Salaries of Baseball Players

Kelso Quan

December 20, 2018

Abstract

Determining one's salary in baseball is a finicky game. There many be algorithms that the MLB uses, but this analysis is trying to develop ways in predicting a baseball player's salary based on their statistics and attributes. Using linear regression, lasso, and bagging, the analysis determined that the bagging without bootstrap method had the lowest MSE, but regression had a similar MSE. A sufficient method would be linear regression, but if the club owner is being frugal about their money, then bagging without bootstrap method would yield slightly better results.

1 Introduction

Baseball, an American pastime, is a complex sports game. There are so many movies on baseball. Money ball is a recent movie which delves into the salaries of baseball players. The general manager of the Oakland A's was faced with a tight budget where he must reinvent his team by outsmarting the richer ball clubs. In 1968, Baseball in America was going through organizational changes. Leagues were changed and Divisions were created within those leagues. In this day and age of baseball, players are becoming more and more expensive with their ever larger salaries. It would be useful for sport stats enthusiasts and club owners to see how much their players are worth and predict how much a player may truly be worth. The analysis will be using three methods: regression, lasso, bagging.

2 Methods

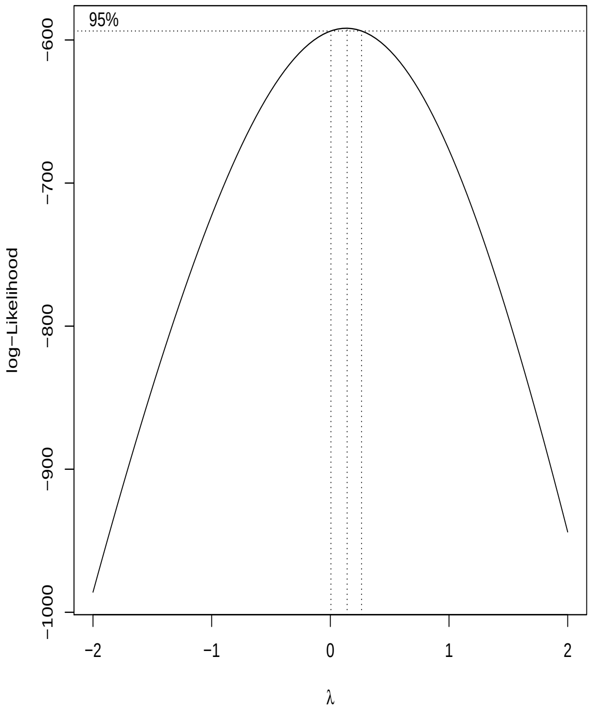
The data was taken in 1986 with 322 observations. There were 59 observations that were omitted because those cases had 'na' values. For those who do not know baseball statistics: AtBat is the number of times at bat, HmRun is the number of home runs, Runs is the number of runs, RBI is the number of runs batted in, Walks is the number of walks, Years is number of years in the major leagues, CAtBat is the number of times at bat during his career, CHits number of hits during his career, CHmRun is number of home runs during his career, CRuns is number of runs during his career, CRBI is number of runs batted in during career, CWalks is number of walks during his career, League has two factors American and National league, Division has two factors East and West, PutOuts is number of put outs, Assists is the number of assists, Errors is the number of errors, Salary is annual salary on opening day in thousands of dollars in 1987, NewLeague is a factor with American and National indicating the player's league at the beginning of 1987. The analysis was done in R/RStudio.

3 Results

3.1 Exploratory Data Analysis

A boxcox transformation showed that the Salary had to be log transformed. Figure 1 shows that λ is close to zero which is an indication that the response should be log transformed. Creating a histogram for the

Figure 1: BoxCox transformation on Salary



```
## [1] 0.1414141
```

response confirmed that salary indeed needed to be log transformed. Then histogram of every covariate must be made to see if they are skewed. If so, the variable may be log transformed to better fit the model. Not every predictor variable had to be log transformed. Univariate analysis was done between single variables and the response to better understand the significance between that covariate and the response. To further see correlation between the response and predictor variables, a correlation matrix was created. Figure 4 shows that most variables are correlated to the response with the exception of League, Assists, Errors, and NewLeague. Division was the only variable that was negatively correlated to Salary.

