Analyzing MLB Player Positions & Offensive Statistics

Britton Green

In this analysis, I chose to review MLB hitting data for the 2023 – 2024 season (Baseball Reference, 2025) filtered by player positions and hitting statistics to gain insight into player values from a different perspective. First, I loaded all of the required libraries I would need for this analysis, including tidyverse, ggplot2 and janitor in RStudio. I imported the data using the read.csv function and created a dataframe called "mlb_batters_original". The raw imported file has 741 observations/rows and 36 variables/columns, such as Player, Age, Team, HR (Home Runs), BA (Batting Average), among many others.

Data Structure:

```
> str(mlb_batters_original)
'data.frame':   741 obs. of  36 variables:
 $ Rk               : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Player           : chr  "Jarren Duran*" "Shohei Ohtani*" "Gunnar Henderson*" "Marcus Semien" ...
 $ Age              : int  27 29 23 33 25 24 32 25 22 29 ...
 $ Team             : chr  "BOS" "LAD" "BAL" "TEX" ...
 $ League           : chr  "AL" "NL" "AL" "AL" ...
 $ WAR              : num  8.7 9.2 9.1 4.1 7.9 9.4 10.8 6.2 5.2 2.6 ...
 $ GP               : int  160 159 159 159 157 161 158 159 160 162 ...
 $ PA               : int  735 731 719 718 713 709 704 697 696 695 ...
 $ AB               : int  671 636 630 650 576 636 559 616 618 608 ...
 $ R                : int  111 134 118 101 128 125 122 98 105 91 ...
 $ H                : int  191 197 177 154 166 211 180 199 160 146 ...
 $ X2B              : int  48 38 31 27 31 45 36 44 36 31 ...
 $ X3B              : int  14 7 7 2 4 11 1 1 10 0 ...
 $ HR               : int  21 54 37 23 41 32 58 30 25 34 ...
 $ RBI              : int  75 130 92 74 109 109 144 103 76 88 ...
 $ SB               : int  34 59 21 8 7 31 10 2 67 3 ...
 $ CS               : int  7 4 4 3 4 12 0 2 16 0 ...
 $ BB               : int  54 81 78 64 129 57 133 72 69 70 ...
 $ SO               : int  160 162 159 105 119 106 171 96 218 172 ...
 $ BA               : num  0.285 0.31 0.281 0.237 0.288 0.332 0.322 0.323 0.259 0.24 ...
 $ OBP              : num  0.342 0.39 0.364 0.308 0.419 0.389 0.458 0.396 0.339 0.329 ...
 $ SLG              : num  0.492 0.646 0.529 0.391 0.569 0.588 0.701 0.544 0.471 0.459 ...
 $ OPS              : num  0.834 1.036 0.893 0.699 0.989 ...
 $ rOBA             : num  0.373 0.449 0.385 0.31 0.424 0.416 0.484 0.399 0.367 0.341 ...
 $ TB               : int  330 411 333 254 328 374 392 335 291 279 ...
 $ GIDP             : int  6 7 2 9 10 4 22 16 12 14 ...
 $ HBP              : int  6 6 7 3 4 8 9 5 6 13 ...
 $ SH               : int  1 0 0 0 0 0 0 0 2 0 ...
 $ SF               : int  3 5 4 1 4 8 2 4 1 4 ...
 $ IBB              : int  1 10 1 2 2 9 20 12 2 4 ...
 $ Pos              : chr  "8" "DH" "6,DH" "4" ...
 $ Awards           : chr  "ASMVP-8" "ASMVP-1SS" "ASMVP-4" "AS" ...
 $ Player.additional: chr  "duranja01" "ohtansh01" "hendegu01" "semiema01" ...
 $ X                : logi  NA NA NA NA NA NA ...
 $ X.1              : logi  NA NA NA NA NA NA ...
 $ X.2              : logi  NA NA NA NA NA NA ...
```

Using the "summarise" function, we can see that 27 variables are numeric, 6 are character and 3 are logical (because they were imported as NA values). Additionally, we are able to gain additional insight into the numeric variables, such as minimum, median, mean, max, 1st quartile and 3rd quartile as well as the number if "NA" values for the logical class variables, which need to be cleansed before further analysis.

