What Variable Influences Baseball Players Salaries The Most?

Aston Pearcy - Linkedin

Context

The dataset I am analysing is the Hitters dataset from the ISLR package in R. The dataset contains information on Major League Baseball players in the 1986 season and 1987 opening day, and is made up of 322 observations on the following 20 variables:

Variable Label	Meaning
AtBat	Number of times at bat in the season
Hits	Number of hits in the season
HmRun	Number of home runs in the season
Runs	Number of runs in the season
RBI	Number of runs batted in in the season
Walks	Number of walks in the season
Years	Number of years in the major leagues
CAtBat	Number of times at bat in career
CHits	Number of hits in career
CHmRun	Number of home runs in career
CRuns	Number of runs in career
CRBI	Number of runs batted in in career
CWalks	Number of walks in career
League	Factor with levels A and N denoting players league at end of season
Division	Factor with levels E and W denoting players division at end of season
PutOuts	Number of putouts in the season
Assists	Number of assists in the season
Errors	Number of errors in the season
NewLeague	Factor with levels A and N indicating league at beginning of 1987
Salary	Annual salary on opening day 1987 in thousands of dollars

Most of these variables are numerical.

I wanted to see what variables had the greatest effect on the salary of players. From an outside perspective sometimes the salaries that players are paid is highly subjective (in any sport) so I thought it would be interesting to see the relationship between a players performance statistics and the salary they're paid.

This work was done as part of my studies at the University of Otago in the STAT312 paper, "Modelling High Dimensional Data".

Analysis

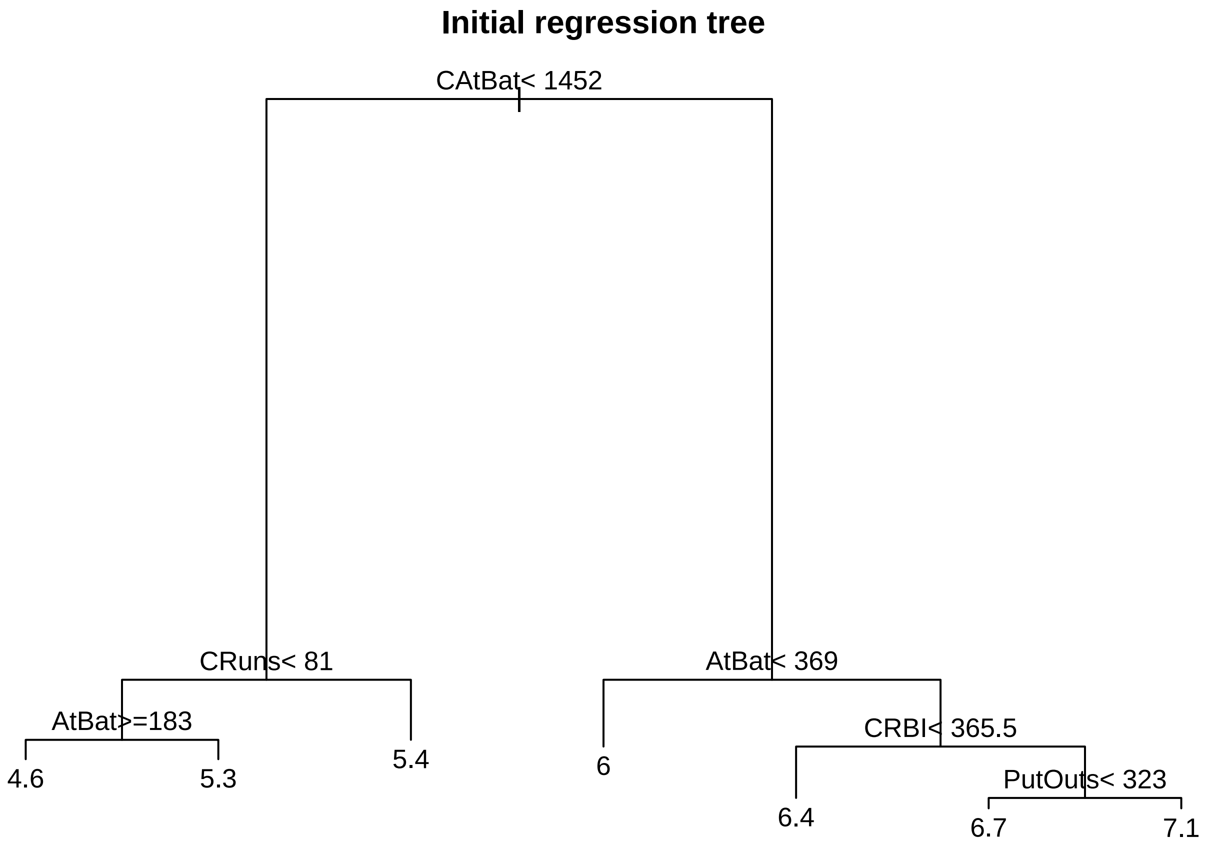
Setting up and cleaning data

```
library(ISLR)
d = na.omit(Hitters)
d$logSalary = log(d$Salary)
library(dplyr)
d = select(d, -Salary)
```

```
set.seed(499)
train = sample(1:nrow(d), 0.7*nrow(d))
test = -train
```

