Batted Ball System Analysis

AJ Fong, ajfong25@gmail.com

2024-12-23

Importing Data

```
project_data <- read.csv("C:/Users/ajfon/OneDrive/Desktop/battedBallData.csv")
head(project_data)
```

```
##   batter pitcher     hittype   speed_A   vangle_A   speed_B   vangle_B
## 1    393     405 ground_ball 110.98757   4.194081 103.84257   3.164307
## 2    366     405 ground_ball  60.09840 -54.652102  28.09220 -28.324082
## 3    448     518  line_drive 102.75760  11.751851  97.84600  11.658800
## 4    140     518    fly_ball  61.95209  33.488154  59.38974  32.798274
## 5    521     518  line_drive 116.69086  22.700762 111.01456  23.164572
## 6    401     518 ground_ball 105.98183 -10.497794  78.50893  -7.471214
```

Similarities and Differences in Systems

Generating summaries based on hit types to easily compare similarities and differences in data collection between the systems.

```
ground_ball_data <- project_data[project_data$hittype == "ground_ball", ]
fly_ball_data <- project_data[project_data$hittype == "fly_ball", ]
line_drive_data <- project_data[project_data$hittype == "line_drive", ]
popup_data <- project_data[project_data$hittype == "popup", ]
unknown_data <- project_data[project_data$hittype == "U", ]

summary(ground_ball_data[4:7])
```

```
##     speed_A          vangle_A          speed_B          vangle_B
##  Min.   : 26.46   Min.   :-91.899   Min.   :  5.152   Min.   :-85.0909
##  1st Qu.: 76.54   1st Qu.:-18.170   1st Qu.: 53.672   1st Qu.:-14.4210
##  Median : 88.68   Median : -7.908   Median : 67.582   Median : -5.9707
##  Mean   : 86.15   Mean   :-10.769   Mean   : 68.224   Mean   : -7.1193
##  3rd Qu.: 97.44   3rd Qu.: -0.217   3rd Qu.: 86.110   3rd Qu.:  0.2621
##  Max.   :121.85   Max.   : 45.633   Max.   :114.403   Max.   : 81.7767
##  NA's   :4591     NA's   :4591      NA's   :681       NA's   :681
```

```
summary(fly_ball_data[4:7])
```

```
##     speed_A          vangle_A         speed_B          vangle_B
##  Min.   : 50.79   Min.   :-41.24   Min.   : 49.48   Min.   : 0.5774
##  1st Qu.: 84.02   1st Qu.: 28.22   1st Qu.: 80.44   1st Qu.:28.3642
##  Median : 90.83   Median : 34.19   Median : 87.13   Median :34.4900
##  Mean   : 90.06   Mean   : 35.40   Mean   : 86.31   Mean   :35.7039
##  3rd Qu.: 96.86   3rd Qu.: 41.94   3rd Qu.: 92.94   3rd Qu.:42.2749
##  Max.   :115.16   Max.   : 65.63   Max.   :111.41   Max.   :83.8714
##  NA's   :276      NA's   :276      NA's   :236      NA's   :236
```

```
summary(line_drive_data[4:7])
```

```
##     speed_A          vangle_A         speed_B          vangle_B
##  Min.   : 35.44   Min.   :-22.51   Min.   : 30.20   Min.   :-8.465
##  1st Qu.: 85.79   1st Qu.: 10.97   1st Qu.: 82.20   1st Qu.:10.309
##  Median : 95.24   Median : 15.52   Median : 91.03   Median :15.050
##  Mean   : 92.84   Mean   : 15.82   Mean   : 88.74   Mean   :15.355
##  3rd Qu.:101.49   3rd Qu.: 20.37   3rd Qu.: 96.99   3rd Qu.:20.105
##  Max.   :118.81   Max.   : 53.46   Max.   :114.40   Max.   :54.640
##  NA's   :258      NA's   :258      NA's   :130      NA's   :130
```

```
summary(popup_data[4:7])
```

```
##     speed_A          vangle_A        speed_B          vangle_B
##  Min.   : 35.06   Min.   :18.78   Min.   : 25.26   Min.   :17.53
##  1st Qu.: 66.04   1st Qu.:50.93   1st Qu.: 64.16   1st Qu.:56.69
##  Median : 74.41   Median :57.94   Median : 72.75   Median :63.86
##  Mean   : 73.93   Mean   :56.12   Mean   : 71.73   Mean   :63.11
##  3rd Qu.: 82.45   3rd Qu.:62.71   3rd Qu.: 79.93   3rd Qu.:71.07
##  Max.   :105.91   Max.   :78.46   Max.   :105.00   Max.   :90.90
##  NA's   :2446     NA's   :2446    NA's   :355      NA's   :355
```

```
summary(unknown_data[4:7])
```

```
##     speed_A        vangle_A       speed_B         vangle_B
##  Min.   :92.8   Min.   :24.4   Min.   :60.52   Min.   :-22.8623
##  1st Qu.:92.8   1st Qu.:24.4   1st Qu.:67.82   1st Qu.:-11.1260
##  Median :92.8   Median :24.4   Median :75.12   Median :  0.6103
##  Mean   :92.8   Mean   :24.4   Mean   :75.12   Mean   :  0.6103
##  3rd Qu.:92.8   3rd Qu.:24.4   3rd Qu.:82.42   3rd Qu.: 12.3467
##  Max.   :92.8   Max.   :24.4   Max.   :89.72   Max.   : 24.0830
##  NA's   :1      NA's   :1
```