Summary:

```
> summary(mlb_batters_original)
       Rk          Player               Age            Team              League               WAR             GP               PA              AB              R                H
 Min.   :  1   Length:741         Min.   :20.00   Length:741         Length:741         Min.   :-2.0   Min.   :  1.00   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0   Min.   :  0.00
 1st Qu.:186   Class :character   1st Qu.:25.00   Class :character   Class :character   1st Qu.:-0.1   1st Qu.: 15.00   1st Qu.: 37.0   1st Qu.: 33.0   1st Qu.:  3.0   1st Qu.:  6.00
 Median :371   Mode  :character   Median :27.00   Mode  :character   Mode  :character   Median : 0.1   Median : 62.00   Median :188.0   Median :172.0   Median : 19.0   Median : 37.00
 Mean   :371                      Mean   :27.97                                         Mean   : 0.8   Mean   : 68.48   Mean   :246.2   Mean   :220.9   Mean   : 28.8   Mean   : 53.74
 3rd Qu.:556                      3rd Qu.:30.00                                         3rd Qu.: 1.3   3rd Qu.:120.00   3rd Qu.:430.0   3rd Qu.:389.0   3rd Qu.: 48.0   3rd Qu.: 93.00
 Max.   :741                      Max.   :40.00                                         Max.   :10.8   Max.   :162.00   Max.   :735.0   Max.   :671.0   Max.   :134.0   Max.   :211.00
                                                                                        NA's   :2
      X2B             X3B               HR              RBI              SB               CS               BB               SO              BA              OBP              SLG
 Min.   : 0.00   Min.   : 0.0000   Min.   : 0.000   Min.   :  0.0   Min.   : 0.000   Min.   : 0.000   Min.   :  0.00   Min.   :  0.0   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.: 1.00   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.:  2.0   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:  2.00   1st Qu.: 10.0   1st Qu.:0.2000   1st Qu.:0.2645   1st Qu.:0.3000
 Median : 7.00   Median : 0.0000   Median : 4.000   Median : 18.0   Median : 1.000   Median : 0.000   Median : 13.00   Median : 48.0   Median :0.2310   Median :0.2980   Median :0.3670
 Mean   :10.49   Mean   : 0.9406   Mean   : 7.359   Mean   : 27.5   Mean   : 4.881   Mean   : 1.297   Mean   : 20.15   Mean   : 55.6   Mean   :0.2206   Mean   :0.2868   Mean   :0.3535
 3rd Qu.:18.00   3rd Qu.: 1.0000   3rd Qu.:12.000   3rd Qu.: 46.0   3rd Qu.: 6.000   3rd Qu.: 2.000   3rd Qu.: 32.00   3rd Qu.: 92.0   3rd Qu.:0.2560   3rd Qu.:0.3250   3rd Qu.:0.4225
 Max.   :48.00   Max.   :14.0000   Max.   :58.000   Max.   :144.0   Max.   :67.000   Max.   :16.000   Max.   :133.00   Max.   :218.0   Max.   :0.6000   Max.   :0.6670   Max.   :1.4000
                                                                                                                                     NA's   :90       NA's   :90       NA's   :90
      OPS              rOBA              TB              GIDP              HBP               SH               SF              IBB               Pos               Awards
 Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00   Min.   : 0.000   Min.   : 0.0000   Length:741         Length:741
 1st Qu.:0.5740   1st Qu.:0.2650   1st Qu.:  9.00   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.0000   Class :character   Class :character
 Median :0.6600   Median :0.2980   Median : 57.00   Median : 3.000   Median : 1.000   Median : 0.00   Median : 1.000   Median : 0.0000   Mode  :character   Mode  :character
 Mean   :0.6403   Mean   :0.2881   Mean   : 88.19   Mean   : 4.355   Mean   : 2.726   Mean   : 0.61   Mean   : 1.702   Mean   : 0.6869
 3rd Qu.:0.7450   3rd Qu.:0.3290   3rd Qu.:149.00   3rd Qu.: 7.000   3rd Qu.: 4.000   3rd Qu.: 1.00   3rd Qu.: 3.000   3rd Qu.: 1.0000
 Max.   :2.0670   Max.   :0.8070   Max.   :411.00   Max.   :25.000   Max.   :22.000   Max.   :11.00   Max.   :13.000   Max.   :20.0000
 NA's   :90       NA's   :90
 Player.additional      x              X.1            X.2
 Length:741         Mode:logical   Mode:logical   Mode:logical
 Class :character   NA's:741       NA's:741       NA's:741
 Mode  :character
```

Data Cleaning:

I began cleaning my data by creating a new datafame called "mlb_batters_all" to perform all of my cleaning functions on so that I had my original dataframe ("mlb_batters_original") to reference when needed. Using the "janitor" package, I leveraged the "clean_names" function to flatten all of the column titles to lowercase for consistency. Next, I wanted to remove all 3 logical columns as they only had NA values ("x", "x_1", "x_2") as well as the "player_additional" column as the values were only strings of numbers and letters and were of no use to any potential analysis. The last cosmetic cleaning I needed to perform was the majority of values under "Player" (player names) either had an * or # next to their name (This was part of the original download). I needed to perform a "mutate" function leveraging "str_remove_all" to remove either of those characters from the cells under the "Player" column.

