“Hitters” Predictive Models

Owen Enestvedt
Dr. Fotios Kokkotos
Data Science II

Executive Summary

The MLB which is short for Major League Baseball, is a professional baseball organization and the oldest major professional sports league in the world. It was founded in 1876 in Cincinnati, Ohio. The main goal of this project was to learn about the longevity of players in the professional scene as well as their salaries. This project used the Hitters data set from the ISLR library in R, which included Major League Baseball Data from the 1986 and 1987 seasons. The following lines include the research objectives:

Objectives:

1. Create a model that predicts a player's salary.
    a. Learning about how a salary can be influenced for each player is important to know for both the player and their team. Finances are extremely heavily considered on both sides of a deal, with each side negotiating to receive the best deal to benefit them. The goal is to provide insight into which factors have the greatest effect on a player's salary.
2. Create a model that classifies players as being a novice or experienced player.
    a. According to "Witnauer, W. D., Rogers, R. G., & Saint Onge, J. M. (2007). Major league baseball career length in the twentieth century. Population research and policy review, 26(4), 371–386. https://doi.org/10.1007/s11113-007-9038-5", between 1902 and 1993 "A rookie position player can expect to play 5.6 years" in the MLB, so another goal is to predict if a player is experienced or not, based off certain factors.

Results:

The best model to predict a player's salary was a regression tree model. This model found that as the number of hits and walks a player had each season increased and the higher the number of career runs they had, the higher their salary was.

The best model to predict MLB experience was a decision tree model. This model found that the three greatest predictors of career length were the number of career at bats, career RBIs (runs batted in), and career runs.

Data and Approach

Data Description:

The Hitters data set includes Major League Baseball Data from the 1986 and 1987 seasons which was used to create the models described in the research objectives. This data set includes 322 observations, each corresponding to a player in the MLB. The data set contains 20 variables which are related to stats for each player in 1986, their lifetime stats, errors made, salary, and years played. The data set was accessed through the ISLR library in RStudio.

Data Engineering and Validation:

The research of this data took place in the application RStudio. The following packages were used: ISLR, rpart, rpart.plot, partykit, randomForest, caret, class, dplyr, e1071, and nnet.

To label each player in the data set as being an experienced player or not, a new variable was mutated. This variable marked anyone who has been playing less than or equal to 5.6 years as "No" under being an experienced player, and any player over 5.6 years was marked as "Yes".

Since there were six variables related to a player's career stats centralized around their hitting performance, many of these predictors were highly correlated with each other. A correlation test was run between every numerical predictor, not just the career stats, to investigate the relationships between the independent variables, and in the case of Salary, the response variable as well. The categorical variables referred to a player's league and division and had no influence on the salary or career length.

While investigating the research objectives, the data set was randomly split into two smaller sets. These were the training (80% of original observations) and the test (20% of original observations). The data exploration and model creation took place within the train data set. Then each model was run on the test data set and compared the predictions made by the model to the actual data. Using the train and test separation allows training models to be made and then the test data confirms it works correctly (or incorrectly).

For a few observations there was a reported salary of NA. Making use of the 'na.omit' function allowed research to continue without the NA players, because salary was a variable of utmost importance.

When working with the numerical response variable salary, training regression models were compared using their R^2 and mean squared error. The R^2 of a model is how much variability is explained by the model with 100% being a perfect (yet unrealistic) model. The mean square error was used for all models related to salary being the response variable. It measures the average squared difference between the estimated values and the actual value.

When working with the classifier of being an experienced player or not, models were rated on their misclassification error rate. This error rate comes from comparing the predictions a model makes to actual outcomes and finding the percent of time it is incorrect.

Models:

The following models were used to explore and explain the data.

Quantitative:

The first model was a multiple regression model. The goal of multiple linear regression is to model the linear relationship between the explanatory variables and response variables and predict a numerical value.

The next model was a regression tree. The base of the tree is the root node. From the root node flows a series of decision nodes that depict decisions to be made. From the decision nodes are leaf nodes that represent the consequences of those decisions. Each decision node represents a split point, and the leaf nodes that stem from a decision node represent the possible answers. This pattern takes place recursively down a tree.

