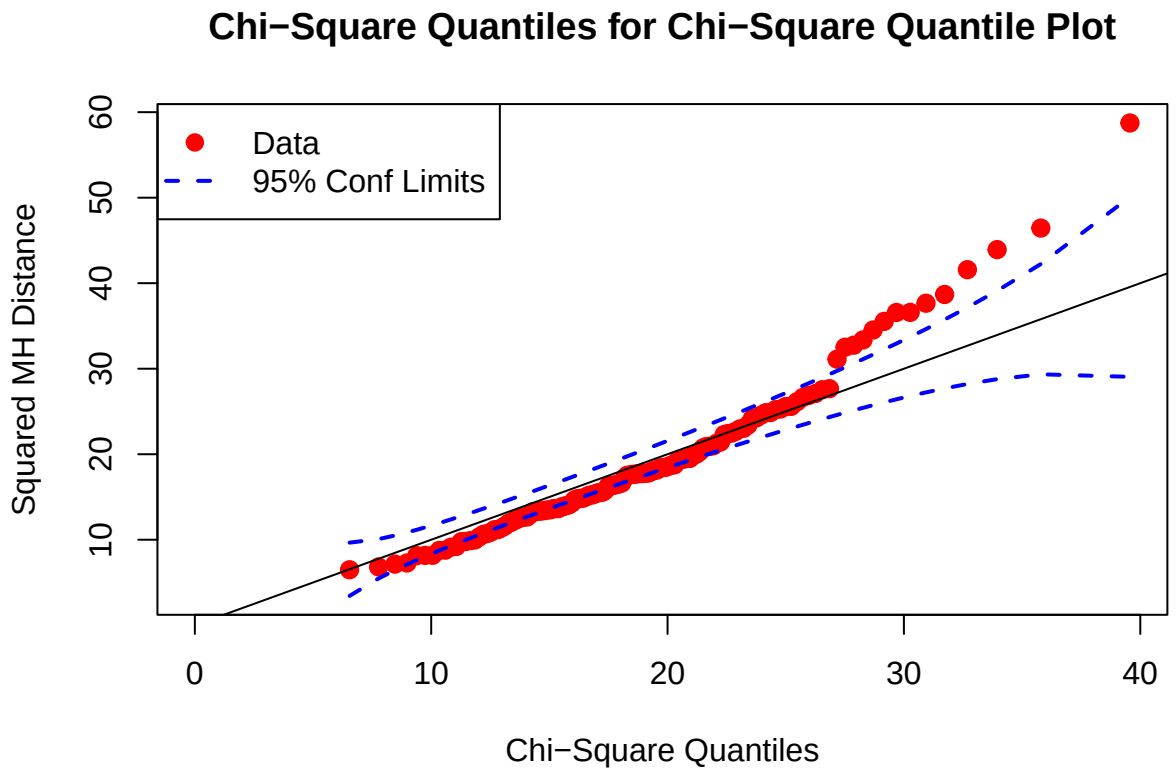
```
##  [1] "player_age"         "pa"                 "k_percent"
##  [4] "bb_percent"         "babip"              "b_intent_walk"
##  [7] "xba"                "xslg"               "xwoba"
## [10] "xobp"               "xiso"               "xbacon"
## [13] "sweet_spot_percent" "barrel_batted_rate" "hard_hit_percent"
## [16] "avg_best_speed"     "avg_hyper_speed"    "whiff_percent"
## [19] "swing_percent"
```

We have 134 players and 19 variables. This is enough data for Principal Component Analysis.

Check for Multivariate Normality

First, we will want to do a chi-square quantile plot for multivariate normality. This is not needed for PCA, but it is a good practice to check for multivariate normality.