Constructing regression trees using rpart package

```
par(mar = c(0,0,3,0), cex = 0.85)
library(rpart)
rparthitters = rpart(logSalary~., data=d, subset=train, method = "anova")
plot(rparthitters)
text(rparthitters, pretty = 1, digits = 2)
title(main = "Initial regression tree")
```



The tree first partitions the data at the CAtBat variable. Seems that with this approach the number of times at bat over the career of a player has the largest effect on the players' salary. This doesn't seem unreasonable as the batting order in baseball is decided by the team manager. The earlier batters (1-4) bat more than the later ones and are generally the better batters, and so bring in more home runs for the team. Therefore, they are likely prioritsed and paid more than some of the other players. Earlier batters are also historically the more famous of the players, and so could be expected to be paid more money as they are a bigger name and carry a certain level of prestige with them.

Next, a bagged model was constructed (20 variables, 19 predictors, 1 outcome):

```
library(randomForest)
set.seed(590)
bhitters = randomForest(logSalary~., data=d, mtry = 19, subset=train, importance = TRUE)
round((bhitters$importance),2)
```

```
##           %IncMSE IncNodePurity
## AtBat        0.05          7.64
## Hits         0.02          3.90
## HmRun        0.00          1.26
## Runs         0.01          2.57
## RBI          0.01          4.14
## Walks        0.01          3.35
## Years        0.01          1.33
## CAtBat       0.40         69.37
## CHits        0.10         14.32
## CHmRun       0.01          2.74
## CRuns        0.11         16.02
## CRBI         0.05          7.30
## CWalks       0.03          4.94
## League       0.00          0.10
## Division     0.00          0.18
## PutOuts      0.01          3.64
## Assists      0.00          1.15
## Errors       0.00          0.88
## NewLeague    0.00          0.17
```

```
bhitters
```

```
## 
## Call:
##  randomForest(formula = logSalary ~ ., data = d, mtry = 19, importance = TRUE,      subset = train) 
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 19
## 
##           Mean of squared residuals: 0.1578381
##                     % Var explained: 80.25
```

The model captures ~80% of the variance contained within the data. The results of this bagged model are pleasingly in line with the regression tree analysis conducted above. The most important variable to the model was again found to be CAtBat, the number of times the player was at bat in their career. This was true with inclusion of the MSE, and inclusion of node purity.

Fitting random forest model:

```
set.seed(7000)
rfhitters = randomForest(logSalary~., data=d, subset = train, importance = TRUE)
rfhitters
```

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

```
##  randomForest(formula = logSalary ~ ., data = d, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.1502639
##                     % Var explained: 81.2
```

Setting values of mtry explicitly:

```
rfhitters10 = randomForest(logSalary~., data=d, mtry = 10,subset = train, importance = TRUE)
rfhitters10
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d, mtry = 10, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 0.1503758
##                     % Var explained: 81.19
```

Testing more values of mtry

```
rfhitters5 = randomForest(logSalary~., data=d, mtry = 5,subset = train, importance = TRUE)
rfhitters5
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d, mtry = 5, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 0.1470207
##                     % Var explained: 81.61
```