System A is more accurate because it tracks the ball closer to the moment of impact. This results in higher exit velocity measurements compared to System B. This is similar to how pitchers' throwing speeds used to be measured at home plate but are now measured at the point of release.

Organizing Data Based on Batter ID

```r
# Initialize an empty list
ordered_project_data <- list()

# Loop through unique batter IDs
for (batter_id in unique(project_data$batter)) {
  # Add each batter's data to the list, using the batter ID as the key
  ordered_project_data[[as.character(batter_id)]] <- project_data[project_data$batter == batter_id, ]
}
```

I created a for loop that made a list containing subsets of data for each batter ID. This list simplifies retrieving a specific player's data for each hit type, exit velocity, launch angle, and system type. It can also be easily replicated for each pitcher ID.

Example Usage

```r
head(ordered_project_data$'1')
```

```
##      batter pitcher     hittype   speed_A  vangle_A  speed_B  vangle_B
## 5926      1     396  line_drive  97.44065  12.43256 92.76435 12.347392
## 6232      1     220  line_drive 102.26700  23.37575 96.30890 24.090033
## 6241      1     220 ground_ball  91.88626 -14.89327 56.39896 -8.235504
## 6261      1     484 ground_ball        NA        NA 26.01221 39.359478
## 6271      1     115  line_drive  93.47066  19.27579 89.04055 19.323983
## 6278      1     396  line_drive  70.13823  14.14474 67.51310 13.394507
```

True Exit Velocity and Launch Angle

```r
batter_data <- function(player_wanted, hittype_wanted) {
  # Ensure the player ID is within the valid range
  if (player_wanted < min(project_data$batter, na.rm = TRUE) ||
      player_wanted > max(project_data$batter, na.rm = TRUE)) {
    stop("Player ID not found")
  }

  # Filter data for the specified player
  datasets <- list(
    Ground_Ball = ground_ball_data[ground_ball_data$batter == player_wanted, ],
    Fly_Ball = fly_ball_data[fly_ball_data$batter == player_wanted, ],
    Line_Drive = line_drive_data[line_drive_data$batter == player_wanted, ],
    Popup = popup_data[popup_data$batter == player_wanted, ],
    Unknown = unknown_data[unknown_data$batter == player_wanted, ],
    General = project_data[project_data$batter == player_wanted, ]
  )

  # Stop if no data is found
  if (nrow(datasets$General) == 0) {
    stop("No data found for the specified player ID")
  }
```