Random forest models were also used to predict salary. This model operates by constructing a multitude of decision trees at training time. Then, the mean prediction of the individual trees is returned. The model uses a chosen smaller number of the predictors at the time of modeling, as well as offering the ability to customize the number of decision trees made before averaging.

Categorical:

The base model is the null model which makes use of the null hypothesis. The null hypothesis assumes that any kind of difference between the chosen characteristics that you see in a set of data is due to chance. The goal is to reject the null hypothesis and outperform the null model which requires convincing evidence in the form of an observed difference that is too large to be explained solely by chance.

The next model is a logistic regression model. It is a statistical analysis method used to predict a binary outcome, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

A classification tree was also used. The classification tree is like a regression tree from the fact that it is a decision tree, but instead has a categorical response instead of quantitative.

Again, a random forest was used. This model is very similar to the quantitative random forest model but instead has a categorical response variable.

Another model is called the KNN (k-nearest neighbors) algorithm. It is a method for estimating the likelihood that a data point will become a member of one group or another based on what group the data points nearest to it belong to. The number of nearby points which will signify the group to join is adjustable.

The naïve bayes algorithm is a classification technique based on Bayes' Theorem which includes an assumption of independence among predictors. In other words, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The final model was called a neural network. It uses a process that mimics the way the human brain operates. They refer to systems of neurons/nodes and the connections are modeled as weights between the nodes. Neural networks can adapt to changing input, so the network generates the best possible result without needing to redesign the output.

Data Tables:

Next is some insight into the predictors that were used in the making of the models.

5 number summaries:

Name	Variable	Minimum	1st Quartile	Median	3rd Quartile	Maximum
Career Runs	CRuns	1.00	100.20	247.00	526.20	2165.00
Hits in the 1986 season	Hits	1.00	64.00	96.00	137.00	238.00
Years played	Years	1.00	4.00	6.00	11.00	24.00
Walks in the 1986 season	Walks	0.00	22.00	35.00	53.00	105.00
Career at bats	CAtBat	19.00	816.80	1928.00	3924.20	14053.00
Career hits	CHits	4.00	209.00	508.00	1059.20	4256.00
Career RBI's	CRBI	0.00	88.75	220.50	426.25	1659.00

Detailed Findings

Salary Prediction Model

The goal of this model was to predict a player's salary based off their statistics while also investigating which statistics have the greatest impact on their salary. First the data was separated into training and testing data sets as mentioned before. Next, any observations missing statistics or salary information were removed. Then a correlation table was run to investigate the relationship between predictors and the response variable salary, while also making sure not to include multiple predictors which were correlated with each other.

Multiple Linear Regression:

The first model created was a multiple regression model. The model was formed originally using the variables CRuns, Hits, Walks, and Years. Then Walks and Years were removed as they were not statistically significant.

```
mod2 <- lm(Salary ~ CRuns + Hits, data = train)
summary(mod2)

##
## Call:
## lm(formula = Salary ~ CRuns + Hits, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -797.20 -182.97  -62.12   77.41 2142.33
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.90887   62.47669  -0.351    0.726
## CRuns         0.64627    0.07396   8.738 8.91e-16 ***
## Hits          3.03655    0.54910   5.530 9.78e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 352.1 on 203 degrees of freedom
## Multiple R-squared:  0.409,  Adjusted R-squared:  0.4032
## F-statistic: 70.24 on 2 and 203 DF,  p-value: < 2.2e-16
```