Seems like mtry = 5 gives the lowest mean squared residuals, and the greatest %variance explained.

```
round((rfhitters5$importance),2)
```

```
##           %IncMSE IncNodePurity
## AtBat        0.03          5.49
## Hits         0.03          5.59
## HmRun        0.00          1.58
## Runs         0.02          4.12
## RBI          0.01          5.08
## Walks        0.01          3.50
## Years        0.03          6.91
## CAtBat       0.19         26.40
## CHits        0.17         24.79
## CHmRun       0.02          4.39
## CRuns        0.13         19.52
## CRBI         0.08         15.40
## CWalks       0.08         16.08
## League       0.00          0.14
## Division     0.00          0.17
## PutOuts      0.01          2.86
## Assists      0.00          1.11
## Errors       0.00          0.88
## NewLeague    0.00          0.23
```

Again, CAtBat is the most important variable, however, there are some that are almost as important, these are CRuns, CHits, CRBI, and CWalks.

Constructing a boosted tree model:

```
library(gbm)
boostedhitters = gbm(logSalary~., data=d[train,], distribution = "gaussian", cv.folds = 10, n.cores=1,
```

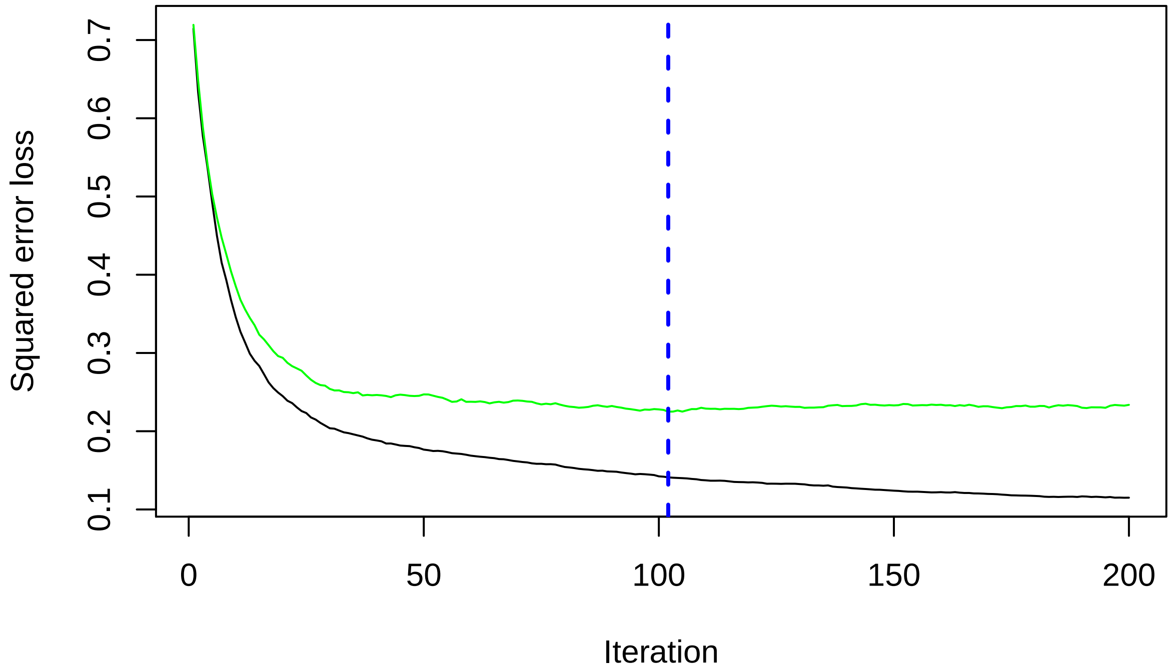
```
## CV: 1
## CV: 2
## CV: 3
## CV: 4
## CV: 5
## CV: 6
## CV: 7
## CV: 8
## CV: 9
## CV: 10
```

```
boostedhitters
```

```
## gbm(formula = logSalary ~ ., distribution = "gaussian", data = d[train,
##     ], n.trees = 200, cv.folds = 10, n.cores = 1)
## A gradient boosted model with gaussian loss function.
## 200 iterations were performed.
## The best cross-validation iteration was 102.
## There were 19 predictors of which 18 had non-zero influence.
```

