MLB Player Evaluation and Prediction through Impact Score based on War

2024-12-16

Problem:

The Philadelphia Phillies are busy in winter meetings attempting to evaluate potential MLB talent to retool their roster. Whether it be free agency, where players in between contracts can sign to a new team, or via trade, where players can be exchanged for one another, there is organizational pressure to find the right pieces to complete a roster capable of winning.

Solution:

I have constructed a linear regression model to predict the WAR of players for the upcoming 2025 season. WAR, also known as Wins Above Replacement, is a statistic that quantifies player performance when compared to that of a replacement level player, which is a readily-available major league or minor league bench player, at their specific position. In other words a WAR of 4.2, tells us this specific player added 4.2 more wins to his team over the course of the season than a replacement level player would have.

WAR Scale for a Single Season:

< 0 WAR: The player performed below the level of a replacement-level player. Essentially, a readily available minor league or bench player could have done better. Examples: Players with poor offensive and defensive performance or who contribute negatively overall.

0 - 1 WAR:

Replacement Level: A fringe MLB player or a bench player. This is the baseline for major league performance.
1 - 2 WAR:

Role Player: A decent but unspectacular contributor. Often a part-time player, a younger everyday player still in development stages, or a respected utility/ defensive minded player.

2 - 3 WAR:

Solid Starter: A regular player who plays an important role to a winning team, but isn't necessarily a standout.

3 - 4 WAR:

All-Star Caliber: A high-quality player who is one of the better performers in the league at their position.
4 - 6 WAR:

Superstar: A top-tier player. These players are among the best in the league and often candidates for MVP voting.

6+ WAR:

MVP-Level: An elite player who provides exceptional value and is often among the best players in the entire league for that season. Examples: Mike Trout, Shohei Ohtani, Mookie Betts in their best years.

Dataset Selection and Cleaning

As many WAR calculations are not made readily available, I have gathered a data set from a respected baseball statistics database called FanGraphs. In compiling my table, I decided to collect data from the past 3 MLB seasons, 2022-2024. This will allow appropriate context for our 2025 predictions, not overly relying on the past 2024 season, which could've just been an up or down season for someone, while not taking in account a season from 4-5 years ago as number will greatly differ, especially for aging or younger players. To allow for promising rookies to enter my model but exlude those with only a few weeks of games, I have placed a 300 PA, or plate appearance, minimum. Any less may overly skew someones performance. In order to follow R's linear model syntax, I had to change some column names to not include a +,/, or % in the header. I created the BB_perK column, dividing BB by the K columns, in order to create a very useable ratio for plate discipline. The Data includes 53 different columns as to not limit my studies, however to begin I will select 6 predictors I believe to have a strong impact on WAR, and a player's overall value to their team.

Predictor Variables:

wRC+ (Weighted Runs Created Plus)

Definition: Measures a player's total offensive value and adjusts for park factors and the league environment. It's scaled so that 100 is league average, and every point above or below represents a percentage point better or worse than the league average.

Interpretation: A player with a wRC+ of 150 is 50% better than league average offensively, while one with wRC+ of 80 is 20% below average.

Hard_hitPercentage (Hard Hit Percentage)

Definition: The percentage of balls a batter hits in play with an exit velocity of 95 mph or higher. Available from Statcast data.

Interpretation: A higher percentage indicates that a player consistently makes hard contact, often leading to more extra-base hits and home runs or at the very least less time for the fielder to react to the ball. A .34 HHP means 34% of the balls hit in play by player X, were hit at 95+ mph

BB_perK (Walks per Strikeout)

Definition: Ratio of walks to strikeouts, representing a player's plate discipline and contact skills.

Interpretation: A higher value suggests that the player is more patient and disciplined at the plate, leading to better on-base potential and fewer wasted at-bats.

ISO (Isolated Power)

Definition: Measures a player's raw power by subtracting batting average from slugging percentage. It isolates the extra bases gained per at-bat. Formula:

$$ISO = SLG - AVG$$

Interpretation:

Low ISO (< 0.100): Singles hitter.

Moderate ISO (0.100 - 0.200): Balanced hitter with some power.

High ISO (> 0.200): Power hitter.

BsR (Base Running Runs)

Definition: A comprehensive baserunning metric from FanGraphs that evaluates a player's value on the bases, including stolen bases, caught stealing, and advancing on balls in play.

Interpretation: Positive BsR values indicate above-average baserunning, while negative values suggest below-average baserunning skills.

Def (Defensive Runs)

Definition: A per-game measure of defensive value, derived from Def (Defensive Runs). Def from FanGraphs is a composite defensive metric combining positional adjustments, fielding runs saved, and other components.

Interpretation: Higher values suggest better defensive contributions. For example:

Positive Def: Indicates above-average defense.

Negative Def: Suggests below-average defense.

Positional Adjustments are significant as SS and CF are much more difficult to play at an elite level than LF or 1B, and must be accounted for in the model

So Why These Predictors?

wRC+ and *ISO** measure offensive performance and power.

Hard_hitPercentage captures the quality of contact.

BB_perK evaluates plate discipline.

BsR accounts for baserunning impact.

Def_perGame assesses defensive value per game.

1) Loading the Dataset

```r
# Load necessary libraries
library(readr)   # For reading CSV files
library(dplyr)   # For data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2) # For visualization (optional)

# Step 1: Load the dataset
MLB_data <- read_csv("MLB_PlayerData22-24.csv")
```

```
## Rows: 856 Columns: 52
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Name, Team, NameASCII
## dbl (48): Season, Age, G, PA, HR, AVG, RBI, R, SB, CS, 2B, 3B, BB, IBB, SO, ...
## lgl  (1): xBA
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(MLB_data)
```

```
## # A tibble: 6 x 52
##   Name  Team  Season   Age     G    PA    HR   AVG   RBI     R    SB    CS  `2B`
##   <chr> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Aaro~ NYY     2022    30   157   696    62 0.311   131   133    16     3    28
## 2 Aaro~ NYY     2024    32   158   704    58 0.322   144   122    10     0    36
## 3 Matt~ ATL     2023    29   162   720    54 0.283   139   127     1     0    27
## 4 Shoh~ LAD     2024    29   159   731    54 0.310   130   134    59     4    38
## 5 Kyle~ PHI     2023    30   160   720    47 0.197   104   108     0     2    19
## 6 Pete~ NYM     2023    28   154   658    46 0.217   118    92     4     1    21
## # i 39 more variables: `3B` <dbl>, BB <dbl>, IBB <dbl>, SO <dbl>,
## #   BB_perK <dbl>, HBP <dbl>, BABIP <dbl>, ISO <dbl>, wRC_plus <dbl>,
## #   WAR <dbl>, BsR <dbl>, Def <dbl>, Off <dbl>, WPA <dbl>, GB <dbl>, FB <dbl>,
## #   `HR/FB` <dbl>, IFH <dbl>, Pull <dbl>, Cent <dbl>, Oppo <dbl>, Soft <dbl>,
## #   `Med%` <dbl>, Hard <dbl>, Clutch <dbl>, Fld <dbl>, Rep <dbl>, Pos <dbl>,
## #   wOBA <dbl>, O_Swing <dbl>, `Z-Swing` <dbl>, Swing <dbl>, Zone <dbl>,
## #   Barrel <dbl>, Hard_hitPercentage <dbl>, xBA <lgl>, NameASCII <chr>, ...
```