3.2 Model Fitting/Inferences

A model was tested to see if all linear terms without transformation was possible. The residual plots looked terrible. Then a model with the log transformed response and a couple of necessarily log transformed predictors along with non transformed predictors made a decent model. The regression model without interaction terms included: Runs, log(CHits), Division, and PutOuts. This model was found by the stepAIC function with both directions, then insignificant terms were eliminated one at a time.

The regression model with interaction terms included: AtBat, HmRun, RBI, log(CRuns), PutOuts, Assists, Errors, AtBat:Walks, AtBat:log(CHits), HmRun:log(CHits), HmRun:League, RBI:League, RBI:Errors, log(CHits):log(CRuns), log(CHits):League, log(CHits):Assists, log(CHits:Errors), log(CHits):NewLeague, log(CRuns):League, log(CRuns):NewLeague, Division:Assists, PutOuts:Assists, and League:Putouts. This model was found by the stepAIC function with both directions, then insignificant terms were eliminated one at a time. The qq plot for the model looked normal enough. After looking at these regression models, the analysis proceeded to train data for the lasso and bagging methods. The training data was half of the data.

Plotting the points for the bagging and no bootstrap bagging showed that the errors were random. By strictly looking at the MSE of all three methods, bagging with no bootstrap was better. There were three potential outliers: Mike Schmidt, Steve Balboni, and Terry Kennedy.

4 Conclusion

The best method was no bootstrap bagging. It had the lowest MSE compared to the other methods. Table 1 shows that Bagging without bootstrap had the lowest MSE. But in terms of ease and interpretability, linear regression is good enough. Club owners should be happy with the linear regression model, but if not, they should go for Bagging No Bootstrap method. This analysis was limited to the regression, bagging, and lasso methods. Perhaps, ridge regression may have been explored. Random Forest was also looked into during this analysis, but was not considered. But it turns out that random forest is the best method in terms of low MSE.