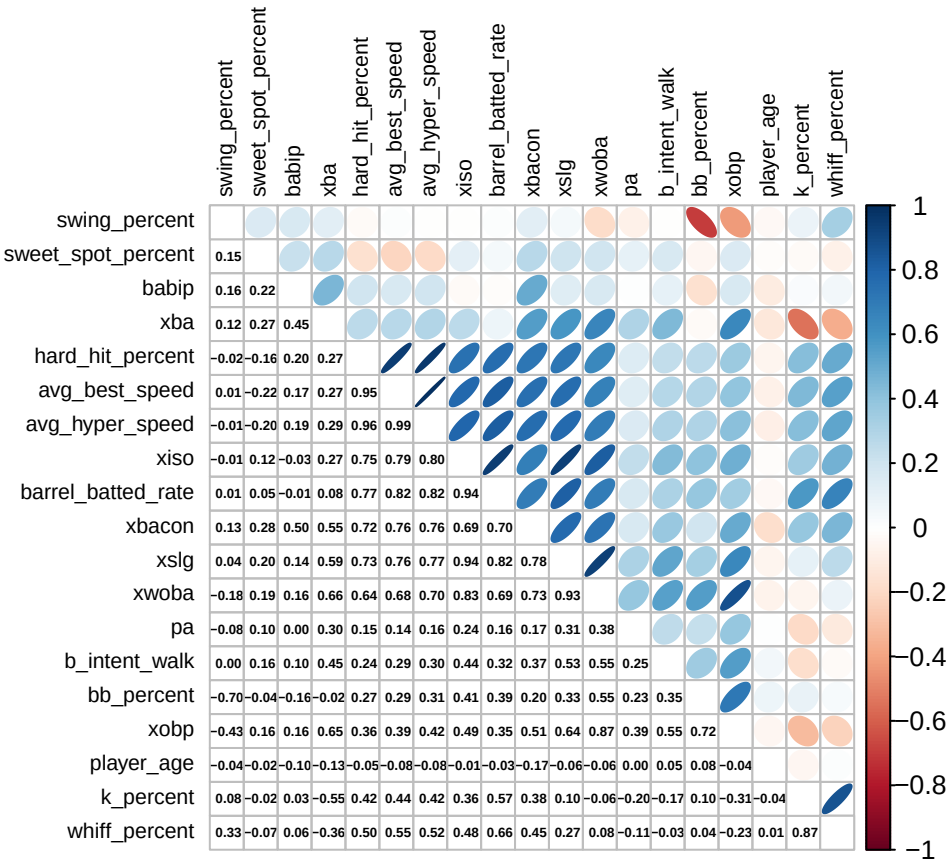
```r
source("http://www.reuningscherer.net/multivariate/R/CSQPlot.r.txt")
CSQPlot(statcast2023)
```



It is not multivariate normal, as a large proportion falls outside the 95% confidence curves. However, it is close, so we will still do Parallel Analysis later.

Next, we will want to compute the correlation matrix for the data. This will allow us to see how the variables are related to each other.

```
corrplot.mixed(cor(statcast2023), lower.col = "black", upper = "ellipse",
tl.col = "black", number.cex = .45, order = "hclust",
tl.pos = "lt", tl.cex = .7)
```



The correlation matrix shows that there are some variables that are highly correlated with each other. This is a good sign for PCA, as it means that there are underlying patterns in the data that can be captured by the principal components.

Next, we can perform Principle Components Analysis on the data.

Perform PCA

Method #1:

We will say somewhat arbitrarily that we want to capture 80% of the variance in the data. This is a common threshold for PCA.

```
library(PerformanceAnalytics)
## Perform PCA
pc1 <- princomp(statcast2023, cor = TRUE)
summary(pc1)
```

```
## Importance of components:
```

```
##                           Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     2.8206861 1.8432023 1.4785272 1.1101748 1.05362179
## Proportion of Variance 0.4187511 0.1788103 0.1150549 0.0648678 0.05842731
## Cumulative Proportion  0.4187511 0.5975613 0.7126162 0.7774840 0.83591131
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.96722914 0.87532666 0.78464119 0.53637771 0.46279756
## Proportion of Variance 0.04923854 0.04032615 0.03240325 0.01514216 0.01127271
## Cumulative Proportion  0.88514985 0.92547600 0.95787925 0.97302141 0.98429412
##                           Comp.11     Comp.12     Comp.13     Comp.14
## Standard deviation     0.370541004 0.252019096 0.203462361 0.165251348
## Proportion of Variance 0.007226349 0.003342822 0.002178786 0.001437264
## Cumulative Proportion  0.991520474 0.994863296 0.997042082 0.998479346
##                           Comp.15      Comp.16      Comp.17      Comp.18
## Standard deviation     0.15009621 0.0622308907 0.0446634940 2.157119e-02
## Proportion of Variance 0.00118573 0.0002038255 0.0001049909 2.449034e-05
## Cumulative Proportion  0.99966508 0.9998689012 0.9999738921 9.999984e-01
##                            Comp.19
## Standard deviation     5.543790e-03
## Proportion of Variance 1.617559e-06
## Cumulative Proportion  1.000000e+00
```