2) Linear Model

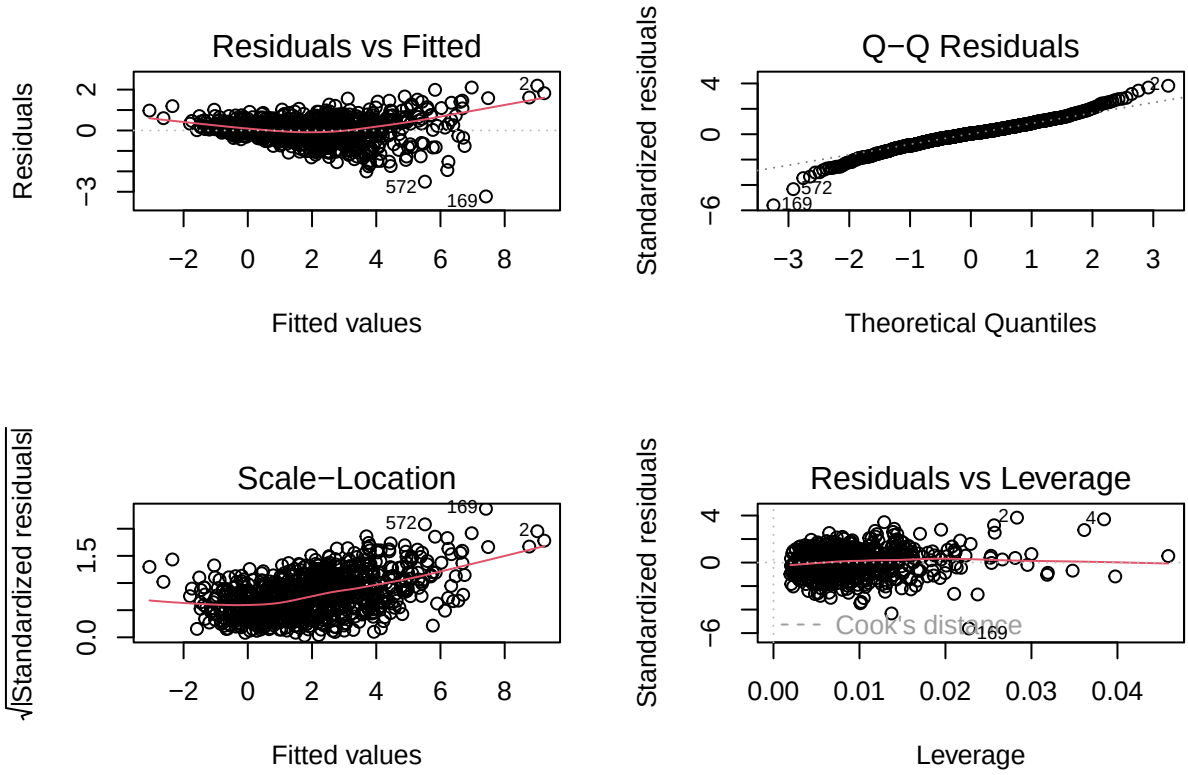
```
hitterModel <- lm(WAR ~ wRC_plus + Hard_hitPercentage + BB_perK + ISO + BsR +
                    Def, data = MLB_data)
summary(hitterModel)
```

```
##
## Call:
## lm(formula = WAR ~ wRC_plus + Hard_hitPercentage + BB_perK +
##     ISO + BsR + Def, data = MLB_data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2283 -0.3087  0.0279  0.3358  2.1929
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -5.524733   0.128210 -43.091  < 2e-16 ***
## wRC_plus            0.061805   0.001508  40.995  < 2e-16 ***
## Hard_hitPercentage  1.533981   0.375633   4.084 4.85e-05 ***
## BB_perK             0.952699   0.136562   6.976 6.10e-12 ***
## ISO                 1.151037   0.668402   1.722   0.0854 .
## BsR                 0.131468   0.007769  16.923  < 2e-16 ***
## Def                 0.102362   0.002439  41.963  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5828 on 849 degrees of freedom
## Multiple R-squared:  0.9052, Adjusted R-squared:  0.9046
## F-statistic:  1351 on 6 and 849 DF,  p-value: < 2.2e-16
```

3): Check diagnostic plots

```
par(mfrow = c(2, 2))
plot(hitterModel)
```



Residuals vs Fitted A curve in the Residuals vs. Fitted (R vs. F) plot suggests that the relationship between the response variable and the predictors may not be strictly linear. In a well-fitted linear model, the residuals should be randomly scattered around zero with a fairly horizontal red line, indicating that the linearity assumption is valid. However, if the residuals display a clear curve or pattern, it implies that the model is not capturing all the underlying structure of the data, possibly because the relationship between the predictors and the response is non-linear. The R v F plot does appear affected by some outlier points that we will look into further detail.

To address this issue, a transformation of the response variable (or predictors) may be necessary. Transformations such as log, square root, or polynomial terms can help linearize the relationship between the variables, making the model more appropriate for the data.

Q-Q Plot The residuals generally follow a normal distribution across most of their range since they lie near the diagonal line in the middle portion of the plot, however they experience deviation at the ends meaning the residuals might have outliers. This may be a sign of non-normality which could impact hypothesis testing if not transformed

Scale-Location The scale-location plot primarily tests the assumption of homoscedasticity in linear regression, which states that the variance of the residuals should be constant across all levels of the fitted values. The points on the plot seem to form a curved shape as fitted values increase, bringing constant variance into question. A transformation seems likely needed.

Residuals vs Leverage Leverage measures the potential influence of a data point based on how far its predictor values are from the mean of all predictor values. Most Points Cluster Around the Horizontal Line

(0):

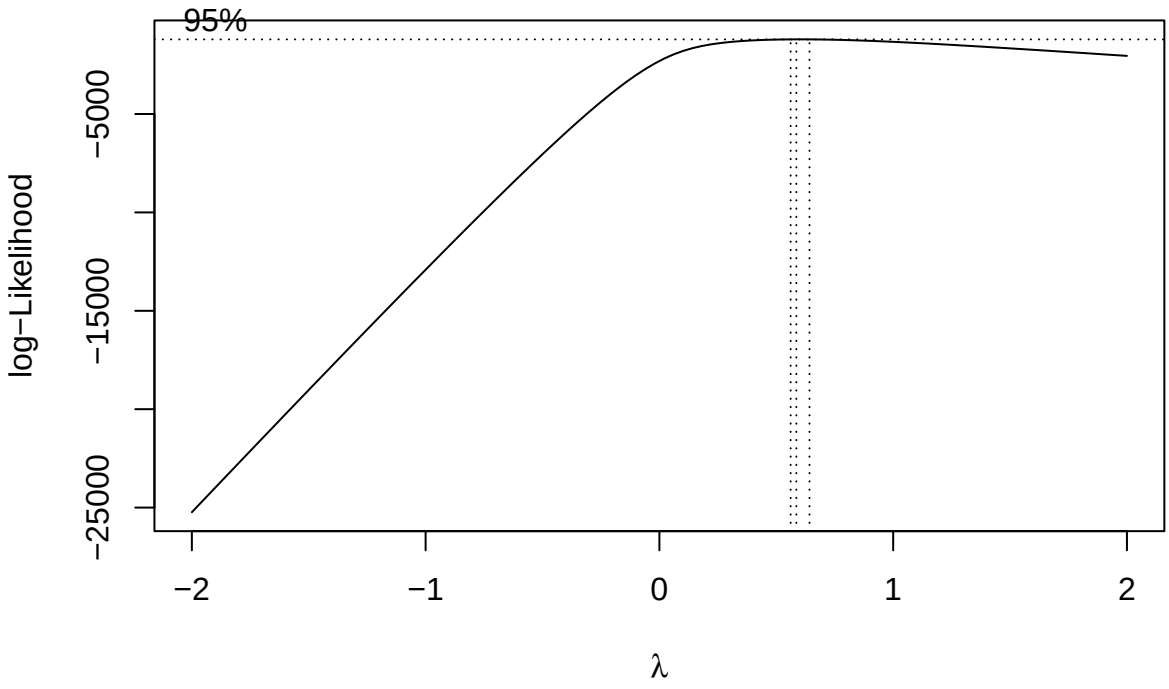
This plot indicates that the majority of observations have small residuals and low leverage as they cluster around the horizontal line, contributing little to the model's instability. However we appear to see 10 or so point on the right side of the plot, indicating high leverage over the model. These points could skew the regression results. Only one point, labeled 169 (Andres Gimenez), appears to be past the line of Cook's Distance. As this data is from a very controlled league and the player qualifies for appropriate playing time, we can really only view him as an outlier and won't undergo any further steps to remove his data point.

4) Optimal Lamda Transformation

```
# Add the shifted response to the dataset
MLB_data <- MLB_data %>%
  mutate(ImpactScore = WAR + abs(min(WAR)) + 1e-6)

# Fit the linear model with the shifted response
hitterModel <- lm(ImpactScore ~ wRC_plus + Hard_hitPercentage + BB_perK +
                    ISO + BsR + Def, data = MLB_data)

# Load the necessary library
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Generate the Box-Cox plot and save the result in boxcox_result
boxcox_result <- boxcox(hitterModel, lambda = seq(-2, 2, by = 0.05))
```




```
# Extract the optimal lambda (the value of lambda that maximizes the log-likelihood)
optimal_lambda <- boxcox_result$x[which.max(boxcox_result$y)]
```

```
# Print the optimal lambda
cat("Optimal Lambda:", optimal_lambda, "\n")
```

```
## Optimal Lambda: 0.5858586
```