Finding the best model:

```
gbm.perf(boostedhitters,method="cv")
```

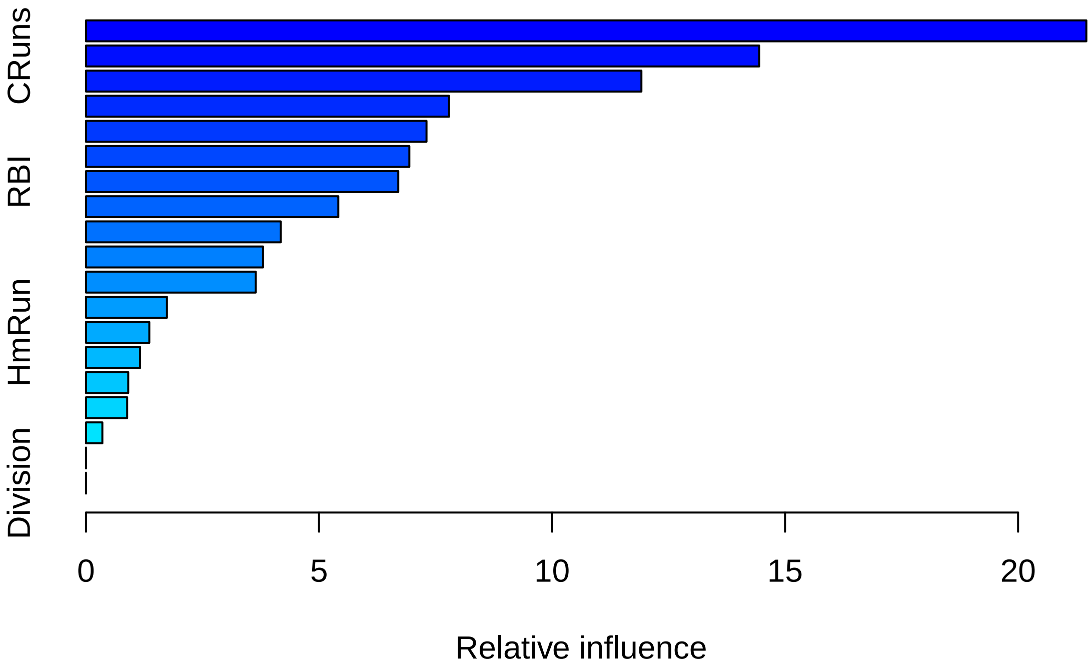


```
## [1] 102
```

n.trees for best model indicated by the dotted blue line. Found to be at a value of n.trees = 102.

Using optimal value for best model:

```
bestboostedhitters = gbm(logSalary~., data=d[train,], distribution = "gaussian", n.cores=1, n.trees = 1
summary(bestboostedhitters)
```