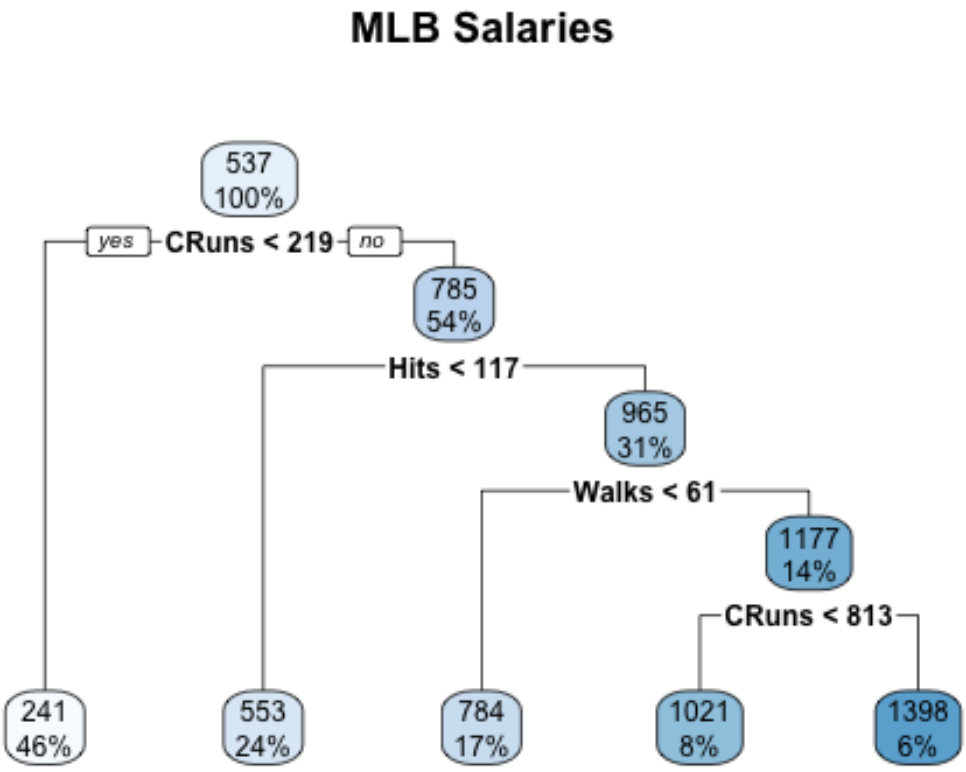
This model shows that the more runs someone has throughout their career and more hits they have during the season, the higher their salary is. It also demonstrated that these are the most important predictors of salary out of the beginning predictors. The R^2 for this model was about 40%.

This model was run on the test data and a resulted in a MSE of 106528.

Regression Tree:

The next model was a decision tree, more specifically a regression tree. The most significant predictors in this model were CRuns, Hits, and Walks.

```
form <- as.formula("Salary ~ CRuns + Hits + Walks")
mod_tree = rpart(form, data = train, control = rpart.control(cp = 0.02))
rpart.plot(mod_tree, main = "MLB Salaries")
```



The model shows the root node to be CRuns, meaning the most important predictor of salary is the number of runs a player has totaled over their career.

This model was run on the test data and a resulted in a MSE of 63164.

Random Forest:

The final model created to model salary was a random forest. The predictors were chosen by investigating the correlation table. Four different random forests were created using different combinations of the number of trees created and the number of predictors tried. The best random forest created 500 trees and used two of the four predictors in each tree.

```
mod_forest <- randomForest(Salary ~ CRuns + Hits + Years + Walks, data = test
, ntree = 500, mtry = 2)
mod_forest

##
## Call:
##  randomForest(formula = Salary ~ CRuns + Hits + Years + Walks,      data =
test, ntree = 500, mtry = 2)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 84767.64
##                     % Var explained: 55
```

The test data was run in this model and resulted in a MSE of 84767.64.

Results:

Model	MSE test
Multiple Linear Regression	106528.00
Regression Tree	63164.00
Random Forest	84767.64

The three models had varying success with predicting the test data. The model which predicted the best was the regression tree as the test MSE was the lowest out of the all the models by a significant margin. In addition, this model also provides insight into which variables are most significant and in which order, while also having a visual which is easy to read and interpret, therefore making it the best model to predict a player's salary.

Experienced players

The goal of these models was to be able to predict if a player was considered experienced or not based off their statistics. A player was considered experienced if they had been played for over 5.6 years. Anyone who has played for 5.6 years or less was considered a novice. Once again, the data was separated into training and testing. Then a correlation table was run to investigate the relationship between predictors and Years (because a player being experienced or not is directly dependent on the number of years they have played), while also making sure not to include multiple predictors which were correlated with each other.