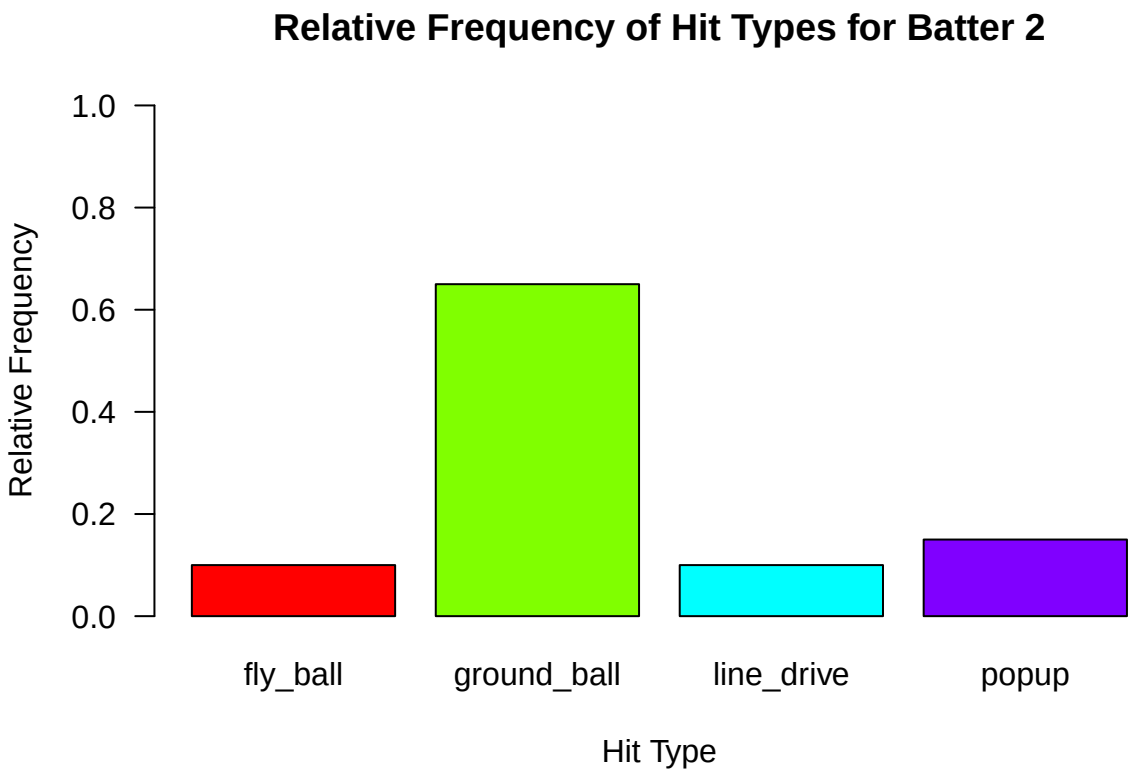
```r
  # Helper function to calculate means
  calculate_means <- function(data, column) {
    sapply(datasets, function(df) mean(df[[column]], na.rm = TRUE))
  }

  # Calculate values for each metric
  batter_info <- data.frame(
    "System A EV" = calculate_means(datasets, "speed_A"),
    "System A LA" = calculate_means(datasets, "vangle_A"),
    "System B EV" = calculate_means(datasets, "speed_B"),
    "System B LA" = calculate_means(datasets, "vangle_B"),
    "True Average EV" = (calculate_means(datasets, "speed_A") +
                           calculate_means(datasets, "speed_B")) / 2,
    "True Average LA" = (calculate_means(datasets, "vangle_A") +
                           calculate_means(datasets, "vangle_B")) / 2
  )

  # Set row names for categories
  rownames(batter_info) <- names(datasets)

  # Return all data or a specific hit type
  if (hittype_wanted == "Show All") {
    return(batter_info)
  } else {
    return(batter_info[hittype_wanted, , drop = FALSE])
  }
}


#player_wanted <- as.numeric(readline("Insert Player ID: "))
#hittype_wanted <- readline("What hit type do you want to see (Ground Ball, Fly Ball, Line Drive, Popup

# Setting player_wanted value = 1 & hittype_wanted = Show All
player_data <- batter_data(2, "Show All")
print(player_data)
```

```
##             System.A.EV System.A.LA System.B.EV System.B.LA True.Average.EV
## Ground_Ball    79.91901  -15.674329    52.95021  0.06786976        66.43461
## Fly_Ball       79.90761   39.932247    76.40806 39.98556244        78.15783
## Line_Drive     94.03789   13.668876    90.63347 12.91145073        92.33568
## Popup          75.04284   61.686476    66.68908 64.70309675        70.86596
## Unknown             NaN         NaN         NaN         NaN             NaN
## General        81.07292    4.614494    61.12515 15.03928117        71.09904
##             True.Average.LA
## Ground_Ball       -7.803230
## Fly_Ball          39.958905
## Line_Drive        13.290163
## Popup             63.194786
## Unknown                 NaN
## General            9.826888
```

Frequency of Hit Type

Creating a relative frequency table to show the proportions of each hit type for a specific batter.

```r
plot_batter_data <- function(player_wanted) {

  # Filter the data for the player_id
  player_id <- project_data[project_data$batter == player_wanted, ]

  # Count the frequency of each hit type for the player
  hit_counts <- table(player_id$hittype)

  # Calculate the total number of hits for the player
  total_hits <- sum(hit_counts)

  # Calculate relative frequency for each hit type
  relative_frequency <- hit_counts / total_hits

  # Plot the relative frequencies
  barplot(relative_frequency,
          names.arg = names(hit_counts),
          col = rainbow(length(hit_counts)),
          main = paste("Relative Frequency of Hit Types for Batter",
                       player_wanted),
          xlab = "Hit Type",
          ylab = "Relative Frequency",
          las = 1,
          ylim = c(0, 1))

  return(relative_frequency)
}

# Example usage
plot_batter_data(2)
```