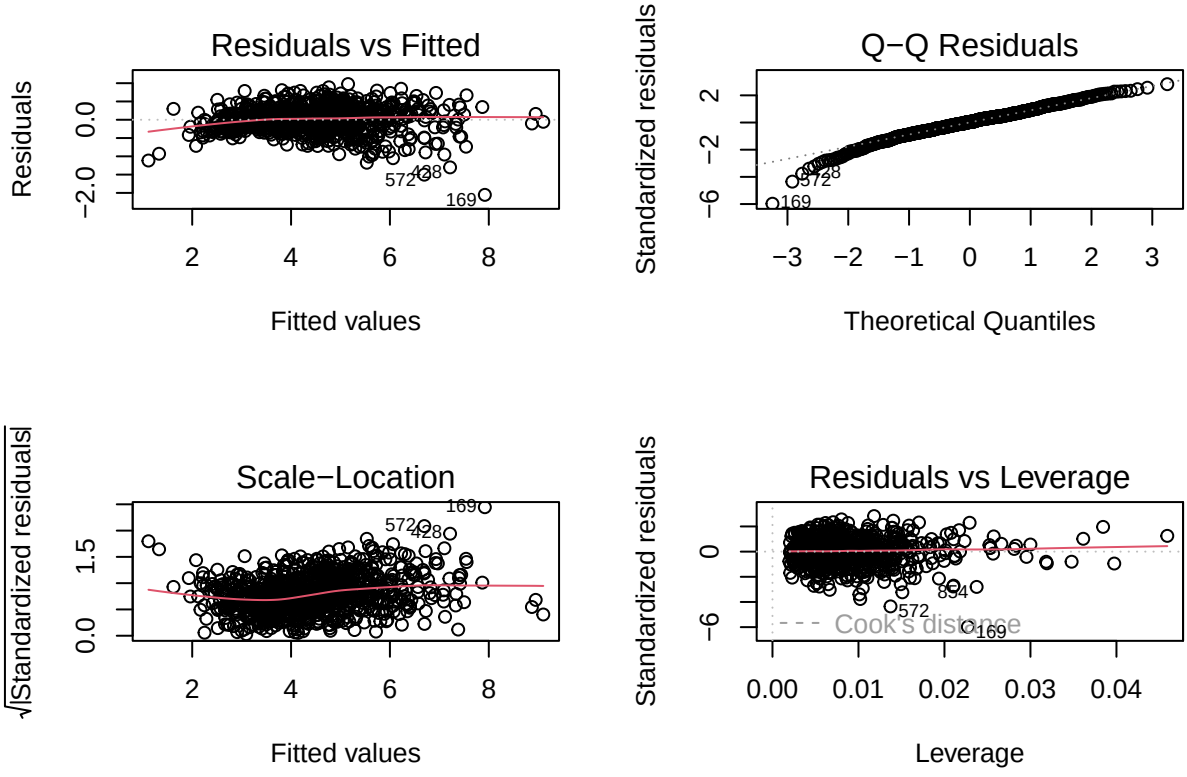
I am choosing to increase every WAR value by the absolute value of the minimum point + an incredibly small number to ensure every point is slightly above zero before undergoing box cox transformation. I decided to multiply the transformed value by 2 to scale the response value more appropriately

```
MLB_data <- MLB_data %>%
  mutate(ImpactScore = ((WAR + abs(min(WAR)) + 1e-6) ^ optimal_lambda)*2)

# Fit the linear model with the transformed response
hitterModel_transformed <- lm(ImpactScore ~ wRC_plus + Hard_hitPercentage +
                                BB_perK + ISO + BsR + Def, data = MLB_data)

# Summary of the transformed model
summary(hitterModel_transformed)
```

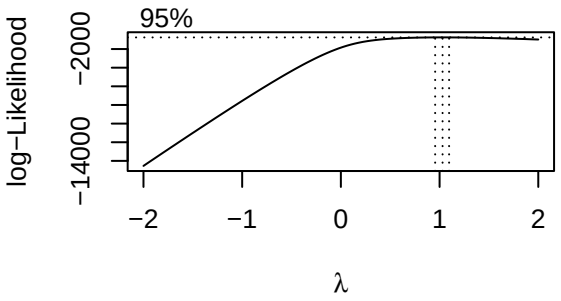
```
##
## Call:
```

```
## lm(formula = ImpactScore ~ wRC_plus + Hard_hitPercentage + BB_perK +
##     ISO + BsR + Def, data = MLB_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05350 -0.20607  0.01495  0.21193  0.97533
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.3173214  0.0764602  -4.150 3.66e-05 ***
## wRC_plus             0.0420130  0.0008991  46.728  < 2e-16 ***
## Hard_hitPercentage   0.5660036  0.2240146   2.527   0.0117 *
## BB_perK              0.4498793  0.0814411   5.524 4.41e-08 ***
## ISO                  0.2837734  0.3986113   0.712   0.4767
## BsR                  0.0797698  0.0046329  17.218  < 2e-16 ***
## Def                  0.0685990  0.0014547  47.156  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3476 on 849 degrees of freedom
## Multiple R-squared:  0.9191, Adjusted R-squared:  0.9185
## F-statistic:  1607 on 6 and 849 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(hitterModel_transformed)
```



```r
# Generate the Box-Cox plot and save the result in boxcox_result
boxcox_result <- boxcox(hitterModel_transformed, lambda = seq(-2, 2, by = 0.1))
```



Residuals vs Fitted The residuals are randomly scattered around zero, without any discernible pattern, suggesting that the linearity assumption is valid and that the relationship between the predictors and the response is well captured by the model.

Q-Q Plot The points closely follow the diagonal line, suggesting that the residuals are normally distributed, which validates the assumption of normality of residuals. There are still some outlier points towards the left of the graph but fixed the right side.

Scale-Location The points form a fairly horizontal line with roughly equal spread across all levels of fitted values. There's no funnel or pattern, it's a sign of good homoscedasticity. The outliers don't seem to have as much effect on the plot as they did previously.

Residuals vs Leverage Most of the points lie within a reasonable range from the horizontal line. The Cook's distance line only flags observation 169, which we will look further into to see if they are overly influential to the model.

Box Cox The Box Cox confidence interval includes $= 1$, meaning that the data does not require another transformation, as $= 1$ suggests that the scale of the response variable is appropriate for modeling. In other words, the response variable now follows a roughly normal distribution and does not need to be transformed.

Diagnostic Conclusion As the diagnostic plots were greatly improved to satisfy the assumptions of linearity after our shift to positive response values with a subsequent optimal lambda shift. We will then multiply the lambda shift by a constant 2 for better scaling of the response variable. We will now move on to hypothesis testing with the response variable Impact Score. A shift from WAR.

5) Linear Model Results

$R^2 = .919$: This value means 91.9% of the variability in our response variable, Impact Score, can be explained by our predictor variables. The Residual Standard Error of .174 indicates that on average, the observed values deviate from the predicted values by .17 units, which given a range of. A high F-value of 1607 indicates that the model is significantly better at explaining variability of the response variable compared to just using the mean of the response variable. and Our model seems to successfully predict IS. We will next go over our hypothesis tests to see if any of our predictors should be dropped.

To perform hypothesis tests for each coefficient in a regression, the null hypothesis (H_0) and the alternative hypothesis (H_1) are typically:

- **Null Hypothesis (H_0)**: The coefficient is equal to 0 (i.e., the predictor has no effect on the response variable).
- **Alternative Hypothesis (H_1)**: The coefficient is not equal to 0 (i.e., the predictor has an effect on the response variable).

The **t-values** and corresponding **p-values** allow us to test these hypotheses.

Results for Each Variable

Hypothesis Testing

Using $\alpha = .05$ we will conduct a t-test on the intercept and all of the predictor variables H_0: The coefficient is equal to 0 (the predictor has no effect on the response variable). H_1: The coefficient is not equal to 0 (the predictor has a significant effect on the response variable).

1. (Intercept)

- H_0: Intercept = 0
- H_1: Intercept 0
- **t-value** = -4.150
- **p-value** = $3.66e-05$

Conclusion: Reject H_0 (Intercept is significantly different from 0).

---

2. wRC_plus

- H_0: Coefficient of `wRC_plus` $= 0$
- H_1: Coefficient of `wRC_plus` $\neq 0$
- **t-value** $= 46.728$
- **p-value** $= < 2e-16$

Conclusion: Reject H_0. `wRC_plus` has a significant positive effect on the response.

---

3. Hard_hitPercentage

- H_0: Coefficient of `Hard_hitPercentage` $= 0$
- H_1: Coefficient of `Hard_hitPercentage` $\neq 0$
- **t-value** $= 2.527$
- **p-value** $= 0.0117$

Conclusion: Reject H_0. `Hard_hitPercentage` has a significant positive effect on the response.

---

4. BB_perK

- H_0: Coefficient of `BB_perK` $= 0$
- H_1: Coefficient of `BB_perK` $\neq 0$
- **t-value** $= 5.524$
- **p-value** $= 4.41e-08$

Conclusion: Reject H_0. `BB_perK` has a significant positive effect on the response.

---

5. ISO

- H_0: Coefficient of `ISO` $= 0$
- H_1: Coefficient of `ISO` $\neq 0$
- **t-value** $= 0.712$
- **p-value** $= 0.4767$

Conclusion: Fail to reject H_0 at the 5% significance level. `ISO` has a weak positive effect, but it is only marginally significant (at the 10% level).

---

6. BsR

- H_0: Coefficient of `BsR` $= 0$
- H_1: Coefficient of `BsR` $\neq 0$
- **t-value** $= 17.218$
- **p-value** $= < 2e-16$

Conclusion: Reject H_0. `BsR` has a significant positive effect on the response.

---

7. Def

- H_0: Coefficient of `Def` $= 0$
- H_1: Coefficient of `Def` $\neq 0$
- **t-value** $= 47.156$
- **p-value** $= < 2e-16$

Conclusion: Reject H_0. `Def` has a significant positive effect on the response.