We can see that the first 5 principal components capture 80% of the variance in the data.

Method #2:

We would like to retain components where the eigenvalues are greater than 1. This will occur when the standard deviations are greater than 1.

```
pc1$sdev
```

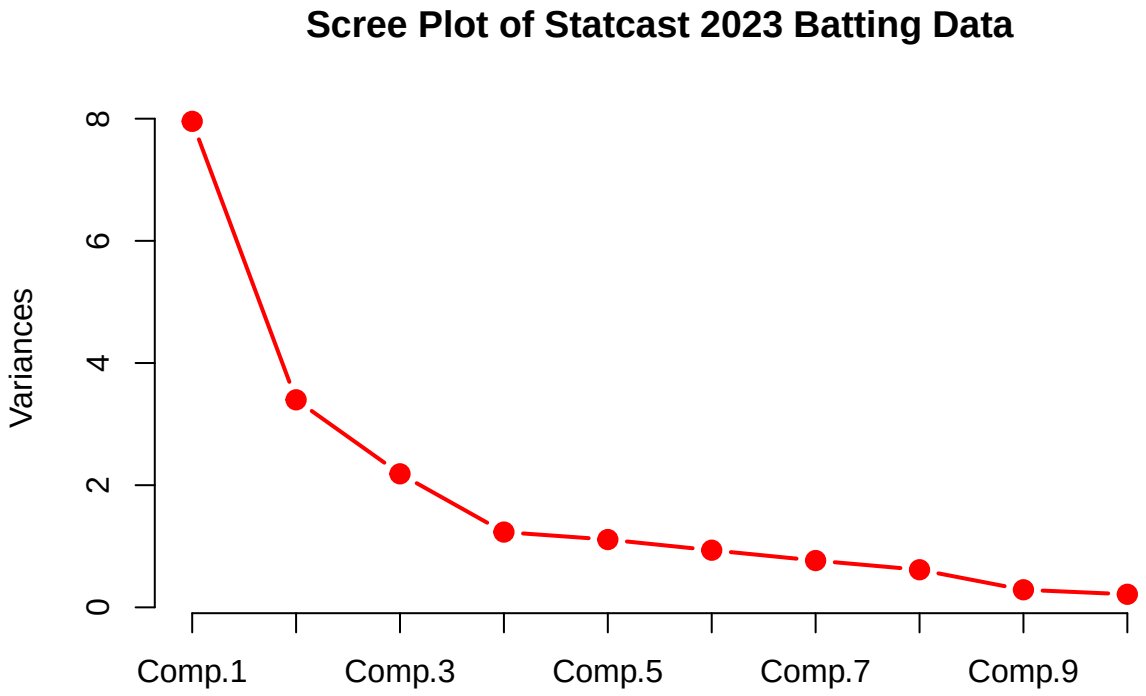
```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 2.82068615 1.84320234 1.47852722 1.11017482 1.05362179 0.96722914 0.87532666
##     Comp.8     Comp.9    Comp.10    Comp.11    Comp.12    Comp.13    Comp.14
## 0.78464119 0.53637771 0.46279756 0.37054100 0.25201910 0.20346236 0.16525135
##    Comp.15    Comp.16    Comp.17    Comp.18    Comp.19
## 0.15009621 0.06223089 0.04466349 0.02157119 0.00554379
```

Using this method, we again will retain 5 components, as they all have standard deviations greater than 1.

