Data Analysis in Baseball

Adam Sherer

2023-04-24

Abstract

Since the 1800s, baseball has been one of the most watched sports in America. In 2019, it had an average of 1.27 million viewers per game, but it was even gaining steam in foreign countries such as South Korea and Japan. As the sport continued to grow, experts looked at analytics and statistics more than ever. Analysts used machine learning and logistic regression methods on large baseball data sets to obtain valuable information and insights into player performance. In this paper, I attempted to determine which statistics were the most influential in determining how successful a player was or would be, as well as how that success could be defined. Then, I took those statistics and built linear models that could be used to evaluate player performance in the future. To do this, I used basic exploratory data analysis techniques to visualize the data and determine which statistics had the highest correlation, and I used simple logistic regression methods to create the models.

Introduction

Baseball is a game that is nearly impossible to predict certain outcomes. Even experts often struggle to accurately predict the winners and losers of games. Predicting individual player success can be just as difficult, as it can be hard to determine what "success" really means for individual players. This "success" can have many different interpretations, but is most often determined by statistics such as batting average (BA),home runs (HR), runs scored (R), runs batted in (RBIs), and many other factors. Data analysis can also be used to predict if/when players will have a breakout year, and when they will start to regress. This paper will primarily look at runs and RBIs to evaluate players, as these are the statistics that help teams win games. I will also use a new statistic called run contributions (RC) that is just a combination of runs and RBIs. I will test this statistic along with runs and RBIs in several different models.

Methods

Unlike most other major sports, it is very easy to find large quantities of baseball data. The resource that provides the most complete and accurate information regarding player and team statistics is the Lahman Database. This database was created by Sean Lahman, a journalist and baseball enthusiast who felt there was no way to obtain complete baseball data, so he decided to create a database that provides people with just that. Collectively these two databases should provide more than enough information to explore the relationship between different variables to determine each player's value.

When looking at baseball data, there are a few things that should be considered. It is important to note that there have been many baseball players that have had very limited experience in the major leagues. We need to determine where we should draw the line between players that are included in our data, and players that are not. For example, consider a player who has only had one plate appearance at the major league level, in which he hit a grand slam. This gives him a run total of 1, and and RBI total of 4, but should this data be considered in our model? If it is, it may show that home runs have the biggest impact on success. However, this data provides us with next to no information, as this player did not have a large enough sample size to effectively determine if home runs truly is the most impactful stat. To counteract this, we will filter our data to include only players that have had 502 plate appearances or more. This may seem like a completely random number, however it actually has great meaning in baseball. 502 plate appearances is the minimum number of required plate appearances to be considered for MLB awards, such as MVP and many others. This is equivalent to approximately 3.1 plate appearances per game a team plays in. The number 3.1 was chosen because a player is guaranteed 3 plate appearances per game, assuming they play the entire game. At the time this rule was made, MLB officials decided that players should have to have at least 0.1 more plate appearances than the minimum. This will ensure that we are only getting information from players that have played at the major league level for a substantial amount of time in a season.

One other important thing to consider is that there are many different statistics that are recorded/calculated for each player, and most fall under two categories: total statistics, and percentages. Total statistics are things that are tracked as a running total, such as runs, plate appearances, home runs, etc. Percentages are things that are tracked as percentages, such as batting average, on base percentage, slugging percentage, etc. Why does it matter that there is a difference between the two? One example would be when using the two different types of statistics together to evaluate one player. If you were to look at the statistics that have the greatest influence on runs scored, the number one statistic will always be plate appearances, because obviously, the more opportunities you have to score runs, the more runs you will score. While this is definitely a true fact, it is not very helpful information as plate appearances are not affected by a player's performance. In order to combat this problem, I created several new statistics that will convert total statistics to percentages. These will include runs, RBIs, and run contributions. I will divde each of them by a player's plate appearances to create runs per plate appearcance (RperPA), RBIs per plate appearance (RBIperPA), and run contributions per plate appearance (RCperPA). These new statistics will eliminate the bias created by using total statistics by instead using percentages that are scaled based on the number of plate appearances a player has.