Summary of Hypothesis Tests

Predictor	t-value	p-value	Conclusion
Intercept	-4.150	3.66e-05	Reject H_0
`wRC_plus`	46.728	$<$ 2e-16	Reject H_0
`Hard_hitPercentage`	2.527	0.0117	Reject H_0
`BB_perK`	5.524	$4.41e-08$	Reject H_0
`ISO`	0.712	0.4767	Fail to reject H_0
`BsR`	17.218	$< 2e-16$	Reject H_0
`Def`	47.156	$< 2e-16$	Reject H_0

From our hypothesis tests, I have decided to drop the ISO predictor from our model and test a few other replacement options. To keep some sort of reflection of a hitter's power in the model, I have decided to replace ISO with HR (Home Runs).

```
# Fit the linear model with the transformed response
hitterModel2 <- lm(ImpactScore ~ wRC_plus + Hard_hitPercentage + BB_perK +
                     HR + BsR + Def, data = MLB_data)

# Summary of the transformed model
summary(hitterModel2)
```

```
##
## Call:
## lm(formula = ImpactScore ~ wRC_plus + Hard_hitPercentage + BB_perK +
##     HR + BsR + Def, data = MLB_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78396 -0.16336  0.01294  0.16696  0.95022
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.3262063  0.0704747   4.629 4.25e-06 ***
## wRC_plus             0.0356221  0.0006748  52.789  < 2e-16 ***
## Hard_hitPercentage -0.6662779  0.1852703  -3.596 0.000341 ***
## BB_perK              0.5826406  0.0655299   8.891  < 2e-16 ***
## HR                   0.0317336  0.0016436  19.308  < 2e-16 ***
## BsR                  0.0801007  0.0038608  20.747  < 2e-16 ***
## Def                  0.0688284  0.0012131  56.740  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2898 on 849 degrees of freedom
## Multiple R-squared:  0.9437, Adjusted R-squared:  0.9433
## F-statistic:  2374 on 6 and 849 DF,  p-value: < 2.2e-16
```

5. HR

- H_0: Coefficient of $\beta_4 = 0$
- H_1: Coefficient of $\beta_4 \neq 0$
- **t-value** $= 19.308$

- **p-value** $= < 2e - 16$

Conclusion: Reject H_0 at the 5% significance level. HR has a significant positive effect on the response.

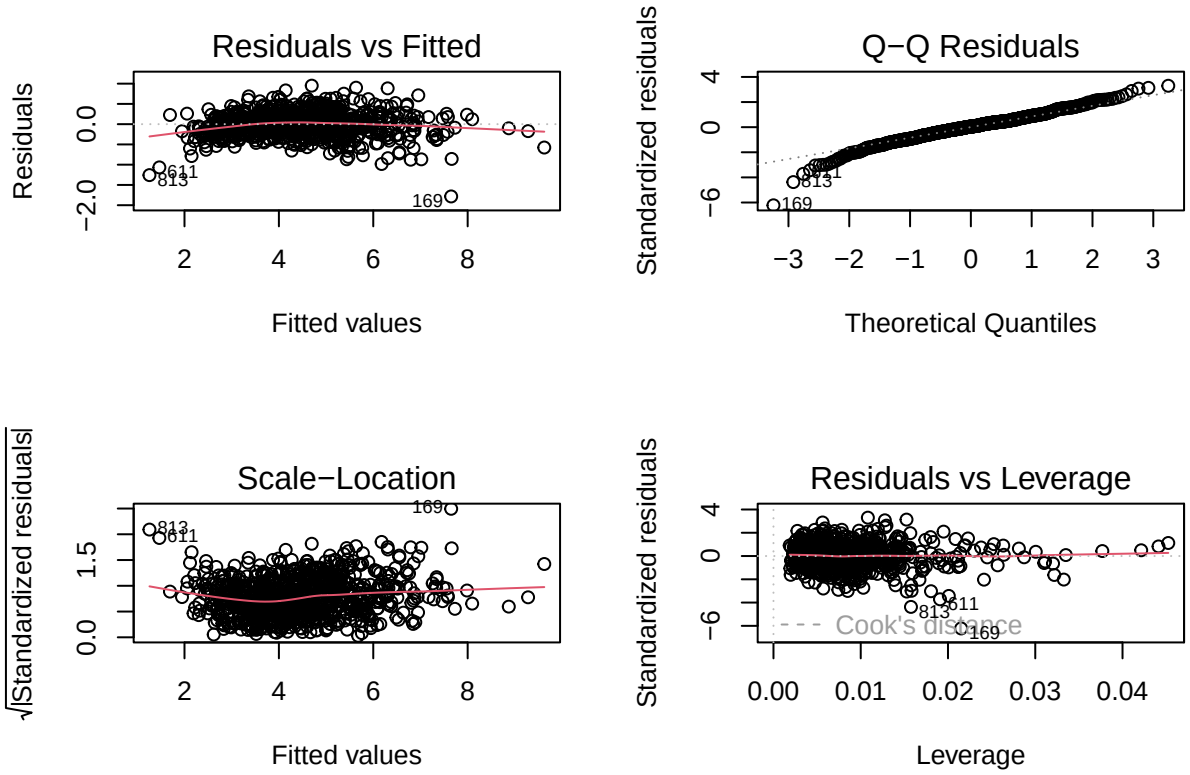
Confidence Intervals

```r
IS_conf =  confint(hitterModel2, level = .95)
print(IS_conf)
```

```
##                           2.5 %      97.5 %
## (Intercept)          0.18788120  0.46453135
## wRC_plus             0.03429760  0.03694652
## Hard_hitPercentage -1.02991950 -0.30263635
## BB_perK              0.45402093  0.71126017
## HR                   0.02850769  0.03495951
## BsR                  0.07252295  0.08767853
## Def                  0.06644750  0.07120936
```

We are 95% confident that the true value of β_0 is between (.188, .465), β_1 is between (.034, .037), β_2 is between (-1.030, .303), β_3 is between (.454, .711), β_4 is between (.029, .035), β_5 is between (.073, .087), and β_6 is between (.066, .071).

```
par(mfrow = c(2, 2))
plot(hitterModel2)
```



Same conclusions as before replacement of ISO to HR

Model 2 Conlusion:

With the dropping of ISO in exchange for HR, the model's R_2 value has increased to .944, explaining about 3% more of the variability in our response variable, Impact Score, the transformation of WAR.

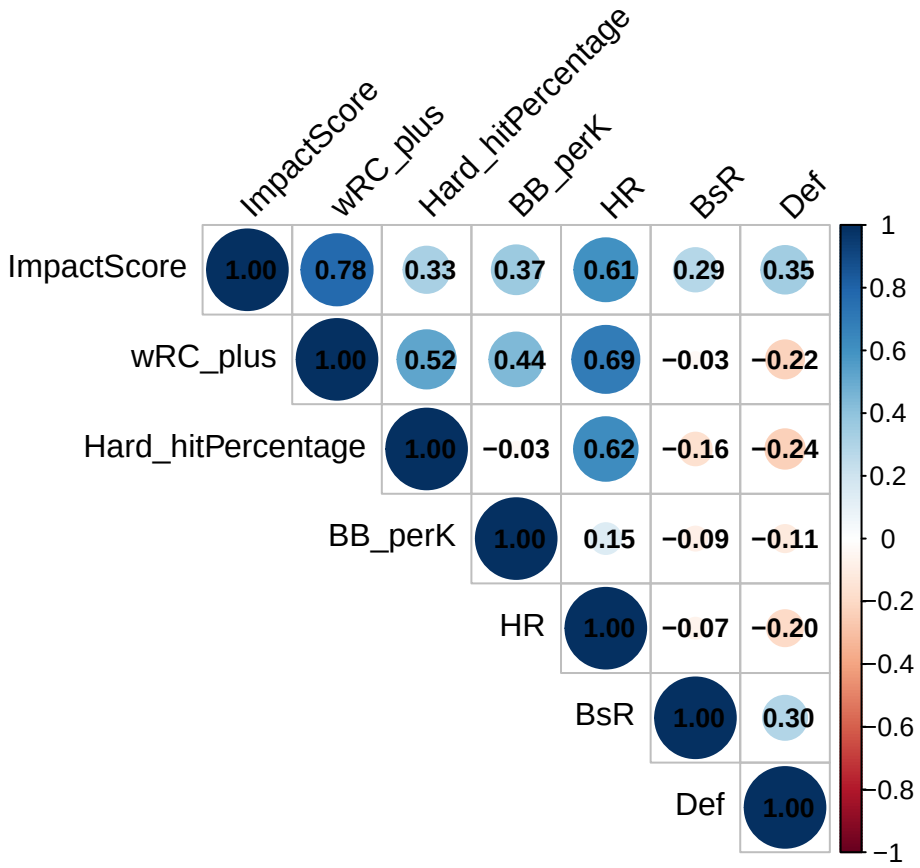
Data Visualizations

Now that we have our complete model we can create some data visualizations of our predictor and response variables, as well as data that plays a role in the calculations of our predictors. This Correlation Matrix is between the predictors and the ImpactScore from 2022-24.

```
predictors <- MLB_data[, c("ImpactScore", "wRC_plus", "Hard_hitPercentage",
                           "BB_perK", "HR", "BsR", "Def")]
cMatrix = cor(predictors)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
# Create a correlation plot
corrplot(cMatrix, method = "circle", type = "upper", tl.col = "black",
         addCoef.col = "black",   number.cex = 0.8,
  number.digits = 2,  tl.srt = 45)
```