	Rk	Player
1	1	Jarren Duran*
2	2	Shohei Ohtani*
3	3	Gunnar Henderson*
4	4	Marcus Semien
5	5	Juan Soto*
6	6	Bobby Witt Jr.
7	7	Aaron Judge
8	8	Vladimir Guerrero Jr.
9	9	Elly De La Cruz#

	rk	player
1	1	Jarren Duran
2	2	Shohei Ohtani
3	3	Gunnar Henderson
4	4	Marcus Semien
5	5	Juan Soto
6	6	Bobby Witt Jr.
7	7	Aaron Judge
8	8	Vladimir Guerrero Jr.
9	9	Elly De La Cruz

Due to players being acquired at the trade deadline during the season, there were some instances where instead of a team name appearing for a particular player (e.g. BOS), a “2TM” or “3TM” value would appear under the “team” variable. In order to conduct any sort of accurate team analysis, this data needed to be cleansed, therefore, I performed a “filter” function to remove any observation/row containing either of those values (I reviewed manually to see if this would affect my data and confirmed it would not). I also performed a “filter” function on the “ba” (batting average) variable to remove any with an “NA” value to ensure this didn’t affect any of my analyses (Players who have an “NA” for batting average means they didn’t play enough games to qualify for one). Additionally, given 2 of my variables began with a number (2b and 3b), they imported with an “X” in front of their title. I performed “colnames” functions to have both of these variables renamed to their original format.

BA	OBP	SLG	OPS	rOBA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA

Team
2TM
2TM
2TM
2TM
2TM
2TM
2TM

X2B	X3B

2b	3b

Data Setup:

I wanted to use player position (“pos”) in my data analysis, however, many of the players played more than 1 position resulting in values such as “8,DH” (i.e. center fielder and designated hitter) in the “pos” column. To mitigate this issue, I leveraged the “mutate” function to create a new column/variable called “pos2”. Using the “if_else” function, I stacked “str_detect” to detect the comma (e.g. 8, DH) followed by "str_trim" and "str_extract" to retrieve the 2nd value in the cell (e.g. DH) and place it in the new "pos2" variable. Next, I needed to remove the 2nd value I had just shifted into its own cell so that there was no overlapping data and so I had my clean data for future analysis.

pos
8
DH
6,DH
4
9,DH
6,DH
8,DH
3,DH
6,DH

pos	awards	pos2
8	ASMVP-8	NA
DH	ASMVP-1SS	NA
6	ASMVP-4	DH
4	AS	NA
9	ASMVP-3SS	DH
6	ASMVP-2GGSS	DH
8	ASMVP-1SS	DH
3	ASMVP-6SS	DH
6	ASMVP-8	DH

In addition to the individual player positions, I also wanted to classify whether or not players were infielders, outfielders or DH (only served as a designated hitter). In order to accomplish this, I built a "mutate" function to create a new variable called "pos_type" leveraging "case_when" and then used the position numbers to categorize the players (e.g. 1-6 = infielder, 7-9 = outfielder, DH = DH). Lastly, I created 1 additional variable called "extra_bases" using the "mutate" function adding the "2b" and "3b" variables in order to potentially rank players on not only their base hitting statistics but their ability to hit the ball in excellent positions on the field.

pos_type	extra_bases
outfielder	37
DH	45
infielder	46
infielder	31
outfielder	49
infielder	32
outfielder	22
infielder	40
infielder	34

Data Analysis

To begin, I created a frequency chart to review the batting averages of all players within the dataset. From the chart, we can see the largest set (252 players or 43.2%) had a BA between .200 - .250 with the second highest between .250 and .300 (161 or 27.6%), which makes sense given any players above .300 after a full season are considered to be among the elite (5.2%).

	ba_range	# of Players	Percent
1	.200 - .250	252	43.2
2	.250 - .300	161	27.6
3	.300 - .350	26	4.5
4	.350 - .400	1	0.2
5	< .200	141	24.1
6	>= .400	3	0.5

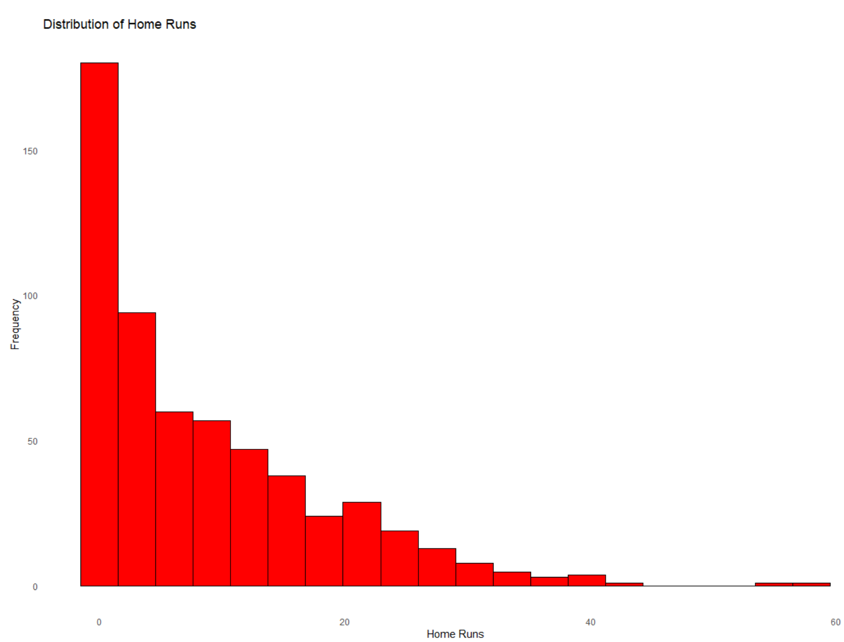
I also created a frequency chart for teams averaging their cumulative batting averages and home runs. This provides an interesting insight as home runs have become an increasingly significant statistic in baseball. Per my analysis, the Los Angeles Dodgers, last season's World Series Champions, were #1 by a large margin averaging 14.6 HRs per player. They also ranked 2nd in Batting Average at 0.255:

team	avg_ba	avg_hr
LAD	0.255	14.600
NYY	0.232	12.556
BAL	0.183	12.278
PHI	0.240	11.353
NYM	0.229	11.167
ATL	0.217	10.333
ARI	0.231	10.300
CLE	0.224	9.833
BOS	0.227	9.737
HOU	0.218	9.632

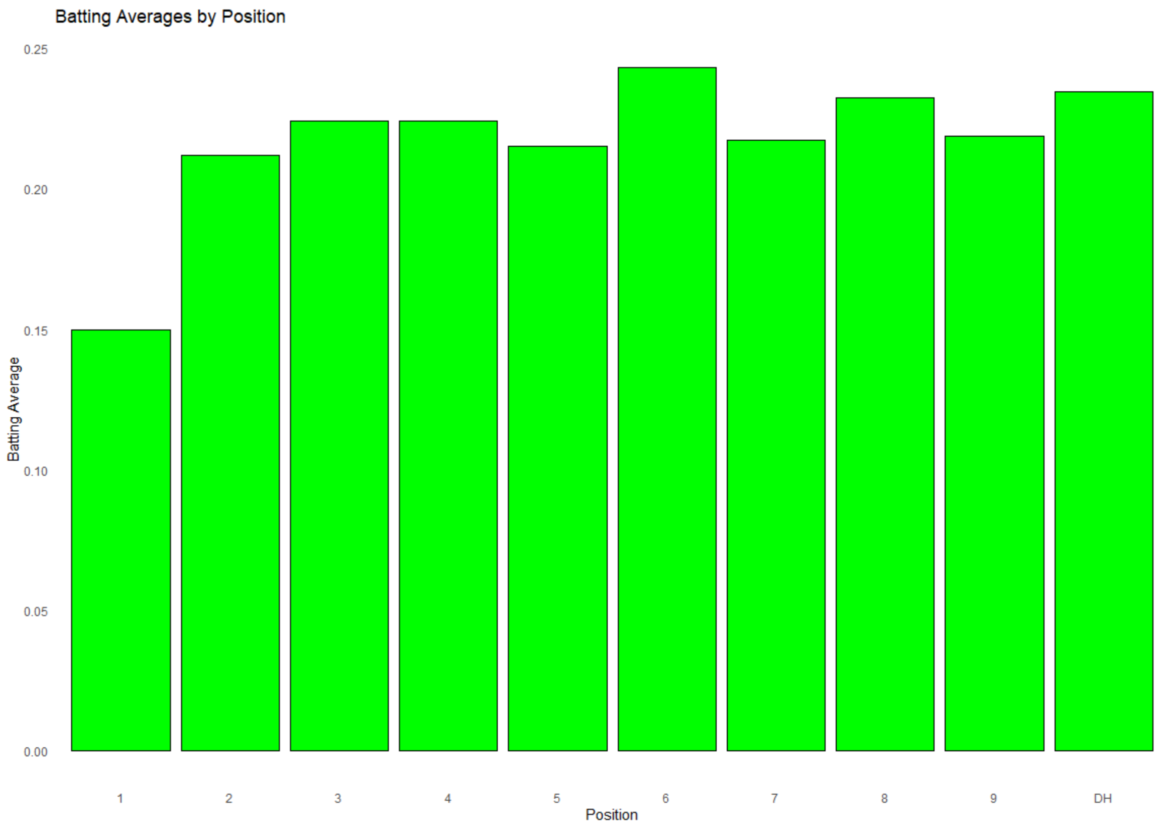
team	avg_ba	avg_hr
WSN	0.269	5.158
LAD	0.255	14.600
MIA	0.243	5.176
PHI	0.240	11.353
SDP	0.238	9.250
TOR	0.237	7.529
MIN	0.234	8.714
NYY	0.232	12.556
ARI	0.231	10.300
NYM	0.229	11.167

I also performed a cross-tabulation of average home runs by position (shown below) to determine if designated hitters do outperform field players when it comes to hitting home runs. Per the data, while infield and outfield players are somewhat aligned (8.37 and 8.81 respectively, DH's clearly outperform them by a wide margin with 11.54 on average. Following up on this theme, I wanted to investigate the distribution of HR's hit throughout a season. Unexpectedly, the majority of players hit less than 20 HRs per season but this chart still provides an interesting perspective as to how few can surpass that threshold (Per a calculation done in RStudio, 19 players hit more than 30 HRs, or 3.25%):