Table 1: Comparing MSE by Methods

	Regression	Bagging	No Bootstrap
MSE	0.38	0.37	0.36

Figure 2: Correlation Matrix between all Variables

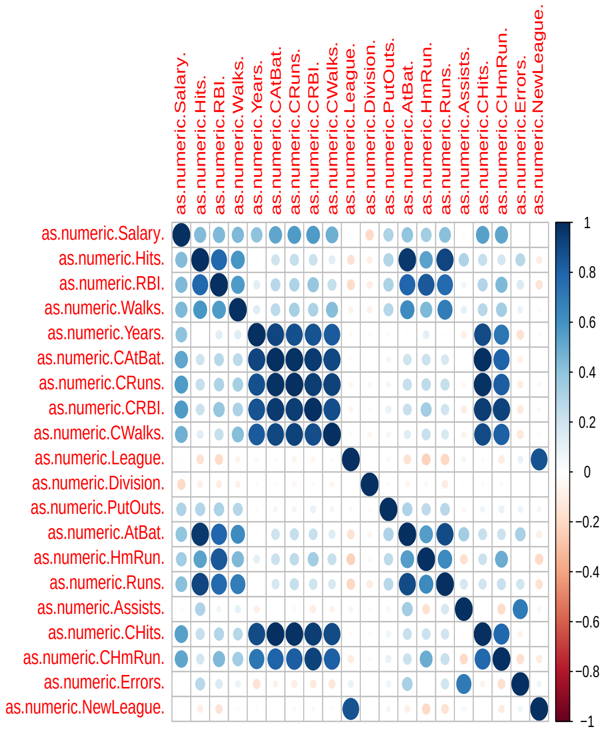
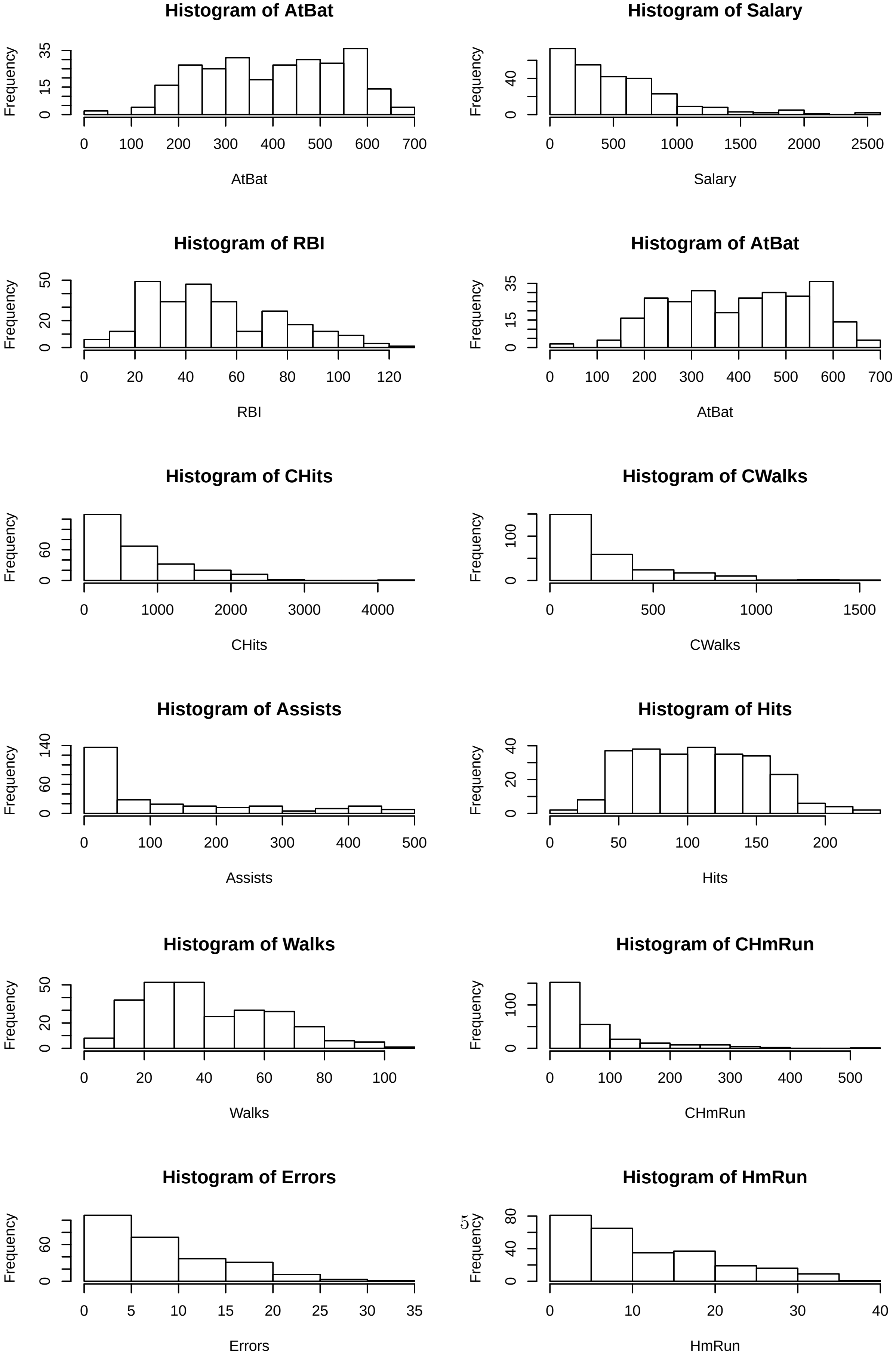