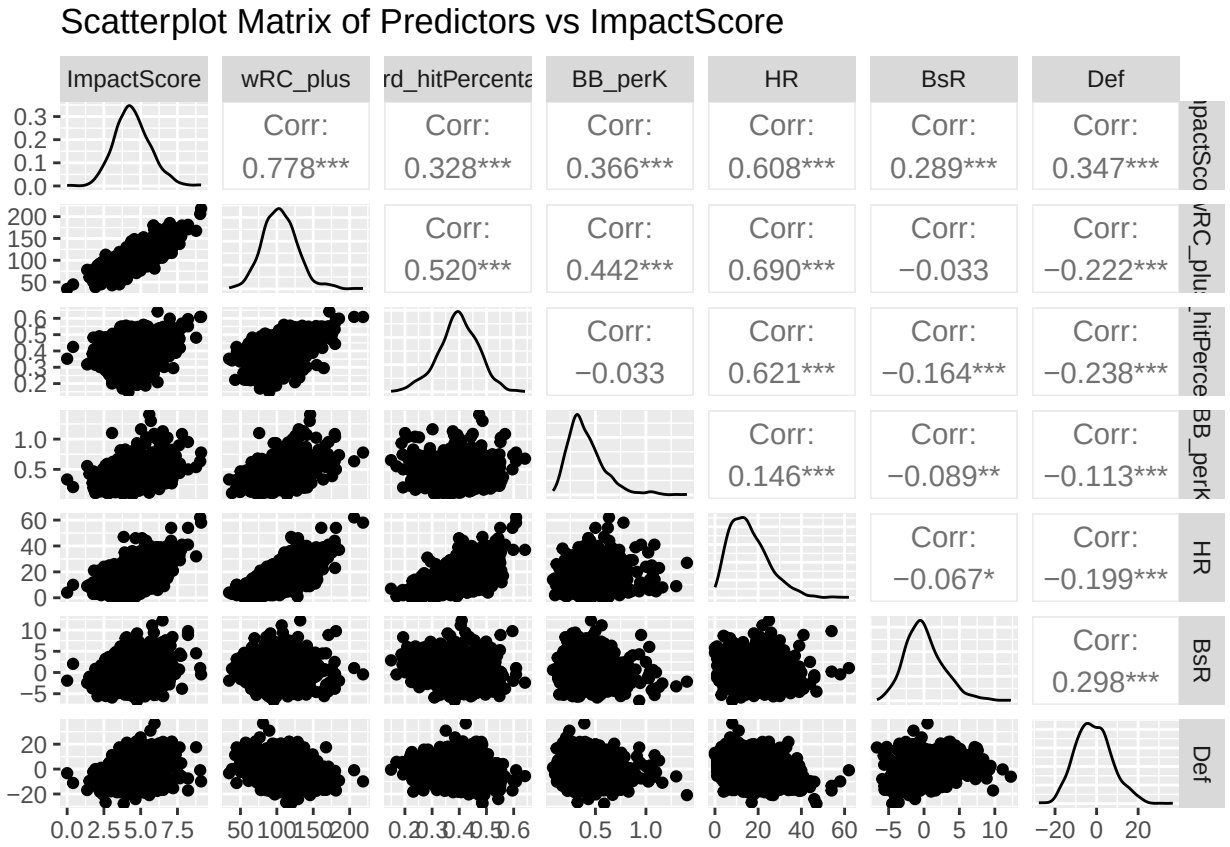


Notably wRC+, offensive production against league average, has the highest correlation with the response, while BsR, baserunning runs created, has the lowest. There are some interesting negative correlations between defense and offense. As you might expect, HR are correlated with offensive production and hard hit perecentage.

```
# Load the GGally package
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
# Example data (replace with your actual dataset)

subset_data <- MLB_data[, c("ImpactScore", "wRC_plus", "Hard_hitPercentage",
                            "BB_perK", "HR", "BsR", "Def")]

# Create the scatterplot matrix
ggpairs(subset_data, title = "Scatterplot Matrix of Predictors vs ImpactScore")
```



Summary Statistics of Each Predictor Variable to Impact Score since 2022 Season

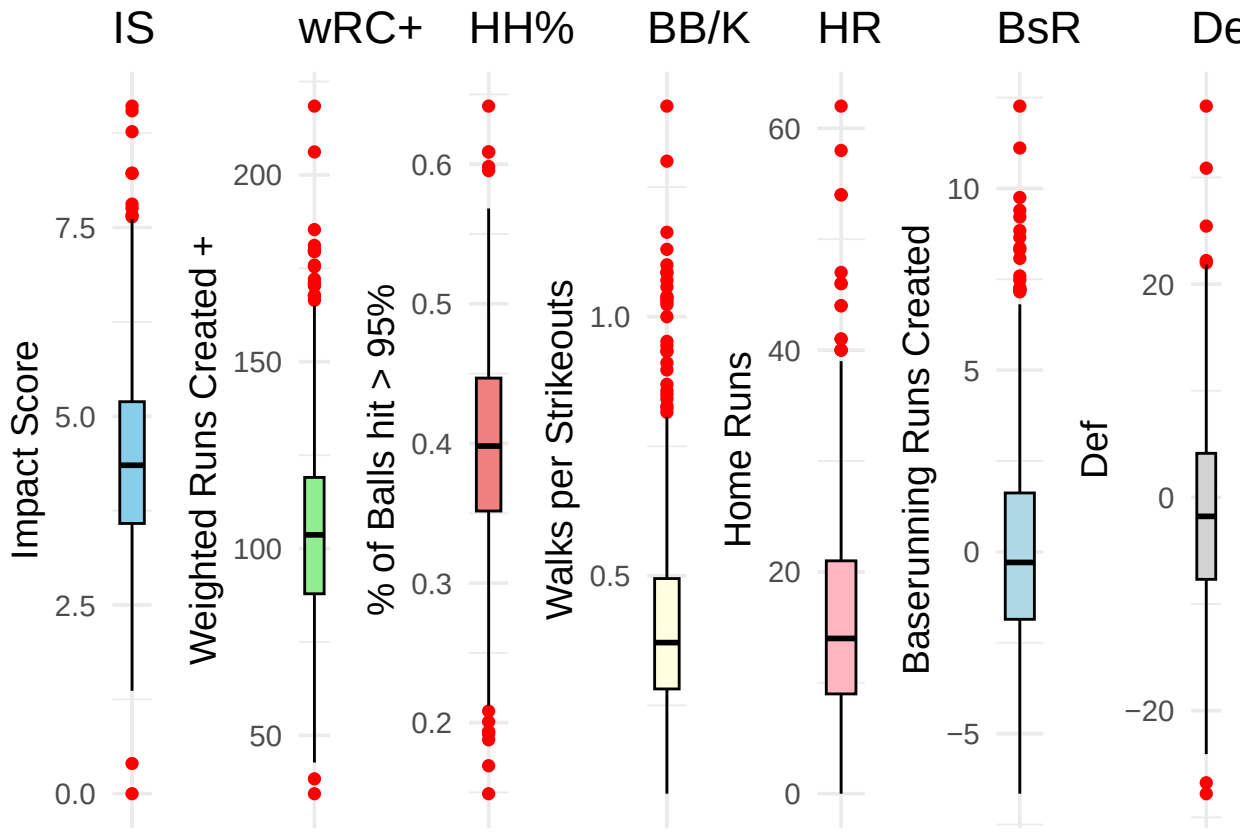
The following box plots show the distribution of the 6 predictors and response variable in order to get a better idea of the scale of each and how far apart the superstar players are from league average.

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Example using your MLB_data
# Create a box plot with proper whiskers for each variable

p1 <- ggplot(MLB_data, aes(x = "", y = ImpactScore)) +
  geom_boxplot(fill = "skyblue", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "IS", y = "Impact Score") +
```

```
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p2 <- ggplot(MLB_data, aes(x = "", y = wRC_plus)) +
  geom_boxplot(fill = "lightgreen", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "wRC+", y = "Weighted Runs Created +") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p3 <- ggplot(MLB_data, aes(x = "", y = Hard_hitPercentage)) +
  geom_boxplot(fill = "lightcoral", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "HH%", y = "% of Balls hit > 95%") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p4 <- ggplot(MLB_data, aes(x = "", y = BB_perK)) +
  geom_boxplot(fill = "lightyellow", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "BB/K", y = "Walks per Strikeouts") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p5 <- ggplot(MLB_data, aes(x = "", y = HR)) +
  geom_boxplot(fill = "lightpink", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "HR", y = "Home Runs") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p6 <- ggplot(MLB_data, aes(x = "", y = BsR)) +
  geom_boxplot(fill = "lightblue", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "BsR", y = "Baserunning Runs Created") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

p7 <- ggplot(MLB_data, aes(x = "", y = Def)) +
  geom_boxplot(fill = "lightgray", color = "black",
               outlier.shape = 16, outlier.size = 2, outlier.colour = "red") +
  labs(title = "Def", y = "Def") +
  theme_minimal(base_size = 14) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())