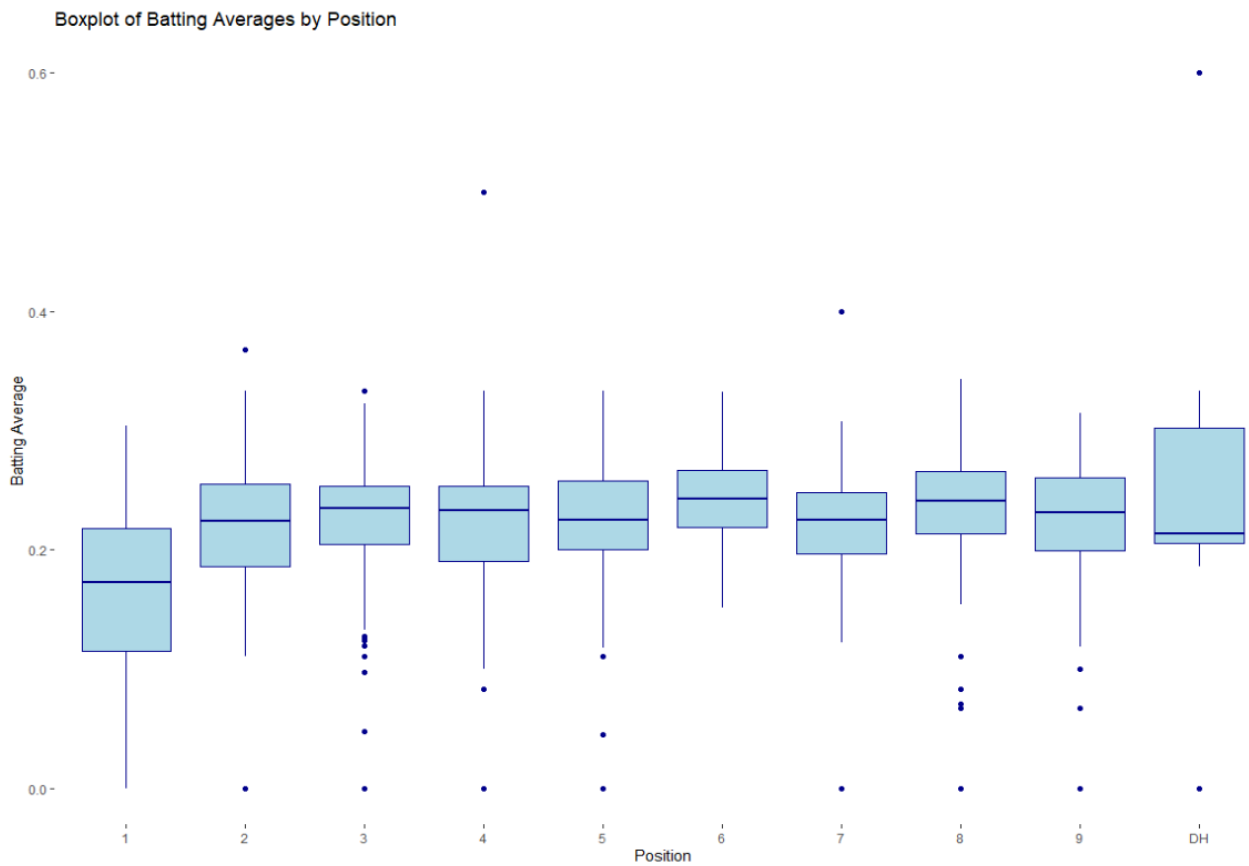
	pos_type	avg_hr
1	DH	11.54
2	infielder	8.37
3	outfielder	8.81