Table 2: VIF of Regression Model without Interaction terms

Runs	lCHits	Division	PutOuts
1.25	1.15	1.01	1.08

Appendix A: Auxiliary Graphics and Tables

Figure 3: Histograms of untransformed variables



Appendix B: R Code

```r
library(ISLR)
library(MASS)
library(corrplot)
library(car)
library(KernSmooth)
library(leaps)
library(xtable)
library(foreach)
library(randomForest)
library(glmnet)
library(tree)
sum(is.na(Hitters))

Hitters2<-na.omit(Hitters)

head(Hitters2)
names(Hitters2)

attach(Hitters2)

hist(AtBat)
hist(Salary)
hist(RBI)
hist(AtBat)
hist(CHits)
hist(CWalks)
hist(Assists)
hist(Hits)
hist(Walks)
hist(CHmRun)
hist(Errors)
hist(HmRun)
hist(Years)
hist(CRuns)
hist(Runs)
hist(CAtBat)
hist(CRBI)
hist(PutOuts)


###########################################
#### Checking for Log Transformation #######
############################################

bc1 <- boxcox(Salary~., data = Hitters2)
bc1$x[bc1$y==max(bc1$y)]


lSalary <- log(Salary)

hist(lSalary, breaks = 20)
hist(Salary)

########################################################
###LOWESS TO FIND FUNCTIONAL FORM OF VARIABLES###########
########################################################

plot(RBI,lSalary)
lines(lowess(RBI,lSalary), col="blue")

plot(AtBat,lSalary)
lines(lowess(AtBat,lSalary), col="blue")

```

```r
plot(CWalks,lSalary)
lines(lowess(CWalks,lSalary), col="blue")

plot(log(CWalks),lSalary)
lines(lowess(log(CWalks),lSalary), col="blue")

plot(CHits,lSalary)
lines(lowess(CHits,lSalary), col="blue")

plot(log(CHits),lSalary)
lines(lowess(log(CHits),lSalary), col="blue")

plot(Assists,lSalary)
lines(lowess(Assists,lSalary), col="blue")

plot(Hits,lSalary)
lines(lowess(Hits,lSalary), col="blue")

plot(Walks,lSalary)
lines(lowess(Walks,lSalary), col="blue")

plot(CHmRun,lSalary)
lines(lowess(CHmRun,lSalary), col="blue")

plot(log(CHmRun),lSalary)
lines(lowess(log(CHmRun),lSalary), col="blue")

plot(Years,lSalary)
lines(lowess(Years,lSalary), col="blue")

plot(log(Years),lSalary)
lines(lowess(log(Years),lSalary), col="blue")

plot(CRuns,lSalary)
lines(lowess(CRuns,lSalary), col="blue")

plot(log(CRuns),lSalary)
lines(lowess(log(CRuns),lSalary), col="blue")

plot(Runs,lSalary)
lines(lowess(Runs, lSalary), col="blue")

plot(log(CAtBat),lSalary)
lines(lowess(log(CAtBat),lSalary), col="blue")

plot(CRBI,lSalary)
lines(lowess(CRBI,lSalary), col="blue")

plot(log(CRBI),lSalary)
lines(lowess(log(CRBI),lSalary), col="blue")

plot(PutOuts,lSalary)
lines(lowess(PutOuts,lSalary), col="blue")

plot(HmRun,lSalary)
lines(lowess(HmRun,lSalary), col="blue")

plot(Errors,lSalary)
lines(lowess(Errors,lSalary), col="blue")


boxplot(lSalary~League)
boxplot(lSalary~Division)

#Log transform below variables
```

```r
lCWalks<-log(CWalks)
lCHits <-log(CHits)
lCRuns <-log(CRuns)
lCAtBat <-log(CAtBat)
lCRBI <- log(CRBI)
lYears <-log(Years)
lCHmRun <-log(CHmRun)

##########################
###CORRELATION MATRIX######
###########################

Hitters2vars = data.frame(as.numeric(Salary), as.numeric(Hits), as.numeric(RBI), as.numeric(Walks),
                          as.numeric(Years), as.numeric(CAtBat), as.numeric(CRuns), as.numeric(CRBI),
                          as.numeric(CWalks), as.numeric(League), as.numeric(Division), as.numeric(PutOuts),
                          as.numeric(AtBat), as.numeric(HmRun), as.numeric(Runs), as.numeric(Assists),
                          as.numeric(CHits), as.numeric(CHmRun), as.numeric(Errors), as.numeric(NewLeague))
Hit2var = cor(Hitters2vars)
corrplot(Hit2var)

###################################
#### Univariate Analysis ##########
###################################

fit.Years<-lm(lSalary~Years, data = Hitters2) #2e-16
summary(fit.Years)
fit.CAtBat<-lm(lSalary~CAtBat, data = Hitters2) #2e-16
summary(fit.CAtBat)
fit.CRuns<-lm(lSalary~CRuns, data = Hitters2) #2e-16
summary(fit.CRuns)
fit.CRBI<-lm(lSalary~CRBI, data = Hitters2) #2e-16
summary(fit.CRBI)
fit.CWalks<-lm(lSalary~CWalks, data = Hitters2) #2e-16
summary(fit.CWalks)
fit.CHits<-lm(lSalary~CHits, data = Hitters2) #2e-16
summary(fit.CHits)
fit.CHmRun<-lm(lSalary~CHmRun, data = Hitters2) #2e-16
summary(fit.CHmRun)
# Keep CHits, remove other variables #

fit.AtBat<-lm(lSalary~AtBat, data = Hitters2)
summary(fit.AtBat)
fit.Hits<-lm(lSalary~Hits, data = Hitters2)
summary(fit.Hits)
# Keep AtBat, remove Hits #

Hitters2.b.vars = data.frame(as.numeric(Salary), as.numeric(RBI), as.numeric(Walks),
                             as.numeric(League), as.numeric(Division), as.numeric(PutOuts),
                             as.numeric(AtBat), as.numeric(HmRun), as.numeric(Runs), as.numeric(Assists),
                             as.numeric(Errors), as.numeric(NewLeague))
Hit2var.b = cor(Hitters2.b.vars)
corrplot(Hit2var.b)


###########################################
#### Model Building no/Interaction##########
###########################################