Method #3:

We will make a scree plot and keep components up to where we see an elbow in the plot.

```
screeplot(pc1, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
main = "Scree Plot of Statcast 2023 Batting Data")
```



Although there is no clear elbow, I concluded that there is a slight elbow at component 4, so this method suggests retaining 3 components.

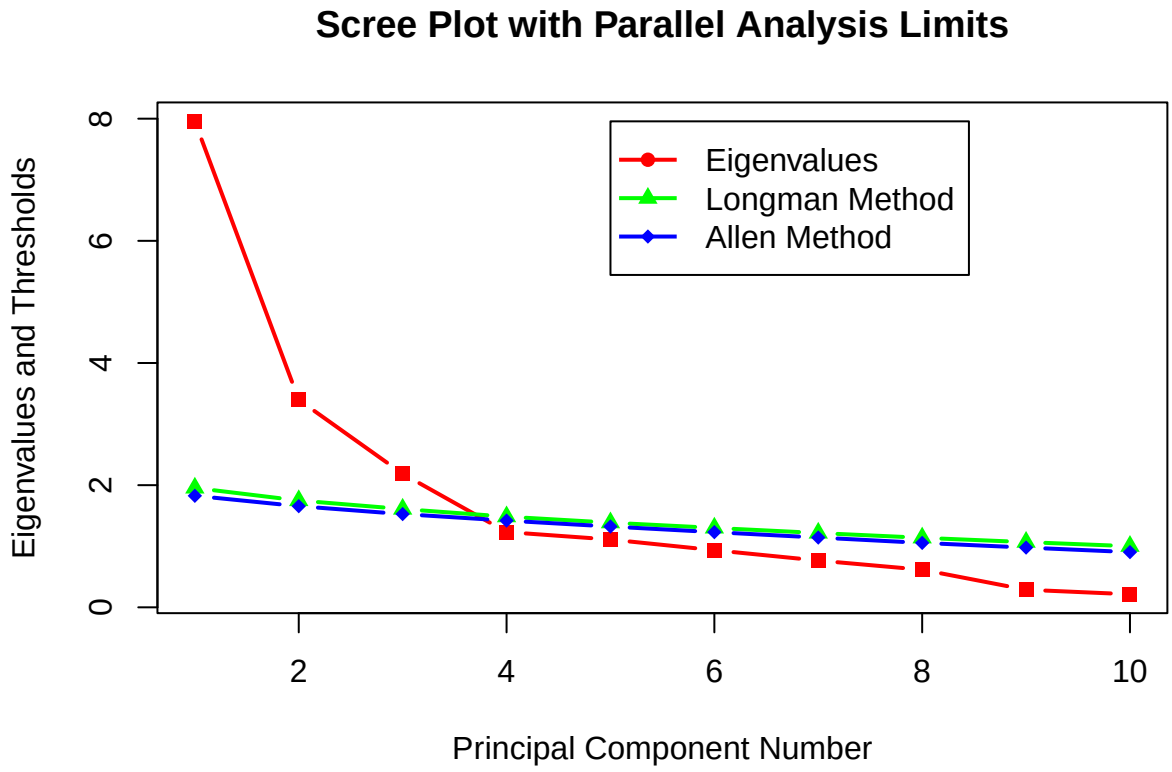
Method #4:

Although the chi-square quantile plot did not indicate precise multivariate normality, we will do Parallel Analysis anyway as it looked close to normal.

```
#get the function online
source("http://www.reuningscherer.net/multivariate/R/parallel.r.txt")

#make the parallel analysis plot using the parallelplot function
parallelplot(pc1)
```

```
##    pcompnum  longman     allen
## 1         1 1.954492 1.8259968
## 2         2 1.748913 1.6559411
## 3         3 1.609462 1.5268299
## 4         4 1.484502 1.4197554
## 5         5 1.385937 1.3230150
## 6         6 1.300056 1.2309200
## 7         7 1.216744 1.1425288
## 8         8 1.138131 1.0549274
## 9         9 1.067589 0.9785564
## 10       10 1.000883 0.9041626
```



This method also indicates that we should use just 3 components, as the eigenvalues are below the predicted eigenvalues for only the first 3 components.