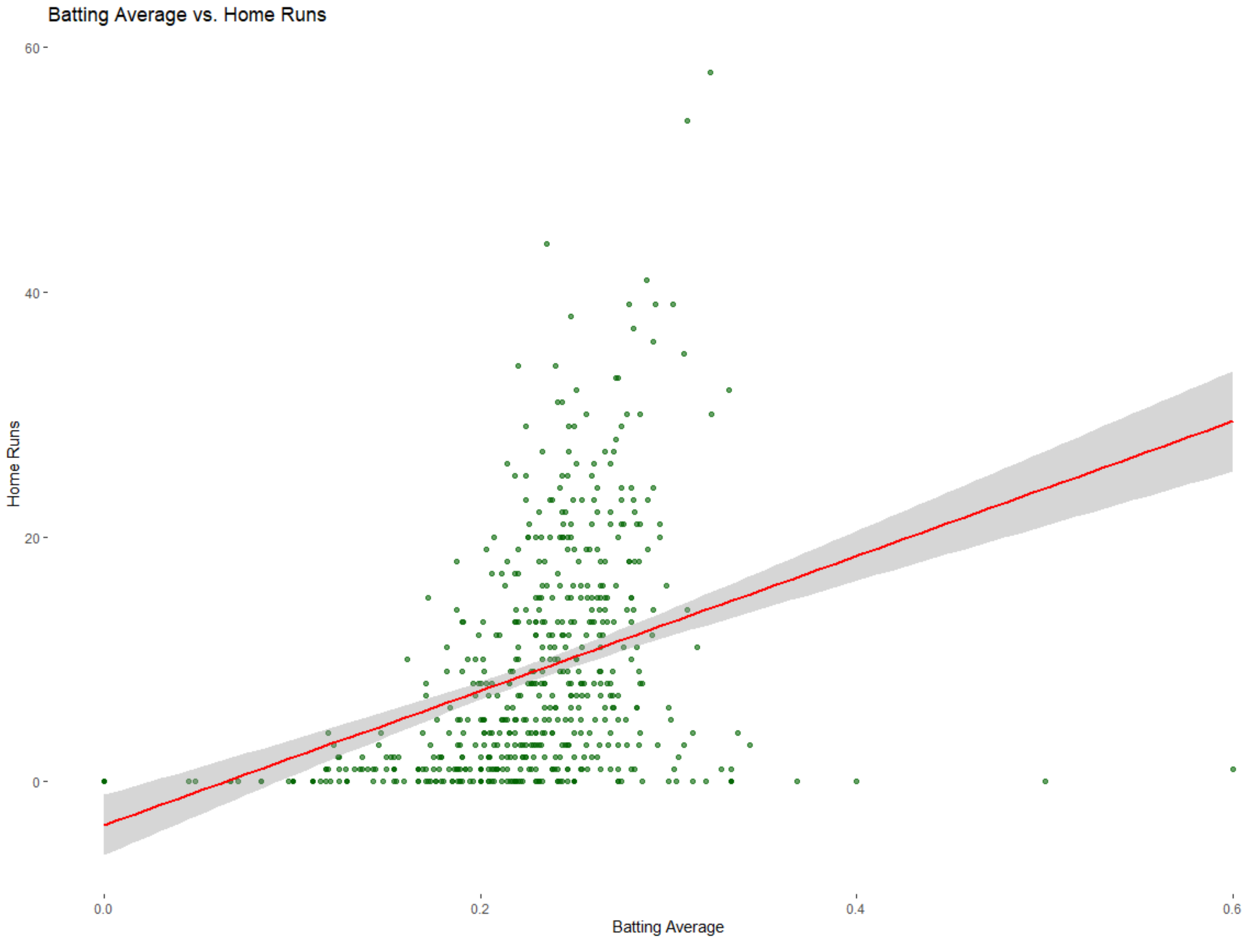
For deeper analysis, I wanted to look at batter performance by player position. I created 2 distinct views to analyze separately as I believe they each provide value – a bar chart and a boxplot. The bar chart clearly shows position 6 (Shortstop) has the highest overall batting average at .243. However, this leaves a lot of questions as to why the DH has a slightly lower BA (.235) based on much of the data we've analyzed previously.