Null Model:

This model was created from the train data set. It assumed that every single person was an experienced player.

```
table(train$experiencedPlayer)

##
##  No Yes
##  89 117

89/(117+89)

## [1] 0.4320388
```

This model predicts with a 43.20% misclassification error rate if it predicts that everyone is an experienced player. This is the baseline error that I am looking to beat with my other models.

Logistic Regression:

The next model was a logistic regression model. It first was trained on four variables, but CRuns and CRBI were insignificant and then removed, leaving CAtBat and CHits.

```
glm.fit2 <- glm(experiencedPlayer ~ CAtBat + CHits, data = train, family = bi
nomial)
summary(glm.fit2)
```

```
##
## Call:
## glm(formula = experiencedPlayer ~ CAtBat + CHits, family = binomial,
##     data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.05002  -0.32705   0.00217   0.17260   2.26799
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.850724   0.744780  -6.513 7.37e-11 ***
## CAtBat       0.010350   0.002619   3.952 7.74e-05 ***
## CHits       -0.028180   0.008748  -3.221  0.00128 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 281.76  on 205  degrees of freedom
## Residual deviance: 103.30  on 203  degrees of freedom
## AIC: 109.3
##
## Number of Fisher Scoring iterations: 7
```

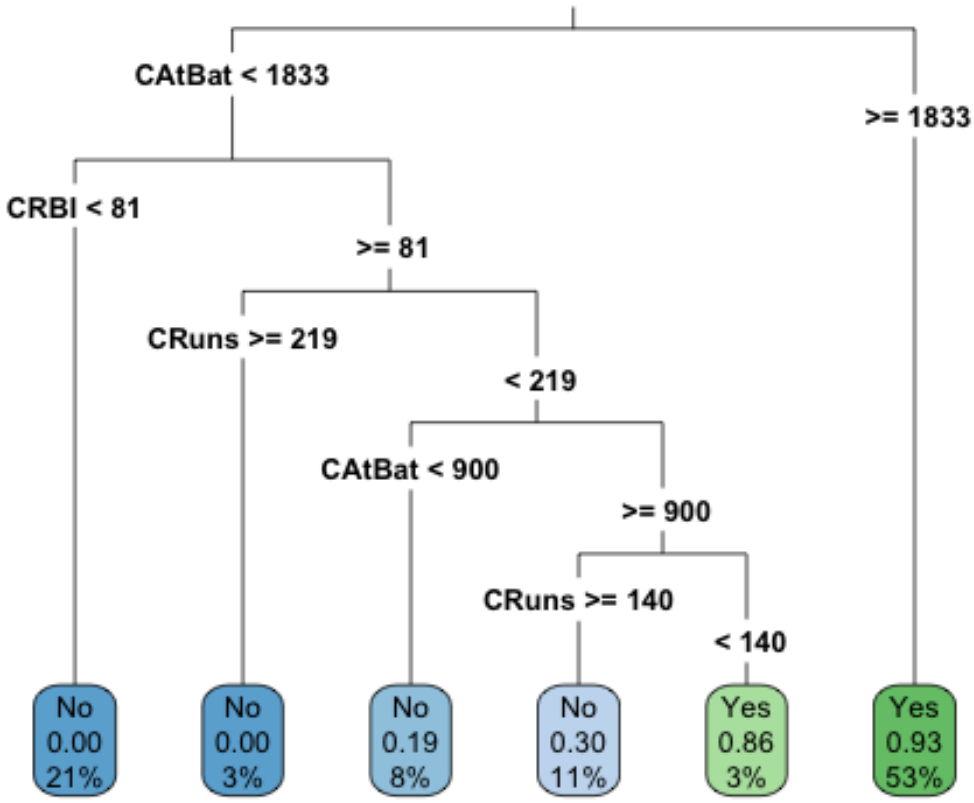
The model was then run with the test data set and a confusion matrix was created. I adjusted the threshold when constructing the confusion matrix which decided if the prediction output was a Yes or No, to find the best accuracy. I found using a threshold of 0.45 facilitated the best predictive power. The logistic regression model had a misclassification error rate of 85.96%, which is significantly worse than the null model.

```
## Prediction No Yes
##        No   5  22
##        Yes 27   3
## Accuracy : 0.1404
```

