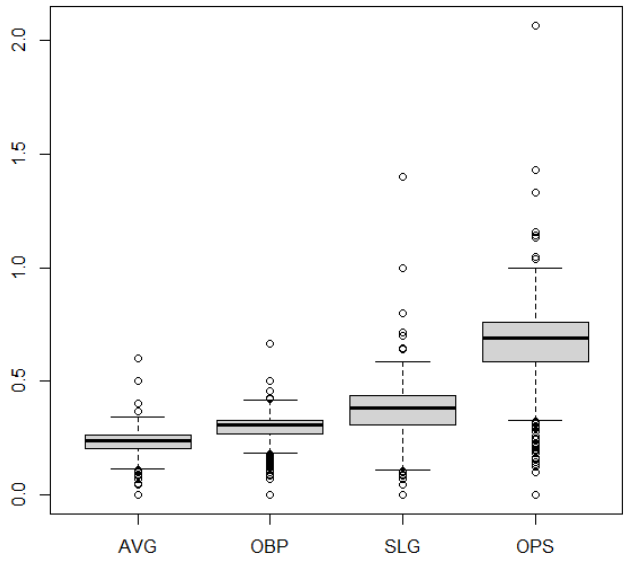
Part 1

- This project focuses on batters' performance statistics in the 2024 Major League Baseball regular season. I use Python to web script the data from the MLB official website (https://www.mlb.com/stats). I posted my original web-scripting code and dataset on my GitHub repository: https://github.com/hermesfu/mlb-data-collector.
- I originally used the data containing all 1092 batters with at least one batting record in the 2024 Major League Baseball regular season.

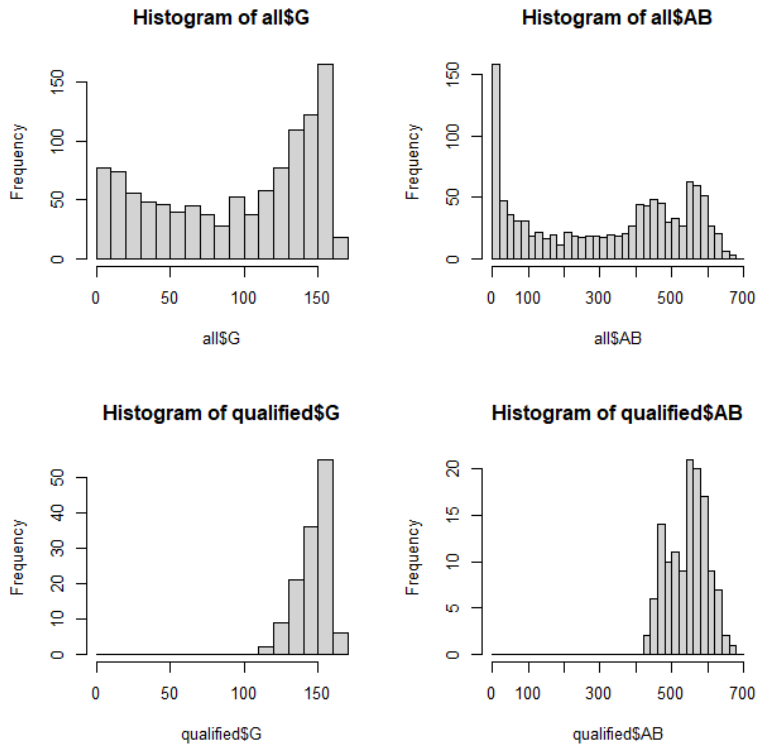
```
> dim(df)
[1] 1092   20
> head(df)
  X           name position TEAM   G  AB   R   H X2B X3B HR RBI BB  SO SB CS   AVG   OBP   SLG   OPS
1 0     B Witt Jr.       SS   KC 161 636 125 211  45  11 32 109 57 106 31 12 0.332 0.389 0.588 0.977
2 1       L Arraez       1B   SD 150 637  83 200  32   3  4  46 24  29  9  3 0.314 0.346 0.392 0.738
3 2 V Guerrero Jr.       1B  TOR 159 616  98 199  44   1 30 103 72  96  2  2 0.323 0.396 0.544 0.940
4 3       S Ohtani       DH  LAD 159 636 134 197  38   7 54 130 81 162 59  4 0.310 0.390 0.646 1.036
5 4        J Duran       CF  BOS 160 671 111 191  48  14 21  75 54 160 34  7 0.285 0.342 0.492 0.834
6 5       J Altuve       2B  HOU 153 628  94 185  31   0 20  65 47 119 22  7 0.295 0.350 0.439 0.789
```

However, when I made the boxplot for AVG, OBP, SLG, and OPS, I found some unreasonable outliers.