# Arrange the plots side by side with proper spacing and adjusted title position
grid.arrange(p1, p2, p3, p4, p5, p6, p7, ncol = 7, widths = rep(1, 7),
             layout_matrix = matrix(1:7, ncol = 7, byrow = TRUE))
```



The red dots symbolize outliers and when on top of the graph superstar players. You can clearly see Aaron Judge's 62 and 58 home run season.

Below is the Top 10 performers of Impact score over the last 3 seasons

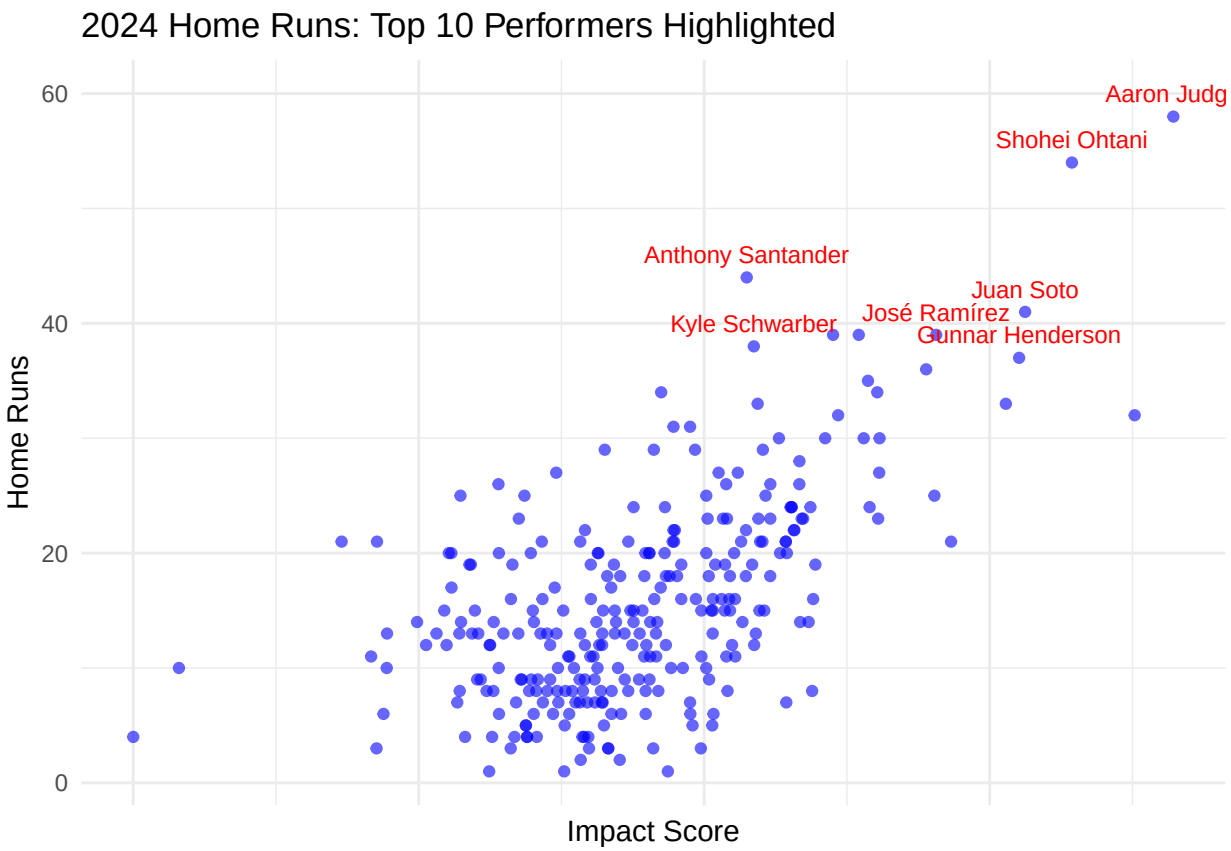
```
Impact_Leaders <- MLB_data %>%
  arrange(desc(ImpactScore)) %>% # Sort by Impact Score (descending order)
  mutate(rank = row_number()) %>% # Create a rank column
  mutate(ImpactScore = round(ImpactScore, 1), wRC_plus = round(wRC_plus, 0),
         BsR = round(BsR, 0),
         Def = round(Def, 0), Hard_hitPercentage = round(Hard_hitPercentage, 2),
         BB_perK = round(BB_perK, 2))%>%
  dplyr::select(rank, Name, Team, Season, ImpactScore, wRC_plus, HR, Def, BsR,
                Hard_hitPercentage, BB_perK)
# Select relevant columns

# View the top 10 leaders
Impact_Leaders_Top10 <- head(Impact_Leaders, 10)
print(Impact_Leaders_Top10)
```

```
## # A tibble: 10 x 11
##     rank Name             Team  Season ImpactScore wRC_plus    HR   Def   BsR
##    <int> <chr>            <chr>  <dbl>       <dbl>    <dbl> <dbl> <dbl> <dbl>
##  1     1 Aaron Judge      NYY     2024         9.1      218    58   -10     0
##  2     2 Aaron Judge      NYY     2022         9        206    62    -1     1
##  3     3 Bobby Witt Jr.   KCR     2024         8.8      168    32    18     5
##  4     4 Shohei Ohtani    LAD     2024         8.2      181    54   -17    10
```

```
##  5     5 Ronald Acuña Jr. ATL     2023         8.2      171    41    -8     9
##  6     6 Juan Soto        NYY     2024         7.8      180    41    -6    -4
##  7     7 Gunnar Henderson BAL     2024         7.8      155    37     5     4
##  8     8 Freddie Freeman  LAD     2023         7.7      162    29    -7     4
##  9     9 Francisco Lindor NYM     2024         7.6      137    33    17     4
## 10    10 Mookie Betts     LAD     2023         7.6      166    39    -4     1
## # i 2 more variables: Hard_hitPercentage <dbl>, BB_perK <dbl>
```

From the best individual 10 seasons from 2022-2024 based on Impact Score, we are able to see some of the best stars of the game. Ohtani and Judge were able to hit the ball hard in 60% of their plate appearances. Ohtani, who just became baseball's first 50/50 member, 50 HR and 50 SB in one season, created 81% more runs than average while creating 10 more runs than average through baserunning alone. While looking at the Def column, you can see the players who excelled are star shortstops in Bobby Witt, Gunnar Henderson, and Francisco Lindor. A quick note on BB/K: This is an extremely significant statistic evaluating plate discripline, the split-second reaction to take or swing at a ball off the plate can be highly influential to not only that player's at bat, but the team's success. The higher the ratio is, the more likely that player is swinging at hittable pitches and taking difficult ones. As you can see in this leaderboard, a lot of these superstar players have great BB/K including Juan Soto, who has famously garnered a reputation in recent years for his superb discipline and swagger while taking pitches. We've also seen in the playoffs how importance it is to work walks and put the ball in play when facing elite pitching.

```r
# Filter and rank players based on home runs for 2024
Top10_HR <- MLB_data %>%
  filter(Season == 2024) %>% # Filter for the 2024 season
  arrange(desc(HR)) %>%      # Sort by Home Runs in descending order
  mutate(rank = row_number()) %>% # Create a rank column
  head(10)                   # Select the top 10 performers

# Scatterplot of all players, labeling the top 10 home run performers
ggplot(MLB_data %>% filter(Season == 2024), aes(x = ImpactScore, y = HR)) +
  geom_point(color = "blue", alpha = 0.6) +                    # Scatterplot points
  geom_text(data = Top10_HR, aes(label = Name),                # Add labels for top 10
            nudge_y = 2, check_overlap = TRUE, size = 3, color = "red") +
  labs(
    title = "2024 Home Runs: Top 10 Performers Highlighted",
    x = "Impact Score",
    y = "Home Runs"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```



This scatterplot further visualizes the positive correlation between Impact Score and Home runs highlighting the top 10 homerun hitters of the 2024 season. Besides the obvious thrill of watching your favorite player smash a ball 450 feet, HR are important to team performance during offensive slumps. When teams are struggling to string hits and walks together in order to score runs, a quick solo shot can go along way in jump starting an offense.

6) Offeason Evaluation of Key Players using Prediction Model

This Winter for the Phillies will be highlighted by roster turnover in aim to stay true World Series Contenders. The organization reportedly offered Alec Bohm and Nick Castellanos in a proposed trade that was reject by the Astros, and have been reported to have interest in free agents Anthony Santander, Teoscar Hernandez, and Alex Bregman. Using the hitterModel we will predict Impact Scores on these 5 notable players for the 2025 season using the averages of their predictors from 2022-2024.

```
# View five players of Note to predict their 2025 Impact Score
alec_bohm_data <- MLB_data %>% filter(grepl("Alec Bohm", Name,
                                            ignore.case = TRUE))
nick_castellanos = MLB_data %>% filter(grepl("Nick Castellanos",
                                             Name, ignore.case = TRUE))
anthony_santander = MLB_data %>% filter(grepl("Anthony Santander",
                                              Name, ignore.case = TRUE))
teoscar_hernandez = MLB_data %>% filter(grepl("Teoscar Hernández",
                                              Name, ignore.case = TRUE))
alex_bregman =  MLB_data %>% filter(grepl("Alex Bregman", Name,
                                          ignore.case = TRUE))
```

```r
# Calculate average predictors
library(dplyr)
bohm_predictors <- alec_bohm_data %>%
  dplyr::select(Name, wRC_plus, Hard_hitPercentage, BB_perK, HR, BsR, Def) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
print(bohm_predictors)
```

```
## # A tibble: 1 x 7
##    Name wRC_plus Hard_hitPercentage BB_perK    HR   BsR   Def
##   <dbl>    <dbl>              <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1    NA     106.              0.433   0.398    16 -2.88 -1.98
```

```r
castellanos_predictors <- nick_castellanos %>%
  dplyr::select(Name, wRC_plus, Hard_hitPercentage, BB_perK, HR, BsR, Def) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
```

```r
print(castellanos_predictors)
```

```
## # A tibble: 1 x 7
##    Name wRC_plus Hard_hitPercentage BB_perK    HR   BsR   Def
##   <dbl>    <dbl>              <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1    NA     103.              0.384   0.238  21.7 -2.70 -16.8
```

```r
santander_predictors <- anthony_santander %>%
  dplyr::select(Name, wRC_plus, Hard_hitPercentage, BB_perK, HR, BsR, Def) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
```

```
print(santander_predictors)
```

```
## # A tibble: 1 x 7
##    Name wRC_plus Hard_hitPercentage BB_perK    HR   BsR   Def
##   <dbl>    <dbl>              <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1    NA     124.              0.436   0.421    35 -1.70 -10.9
```

```
hernandez_predictors <- teoscar_hernandez %>%
  dplyr::select(Name, wRC_plus, Hard_hitPercentage, BB_perK, HR, BsR, Def) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
```

```
print(hernandez_predictors)
```

```
## # A tibble: 1 x 7
##    Name wRC_plus Hard_hitPercentage BB_perK    HR   BsR   Def
##   <dbl>    <dbl>              <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1    NA     123.              0.496   0.229    28 -1.09 -8.99
```

```
bregman_predictors <- alex_bregman %>%
  dplyr::select(Name, wRC_plus, Hard_hitPercentage, BB_perK, HR, BsR, Def) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
```

```
print(bregman_predictors)
```

```
## # A tibble: 1 x 7
##    Name wRC_plus Hard_hitPercentage BB_perK    HR   BsR   Def
##   <dbl>    <dbl>              <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1    NA     127.              0.386   0.900  24.7 -3.28  5.73
```