Loadings of Retained PCA Components

I will somewhat arbitrarily decide to use just 3 components, as two out of four methods of PCA suggested this, and it will lead to easier analysis. Let's print the loadings of the PCA components so that we can see how the variables are related to the components.

```
print(pc1$loadings, cutoff = 0, digits = 2)
```

```
##
## Loadings:
##                    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## player_age          0.03   0.00   0.15   0.33   0.39   0.80   0.13   0.21
## pa                 -0.11  -0.21   0.04   0.13   0.24  -0.29   0.87  -0.13
## k_percent          -0.11   0.46   0.06   0.16  -0.29   0.02   0.12  -0.10
## bb_percent         -0.16  -0.14   0.50   0.10  -0.28   0.07   0.02  -0.17
## babip              -0.08  -0.06  -0.41  -0.21  -0.42   0.43   0.27  -0.20
## b_intent_walk      -0.18  -0.23  -0.01   0.22   0.17   0.09  -0.28  -0.78
## xba                -0.16  -0.37  -0.33  -0.16   0.09   0.05  -0.06   0.12
## xslg               -0.33  -0.08  -0.07   0.10   0.14  -0.09  -0.12   0.16
## xwoba              -0.32  -0.21   0.03   0.04   0.01  -0.04  -0.07   0.14
## xobp               -0.23  -0.36   0.15  -0.03  -0.14   0.05  -0.03  -0.02
## xiso               -0.32   0.06   0.06   0.19   0.13  -0.13  -0.12   0.14
## xbacon             -0.31   0.02  -0.24   0.00  -0.23   0.08   0.04   0.00
## sweet_spot_percent -0.03  -0.15  -0.28   0.63  -0.37  -0.07  -0.01   0.28
## barrel_batted_rate -0.31   0.18   0.08   0.17   0.04  -0.11  -0.05   0.07
## hard_hit_percent   -0.31   0.14   0.01  -0.25   0.07   0.09   0.07   0.12
## avg_best_speed     -0.32   0.15   0.01  -0.24   0.10   0.05   0.01   0.03
```

```
## avg_hyper_speed    -0.32   0.14   0.01  -0.25   0.09   0.06   0.02   0.03
## whiff_percent      -0.16   0.44  -0.04   0.18  -0.05   0.03   0.09  -0.21
## swing_percent       0.02   0.17  -0.52   0.16   0.38  -0.11  -0.06  -0.16
##                    Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15
## player_age          0.02   0.04    0.09    0.05    0.01    0.01    0.01
## pa                 -0.08   0.00    0.05    0.01    0.01   -0.01    0.00
## k_percent          -0.09   0.09    0.28    0.37   -0.21   -0.06    0.20
## bb_percent          0.33   0.15   -0.29    0.15   -0.17   -0.10   -0.55
## babip               0.03  -0.49   -0.25    0.05    0.01   -0.03    0.01
## b_intent_walk      -0.37   0.02    0.06   -0.03    0.01    0.02    0.05
## xba                 0.08   0.15    0.28   -0.21   -0.02    0.02   -0.37
## xslg                0.00  -0.28    0.06   -0.04   -0.23   -0.23   -0.08