```

```
fit.cor<-lm(lSalary~AtBat+HmRun+Runs+RBI+Walks+lCHits+
              League+Division+PutOuts+Assists+Errors+NewLeague, data = Hitters2)
summary(fit.cor)

stepAIC(fit.cor, direction="both")  ##AIC = -285


fit.cor.1<-lm(lSalary ~ AtBat + Runs + RBI + Walks + lCHits +
                League + Division + PutOuts + Assists + Errors, data = Hitters2)
summary(fit.cor.1)


#Remove Walks

fit.cor.2<-lm(lSalary ~ AtBat + Runs + RBI + lCHits +
                League + Division + PutOuts + Assists + Errors, data = Hitters2)
summary(fit.cor.2)

#Remove Assists

fit.cor.3<-lm(lSalary ~ AtBat + Runs + RBI + lCHits +
                League + Division + PutOuts + Errors, data = Hitters2)
summary(fit.cor.3)

#Remove Errors

fit.cor.4<-lm(lSalary ~ AtBat + Runs + RBI + lCHits +
                League + Division + PutOuts, data = Hitters2)
summary(fit.cor.4)

#Remove RBI

fit.cor.5<-lm(lSalary ~ AtBat + Runs + lCHits +
                League + Division + PutOuts, data = Hitters2)
summary(fit.cor.5)

#Remove AtBat

fit.cor.6<-lm(lSalary ~ Runs + lCHits +
                League + Division + PutOuts, data = Hitters2)
summary(fit.cor.6)

#Remove League

fit.cor.7<-lm(lSalary ~ Runs + lCHits +
                Division + PutOuts, data = Hitters2)
summary(fit.cor.7)

vif(fit.cor.7)

########################################
#### Model Building w/interactions #####
#######################################

logtransforms = data.frame(AtBat, HmRun, RBI, Walks,
                           lCHits, lCRuns, League,
                           Division, PutOuts, Assists, Errors, NewLeague)


fit.interaction<-lm(lSalary~.*., data = logtransforms)
summary(fit.interaction)

stepAIC(fit.interaction, direction="both")  ##AIC = -977.24

fit.with.interactions<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                             lCRuns + League + Division + PutOuts + Assists + Errors +
                             NewLeague + AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                             HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                             Walks:lCRuns + Walks:Division + lCHits:lCRuns + lCHits:League +
                             lCHits:Division + lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                             lCRuns:League + lCRuns:Division + lCRuns:NewLeague + League:Division +
                             League:NewLeague + Division:PutOuts + Division:Assists +
                             PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions)

#Remove Walks:Division

fit.with.interactions1<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + Division + PutOuts + Assists + Errors +
                              NewLeague + AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Division + lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:Division + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:PutOuts + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions1)

#Remove PutOut:Division

fit.with.interactions2<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + Division + PutOuts + Assists + Errors +
                              NewLeague + AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Division + lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:Division + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions2)


#Remove CHits:Division

fit.with.interactions3<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + Division + PutOuts + Assists + Errors +
                              NewLeague + AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:Division + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions3)

#Remove CRuns:Division

fit.with.interactions4<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + Division + PutOuts + Assists + Errors +
```

```r
                              NewLeague + AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions4)

#Remove NewLeague

fit.with.interactions5<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + Division + PutOuts + Assists + Errors +
                              AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions5)

#Remove Division

fit.with.interactions6<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + lCHits +
                              lCRuns + League + PutOuts + Assists + Errors +
                              AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions6)

#Remove CHits

fit.with.interactions7<-lm(lSalary ~ AtBat + HmRun + RBI + Walks +
                              lCRuns + League + PutOuts + Assists + Errors +
                              AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague + League:Division +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions7)

#Remove League:Division

fit.with.interactions8<-lm(lSalary ~ AtBat + HmRun + RBI + Walks +
                              lCRuns + League + PutOuts + Assists + Errors +
                              AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions8)

#Remove Walks
fit.with.interactions9<-lm(lSalary ~ AtBat + HmRun + RBI  +
                              lCRuns + League + PutOuts + Assists + Errors +
                              AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                              HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
                              Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                              lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                              lCRuns:League + lCRuns:NewLeague +