I will explain what these values mean later, but in short, these data are in terms of rate, and thus players who played only few games often have extreme values. Higher OPS normally means better performance, but apparently, those outliers with extremely high values are not the best players (if they are, they would not just play few games). Therefore, I use alternate data only containing players who met the Minimum Standards for Individual Championships according to Official Baseball Rules (Only those players who played enough games).

- These histograms can show the difference clearly. G(games) means how many games a batter plays, and AB(at-bats) means how many times a batter is at-bats (where sacrifice plays and walks do not count). The G histogram for all players is a U-shape with more players on the right, meaning most batters regularly played in their teams, and some bad players just played few times and were sent back to minor league 🙁. In the AB histogram for all players, we can see that many players have little at-bat times, which means they are most likely to be substitute batters, so they only came to play about once in a game. There are also more players on the right than in the middle. Both histograms for qualified players are basically the right half of their corresponding histograms for all players, agreeing with the expectations that qualified players are those players who play enough games. Both histograms look like bell shapes.



- The new data contains 129 players and 20 columns.

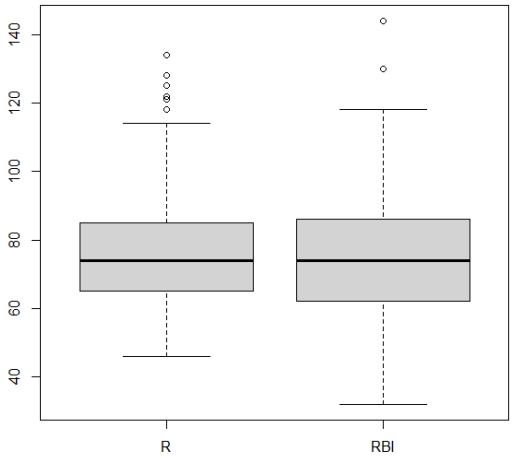
```
> dim(df)
[1] 129  20
> head(df)
  X           name position TEAM   G  AB   R   H X2B X3B HR RBI  BB  SO SB CS   AVG   OBP   SLG   OPS
1 0       A Judge       CF  NYY 158 559 122 180  36   1 58 144 133 171 10  0 0.322 0.458 0.701 1.159
2 1      S Ohtani       DH  LAD 159 636 134 197  38   7 54 130  81 162 59  4 0.310 0.390 0.646 1.036
3 2        J Soto       RF  NYY 157 576 128 166  31   4 41 109 129 119  7  4 0.288 0.419 0.569 0.988
4 3    B Witt Jr.       SS   KC 161 636 125 211  45  11 32 109  57 106 31 12 0.332 0.389 0.588 0.977
5 4      Y Alvarez      DH  HOU 147 552  88 170  34   2 35  86  69  95  6  0 0.308 0.392 0.567 0.959
6 5 V Guerrero Jr.      1B  TOR 159 616  98 199  44   1 30 103  72  96  2  2 0.323 0.396 0.544 0.940
```

- Position and TEAM are categorical data with 11 and 30 options respectively.

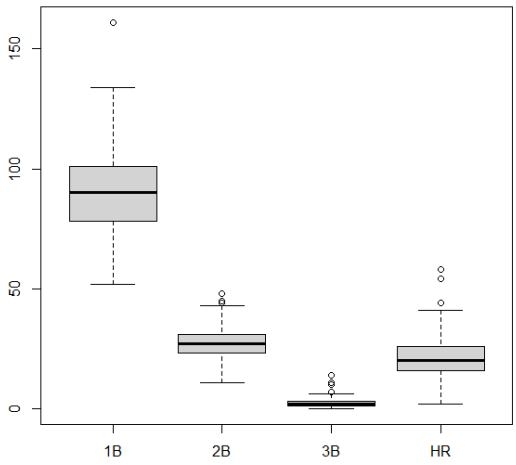
```
> unique(position)
 [1] "SS" "1B" "DH" "CF" "2B" "C"  "3B" "LF" "RF" "X"  "P"
> unique(TEAM)
 [1] "KC"  "SD"  "TOR" "LAD" "BOS" "HOU" "ATL" "NYY" "BAL" "COL" "CLE" "PIT" "NYM" "MIL" "STL" "CIN"
[17] "OAK" "CHC" "TB"  "PHI" "SEA" "TEX" "AZ"  "LAA" "SF"  "WSH" "CWS" "MIN" "MIA" "DET"
```

Boxplot for R and RBI

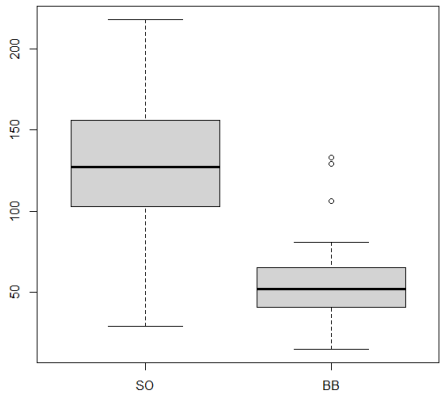
R(runs) means how many times a player reaches the home plate to score. RBI(runs batted in) means how many runs got because of the player's hit. They have similar values because each R is generally caused by one RBI. The range for RBI is wider because a player can get several RBI in one single play but at most one R, making RBI have a higher variation between batters (but almost the same means).