Results

Correlation

In order to determine which statistics have the greatest correlation with runs per plate appearance, RBIs per plate appearance, and run contributions per plate appearance, I will look at several correlation tests (see Appendix 1). These tests will test the correlation between the three aforementioned variables against all other variables in the data set, and display an ordered correlation matrix for each test that contains each variable that is at least 70% correlated with each of RperPA, RBIperPA, and RCperPA.

RperPA & RBIperPA

From these correlation matrices, there are a few conclusions that can be drawn about RperPA and RBIperPA and the variables that have a strong correlation with them. If you look at the matrix for RperPA (see Appendix 1), you will notice that it has no variables in it, other than itself. This means that there are no variables that are correlated with RperPA to a high enough degree to make it significant enough to consider further. As for RBIperPA, we get two variables: on base plus slugging percentage (OPS) and slugging percentage (SLG), each between 70% and 80% correlated. While this is somewhat helpful for predicting RBIs, we want to predict RBIs and Runs, so predicting RBIs on its own is essentially useless without the ability to predict runs as well.

RCperPA

This correlation matrix (see Appendix 1) gives us the same variables as the RBIperPA matrix, with the addition of total bases (TB), however we should note that total bases is not a percentage stat, and as mentioned before, this means it is not very useful to us, and is likely just a coincidence that it has this high of a correlation as there is no logical reason why your total number of bases should have a significant influence on how many runs/RBIs you achieve per plate appearance. If we then disregard TB, we are left with the same variables as the RBI matrix, however these have higher correlation rates, with both variables being around 82% correlated.

Models

Now that I have determined the variables that have a significant influence on a player's RCperPA, the next step is to actually use these variables to build some models that will hopefully predict what a player's RCperPA will be. I create these models by using simple linear regression techniques. I will also split the data up into train and test sets using a 70/30 split in order to have some data that can be used to test the accuracy of these models.

Slugging Percentage

The model for slugging percentage and the summary for this model are included in appendix 2. We can see that the intercept is -0.0003587 and the coefficient is 0.5870764, giving us an equation of:
RCperPA = 0.5870764(SLG) - 0.0003587
This equation can be used to predict the RCperPA of any player, given their SLG. Now that we have found the equation and its coefficients, there are several methods that can be used to evaluate the model. The first is a simple p-test, which can also be found in the summary of the model in appendix 2. This summary gives us a p-value of 2.2e-16. Due to the computer's rounding error, most numbers that are very close to 0 are represented as 2.2e-16, so knowing this information tells us that our p-value is very close to 0. This value is significantly smaller than the generally accepted p-value of 0.05, so this would tell us that our model is statistically significant, meaning it is accurate at predicting RCperPA.

Another method used to evaluate linear models is mean squared error (MSE). Mean squared error takes the residuals and squares them, then finds the mean of those values. This tells us how far on average a given point would be from the regression line, or in other words, the average distance a point is from being predicted completely correctly. For these models, we will calculate the MSE for the test data set using the models we created. The MSE for the SLG model (see appendix 3) is 0.00099. In general, the closer to 0 a MSE value is, the more accurate the model is, which again tells us that our model is very accurate at predicting the points.

On Base Plus Slugging Percentage

The model for on base plus slugging percentage and the summary for this model will be shown in appendix 4. This model has an intercept of -0.077228 and a coefficient of 0.421072. This gives us an equation of:
RCperPA = 0.421072(OPS) - 0.077228
This is how we can use OPS to predict the RCperPA for any player. We will again use a p-test to evaluate the accuracy of this model (see appendix 4). This summary gives us the exact same p-value as the model for SLG of 2.2e-16. This is again because of the computer's rounding approach, which means this number is again very close to 0, meaning our model is statistically significant and can be used to accurately predict RCperPA.

The MSE for this model is very close to that of the SLG model as well, however it differs slightly. This time our MSE value is 0.0009718268 (see appendix 5). While this may be a different value than that of the SLG model, it is still very close to 0, meaning that our model is very accurate at predicting the RCperPA of a test set of players.