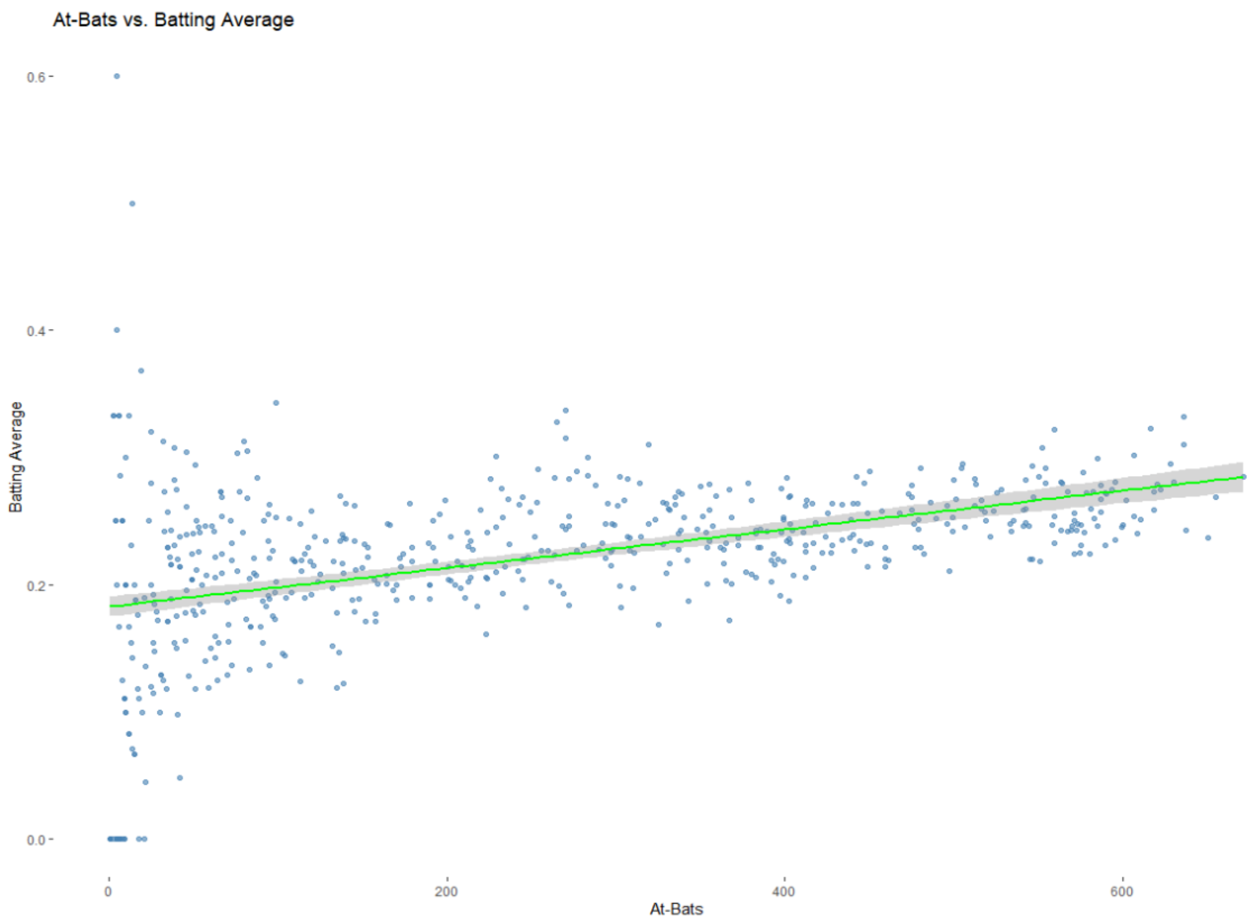
When reviewing the boxplot, we can get a much better view as to why the DH's are actually much better overall hitters. As shown in the plot below, their overall average might be slightly lower but the group as a whole has a much higher quartile and a very high outlier.



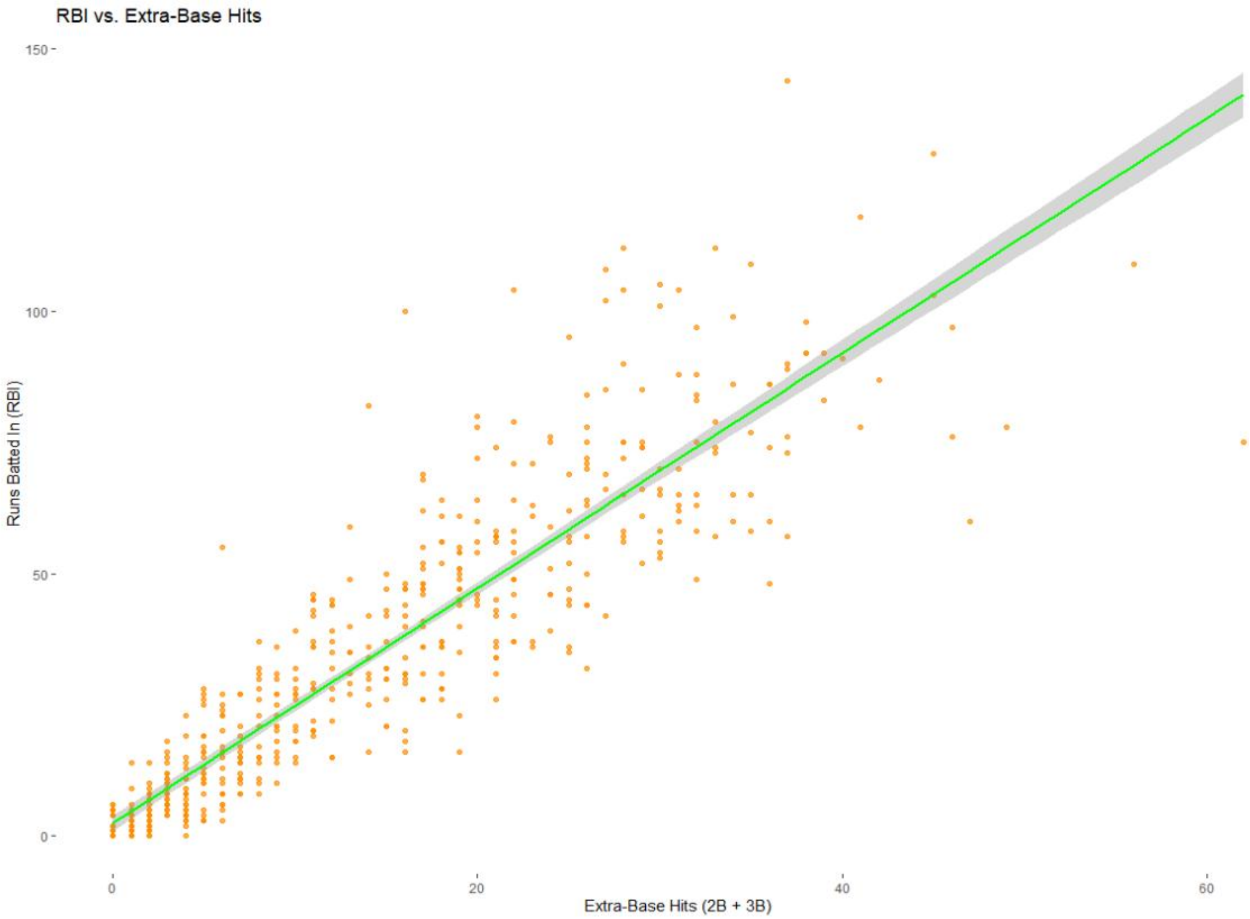
Continuing my analysis on player Batting Averages (BA), I wanted to look at the number of Home Runs hit by BA to see if there was a positive correlation (Understanding the increasing importance of HRs in baseball). Per the scatterplot shown below, there is a clear correlation between the 2 variables indicating players with higher BA's tend to hit a higher number of HR's.