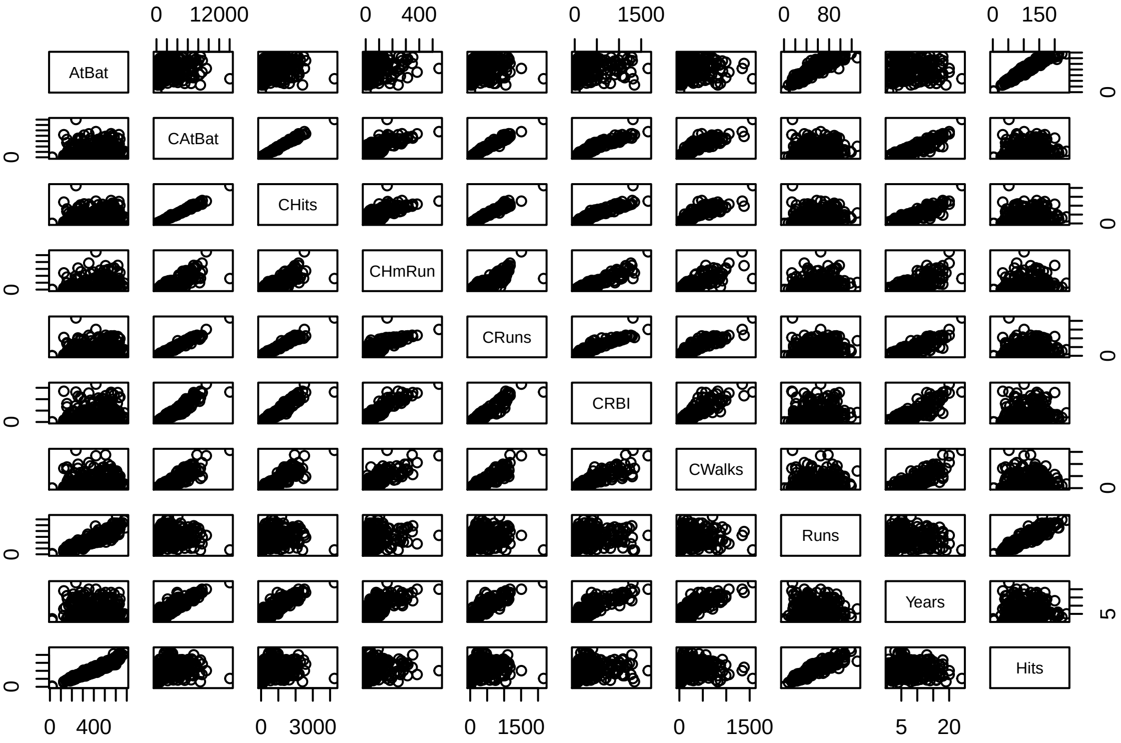


```
##                 var    rel.inf
## CAtBat       CAtBat 21.4650219
## CRuns         CRuns 14.4451475
## CWalks       CWalks 11.9179528
## CHits         CHits  7.7898355
## Years         Years  7.3067287
## CHmRun       CHmRun  6.9384683
## RBI             RBI  6.7011857
## PutOuts     PutOuts  5.4144391
## Hits           Hits  4.1793607
## CRBI           CRBI  3.7995203
## Walks         Walks  3.6423828
## AtBat         AtBat  1.7372979
## HmRun         HmRun  1.3587618
## Assists     Assists  1.1615789
## Errors       Errors  0.9069104
## Runs           Runs  0.8840250
## NewLeague NewLeague  0.3513826
## League       League  0.0000000
## Division   Division  0.0000000
```

Boosted tree model shows that the variable CAtBat has a very high relative influence, about two times that of the second most influential variable, which is CRBI. Perhaps unsurprisingly, the career long variables generally show more influence on the salary of a player than just the year long variables. The Years variable shows a moderate influence on the salary of the player, which makes sense as the longer a player has been in the major leagues the more experienced they are and the more value they can provide to a team, therefore earning a better salary.

Variables were plotted against each other to determine if there were any problems with collinearity. The dataset contains many career variables that could show an interaction with each other such as CAtBat and CHits. It is expected that the more times a player is at bat in their career then the more times they would have hit the ball.

```
plot(d[,c(1,8:13,4,7,2)])
```



As expected, the career variables show a high level of collinearity with other career variables. The variable AtBat displays a high level of collinearity with the Years and Hits variables.

Addressing the collinearity by removing the career variables and AtBat:

```
d2 = d[,-c(1, 8:13)]
```

Redoing previous analyses with newly generated dataset:

```
set.seed(499)
train = sample(1:nrow(d2), 0.7*nrow(d2))
test = -train
```

The bagged method:

```
set.seed(590)
bhitters = randomForest(logSalary~.,data=d2, mtry = 12,subset=train, importance = TRUE)
bhitters
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d2, mtry = 12, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 12
##
##           Mean of squared residuals: 0.1965651
##                     % Var explained: 75.41
```

```
round((bhitters$importance),2)
```

```
##           %IncMSE IncNodePurity
## Hits         0.13         23.79
## HmRun        0.00          2.88
## Runs         0.04          9.53
## RBI          0.04         10.25
## Walks        0.04          8.52
## Years        0.70         78.08
## League       0.00          0.30
## Division     0.00          0.46
## PutOuts      0.02          5.96
## Assists      0.00          2.13
## Errors       0.00          1.72
## NewLeague    0.00          0.37
```

The model explains ~75% of the variance in the dataset. The most important variable is Years, including both MSE and NodePurity.