Discussion

There are several interesting takeaways of this analysis, however one that sticks out above the rest. The two variables that were used for the models due to their high levels of correlation were SLG and OPS. This is interesting because OPS stands for on base plus slugging percentage, which is exactly what it sounds like. It takes a player's on base percentage and adds it to their slugging percentage. This makes some sense, because if the slugging percentage is accurate, it is logically that the OPS which is dependent on the slugging percentage would also be accurate, however it is interesting to note that OBP (the other half of OPS) actually has very little correlation with RCperPA. Because of this fact, the assumption would be that slugging percentage would be more accurate than OPS because OPS contains one variable that has a strong correlation, and one variable that has a very weak correlation, however this is not the case. These models are nearly identical in their error testing. This raises a few questions that would be interesting to research more in the future.

This paper has provided a very analytical approach to baseball and evaluating baseball player's performance. It has done this by describing how we can determine a player's "success." This is done by choosing runs and RBIs as the statistics that promote success, as they are the statistics that are directly related to scoring team runs, but also by explaining why it is more beneficial to combine the two statistics under one variable. It then goes on to use statistical methods to determine which variables have the strongest correlation with our dependent variable, and builds and tests two separate models that can accurately predict how many runs and RBIs per plate appearance a player will have based on two separate independent variables. Using the p-tests and mean squared errors, we can see that these models are extremely effective in predicting a player's success. Due to how close these models are in their accuracy levels, it is extremely difficult to determine if one model is more effective than the other, however in many cases the preference would be to use the model that uses OPS as the predictor. This is because slugging percentage is just dependent on itself, whereas OPS is dependent on several other things. There may be players who do not have a very good slugging percentage, but have a high enough on base percentage that it brings their OPS up to a high enough level to more accurately predict their RCperPA. Ultimately, both models are very accurate and either can be used to help determine the success of any given player.

Appendix

Appendix 1

```
Batting_num <- select(Batting, -playerID, -yearID, -stint, -teamID, -lgID, -R, -RBIperPA, -RC, -RCperPA

cor_matrix <- cor(Batting_num[, c("RperPA")], Batting_num)

high_corID <- which(abs(cor_matrix) > 0.7 & cor_matrix != 1, arr.ind = TRUE)


high_cor <- data.frame(var1 = rownames(cor_matrix)[high_corID[, 1]],
                       var2 = colnames(cor_matrix)[high_corID[, 2]],
                       cor = cor_matrix[high_corID])

high_cor[order(high_cor$cor, decreasing = TRUE),]
```

```
##     var1   var2 cor
## 1 RperPA RperPA   1
```

```
Batting_num2 <- select(Batting, -playerID, -yearID, -stint, -teamID, -lgID, -R, -RBI, -RC, -RCperPA, -Rp

cor_matrix2 <- cor(Batting_num2[, c("RBIperPA")], Batting_num2)

high_corID2 <- which(abs(cor_matrix2) > 0.7 & cor_matrix2 != 1, arr.ind = TRUE)

high_cor2 <- data.frame(var1 = rownames(cor_matrix2)[high_corID2[, 1]],
                       var2 = colnames(cor_matrix2)[high_corID2[, 2]],
                       cor = cor_matrix2[high_corID2])

high_cor2[order(high_cor2$cor, decreasing = TRUE),]
```

```
##       var1    var2       cor
## 1 RBIperPA SlugPct 0.7788327
## 2 RBIperPA     OPS 0.7077729
```

```
Batting_num3 <- select(Batting, -playerID, -yearID, -stint, -teamID, -lgID, -R, -RBI, -RperPA, -RBIperPA

cor_matrix3 <- cor(Batting_num3[, c("RCperPA")], Batting_num3)

high_corID3 <- which(abs(cor_matrix3) > 0.7 & cor_matrix3 != 1, arr.ind = TRUE)


high_cor3 <- data.frame(var1 = rownames(cor_matrix3)[high_corID3[, 1]],
                        var2 = colnames(cor_matrix3)[high_corID3[, 2]],
                        cor = cor_matrix3[high_corID3])

high_cor3[order(high_cor3$cor, decreasing = TRUE),]
```

```
##      var1    var2       cor
```

```
## 2 RCperPA SlugPct 0.8282720
## 3 RCperPA     OPS 0.8224558
## 1 RCperPA      TB 0.7222246
```