## xwoba               0.21  -0.08    0.00    0.00   -0.06   -0.03    0.30
## xobp                0.36   0.19   -0.07    0.03    0.12    0.17    0.59
## xiso               -0.04  -0.40   -0.05    0.04   -0.26   -0.28    0.06
## xbacon              0.07   0.27    0.54    0.16   -0.14    0.11   -0.13
## sweet_spot_percent -0.30   0.25   -0.29   -0.08    0.18   -0.08   -0.02
## barrel_batted_rate  0.03  -0.34    0.04    0.10    0.44    0.66   -0.22
## hard_hit_percent   -0.34   0.26   -0.38   -0.19   -0.48    0.41    0.05
## avg_best_speed     -0.10   0.19   -0.07    0.12    0.44   -0.40   -0.01
## avg_hyper_speed    -0.16   0.16   -0.17    0.10    0.31   -0.18   -0.04
## whiff_percent       0.35   0.06    0.03   -0.74    0.06   -0.10    0.03
## swing_percent       0.44   0.19   -0.34    0.37   -0.10    0.05   -0.04
##                    Comp.16 Comp.17 Comp.18 Comp.19
## player_age          0.01    0.00    0.00    0.00
## pa                  0.00    0.00    0.00    0.00
## k_percent           0.00   -0.56    0.03    0.01
## bb_percent         -0.03   -0.01    0.03    0.00
## babip              -0.03    0.00    0.00    0.00
## b_intent_walk      -0.01   -0.01    0.05    0.00
## xba                -0.01   -0.55   -0.04   -0.26
## xslg                0.03   -0.07   -0.22    0.74
## xwoba              -0.04   -0.01    0.82    0.00
## xobp                0.04   -0.09   -0.45    0.00
## xiso                0.03    0.13   -0.24   -0.62
## xbacon              0.03    0.58   -0.04   -0.01
## sweet_spot_percent -0.01    0.00    0.00    0.00
## barrel_batted_rate -0.07   -0.07   -0.02    0.01
## hard_hit_percent   -0.15    0.00   -0.01    0.00
## avg_best_speed     -0.62    0.03   -0.04    0.00
## avg_hyper_speed     0.76   -0.01    0.05    0.00
## whiff_percent       0.03    0.00    0.00    0.00
## swing_percent      -0.01    0.00    0.00    0.00
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings      1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
## Proportion Var   0.05   0.05   0.05   0.05   0.05   0.05   0.05   0.05   0.05
## Cumulative Var   0.05   0.11   0.16   0.21   0.26   0.32   0.37   0.42   0.47
##                Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17
## SS loadings       1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
## Proportion Var    0.05    0.05    0.05    0.05    0.05    0.05    0.05    0.05
## Cumulative Var    0.53    0.58    0.63    0.68    0.74    0.79    0.84    0.89
##                Comp.18 Comp.19
## SS loadings       1.00    1.00
```

```
## Proportion Var    0.05    0.05
## Cumulative Var    0.95    1.00
```