Boxplot for Different Hits



The original data used H(hits) to record the sum of all kinds of hits. I replaced it with 1B by subtracting H from all the other kinds of hits, making a clearer comparison. We can easily see from the boxplot that 1B is the easiest to get and 3B is the hardest. (HR means home runs, 1B means a hit enables the batter to reach 1B, etc.)

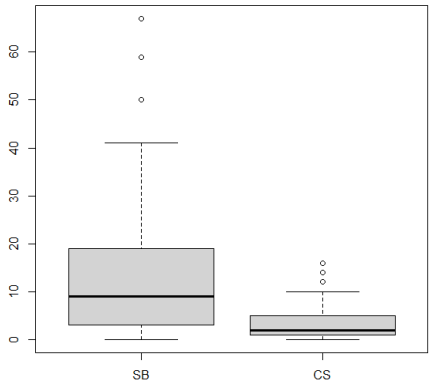


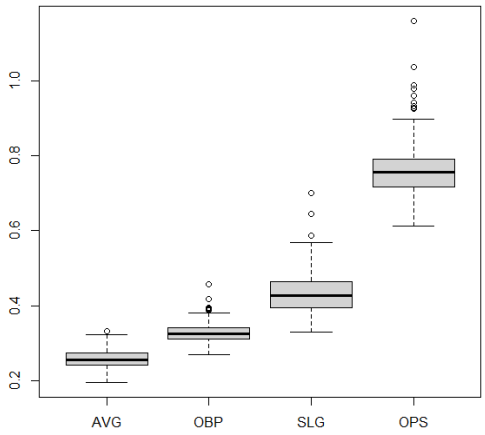
Boxplot for SO and BB

SO means strikeout, which is bad for batters. BB means base on ball, which is good for batters. From the graph, strikeout happens more than base on ball. Some players are good at getting base on ball really much(Aaron Judge because every pitcher is afraid of his power and Juan Soto because he has ordinary plate discipline). There are no outliers for strikeouts, and I guess it is because if a player is stricken out too often, the coach would not let him play anymore.

Boxplot for SB and CS

SB means successful stolen base, and CS means caught at stolen base (unsuccessful stolen base, in other words). Successful stolen bases are much more than unsuccessful stolen bases. I interpret it not because it is easy to steal a base but because only players confident in stealing bases would try to.





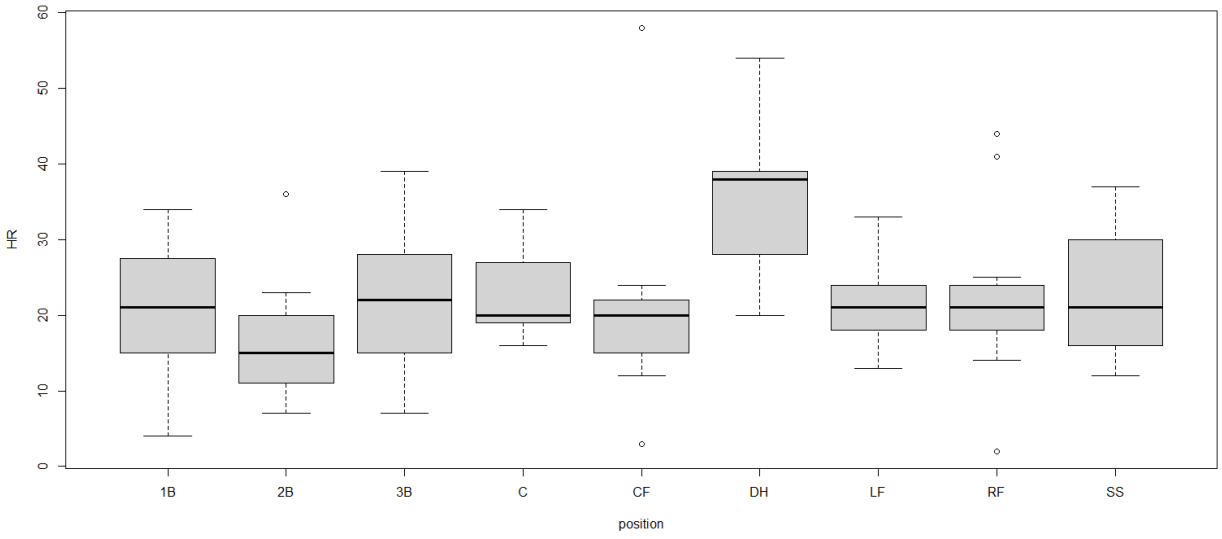
AVG (batting average) means how many hits a player can get per play. OBP (on-base percentage) means how many times a player can get on base per play (it includes both base on ball and hits and thus is almost always larger than AVG). SLG (Slugging percentage) means how many extra bases a player can get per play (1B counts 1, 2B count 2, etc.) OPS is the sum of OBP and SLG and is viewed as one representing indicator for players' performance because it takes two main factors (plate discipline from OBP and power from SLG) for batters into account.