Appendix 2

```
logistic_slg <- lm(RCperPA ~ SlugPct, data=bat_train)
prediction_slg <- predict(logistic_slg, bat_test)
summary(logistic_slg)
```

```
##
## Call:
## lm(formula = RCperPA ~ SlugPct, data = bat_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104027 -0.020400 -0.004109  0.014171  0.179156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.001293   0.001839   0.703    0.482
## SlugPct     0.583241   0.004230 137.885   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03132 on 8934 degrees of freedom
## Multiple R-squared:  0.6803, Adjusted R-squared:  0.6803
## F-statistic: 1.901e+04 on 1 and 8934 DF,  p-value: < 2.2e-16
```

Appendix 3

```
mean((bat_test$RCperPA - prediction_slg)^2)
```

```
## [1] 0.000918424
```

Appendix 4

```
logistic_ops <- lm(RCperPA ~ OPS, data=bat_train)
prediction_ops <- predict(logistic_ops, bat_test)
summary(logistic_ops)
```

```
##
## Call:
## lm(formula = RCperPA ~ OPS, data = bat_train)
##
## Residuals:
```

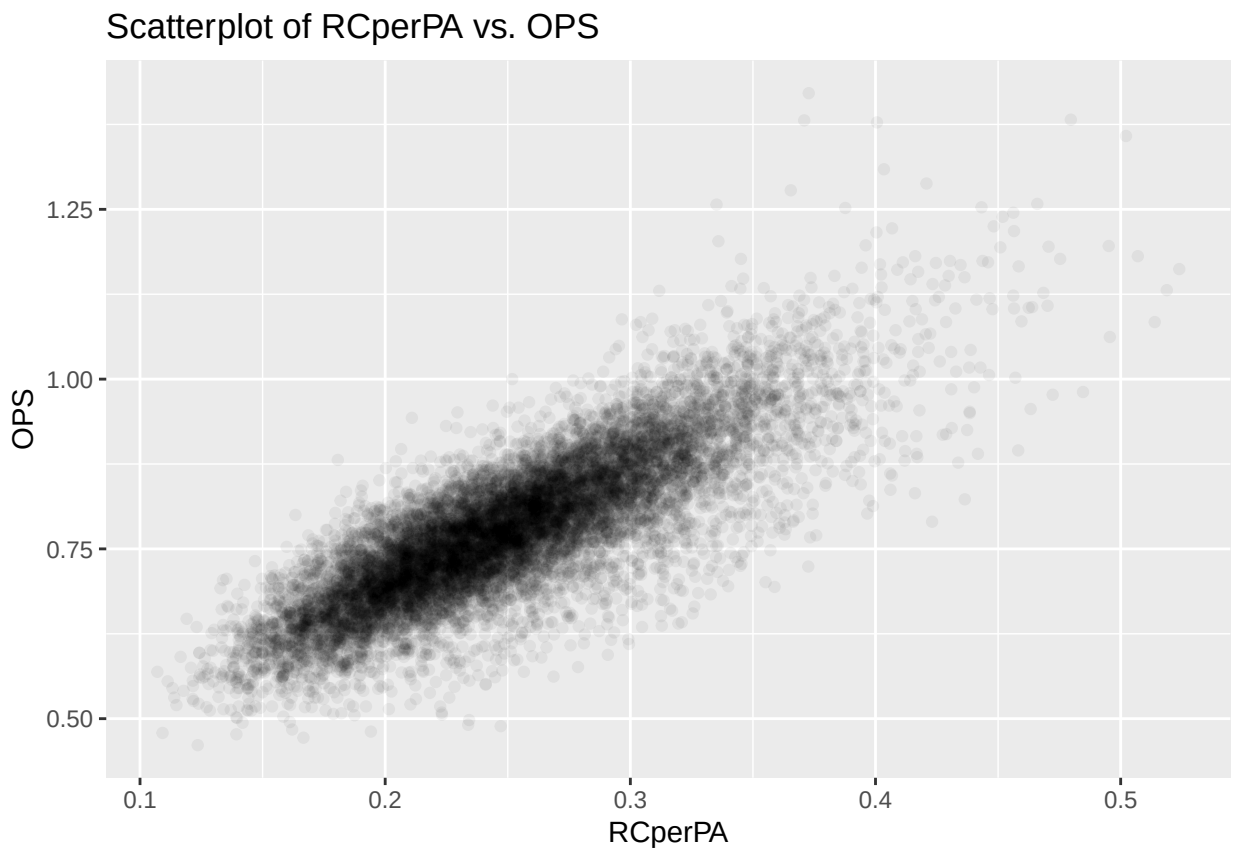
```
##       Min        1Q    Median        3Q       Max
## -0.147859 -0.020935 -0.002891  0.017069  0.167582
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.076450   0.002444  -31.28   <2e-16 ***
## OPS          0.420183   0.003109  135.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03175 on 8934 degrees of freedom
## Multiple R-squared:  0.6716, Adjusted R-squared:  0.6716
## F-statistic: 1.827e+04 on 1 and 8934 DF,  p-value: < 2.2e-16
```