Random forest method:

```
set.seed(7000)
rfhitters = randomForest(logSalary~., data=d2, subset = train, importance = TRUE)
rfhitters
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d2, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 0.2073649
##                     % Var explained: 74.06
```

Will test other values of mtry:

```
rfhitters6 = randomForest(logSalary~., data=d2, mtry = 6, subset = train, importance = TRUE)
rfhitters6
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d2, mtry = 6, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.1925993
##                     % Var explained: 75.91
```

```
rfhitters10 = randomForest(logSalary~., data=d2, mtry = 10, subset = train, importance = TRUE)
rfhitters10
```

```
##
## Call:
##  randomForest(formula = logSalary ~ ., data = d2, mtry = 10, importance = TRUE,      subset = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 0.1919526
##                     % Var explained: 75.99
```

mtry= 10 gives the highest variance explained and the lowest mean of squared residuals.

Lastly, using the boosted tree method:

```
boostedhitters = gbm(logSalary~., data=d2[train,], distribution = "gaussian", cv.folds = 10, n.cores=1,
```

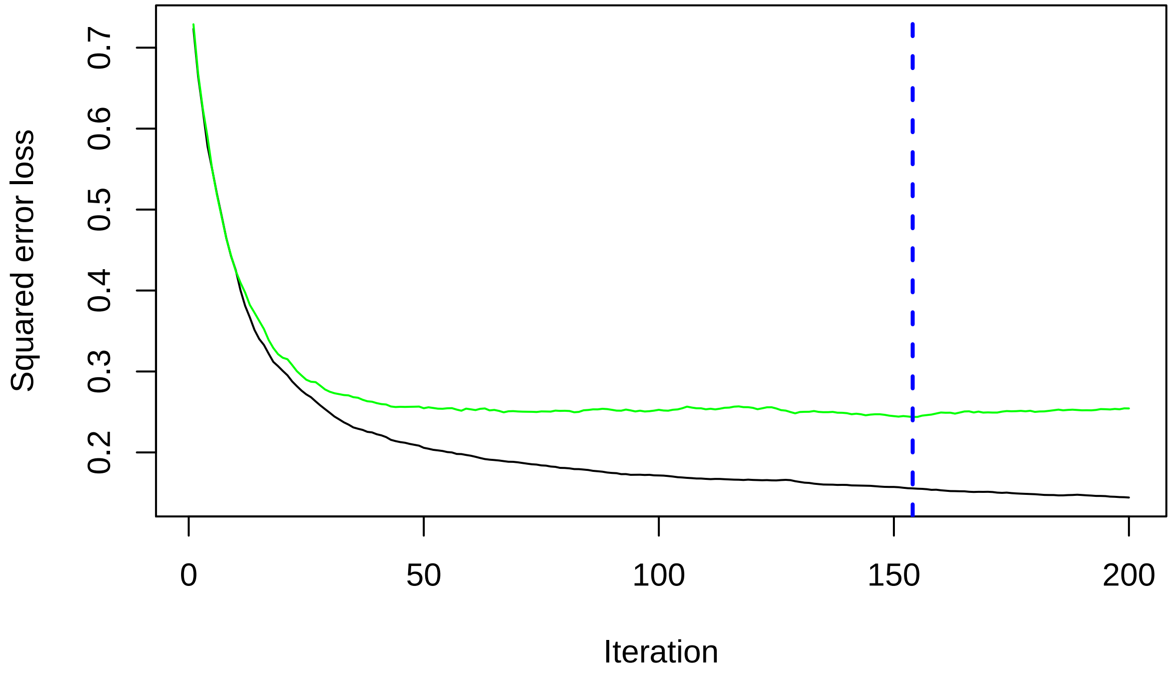
```
## CV: 1
## CV: 2
## CV: 3
## CV: 4
## CV: 5
## CV: 6
## CV: 7
## CV: 8
## CV: 9
## CV: 10
```

```
boostedhitters
```

```
## gbm(formula = logSalary ~ ., distribution = "gaussian", data = d2[train,
##     ], n.trees = 200, cv.folds = 10, n.cores = 1)
## A gradient boosted model with gaussian loss function.
## 200 iterations were performed.
## The best cross-validation iteration was 154.
## There were 12 predictors of which 12 had non-zero influence.
```