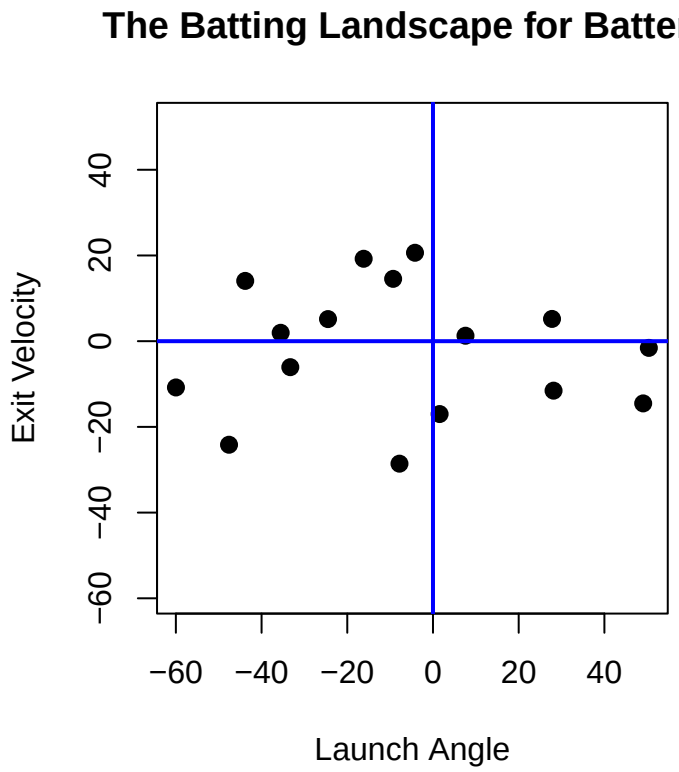

```
##
##    fly_ball ground_ball  line_drive       popup
##        0.10        0.65        0.10        0.15
```

Scatterplot of Exit Velocity and Launch Angle

Creating a scatterplot for an individual batter relative to the True Average LA/EV of the entire dataset.

```
# Calculating True Avg EV/LA
true_average_EV <- ((mean(project_data$speed_A, na.rm = T) +
                       mean(project_data$speed_B, na.rm = T)) / 2)
true_average_LA <- ((mean(project_data$vangle_A, na.rm = T) +
                       mean(project_data$vangle_B, na.rm = T)) / 2)

scatterplot_batter_data <- function(player_wanted){

  # Filter the data for the player_id
  player_id <- project_data[project_data$batter == player_wanted, ]

  par(pty = "s")
  # Creating scatterplot based on LA - True Avg LA & EV - True Avg EV
  scatterplot_batter <- plot((player_id$vangle_A - true_average_LA),
                             (player_id$speed_A - true_average_EV),
          xlab = "Launch Angle",
          ylab = "Exit Velocity",
```

```
          main = paste("The Batting Landscape for Batter", player_wanted),
          cex = 1.25,
          asp = 1,
          pch = 16
  )
  abline(v = 0, col = "blue", lwd = 2)
  abline(h = 0, col = "blue", lwd = 2)
}

#Example Usage
scatterplot_batter_data(2)
```



In the scatterplot, the upper-right quadrant is ideal because it represents data points where the batter excels in both LA and EV. A higher-than-average LA and EV tends to correlate to more extra base hits. (Note: Data points are plotted from System A only)

Conclusion/Findings

Through this project, I discovered that System A reports a higher average Exit Velocity (EV) compared to System B. This difference happens because System A measures EV closer to the point of contact with the bat. This allows for more precise data. The largest discrepancies between the two systems occur with ground balls for average EV and with popups for average Launch Angle (LA).

The significant difference in ground ball EV is likely due to the difficulty in tracking when the ball is contacted. Ground balls often have high levels of unusual spin which makes them harder to track. For popups, the

largest variation in LA are due to their short horizontal travel and extreme angles. Tracking systems rely on both horizontal and vertical movement to calculate LA. When there is minimal horizontal travel, accuracy diminishes. Conversely, more "squared-up" hits, fly balls and line drives, have minimal variability because their trajectories and velocities are easier for sensors to track.

Another observation is the difference in the number of missing values (NA) between the systems. System A recorded significantly more NA values than System B. This is likely because System A focuses on being more accurate than complete. This probably leads to System A discarding data points that do not meet its stricter quality standards. This trend is particularly evident for popups and ground balls, which are more challenging to measure.

To enhance user accessibility, I included code that allows user input. This design enables individuals to interact with the data directly by entering a player's ID to access statistics. Prioritizing usability allows for the data and code to be practical and shareable.

For the calculation of "True Average Exit Velocity" and "True Average Launch Angle," I used a weighted average with a 50-50 ratio between System A and System B. While System A is more accurate, its higher number of NA values requires implementing data from System B to provide a more comprehensive estimate. System B's values help fill gaps and provide a reasonable "ballpark" figure, especially when System A does not return a result.