Classification Tree:

This next model was a decision tree, more specifically a classification tree. Using this model, it was demonstrated that CAtBat, CRBI, and CRuns were the most significant predictors. The node in this tree was from CAtBat, showing the number of at bats a player has had in their career is a strong explainer of career length.

```
form <- as.formula("experiencedPlayer ~ CAtBat + CRuns + CRBI")
fit = rpart(form, data = train, control = rpart.control(cp = 0.001))
rpart.plot(fit, type=3, digits=2, fallen.leaves = T)
```



This model was run on the test data set and a confusion matrix was created. The classification tree had a misclassification error rate of 10.53% which is an improvement from the null model.

```
## Prediction No Yes
##        No  24   3
##        Yes  3  27
## Accuracy : 0.8947
```

Random Forest:

The next model used was a random forest. The predictors were chosen by investigating the correlation table again. Four different random forests were created using different combinations of the number of trees created and the number of predictors tried. The best random forest created 200 trees and used two of the three predictors in each tree.

```
mod_forest <- randomForest(experiencedPlayer ~ CAtBat + CRuns + CRBI, data =
test, ntree = 200, mtry = 2)
mod_forest

##
## Call:
##  randomForest(formula = experiencedPlayer ~ CAtBat + CRuns + CRBI,      da
ta = test, ntree = 200, mtry = 2)
##                Type of random forest: classification
```

```
##                       Number of trees: 200
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 12.28%
## Confusion matrix:
##     No Yes class.error
## No  24   3   0.1111111
## Yes  4  26   0.1333333
```

The test data was run through this model and the random forest had a misclassification error rate of 12.28%, which is another improvement from the null model.

KNN (k-nearest neighbors) Algorithm:

Next a knn algorithm was trained. Similarly to the random forest model there is some adjustability that comes with this algorithm. The number of nearby neighbors can be altered to help improve the model, usually not going over eight or ten.

```
train_knn <- train %>% select(CAtBat, CRuns, CRBI)
test_knn <- test %>% select(CAtBat, CRuns, CRBI)
train_experienced <- train %>% select(experiencedPlayer) %>%
.$experiencedPlayer
knn_pred <- knn(train_knn, test_knn, train_experienced, k = 3)
```

This model was run on the test data set and the accuracy rate was calculated. The knn model had a misclassification error rate of 12.29% which is an improvement from the null model.

```
table(knn_pred, test_experienced)

## test_experienced
## knn_pred No Yes
##      No  25   5
##      Yes  2  25

Res <- mean(knn_pred == test_experienced) #<-accuracy

## [1] 0.877193

round(1 - res, 2)

## 12.29
```

Naïve Bayes Algorithm:

A model was created using the bayes algorithm. The three predictors chosen to be used are the ones the classification tree designated as the most significant.

```
form <- as.formula("experiencedPlayer ~ CAtBat + CRuns + CRBI")
mod_nb <- naiveBayes(form, data = train)
```

Like every other model, the bayes model was used to predict the test data. The created of a confusion matrix showed the misclassification error rate to be 14.04%, once again beating the null error rate.

```
## Prediction No Yes
##        No  25   2
##        Yes  6  24
## Accuracy : 0.8596
```

Neural Network:

The final model created to classify a player's experience level was a neural network. Similarly to the random forest and knn models there is some flexibility in the neural network with the number of neurons in the model being customizable, in addition to the number of max iterations having the ability to be manipulated.

```
train$experiencedPlayer <- as.factor(train$experiencedPlayer)
test$experiencedPlayer <- as.factor(test$experiencedPlayer)
set.seed(400)
netmod <- nnet(experiencedPlayer ~ CAtBat + CRuns + CRBI, data = train, size
= 5, maxit = 200)
```