Part 2

- There is a common impression for baseball players: center fielders, shortstops, and second basemen run fast but have weak power, while first basemen, catchers, and designated hitters have strong power but run slowly. These statements come from some interesting inferences:
    1. Center fielders, shortstops, and second basemen often need to run fast to catch the ball when fielding, but first basemen, catchers, and designated hitters do not. Therefore, the former run fast and the latter run slow.
    2. No one can be perfect, so fast players should have weak power. No one is useless, so slow players should have stronger power.
- To test whether these statements are significantly true or just a myth, I decided to do ANOVA tests for both home runs (the main indicator of power) and stolen bases (the main indicator of speed) based on players' positions.

ANOVA for Home Runs based on Positions

Firstly, I drew a boxplot to get a general idea. Most positions have similar home runs, except designated hitters have the most. The previous assumption seems to be half-correct.

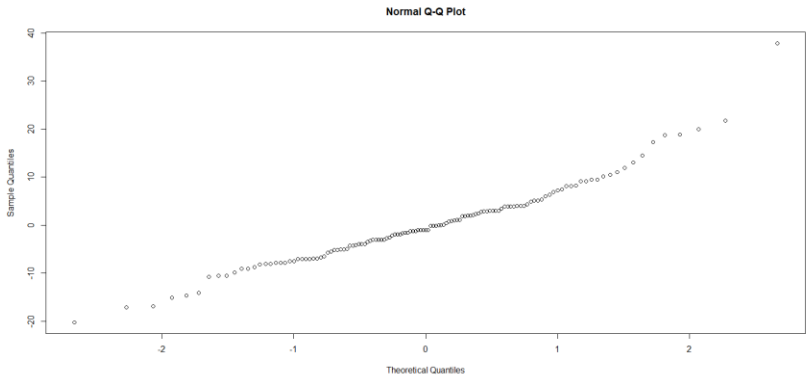


Before conducting the ANOVA test, here are assumption checks.

1. The variances within groups are slightly different from groups but can be tolerated.

```
> aggregate(HR, by=list(position), sd)
  Group.1         x
1      1B  7.680631
2      2B  7.600246
3      3B  8.800599
4       C  5.830952
5      CF 11.801130
6      DH 11.710801
7      LF  5.092526
8      RF 11.128383
9      SS  7.686675
```

2. The residual seems normal (except for the crazy outliers Aaron Judge).



Then, I used ANOVA to test whether there were any significant differences.

H_0: means of HR are all equal, H_1: some means of HR are different

p-value = 0.00382 is small, reject H_0, we have enough evidence to show that some means of HR are different (but not too strong).

```
> summary(HRanova)
                   Df Sum Sq Mean Sq F value  Pr(>F)
as.factor(position)   8   1798  224.81   3.037 0.00382 **
Residuals           120   8882   74.02
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

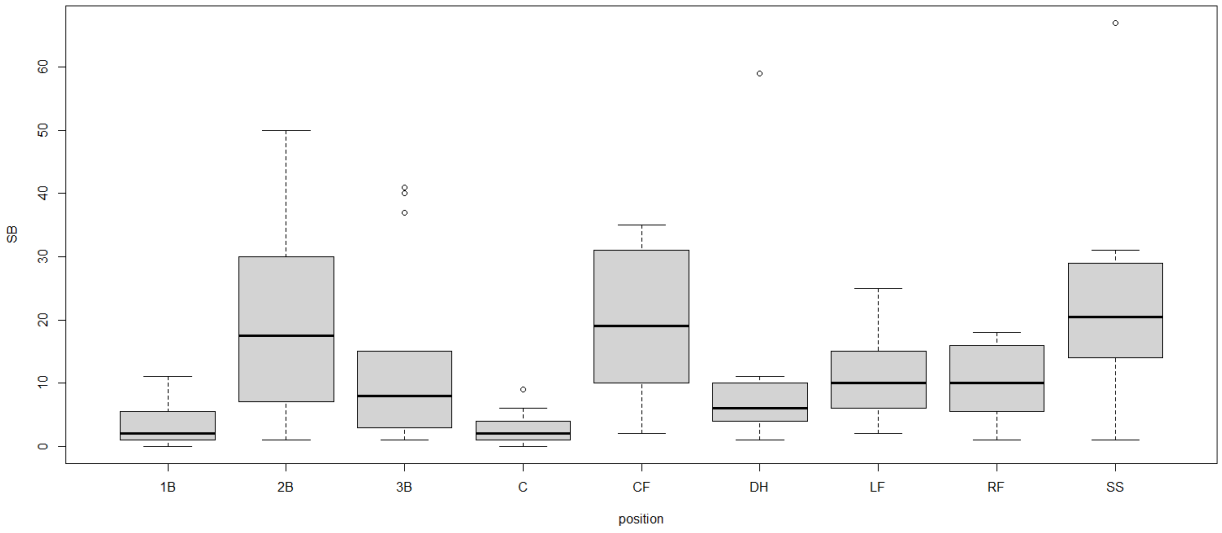
Lastly, I used a t-test with Bonferroni correction to check which pairs have differences. According to the pairwise t-test and the previous boxplot, DH is significantly higher in HR than 2B, which agrees with the earlier statement. Although other pairs do not have too significant differences, C and SS also have higher values than 2B, and 1B, 3B, and CF have lower values than DH.

