Predicting On-base Percentage Using April + May Hitting Statistics

Introduction

Goal

The goal of this project was to be able to predict each player's on-base percentage at the end of the 2019 season given his batting statistics in March/April 2019. I started with a dataset containing various offensive metrics, which included each player's final 2019 season on-base percentage.

Plan

My plan is to predict this using multiple linear regression. Because some statistics might not assist in the prediction of full season on-base percentage, I had to determine what the best predictors would be. Once I am able to figure out the correct statistics, I can find each statistic's coefficient value to create a multiple linear regression model that can predict each player's full season on-base percentage. To finish, I will compare my predicted on-base percentages against the actual ones provided in the dataset.

Definition of Summary Statistics

In this analysis, I created three different linear regression models. The accuracy of these models can be measured with various statistics. The ones that I will use to compare my model will be as follows:

Residual Standard Error: This measures the standard deviation of the residuals. Lower is better for the residual standard error because it means more consistent/ accurate predictions.

Multiple R-Squared: This R-squared value measures the proportion of variation in the response variable that is accounted for by this model. This is also the squared value of the correlation variable. Closer to 1 is better overall for this model, but the biggest shortcoming of multiple R-squared is that even adding a non-meaningful variable will increase the value of multiple R-squared.

Adjusted R-Squared: This is an adjusted version of multiple R-squared. The main difference is that adjusted R-squared only increases when the added variable improves the model. In comparing my three models in this report, I will use adjusted R-squared to choose the best model for prediction.

F-Statistic (and resulting p-value): This is the F statistic for the global F test that will check the overall validity of each model. All of the models have a very small p-value resulting from each F statistic which indicates all of the models are strong enough in their prediction of full season on-base percentage.

Data Cleaning

The first thing I did was read in the dataset. After that, the variables with percentages were listed as character variables. Therefore, I had to remove the percent from each cell and convert the percentages to numeric variables. Now the dataset was ready to use. (The code for this is not included in this report.)

First Model

All Variables

To create my first model, I wanted to use linear regression including every offensive statistic in the dataset and measure how well this predicted the final season on-base percentage. I used the code below to make this model and analyzed the outcome.

Summary Statistics of First Model

Residual Standard Error: 0.02741 on 294 degrees of freedom
Multiple R-squared: 0.511
Adjusted R-squared: 0.4694
F-statistic: 12.29 on 25 and 294 DF, p-value: < 2.2e-16

```
# First Model

model <- lm(FullSeason_OBP ~ MarApr_PA + MarApr_AB + MarApr_H + MarApr_HR + MarApr_R + MarApr_RBI + Mar

# http://www.sthda.com/english/articles/40-regression-analysis/168-multiple-linear-regression-in-r/
```

Coefficients

```
head(summary(model)$coefficients, 10)
```

```
##                 Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)  0.7909880435 3.1088021979  0.2544350 0.7993373941
## MarApr_PA    0.0022974799 0.0007889937  2.9119116 0.0038673197
## MarApr_AB   -0.0031219644 0.0009279445 -3.3643870 0.0008689622
## MarApr_H     0.0029380835 0.0011720220  2.5068501 0.0127208136
## MarApr_HR   -0.0010557898 0.0020827014 -0.5069329 0.6125819112
## MarApr_R     0.0002323137 0.0006042465  0.3844684 0.7009092369
## MarApr_RBI   0.0003952773 0.0005543215  0.7130831 0.4763599505
## MarApr_SB    0.0001244225 0.0009543223  0.1303778 0.8963566573
## MarApr_BB.  -0.0006548742 0.0012536286 -0.5223829 0.6017971262
## MarApr_K.   -0.0012037519 0.0007820349 -1.5392561 0.1248173540
```

Second Model

Subset of Variables

From the first model, I analyzed the p-values of the individual t-tests for all of the statistics that were used to predict the full season on-base percentage. I removed all variables that had a p-value of less than 0.05 and created a new model with this new subset of variables. I then repeated the code from above, but only included the more significant variables. The new variable coefficients (and related statistics) are shown below the code.