The final creation of a confusion matrix came with the training and testing of this neural network. Altering the number of neurons and iterations showed having five neurons and 200 iterations to be the most accurate model. The misclassification rate came to be 10.53%.

```
## Prediction No Yes
##        No  24   3
##        Yes  3  27
## Accuracy : 0.8947
```

Results:

Model	Test Misclassification Error
Null Model	43.20%
Logistic Regression	85.96%
Classification Tree	10.53%
Random Forest	12.28%
KNN (k-nearest neighbors)	12.29%
Naïve Bayes	14.04%
Neural Network	10.53%

Out of the six models which does not include the null model, they all performed relatively similarly excluding the logistic regression model. This model struggled greatly with predicting a player's experience level. The misclassification rate was an extremely high 85.96% rate. However, two models shared the lowest error rate being the classification tree and neural network. Although both models performed similarly, the classification tree will be the best model to predict whether a player is a novice or experienced. This is since decision trees are easy to interpret visually and demonstrate the significance of predictors.

Conclusions

The Hitters data set from the ISLR library in RStudio was used to predict an MLB player's salary and if they were an experienced player or not.

The best model to predict the salary of a player was the regression tree. The model demonstrated CRuns (career runs) to be the most important predictor of salary, with Hits and

Walks also being important. Those with over 813 career runs are most likely to have a very high salary. For those who understand baseball, it makes sense that someone who can consistently perform well hitting and base running will be paid the most overtime. Baseball is a game of maximizing the number of runs your team scores and minimizing the number the other team scores. Drafting or hiring a player which can help increase scoring output is a great investment for a team.

The best model to predict whether a player is experienced or not was the classification tree. The model found that CAtBat (career at bats) was the most significant predictor of a player's experience. Other supporting variables included CRuns (career runs) and CRBI (career RBIs). Players with 1833 career at bats or more almost always were considered experienced players. Again, these findings make sense. Throughout a player's career, they will naturally have more opportunity to have at bats, and better players have a chance to be able to score runs and RBIs.

Reflection

The seasonal data used was from the 1986-1987 season and the career data was from the same period. Having access to current MLB data would allow research to see if the results still stand, or if different variables have a bigger impact on salary or demonstrate career longevity. Also, there are many more statistics recorded today, and to a much more accurate degree. Being able to see the positions of the players in this study would have also been interesting. Seeing how position may affect a salary—which position is valued more. In addition, being able to see if a certain career has a shorter or longer timeline compared to others (pitcher versus outfield for example) may have been evident.

Appendix

The mentioned aspects of the R code for this research:

```
library(ISLR)
summary(Hitters)
```

```
##      AtBat            Hits         HmRun            Runs      
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00  
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25  
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00  
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91  
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00  
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00  
##                                                                
##       RBI             Walks            Years            CAtBat       
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0  
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8  
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0  
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7  
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2  
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0  
##                                                                      
##      CHits            CHmRun           CRuns             CRBI        
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00  
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75  
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50  
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12  
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25  
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00  
##                                                                      
##      CWalks        League  Division    PutOuts          Assists      
##  Min.   :   0.00   A:175   E:157    Min.   :   0.0   Min.   :  0.0  
##  1st Qu.:  67.25   N:147   W:165    1st Qu.: 109.2   1st Qu.:  7.0  
##  Median : 170.50                   Median : 212.0   Median : 39.5  
##  Mean   : 260.24                   Mean   : 288.9   Mean   :106.9  
##  3rd Qu.: 339.25                   3rd Qu.: 325.0   3rd Qu.:166.0  
##  Max.   :1566.00                   Max.   :1378.0   Max.   :492.0  
##                                                                    
##      Errors          Salary       NewLeague
##  Min.   : 0.00   Min.   :  67.5   A:176    
##  1st Qu.: 3.00   1st Qu.: 190.0   N:146    
##  Median : 6.00   Median : 425.0            
##  Mean   : 8.04   Mean   : 535.9            
##  3rd Qu.:11.00   3rd Qu.: 750.0            
##  Max.   :32.00   Max.   :2460.0            
##                  NA's   :59
```