For each component, we want to choose the variables with the highest absolute coefficients, as they are contributing most to that component.

Component #1: The "Quality of Contact" Component

The largest absolute coefficients for this component are Expected Slugging (xslg), Expected Weighted On Base Average (xwoba), Expected Isolated Power (xiso), EV50, and Adjusted EV. This component seems to be related to a player's quality of contact. By definition of these variables, they are all calculated from things like exit velocity and launch angle over a season, which are related to how well a batter is hitting balls they make contact with. Most at-bats end in some sort of contact, so it makes sense that this would be the first (and most important) component.

Component #2: The "Get on Base" Component

The largest absolute coefficients for this component are Strikeout Percentage (k_percent), Whiff Percentage (whiff_percent), Expected Batting Average (xba), and Expected On-Base Percentage (xobp). This component seems to be related to a player's ability to get on base. That is, it is related to how often a player swing and miss (and from that how often they strike out), and how often they get hits or walks. This is the second most important component.

Component #3: The "Plate Approach" Component

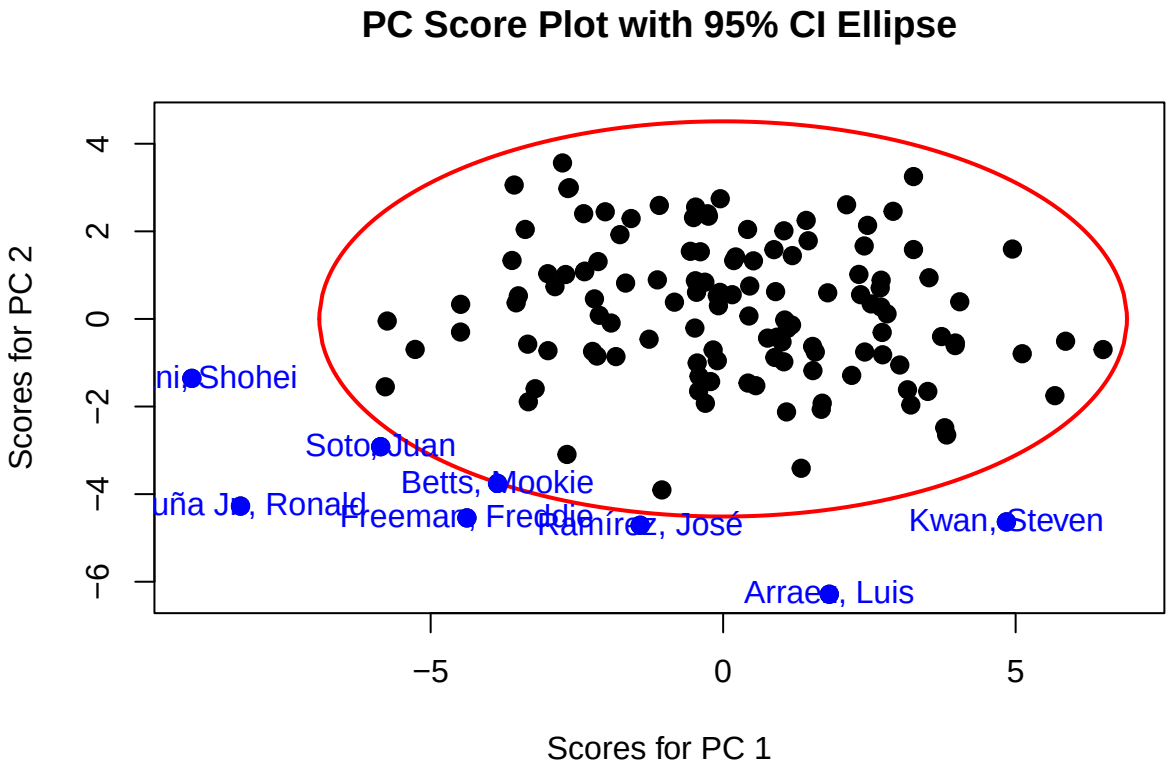
The largest absolute coefficients for this component were Swing Percentage (swing_percent), and Walk Percentage (bb_percent). This component seems to encompass how a batter approaches the plate. Do they swing a lot or try to draw walks? This is the third most important component.

Principal Component Score Plot

Next, we will want to create a Principal Component Score Plot for the first two components.

```r
source("http://reuningscherer.net/multivariate/r/ciscoreplot.R.txt")

# Using the first two components
statcast2023_2 = statcast2023_1[, -c(2, 3)]
ciscoreplot(pc1, c(1, 2), statcast2023_2[, 1])
```



This plot shows us the scores for the first two principal components, along with a 95% CI ellipse. Based on the signs of our coefficients, we need to keep in mind that stats like Strikeout Percentage and Whiff Percentage should have an opposite sign to stats like Expected Slugging (xslg), Expected Weighted On Base Average (xwoba), as better hitters will strikeout less and get better contact. With that said, we have multiple outliers to the data, which are in general, players who make terrific contact and get on base well. These include Juan Soto, Mookie Betts, Ronald Acuña Jr., Jose Ramírez, Freddie Freeman, and Shohei Ohtani. Then there is Luis Arraez, who does not make incredible contact with the ball, however he makes contact at a high rate, does not whiff a lot, and rarely strikes out. Steven Kwan is similar, although not as good in either category.

Conclusion

I found that Statcast data from qualified hitters in 2023 can be broken down using Principal Component Analysis into 3 main components, The "Quality of Contact" Component, The "Get on Base" Component, and The "Plate Approach" Component. These components seem to capture most of the data, allowing for us to simplify from the 19 original variables we started with. We also found that there are some players who are outliers in the data, such as Juan Soto, Mookie Betts, Ronald Acuña Jr., Jose Ramírez, Freddie Freeman, and Shohei Ohtani. These players are all known for their ability to make exceptionally good contact and get on base.