Summary Statistics of Second Model

Residual Standard Error: 0.02726 on 310 degrees of freedom
Multiple R-squared: 0.4901
Adjusted R-squared: 0.4752
F-statistic: 33.1 on 9 and 310 DF, p-value: < 2.2e-16

```
# Second Model

coefficients <- summary(model)$coefficients
coefficients.df <- as.data.frame(coefficients)
names(coefficients.df)[4] <- "pVal"
coefficients_reduced <- coefficients.df %>% filter(pVal < 0.05)
# used only variables included in 'coefficients_reduced'

new_model <- lm(FullSeason_OBP ~ MarApr_PA + MarApr_AB + MarApr_H + MarApr_O.Swing. + MarApr_Z.Swing. +
```

Coefficients

```
summary(new_model)$coefficients
```

```
##                      Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)        0.264847960 0.0312120839  8.485430 8.979744e-16
## MarApr_PA          0.002627718 0.0004524701  5.807496 1.570363e-08
## MarApr_AB         -0.003555236 0.0005116990 -6.947905 2.190022e-11
## MarApr_H           0.003326760 0.0003862353  8.613298 3.664317e-16
## MarApr_O.Swing.    0.004318361 0.0015034811  2.872241 4.356720e-03
## MarApr_Z.Swing.    0.002839450 0.0007975205  3.560347 4.284602e-04
## MarApr_Swing.     -0.007022217 0.0020741368 -3.385610 8.016373e-04
## MarApr_O.Contact. -0.002342087 0.0009461767 -2.475317 1.384606e-02
## MarApr_Z.Contact. -0.004710694 0.0016487174 -2.857187 4.563137e-03
## MarApr_Contact.    0.007672032 0.0025693248  2.986011 3.051679e-03
```

Third Model

Stepwise Selection I wanted to try a slightly different variable screening process for my third model. Using stepwise selection, the program adds variables one at a time, checking the significance of each variable after a new one is added. If a variable is deemed insignificant, it is then removed from the model. The model produced had similar variables to the second model, but did not include any of the swing percentages (in/out of the zone and overall) and did include the home runs per fly ball statistic.

Summary Statistics of Third Model

Residual Standard Error: 0.0275 on 312 degrees of freedom
Multiple R-squared: 0.4775
Adjusted R-squared: 0.4658
F-statistic: 40.74 on 7 and 312 DF, p-value: < 2.2e-16

```r
# Third Model

intercept <- lm(FullSeason_OBP ~ 1, data = BattingData)

all <- lm(FullSeason_OBP ~ MarApr_PA + MarApr_AB + MarApr_H + MarApr_HR + MarApr_R + MarApr_RBI + MarAp

stepwise <- step(intercept, direction = "both", scope = formula(all), trace = 0)
stepwise$coefficients

# https://www.statology.org/stepwise-regression-r/

step_model <- lm(FullSeason_OBP ~ MarApr_PA + MarApr_AB + MarApr_H + MarApr_HR.FB + MarApr_O.Contact. +
```

Coefficients

```r
summary(step_model)$coefficients
```

```
##                       Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)        0.2539806884 0.0236088976 10.757838 3.632408e-23
## MarApr_PA          0.0030972261 0.0003644772  8.497722 8.070359e-16
## MarApr_AB         -0.0040182366 0.0004270109 -9.410150 1.124634e-18
## MarApr_H           0.0031139541 0.0004066648  7.657299 2.401340e-13
## MarApr_HR.FB       0.0004189119 0.0001828467  2.291055 2.262659e-02
## MarApr_O.Contact. -0.0006156153 0.0004263037 -1.444077 1.497208e-01
## MarApr_Z.Contact. -0.0017341697 0.0007940360 -2.183994 2.970704e-02
## MarApr_Contact.    0.0029914706 0.0011305544  2.646021 8.557329e-03
```

Final Analysis

Since the second model with just nine of the 28 original variables had the highest adjusted R^2 value, that is the model that I am most confident moving forward with. I now take the coefficients calculated above in the "Second Model" section and create a new dataset which adds a column to the batting statistics dataset using the variables of interest to predict the full season on-base percentage.

```r
# Prediction

Final_BattingData <- BattingData
Final_BattingData <- Final_BattingData %>% mutate(Calculated_FullSeason_OBP = round((0.2648480 + 0.0026
```