Given the dataset provides a large number of players, some of which have played more games than others and/or have Designated Hitters hit for them at certain times, I wanted to analyze whether or not At-Bats (AB) had a negative or positive effect on Batting Average (BA). Per the analysis, it clearly shows that players who have more At-Bats have higher Batting Averages (BA). This is a clear indicator that better hitters are always in the batting lineup and, if they are not for a particular game, will be assigned as a Designated Hitter (DH) for that game (Note: National League (NL) and American League (AL) have different Designated Hitter rules. National League doesn't allow Designated Hitters to play, however, when they play an American League team they are able to play a DH. This will result in a higher number of AB's for DH's in the AL.)



Lastly, I wanted to utilize another one of the variables I created in my Data Setup to examine the correlation between the hitters who were able to consistently gain Extra Bases on their hits (i.e. 2B + 3B) against the number of Runs Batted In (RBI) they accrued. This seems like a natural correlation but there are several factors at play such as whether or not other players were on base before them (e.g. place in the line-up) and wanting to understand how steep the correlation actually is. Per the analysis shown below, the correlation is extremely steep indicating the batters who achieved more Extra Bases had far greater RBI's.



Conclusion

For this 1st R Practice assignment, I leveraged the MLB hitting data from the 2023 – 2024 season to clean, prepare and plot a variety of different charts. During the data cleansing process, I utilized the "tidyverse" and "janitor" packages to remove variables via the "select" function, clean the "player" variable to remove specific characters (* and #), and "filter" team names and null values. In addition, I structured the data with new variables for my analysis by creating a primary "pos" and secondary "pos2" variable to ensure I had each player's position available for analysis. Lastly, I created an "extra_bases" variable to further analyze which batters were most effective at the plate (i.e. bypassing 1st base and potentially scoring more runs).

Through my analysis and charts, I was able to thoroughly identify that players with a higher Batting Average were far more effective across the board in key categories, such as HR's, RBI's and Extra Bases – Not just at getting on base. Another key insight was that Designated Hitters, despite their title, are not far and away the greatest hitters when it comes to Batting Average. The "Boxplot of Batting Average by Position" does indicate they are slightly above average in comparison to all other positions in hitting despite the bar chart indicating their BA being slightly under (.235) the Shortstop's BA (.243). However, per my cross-tabulation of average home runs by "pos_type" (infielder/outfielder/DH), they hit far more home runs than either other category (11.54 vs 8.81 and 8.37).

References

Baseball Reference. (2024). 2024 Major League Baseball Standard Batting | Baseball-Reference.com. Retrieved from Baseball-Reference.com website: https://www.baseball-reference.com/leagues/majors/2024-standard-batting.shtml

Bluman, A. G. (2018). *Elementary statistics : a step by step approach*. New York, Ny: Mcgraw-Hill Education.

crosstabs. (2016). Retrieved April 10, 2025, from R-project.org website: https://cran.r-project.org/web/packages/pollster/vignettes/crosstabs.html

filter function - RDocumentation. (n.d.). Retrieved from www.rdocumentation.org website: https://www.rdocumentation.org/packages/dplyr/versions/0.7.8/topics/filter

Frequencies and Crosstabs in R. (2024). Retrieved from DataCamp website: https://www.datacamp.com/doc/r/frequencies

Holtz, Y. (2018). The R Graph Gallery – Help and inspiration for R charts. Retrieved from The R Graph Gallery website: https://r-graph-gallery.com/

Kabacoff, R. I. (2015). *R in Action*. Simon and Schuster.

Northeastern University – Canvas – Module 1 learning materials provided by Professor Goulding

torakxkz. (2020, February 8). Special symbols pattern search with str_detect. Retrieved from Stack Overflow website: https://stackoverflow.com/questions/60129630/special-symbols-pattern-search-with-str-detect