```
> pairwise.t.test(HR, position, p.adj='bonferroni', pool.sd=FALSE)

        Pairwise comparisons using t tests with non-pooled SD

data:  HR and position

   1B   2B   3B   C    CF   DH   LF   RF
2B 1.00 -    -    -    -    -    -    -
3B 1.00 1.00 -    -    -    -    -    -
C  1.00 0.82 1.00 -    -    -    -    -
CF 1.00 1.00 1.00 1.00 -    -    -    -
DH 0.63 0.14 0.87 1.00 0.59 -    -    -
LF 1.00 1.00 1.00 1.00 1.00 0.66 -    -
RF 1.00 1.00 1.00 1.00 1.00 1.00 1.00 -
SS 1.00 0.87 1.00 1.00 1.00 1.00 1.00 1.00

P value adjustment method: bonferroni
```

In conclusion, the previous statement about HR is partially correct. We can only confidently say that DH has a significantly higher HR value than most other positions. The statements for C and 2B probably make sense but with only slight evidence. On the contrary, SS has more HR and 1B has fewer HR than the statement assumes.

ANOVA for Stolen Bases based on Positions

The procedure the analyze stolen bases is similar to the one for home runs. Firstly, I drew a boxplot to get a general idea. We can generally classify the number of stolen bases into three groups. Second basemen, center fielders, and left fielders have the most stolen bases. Third basemen, left fielders, and right fielders have moderate numbers. First basemen, catchers, and designated hitters (except Shohei Ohtani the crazy outlier) have the least. The previous assumption seems accurate in terms of speed.

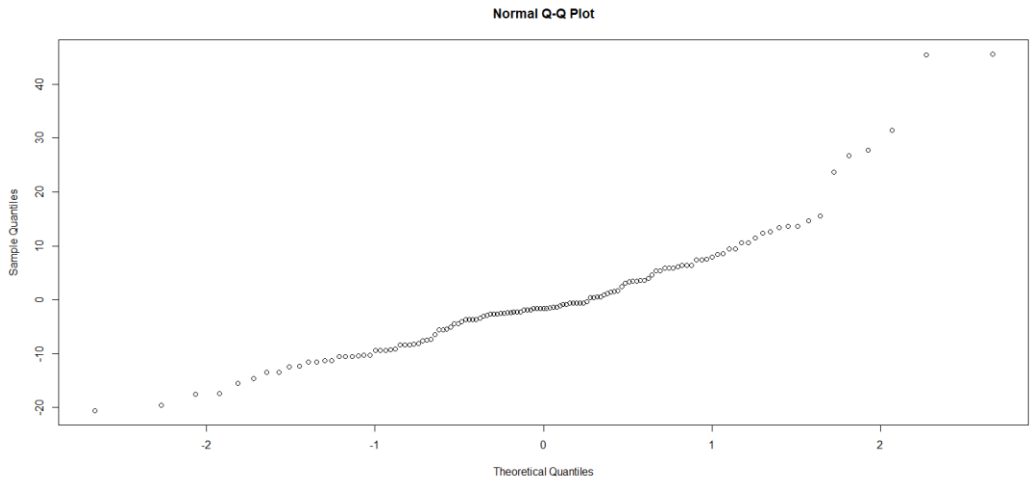


Before conducting the ANOVA test, here are assumption checks.

1. The variances within groups are different for different groups. The main reason is that most players in 1B, C, and DH rarely steal bases and thus have low values in both means and variations. Therefore, the result of ANOVA might not be strongly valid. However, it can still provide some information, especially alongside Bonferroni pairwise t-tests.