Accuracy Analysis

Residual Standard Error: 0.02726 on 310 degrees of freedom
Multiple R-squared: 0.4901
Adjusted R-squared: 0.4752
F-statistic: 33.1 on 9 and 310 DF, p-value: < 2.2e-16
Root Mean Squared Error: 0.02622

Along with the summary statistics noted with the second model above, I added the root mean squared error statistic. This statistic measures the average error of my model in predicting the full season on-base

percentage of a player from just just his March and April hitting statistics. In the code below, I created a new dataset that just included the real full season on-base percentage, the predicted full season on-base percentage, the residual, and the squared residual for each player. From there, I was able to calculate the root mean squared error. A lower value is always better for this statistic, so, taking into account the extreme variability that could occur over the course of the season, this is a very accurate prediction model. This statistic is the best measurement of the accuracy of my predictions because it measures how far the typical prediction is from the real full season on-base percentage value.

```
# Analysis

Final_BattingData <- Final_BattingData %>% mutate(Residual = round((Calculated_FullSeason_OBP-FullSeaso

Real <- Final_BattingData[, c('Name', 'FullSeason_OBP')]
Predicted <- Final_BattingData[, c('Name', 'Calculated_FullSeason_OBP')]
Residual <- Final_BattingData[, c('Name', 'Residual')]

Real_vs_Predicted <- merge(Real, Predicted, by = "Name")
Real_vs_Predicted <- merge(Real_vs_Predicted, Residual, by = "Name")

Final_BattingData <- Final_BattingData %>% mutate(Residual_Squared = round((Calculated_FullSeason_OBP-Fu

Residual_Squared <- Final_BattingData[, c('Name', 'Residual_Squared')]
Real_vs_Predicted <- merge(Real_vs_Predicted, Residual_Squared, by = "Name")

Mean_Residual_Squared <- mean(Real_vs_Predicted$Residual_Squared)
Root_Mean_Residual_Squared <- sqrt(Mean_Residual_Squared)
Root_Mean_Residual_Squared
```

```
## [1] 0.02622022
```

```
# http://www.sthda.com/english/articles/38-regression-model-validation/158-regression-model-accuracy-me
```

Final Findings

To display my final results, I outputted the first 20 observations of the dataset that compares the real and predicted full season on-base percentages using just offensive data from March and April. As you can see below, the model is accurate (relatively) in predicting this value for each player.

```
head(Real_vs_Predicted, 20)
```

```
##                 Name FS_OBP Calculated_FS_OBP Residual Residual_Squared
## 1       A.J. Pollock  0.327             0.321   -0.006            0.000
## 2        Aaron Judge  0.381             0.347   -0.034            0.001
## 3  Adalberto Mondesi  0.291             0.322    0.031            0.001
## 4         Adam Eaton  0.365             0.335   -0.030            0.001
## 5         Adam Engel  0.304             0.293   -0.011            0.000
## 6       Adam Frazier  0.336             0.335   -0.001            0.000
## 7         Adam Jones  0.313             0.339    0.026            0.001
## 8  Albert Almora Jr.  0.271             0.303    0.032            0.001
## 9      Albert Pujols  0.305             0.327    0.022            0.000
## 10      Aledmys Diaz  0.356             0.285   -0.071            0.005
```

```
##  11       Alex Bregman  0.423            0.388   -0.035            0.001
##  12        Alex Gordon  0.345            0.383    0.038            0.001
##  13       Alex Verdugo  0.342            0.334   -0.008            0.000
##  14       Amed Rosario  0.323            0.318   -0.005            0.000
##  15 Andrelton Simmons  0.309            0.318    0.009            0.000
##  16 Andrew Benintendi  0.343            0.350    0.007            0.000
##  17  Andrew McCutchen  0.378            0.351   -0.027            0.001
##  18    Anthony Rendon  0.412            0.371   -0.041            0.002
##  19     Anthony Rizzo  0.405            0.355   -0.050            0.003
##  20  Asdrubal Cabrera  0.342            0.312   -0.030            0.001
```