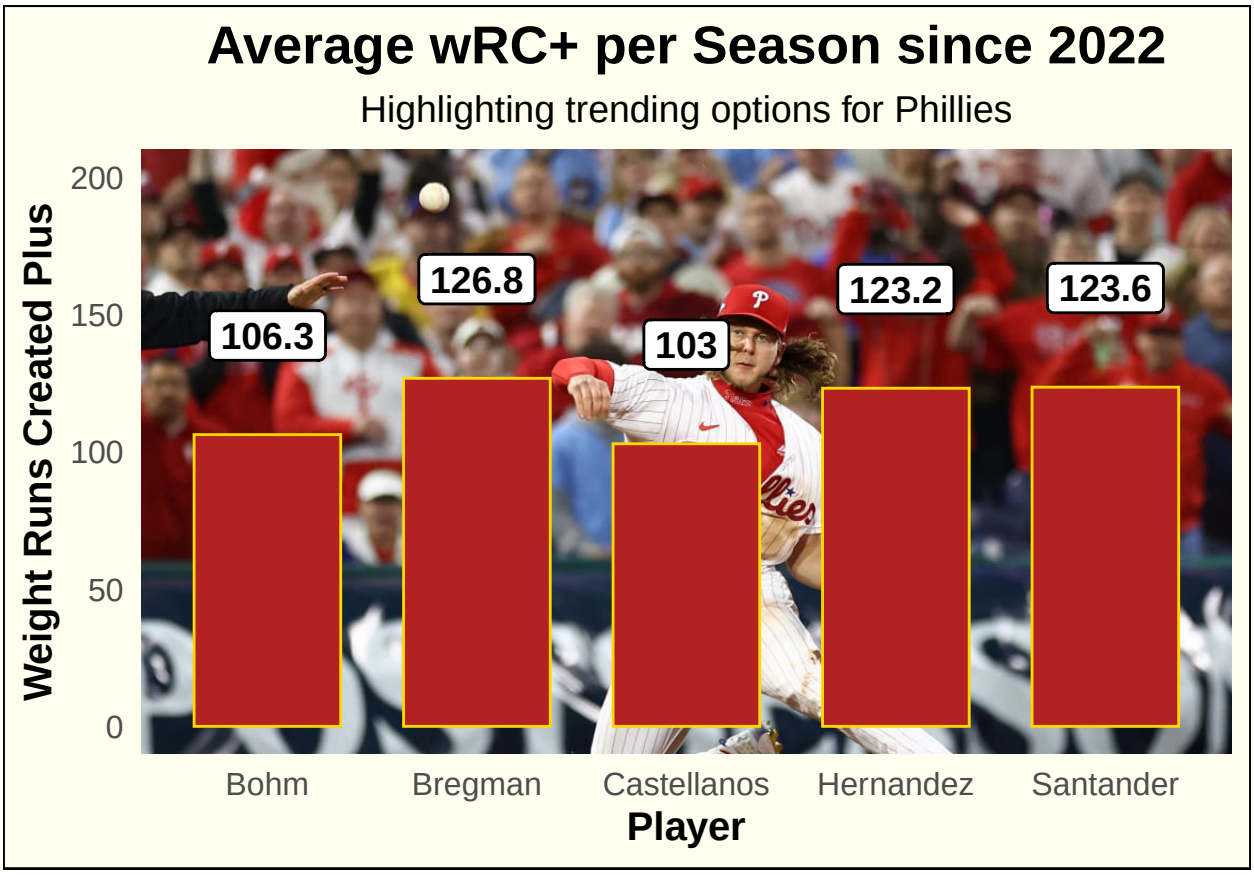
Bar Chart of wRC+

Let's quickly view the comparison of one of the averaged predictors wRC+ as we recently discovered the offensive production analysis tool had the strongest correlation with Impact Score.

```
library(ggplot2)
library(ggimage)
library(grid)
library(jpeg)
```

```r
# Load baseball field background
field_image <- rasterGrob(
  readJPEG("ab1.jpeg"),
  width = unit(1, "npc"),
  height = unit(1, "npc"),
  interpolate = TRUE
)

# Player data with Impact Score (IS) values
player_IS <- c("Bohm" = bohm_predictors$wRC_plus,
               "Castellanos" = castellanos_predictors$wRC_plus,
               "Santander" = santander_predictors$wRC_plus,
               "Hernandez" = hernandez_predictors$wRC_plus,
               "Bregman" = bregman_predictors$wRC_plus)

# Convert to a data frame for ggplot and ensure Player is a character vector
graph_data <- data.frame(
  Player = as.character(names(player_IS)),  # Ensure Player is a character vector
  WAR = as.numeric(player_IS)
)
graph_data$Player <- gsub("\\.1$", "", graph_data$Player)
# Create the bar chart with adjusted Y-axis, full player names, and horizontal text
ggplot(graph_data, aes(x = Player, y = WAR)) +
  # Add background image
  annotation_custom(field_image, xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = Inf) +

  # Add bars for players
  geom_col(width = 0.7, fill = "firebrick", color = "gold", linewidth = .5) +

  # Add text labels for WAR values
  geom_label(aes(label = round(WAR, 1)),
             vjust = -1.5, size = 5, color = "black",
             fill = "white", fontface = "bold", label.size = 0.5, label.padding =
               unit(0.25, "lines")) +

  # Add titles and labels
  labs(
    title = "Average wRC+ per Season since 2022",
    subtitle = "Highlighting trending options for Phillies",
    x = "Player",
    y = "Weight Runs Created Plus"
  ) +

  # Set theme
  theme_minimal(base_size = 15) +
  theme(
    plot.background = element_rect(fill = "ivory"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 12),
    axis.text.x = element_text(angle = 0, hjust = 0.5),  # Make player names horizontal
    plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
```

```
    plot.subtitle = element_text(size = 14, hjust = 0.5),
    legend.position = "none"
  ) +

  # Set Y-axis limits to go from 0 to 10
  scale_y_continuous(limits = c(0, 200))
```



wRC+ is perhaps the best overall purely offensive statistic to evaluate a player's hitting relative to league average. This visual shows the three free agents Alex Bregman, Teoscar Hernandez, and Anthony Santander with better offensive performance than the Phillies incumbents with Bregman leading the pack. Alex Bregman, the free agent 3B, is 26.8% better than average while Bohm, the Phillies current third baseman is only 6.3% better than average over the last 3 seasons. Hernandez and Santander, free agent corner outfield options, are both about 23% better than average while the current phillies outfielder Castellanos, is only 3% better than average. From an offensive stand point, if the Phillies can financially afford to move on from Bohm and Castellanos, they probably should. Next we will look at Def and our response variable, Impact Score.

Heat Map of Def

```
# Alternatively, use dplyr for cleaner manipulation (if dplyr is loaded)
library(dplyr)
Def_Graph <- MLB_data %>%
  filter(Name %in% c("Alec Bohm", "Nick Castellanos", "Anthony Santander",
```

```
                    "Teoscar Hernández", "Alex Bregman") &
           Season %in% c(2022, 2023, 2024)) %>%
  dplyr::select(Name, Season, Def)
# Create the heatmap
ggplot(Def_Graph, aes(x = Season, y = Name, fill = Def)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  geom_text(aes(label = round(Def)), color = "white", size = 5,
            fontface = "bold") +  # Add wRC+ values as text
  labs(title = "Def for Selected Players (2022-2024)",
       x = "Season",
       y = "Player",
       fill = "Def") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels
    plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 14, hjust = 0.5)
  )
```

The "Def" stat from FanGraphs is a measure of a player's defensive contribution in terms of runs saved or cost compared to a league-average player at their position. It aggregates several advanced defensive metrics to provide a single number reflecting a player's overall defensive ability. Def = +5 means the player saved 5 runs compared to an average defender at their position. In this heat map of Def, we are able to see the last 3 years of the players in question. Once again, the free agent players outperform the current Phillies with Bregman being the best fielder out of them all. It should be noted that Bohm has seen considerable defensive

improvement however had some crutial mistakes this past playoffs that has the organization looking at a proven playoff performer in Bregman.