#Data Wrangling / Planning

```
Hitters$experiencedPlayer <- ifelse(Hitters$Years <= 5.6, "No", "Yes")
Hitters$experiencedPlayer <- as.factor(Hitters$experiencedPlayer)
```

#Correlation Table

```
trainCor <- subset(train, select=-c(League, Division, NewLeague, experiencedP
layer))
res <- cor(trainCor)
round(res, 2)
```

```
##         AtBat Hits HmRun Runs  RBI Walks Years CAtBat CHits CHmRun CRuns
CRBI
## AtBat    1.00 0.97  0.58 0.91 0.82  0.66  0.03   0.22  0.23   0.22  0.25
0.23
## Hits     0.97 1.00  0.55 0.91 0.81  0.62  0.04   0.22  0.25   0.21  0.25
0.24
## HmRun    0.58 0.55  1.00 0.66 0.85  0.46  0.12   0.22  0.23   0.48  0.27
0.35
## Runs     0.91 0.91  0.66 1.00 0.80  0.72  0.00   0.19  0.21   0.25  0.25
0.22
## RBI      0.82 0.81  0.85 0.80 1.00  0.58  0.14   0.28  0.30   0.42  0.31
0.38
## Walks    0.66 0.62  0.46 0.72 0.58  1.00  0.14   0.29  0.29   0.37  0.36
0.33
## Years    0.03 0.04  0.12 0.00 0.14  0.14  1.00   0.91  0.89   0.73  0.87
0.86
## CAtBat   0.22 0.22  0.22 0.19 0.28  0.29  0.91   1.00  0.99   0.81  0.98
0.95
## CHits    0.23 0.25  0.23 0.21 0.30  0.29  0.89   0.99  1.00   0.80  0.98
0.95
## CHmRun   0.22 0.21  0.48 0.25 0.42  0.37  0.73   0.81  0.80   1.00  0.83
0.93
## CRuns    0.25 0.25  0.27 0.25 0.31  0.36  0.87   0.98  0.98   0.83  1.00
0.95
## CRBI     0.23 0.24  0.35 0.22 0.38  0.33  0.86   0.95  0.95   0.93  0.95
1.00
## CWalks   0.15 0.14  0.23 0.18 0.23  0.44  0.83   0.91  0.89   0.81  0.93
0.88
## PutOuts  0.33 0.32  0.23 0.28 0.30  0.27 -0.02   0.06  0.08   0.08  0.07
0.10
## Assists  0.36 0.33 -0.14 0.22 0.08  0.14 -0.08  -0.01 -0.01  -0.19 -0.03 -
0.10
## Errors   0.34 0.31 -0.02 0.22 0.16  0.10 -0.18  -0.09 -0.08  -0.20 -0.11 -
0.14
## Salary   0.38 0.43  0.31 0.41 0.42  0.43  0.40   0.53  0.56   0.53  0.57
0.57
##         CWalks PutOuts Assists Errors Salary
## AtBat     0.15    0.33    0.36   0.34   0.38
## Hits      0.14    0.32    0.33   0.31   0.43
## HmRun     0.23    0.23   -0.14  -0.02   0.31
## Runs      0.18    0.28    0.22   0.22   0.41
## RBI       0.23    0.30    0.08   0.16   0.42
## Walks     0.44    0.27    0.14   0.10   0.43
```

```
## Years     0.83   -0.02   -0.08  -0.18   0.40
## CAtBat    0.91    0.06   -0.01  -0.09   0.53
## CHits     0.89    0.08   -0.01  -0.08   0.56
## CHmRun    0.81    0.08   -0.19  -0.20   0.53
## CRuns     0.93    0.07   -0.03  -0.11   0.57
## CRBI      0.88    0.10   -0.10  -0.14   0.57
## CWalks    1.00    0.05   -0.06  -0.15   0.49
## PutOuts   0.05    1.00   -0.02   0.10   0.32
## Assists  -0.06   -0.02    1.00   0.68   0.05
## Errors   -0.15    0.10    0.68   1.00   0.03
## Salary    0.49    0.32    0.05   0.03   1.00
```