```
> aggregate(SB, by=list(position), sd)
  Group.1         x
1      1B  3.434307
2      2B 14.036843
3      3B 14.767056
4       C  2.934469
5      CF 11.549892
6      DH 20.378560
7      LF  6.883484
8      RF  5.915439
9      SS 14.701830
```

2. The residual seems normal (except for the crazy outliers De La Cruz and Ohtani).



Then, I used ANOVA to test whether there were significant differences.

H_0: means of SB are all equal, H_1: some means of SB are different

p-value = 0.6.54e-6 is extremely small, reject H_0, we have significant evidence to show that some means of SB are different.

```
> summary(SBanova)
                    Df Sum Sq Mean Sq F value   Pr(>F)
as.factor(position)   8   5454   681.7    5.49 6.54e-06 ***
Residuals           120  14901   124.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lastly, I used t-test with Bonferroni correction to check which pairs have differences. Many pairs have significant differences. According to the t-test and the previous boxplot, SS, 2B, and CF have significantly more SB, while 1B and C have significantly less SB, which agrees with the earlier statement. The results are much more significant than the results for HR. The only position not significantly different from any other is DH.

```
> pairwise.t.test(SB, position, p.adj='bonferroni', pool.sd=FALSE)

        Pairwise comparisons using t tests with non-pooled SD

data:  SB and position

   1B     2B     3B     C      CF     DH     LF     RF
2B 0.0560 -      -      -      -      -      -      -
3B 1.0000 1.0000 -      -      -      -      -      -
C  1.0000 0.0395 0.8021 -      -      -      -      -
CF 0.0038 1.0000 1.0000 0.0024 -      -      -      -
DH 1.0000 1.0000 1.0000 1.0000 1.0000 -      -      -
LF 0.0107 1.0000 1.0000 0.0072 1.0000 1.0000 -      -
RF 0.1171 1.0000 1.0000 0.0710 0.4625 1.0000 1.0000 -
SS 0.0027 1.0000 1.0000 0.0017 1.0000 1.0000 0.5156 0.2397

P value adjustment method: bonferroni
```

In conclusion, the previous statement about SB is proved to be correct for SS, CF, 2B, C, and 1B. The only exception is DH. DH is shown not significantly lower than other positions.

(One funny explanation is that DH is bad at defense and thus is good at both running and power. The statement “no one is perfect” still holds true.)

Classification of positions

Similar to the previous part, we assume there might be differences between positions (we have already concluded that there are significant differences in SB, and there may also be other differences). Then, is it possible to predict players’ positions from their data? Therefore, I use multinomial regression to test this.