                              League:NewLeague + Division:Assists +
                              PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions9)

#Remove Walks:Hits
fit.with.interactions10<-lm(lSalary ~ AtBat + HmRun + RBI  +
                                   lCRuns + League + PutOuts + Assists + Errors +
                                   AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                                   HmRun:NewLeague + RBI:League + RBI:Errors +
                                   Walks:lCRuns + lCHits:lCRuns + lCHits:League +
                                   lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                                   lCRuns:League + lCRuns:NewLeague +
                                   League:NewLeague + Division:Assists +
                                   PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions10)

#Remove CRuns:Walks
fit.with.interactions11<-lm(lSalary ~ AtBat + HmRun + RBI  +
                                   lCRuns + League + PutOuts + Assists + Errors +
                                   AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                                   HmRun:NewLeague + RBI:League + RBI:Errors +
                                   lCHits:lCRuns + lCHits:League +
                                   lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                                   lCRuns:League + lCRuns:NewLeague +
                                   League:NewLeague + Division:Assists +
                                   PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions11)

#Remove HmRun:NewLeague
fit.with.interactions12<-lm(lSalary ~ AtBat + HmRun + RBI  +
                                   lCRuns + League + PutOuts + Assists + Errors +
                                   AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                                   RBI:League + RBI:Errors +
                                   lCHits:lCRuns + lCHits:League +
                                   lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                                   lCRuns:League + lCRuns:NewLeague +
                                   League:NewLeague + Division:Assists +
                                   PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions12)

#Remove League
fit.with.interactions13<-lm(lSalary ~ AtBat + HmRun + RBI  +
                                   lCRuns + PutOuts + Assists + Errors +
                                   AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                                   RBI:League + RBI:Errors +
                                   lCHits:lCRuns + lCHits:League +
                                   lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                                   lCRuns:League + lCRuns:NewLeague +
                                   League:NewLeague + Division:Assists +
                                   PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions13)

#Remove League:NewLeague
fit.with.interactions14<-lm(lSalary ~ AtBat + HmRun + RBI  +
                                   lCRuns + PutOuts + Assists + Errors +
                                   AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
                                   RBI:League + RBI:Errors +
                                   lCHits:lCRuns + lCHits:League +
                                   lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
                                   lCRuns:League + lCRuns:NewLeague +
                                Division:Assists +
                                PutOuts:Assists + League:PutOuts, data = logtransforms)
summary(fit.with.interactions14)

vif(fit.with.interactions14)

##############################################
#### MODEL DIAGNOSTICS no Interactions ######
##############################################

plot(predict(fit.cor.7), rstudent(fit.cor.7), ylab="Studentized Residuals", xlab="Predicted")
identify(predict(fit.cor.7), rstudent(fit.cor.7), labels=row.names(Hitters2)) # 'escape to finish'
predict(fit.cor.7)[rstudent(fit.cor.7)==min(rstudent(fit.cor.7))]


sresid <- studres(fit.cor.7)
hist(sresid, freq=FALSE, xlab = "Residuals", main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit, col = "blue")


qqPlot(fit.cor.7, main="QQ Plot", ylab="Studentized Residuals")


cutoff <- 4/((nrow(set2)-length(fit.cor.7$coefficients)-2))
plot(fit.cor.7, which=4, cook.levels=cutoff) # influence Plot


influencePlot(fit.cor.7, id.method="identify",
              main="Influence Plot", sub="Circle size is proportial to Cook's Distance")

varif = vif(fit.cor.7)
varif



##############################################
#### MODEL DIAGNOSTICS w/Interactions ######
##############################################

plot(predict(fit.with.interactions14), rstudent(fit.with.interactions14), ylab="Studentized Residuals", xlab="Predicted")
identify(predict(fit.with.interactions14), rstudent(fit.with.interactions14), labels=row.names(Hitters2)) # 'escape to finish'
predict(fit.with.interactions14)[rstudent(fit.with.interactions14)==min(rstudent(fit.with.interactions14))]


sresid <- studres(fit.with.interactions14)
hist(sresid, freq=FALSE, xlab = "Residuals", main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit, col = "blue")


qqPlot(fit.with.interactions14, main="QQ Plot", ylab="Studentized Residuals")


cutoff <- 4/((nrow(set2)-length(fit.with.interactions14$coefficients)-2))
plot(fit.cor.7, which=4, cook.levels=cutoff) # influence Plot