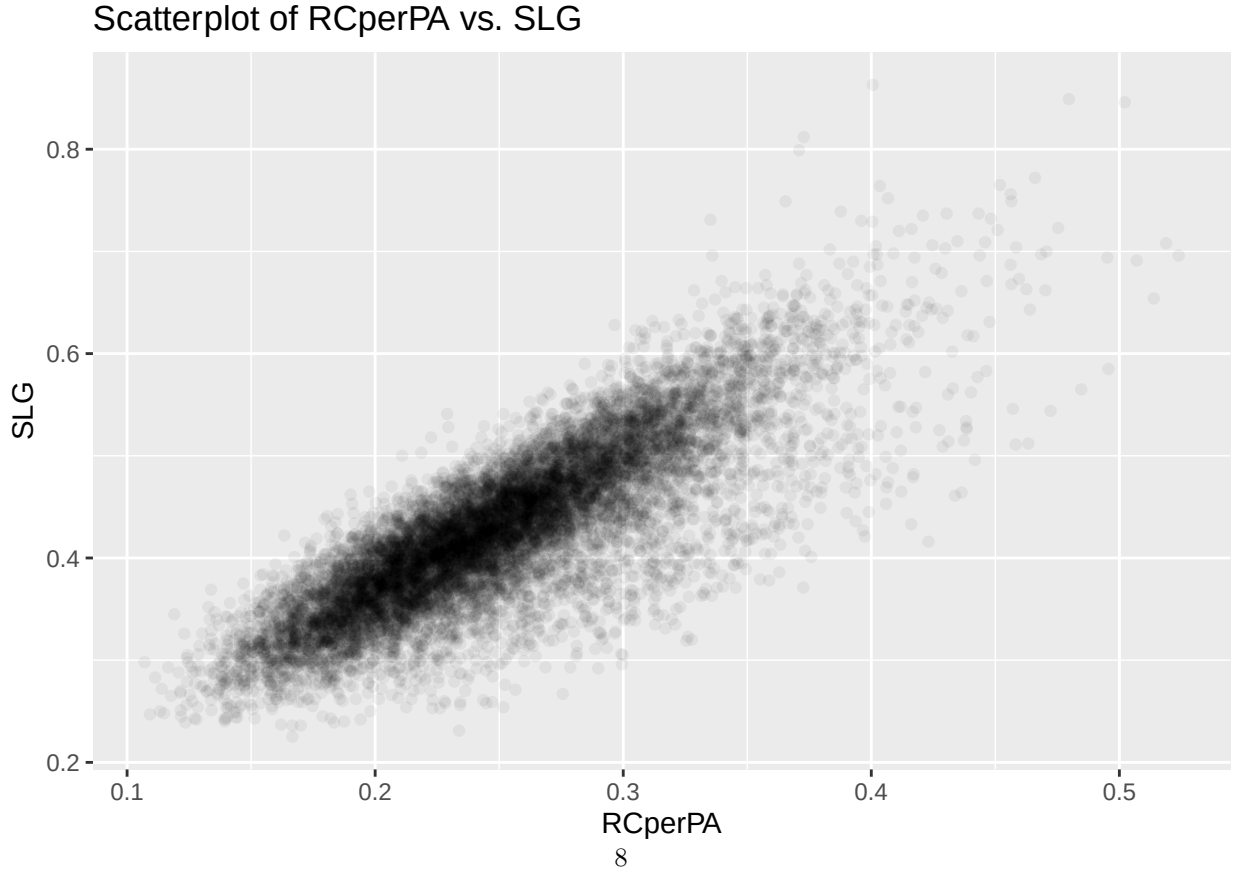
Appendix 5

```
mean((bat_test$RCperPA - prediction_slg)^2)
```