Firstly, I include all the possible columns (G, AB, R, H, X2B, X3B, HR, RBI, BB, SO, SB, CS, AVG, OBP, SLG, and OPS) into the model.

```
Call:
multinom(formula = position ~ G + AB + R + H + X2B + X3B + HR +
    RBI + BB + SO + SB + CS + AVG + OBP + SLG + OPS)

Coefficients:
   (Intercept)           G          AB           R           H         X2B         X3B          HR         RBI          BB          SO          SB          CS         AVG         OBP
2B  -0.7589559 -0.001042448  0.006352537 0.06555330  0.015214520 -0.2323176608 0.13652888  0.12522764 -0.102713807 -0.02643549  0.004379812  0.19437194 0.3753548 -0.12735453 -0.43389408
3B  -2.0521824  0.032710388 -0.002090522 0.05155403 -0.038639581  0.031209684 0.57633954  0.08787830  0.007863611 -0.05336017 -0.012085935  0.11458235 0.5419331 -0.55887548 -0.72497211
C   -0.2785816  0.045662291 -0.018917943 0.06523096  0.010749517 -0.220835307 0.49435585 -0.03335404  0.061983916 -0.06408302  0.014330594 -0.20120517 0.3663855 -0.15729418 -0.15605089
CF   0.9804904  0.097831010 -0.037964908 0.06256323  0.007968932 -0.040846946 1.01547925  0.28492099 -0.103671738 -0.09156508  0.010058967  0.13619732 0.6222324  0.22412842 -0.13045871
DH  -0.8533682  0.025100362 -0.024863242 0.07709453  0.090198022 -0.288557501 0.07607425  0.41309442 -0.145069488 -0.05780813  0.014404545  0.16220762 0.4726944 -0.35842334 -0.31751392
LF  -0.4049740 -0.015852746  0.005575496 0.05311919 -0.031869658 -0.067366754 0.61504588  0.03899034 -0.007703303 -0.02592489  0.001942659  0.08549266 0.6405141 -0.01683492  0.27502610
RF  -0.3931214  0.053450670 -0.008016171 0.08117618 -0.037257462 -0.088174721 0.61379505  0.18191297 -0.065556928 -0.04685855 -0.003640141  0.10060275 0.4471893  0.01043323 -0.09574748
SS   0.1697755 -0.053412158  0.006363005 0.11261126 -0.026130238  0.002521746 0.58620979  0.19690105 -0.086638397 -0.09180162  0.010361397  0.13087543 0.7300656  0.22693230  0.22328246
         SLG         OPS
2B -0.07219478 -0.5060889
3B -0.97101798 -1.6959901
C  -0.07142939 -0.2274803
CF  0.37092800  0.2404693
DH -0.74603043 -1.0635443
LF -0.15418189  0.1208442
RF -0.29286648 -0.3886140
SS  0.48084608  0.7041285

Std. Errors:
   (Intercept)          G         AB          R          H         X2B         X3B         HR         RBI         BB         SO         SB         CS         AVG         OBP         SLG         OPS
2B   0.5130899 0.06257706 0.02266804 0.06177482 0.04955325 0.09935656 0.3608253 0.1319694 0.05530100 0.03998566 0.01895278 0.08696825 0.3007561 0.13488996 0.17714644 0.2229528 0.4000908
3B   1.2985360 0.06468481 0.02181598 0.05710916 0.04405416 0.08032874 0.3242427 0.1105394 0.04114497 0.03814995 0.01715849 0.08558768 0.2929668 0.34368883 0.42559679 0.5857379 1.0113337
C    0.2696416 0.07702812 0.02645000 0.06286723 0.04868214 0.10451995 0.4178505 0.1383234 0.05470100 0.04044745 0.02150145 0.17926275 0.4674953 0.06692860 0.08760789 0.1200464 0.2076446
CF   0.6211546 0.06330617 0.02381578 0.06321793 0.05475058 0.09956630 0.3338373 0.1446776 0.06327618 0.04358102 0.01985375 0.08681024 0.2940237 0.16379198 0.20416111 0.2806201 0.4847753
DH   0.2549182 0.09688758 0.03149122 0.08284239 0.05588606 0.14058922 0.4549494 0.1832338 0.08150910 0.05240449 0.02165190 0.10412532 0.4435710 0.07075474 0.08847216 0.1227484 0.2112128
LF   1.7287496 0.06398533 0.02106533 0.05286799 0.04413057 0.07920999 0.3145077 0.1129433 0.04401915 0.03541854 0.01663060 0.08455981 0.2800066 0.45301237 0.57463024 0.7714912 1.3461105
RF   2.0484460 0.06565136 0.02114686 0.05849497 0.04933453 0.08961807 0.3306092 0.1145552 0.04923360 0.03854986 0.01767731 0.08888441 0.3000055 0.52481265 0.68858501 0.9147604 1.6033408
SS   1.8953030 0.07261631 0.02351223 0.06120663 0.05014375 0.09039601 0.3207735 0.1292727 0.05501722 0.04086669 0.01825650 0.08557232 0.2856875 0.50565156 0.63049685 0.8518917 1.4823869

Residual Deviance: 377.2012
AIC: 633.2012
```

The coefficients reveal several interesting things. As we concluded from the previous ANOVA tests, more SB means less likely to be C (the position with the least SB). The possibility of being DH (the position with the most HR) increases the most when HR increases. They also reveal some discoveries. For example, 3B and RF might have less SO, while SS and LF might have higher OBP.

I then tested the percentages of the model to predict players' positions correctly from the trained data (2024) and test data (2023). The model successfully predicts 2024 data with 0.496124 accuracy. It is actually an impressive result because there are nine different positions, so the accuracy from random guess is only 1/9 = 0.11. From the two-sample proportions z-test, the model is significantly better than randomly guessing (with p = 2.211e-11)

```
        2-sample test for equality of proportions with continuity correction

data:  c(sum(position == apply(results, 1, function(row) names(row)[which.max(row)])), length(position)/9) out of c(length(position), length(position))
X-squared = 43.418, df = 1, p-value = 2.211e-11
alternative hypothesis: greater
95 percent confidence interval:
 0.2917366 1.0000000
sample estimates:
   prop 1    prop 2
0.4961240 0.1111111
```

However, the accuracy drops to 0.2686567 for 2023 data. It makes sense because the model is trained from 2024 data. Overfitting and changes in the MLB environment each year can reduce the accuracy of predicting other than the trained data. 27% is not terrible accuracy (still significantly better than randomly guessing with a p-value of 0.0008675), but it can be improved.

```
        2-sample test for equality of proportions with continuity correction

data:  c(sum(df2$position == apply(results, 1, function(row) names(row)[which.max(row)])), length(df2$position)/9) out of c(length(df2$position), length(df2$position))
X-squared = 9.8107, df = 1, p-value = 0.0008675
alternative hypothesis: greater
95 percent confidence interval:
 0.07287414 1.00000000
sample estimates:
   prop 1    prop 2
0.2686567 0.1111111
```