Finding the best model:

```
gbm.perf(boostedhitters,method="cv")
```

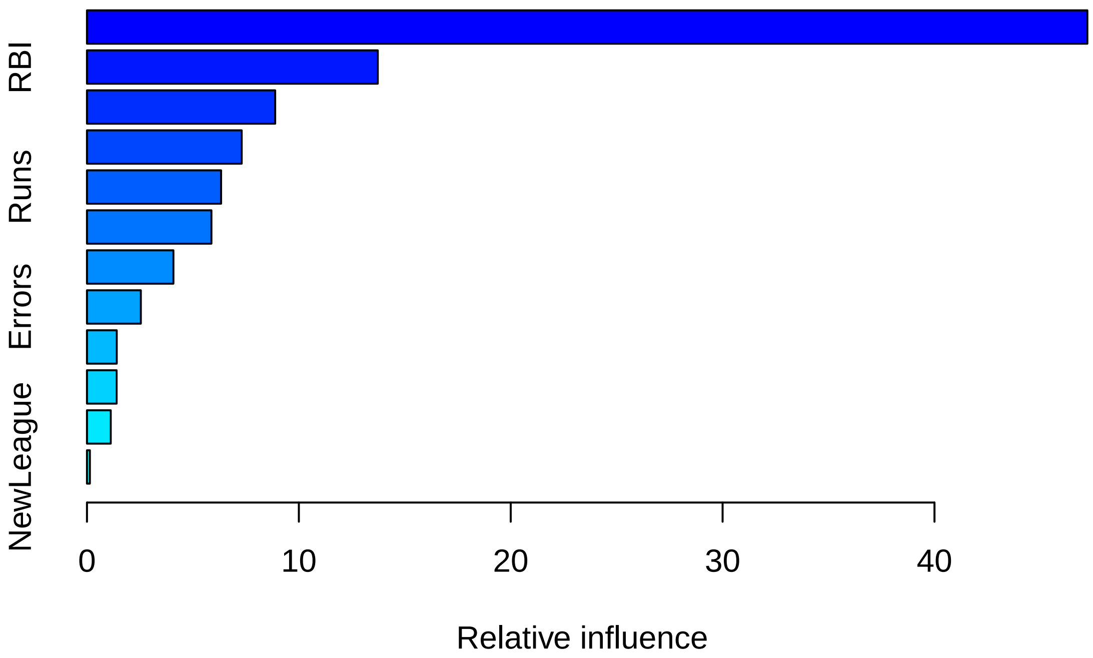


```
## [1] 154
```

n.trees for best model indicated by the dotted blue line. Found to be at a value of n.trees = 154.

Using optimal value for best model:

```
bestboostedhitters = gbm(logSalary~., data=d2[train,], distribution = "gaussian", n.cores=1, n.trees = 
summary(bestboostedhitters)
```



```
##                 var    rel.inf
## Years         Years 47.2154973
## RBI             RBI 13.7242367
## Hits           Hits  8.8821675
## Walks         Walks  7.3007125
## Runs           Runs  6.3281532
## PutOuts     PutOuts  5.8725659
## HmRun         HmRun  4.0779541
## Errors       Errors  2.5413457
## Assists     Assists  1.4045334
## League       League  1.3973224
## Division   Division  1.1200602
## NewLeague NewLeague  0.1354511
```

The variable with the greatest influence is the Years variable. The Years variable is very influential, having about four times the influence of the second variable. The Years variable being highly influential could suggest that, like before, the more experienced players are paid more due to the value that they provide the team.

Another influential variable is the Hits variable. The influence of the Hits variable could be viewed from the perspective that highly paid players tend to be at bat more as they are generally placed earlier in the batting order and so have more opportunities to gain a hit. They also could have a higher hit rate, and that could be a driving factor for them being paid a higher salary.

Both of these observations are consistent with the bagged method used prior.

Summary

Using a wide variety of regression tree methods an investigation into the influence of a range of variables on the salary of baseball players was conducted. Initial analyses suggested that the variable CAtBat - the number of times at bat in a players career - had the greatest influence on the salary of a player. The collinearity of the data was looked at, and it was determined that the career variables (Variables with a C in front of their label) displayed a high level of collinearity with each other. These variables were removed, along with others showing collinear behaviour, and analyses were repeated. The most important variable to a players salary was then determined to be the Years variable, which is the number of years in the major leagues. This could be due to their reputation and the trust that can be placed in their ability, due to a proven track record.