Next we will look at Impact Score, our transformed WAR statistic generated by our averaged predictors, and final evaluation of the five players.

Prediction Intervals

```
# Predict Alec Bohm's 2025 Impact Score
bohm_IS <- predict(hitterModel2, newdata = bohm_predictors)
bohm_Int = predict(hitterModel2, newdata = bohm_predictors,
                   interval = "prediction", level = .95)
cat("Alec Bohm Prediction Interval: ", bohm_Int, "\n")
```

```
## Alec Bohm Prediction Interval:  4.196863 3.627225 4.7665
```

```
# Predict Nick Castellanos 2025 Impact Score
castellanos_IS <- predict(hitterModel2, newdata = castellanos_predictors)
castellanos_Int = predict(hitterModel2, newdata = castellanos_predictors,
                          interval = "prediction", level = .95)
cat("Nick Castellanos Prediction Interval: ", castellanos_Int, "\n")
```

```
## Nick Castellanos Prediction Interval:  3.193328 2.62203 3.764626
```

```
# Predict Anthony Santander's 2025 Impact Score
santander_IS <- predict(hitterModel2, newdata = santander_predictors)
santander_Int = predict(hitterModel2, newdata = santander_predictors,
                        interval = "prediction", level = .95)
cat("Anthony Santander Prediction Interval: ", santander_Int, "\n")
```

```
## Anthony Santander Prediction Interval:  4.905162 4.333792 5.476532
```

```
# Predict Teoscar Hernandez' 2025 Impact Score
hernandez_IS <- predict(hitterModel2, newdata = hernandez_predictors)
hernandez_Int = predict(hitterModel2, newdata = hernandez_predictors,
                        interval = "prediction", level = .95)
cat("Teoscar Hernandez Prediction Interval: ", hernandez_Int, "\n")
```

```
## Teoscar Hernandez Prediction Interval:  4.699422 4.128969 5.269874
```

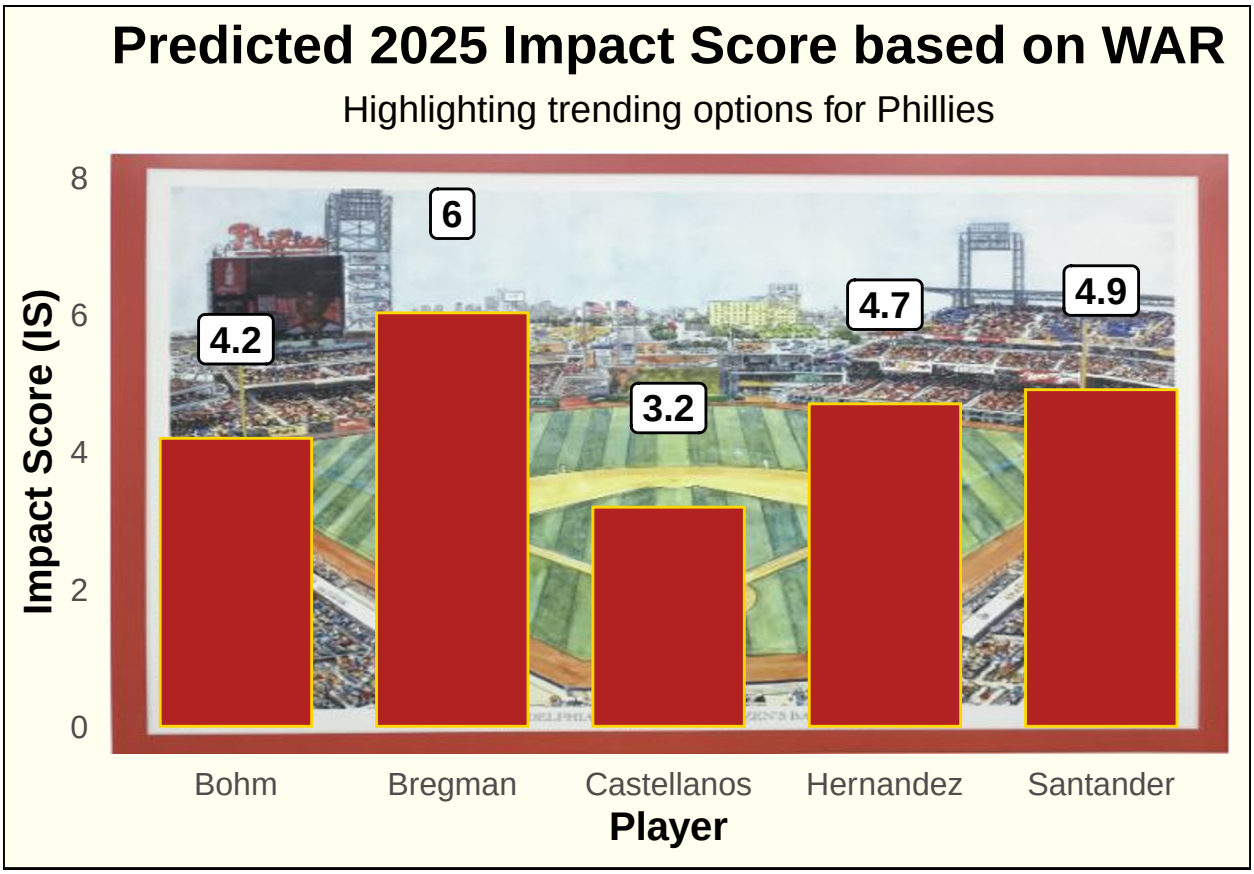
```
# Predict Alex Bregman Impact Score
bregman_IS <- predict(hitterModel2, newdata = bregman_predictors)
bregman_Int = predict(hitterModel2, newdata = bregman_predictors,
                      interval = "prediction", level = .95)
cat("Alex Bregman Prediction Interval: ", bregman_Int, "\n")
```

```
## Alex Bregman Prediction Interval:  6.024872 5.45189 6.597854
```

Based on both individual variability and uncertainty of the model's predictions, if we repeated this process many times, 95% of those future observed Impact Score values would fall within this range. The first number is a point estimate predicted from the model, while the 2nd and 3rd numbers are the lower and upper bounds we are confident the prediction is included in.

```
library(ggplot2)
library(ggimage)
library(grid)
library(jpeg)

# Load baseball field background
field_image <- rasterGrob(
  readJPEG("cbp1.jpg"),
  width = unit(1, "npc"),
  height = unit(1, "npc"),
  interpolate = TRUE
)

# Player data with Impact Score (IS) values
player_IS <- c("Bohm" = bohm_IS,
               "Castellanos" = castellanos_IS,
               "Santander" = santander_IS,
               "Hernandez" = hernandez_IS,
               "Bregman" = bregman_IS)

# Convert to a data frame for ggplot and ensure Player is a character vector
graph_data <- data.frame(
  Player = as.character(names(player_IS)),  # Ensure Player is a character vector
  WAR = as.numeric(player_IS)
)
graph_data$Player <- gsub("\\.1$", "", graph_data$Player)
# Create the bar chart with adjusted Y-axis, full player names, and horizontal text
ggplot(graph_data, aes(x = Player, y = WAR)) +
  # Add background image
  annotation_custom(field_image, xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = Inf) +

  # Add bars for players
  geom_col(width = 0.7, fill = "firebrick", color = "gold", linewidth = .5) +

  # Add text labels for WAR values
  geom_label(aes(label = round(WAR, 1)),
             vjust = -1.5, size = 5, color = "black",
             fill = "white", fontface = "bold", label.size = 0.5,
             label.padding = unit(0.25, "lines")) +

  # Add titles and labels
  labs(
    title = "Predicted 2025 Impact Score based on WAR",
    subtitle = "Highlighting trending options for Phillies",
    x = "Player",
    y = "Impact Score (IS)"
  ) +

  # Set theme
```

```r
theme_minimal(base_size = 15) +
theme(
  plot.background = element_rect(fill = "ivory"),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  axis.title = element_text(face = "bold"),
  axis.text = element_text(size = 12),
  axis.text.x = element_text(angle = 0, hjust = 0.5),  # Make player names horizontal
  plot.title = element_text(face = "bold", size = 20, hjust = 0.5),
  plot.subtitle = element_text(size = 14, hjust = 0.5),
  legend.position = "none"
) +

# Set Y-axis limits to go from 0 to 10
scale_y_continuous(limits = c(0, 8))
```



Conclusion

After transforming the response variable to ensure linearity in the model, we successfully developed an accurate predictive framework for evaluating a players performance in Impact Score based on six different predictor variables, wRC+, BsR, Def, HR, HH%, and BB/K. The model's prediction intervals provide a range of expected performance, reinforcing its reliability for decision-making. Based on the results, Alex Bregman stands out as a high-impact player, with a predicted Impact Score that highlights his value on the

field. Given his consistent performance and potential to enhance team success, the Phillies should strongly consider signing Bregman to bolster their roster for the upcoming season.