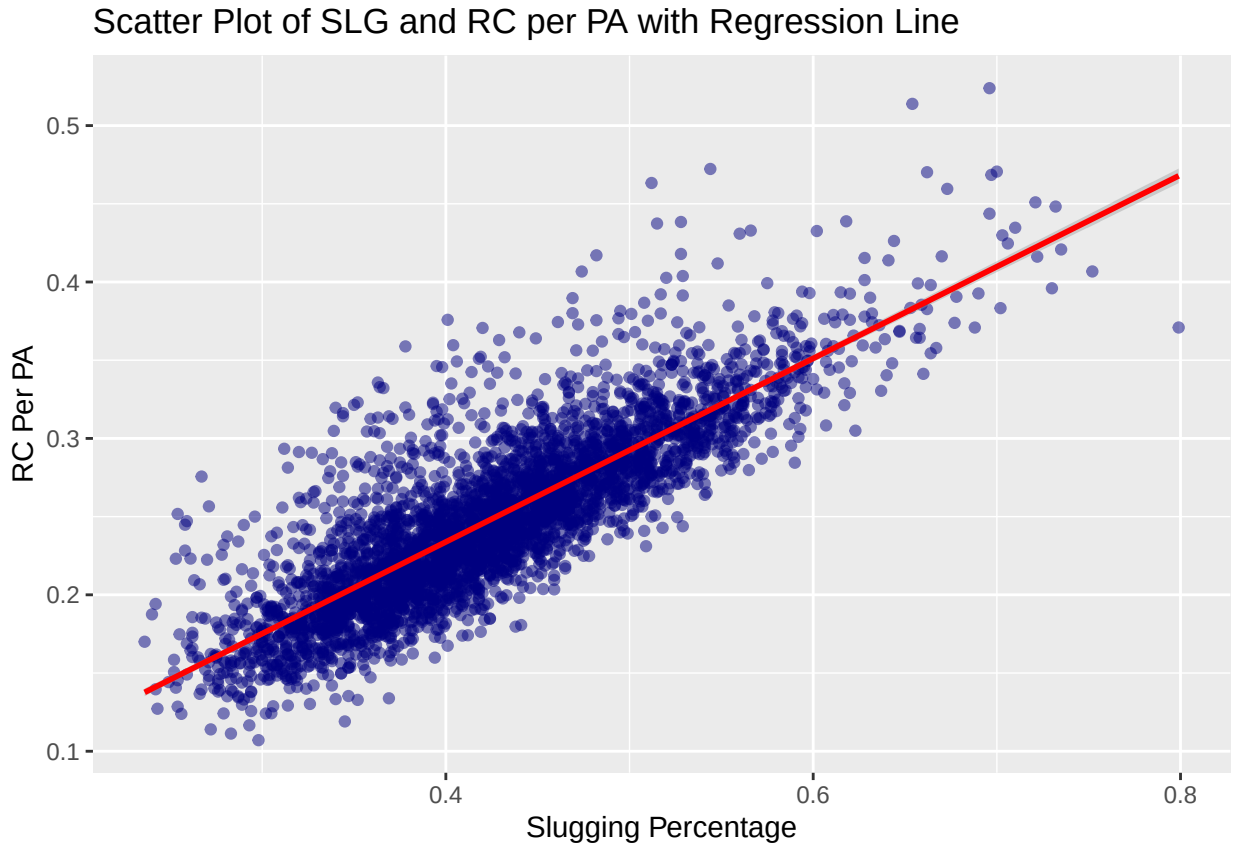
```
## [1] 0.000918424
```

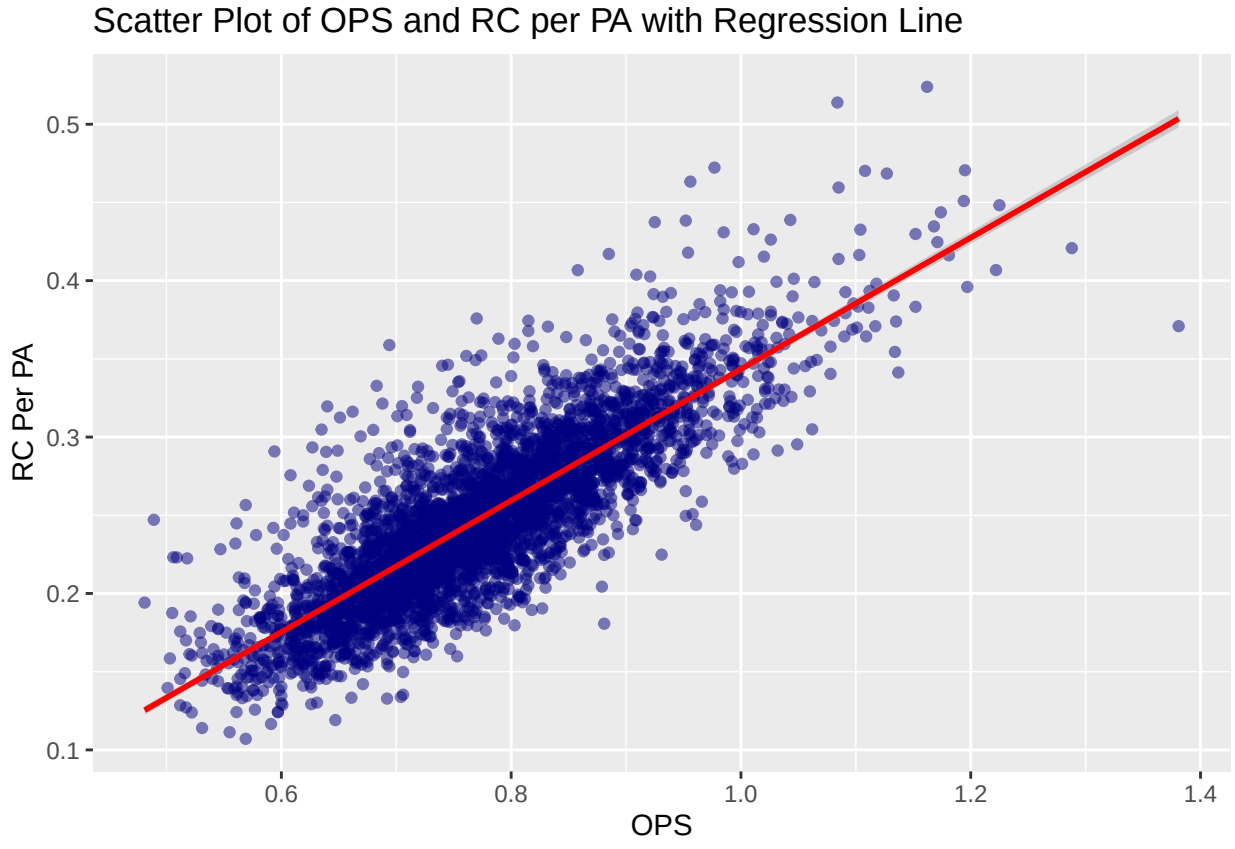
Appendix 6


Scatterplot of RCperPA vs. OPS
OPS
1.25
1.00
0.75
0.50
0.1
0.2
0.3
0.4
0.5
RCperPA


Scatterplot of RCperPA vs. SLG
SLG
0.8
0.6
0.4
0.2
0.1
0.2
0.3
0.4
0.5
RCperPA

Appendix 7


Scatter Plot of SLG and RC per PA with Regression Line
RC Per PA
0.5
0.4
0.3
0.2
0.1
0.4
0.6
0.8
Slugging Percentage


Scatter Plot of OPS and RC per PA with Regression Line
RC Per PA
0.5
0.4
0.3
0.2
0.1
0.6
0.8
1.0
1.2
1.4
OPS