Therefore, I use stepwise model selection to construct a new model.

```
Call:
multinom(formula = position ~ X2B + RBI + SB + OBP + SLG + OPS)

Coefficients:
   (Intercept)         X2B          RBI         SB        OBP       SLG        OPS
2B   2.6746477 -0.10923612 -0.041204274  0.3057121 -12.497102 11.625965 -0.8711366
3B   0.6537344  0.03792199  0.010641127  0.2661089 -20.746198 14.873058 -5.8731397
C    3.0878953 -0.15629102  0.038122029 -0.1026594 -13.047815  9.706236 -3.3415789
CF   2.5041629 -0.06384454 -0.106913546  0.3290628 -59.373233 57.236243 -2.1369900
DH  -8.1780063 -0.22583343 -0.013038991  0.2947920 -25.889598 32.958188  7.0685907
LF  -1.4357174 -0.04255932  0.001511364  0.2506563  -5.620361  5.668617  0.0482564
RF   0.4118886 -0.08036945 -0.028505609  0.2394521 -21.531245 21.212105 -0.3191394
SS  -0.2103744  0.02912290 -0.084756708  0.3255002 -47.604098 45.521160 -2.0829383

Std. Errors:
   (Intercept)        X2B        RBI         SB      OBP       SLG      OPS
2B    5.021413 0.07345050 0.03782915 0.07648356 12.48186 11.461727 5.570242
3B    4.448351 0.06482360 0.03150995 0.07561264 12.03424 10.584165 5.210413
C     5.596177 0.08117539 0.03576414 0.15231676 14.34791 12.530972 6.307476
CF    4.532383 0.06829886 0.04186417 0.07494030 13.54840 12.379058 5.376445
DH    5.029946 0.10638178 0.04400824 0.08084879 16.50444 14.796584 6.074751
LF    4.066090 0.06199757 0.03020423 0.07515263 10.74430  9.706461 4.675181
RF    4.428025 0.06893635 0.03377650 0.07785910 12.03264 10.775986 5.115744
SS    4.422012 0.06385774 0.03779249 0.07451543 12.42326 11.188925 5.256813

Residual Deviance: 434.158
AIC: 530.158
```

The new model contains only 2B, RBI, SB, OBP, SLG, and OPS. It has a better AIC 530.158 than the previous model's 633.2012. As the previous ANOVA test indicated, SB is a representative indicator of positions. All other columns are representative indicators of players' performances, and some irrelevant columns, such as G and AB, were removed. This means the reduced model takes players' performances as the main consideration when determining their positions. One noticeable thing is that as the OPS (the most representative indicator for overall batting performance) increases, the likelihood of being DH increases the most. This result agrees with the impression that DH is usually a better batter.

I then did the same test for the new model. Surprisingly, the new model only has an accuracy of 0.3565891 for 2024 data and 0.2462687 for 2023. The new accuracy for 2024 data is much worse than the previous model, while the accuracy for 2023 is similar. But still, according to the z-test, the accuracies are still significantly better than randomly guessing.

```
        2-sample test for equality of proportions with continuity correction

data:  c(sum(position == apply(results, 1, function(row) names(row)[which.max(row)])), length(position)/9) out of c(length(position), length(position))
X-squared = 20.345, df = 1, p-value = 3.233e-06
alternative hypothesis: greater
95 percent confidence interval:
 0.15476 1.00000
sample estimates:
   prop 1    prop 2
0.3565891 0.1111111
```

```
        2-sample test for equality of proportions with continuity correction

data:  c(sum(df2$position == apply(results, 1, function(row) names(row)[which.max(row)])), length(df2$position)/9) out of c(length(df2$position), length(df2$position))
X-squared = 7.4441, df = 1, p-value = 0.003182
alternative hypothesis: greater
95 percent confidence interval:
 0.05191935 1.00000000
sample estimates:
   prop 1    prop 2
0.2462687 0.1111111
```

In conclusion, both the model considering all information and the stepwise selection model can provide a significantly better prediction for players' positions, which means that players' performances may differ according to their positions. However, their interactions are complex, so neither of the models can provide an accurate result. A better model or more meaningful data may be able to provide more reliable predictions, or maybe the relationships between players' performances and positions are not strong enough.

Extra Discoveries

This is not technically part of the project. I just wrote down some interesting discoveries when I played around with the data randomly:

1. OPS = OBP + SLG. When I made a model to predict OPS from all other values, the slopes for OBP and SLG were perfect 1. Interestingly, When I used a backward stepwise procedure to select a reduced model, it included more things than just OBP and SLG, when the perfect model should only include these two.
2. When I made a model to predict SO from all other values, the most significant value was CS. I do not know what their relationships are at all, I think it is just a funny coincidence (the only reasonable explanation is that careless players would have more CS and SO at the same time). HR also has a strong relationship with SO, which agrees with the impression of most people.