```

```
influencePlot(fit.with.interactions14, id.method="identify",
              main="Influence Plot", sub="Circle size is proportial to Cook's Distance")
vif(fit.with.interactions14)
#########################
######## Lasso #########
#########################

x=model.matrix(Salary~., Hitters2)[,-1]
y=Salary

grid=10^seq(10,-2,length=100)
lasso.mod = glmnet(x, Salary, alpha=1, lambda=grid) # alpha=1 is L1 norm, lasso penalty
plot(lasso.mod)
lasso.coef = predict(lasso.mod,type="coefficients")
lasso.coef = predict(lasso.mod,type="coefficients", s=0)  # s is penalty parameter lambda
summary(lm(Salary~., Hitters2)) # same coeff as above lasso

set.seed(1)
train=sample(1:nrow(x), nrow(x)/2) # split data in half
test=(-train)
y.test=y[test]
lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)
plot(lasso.mod)
set.seed(1)
# minimizes squared-error loss (prediction error)
# run lasso on training set

test=(-train)
y.test=y[test]
cv.out=cv.glmnet(x[train,],y[train],alpha=1)

plot(cv.out)
bestlam=cv.out$lambda.min  #16.78016
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
mean((lasso.pred-y.test)^2) # prediction error for best lambda
out=glmnet(x,y,alpha=1,lambda=grid) # run lasso on whole data set
lasso.coef=predict(out,type="coefficients",s=bestlam)
lasso.coef
lasso.coef[lasso.coef!=0]

mean((lasso.pred-y.test)^2) #100743.4

set.seed (1)
train = sample(1:nrow(Hitters2), nrow(Hitters2)/2)
tree.Hitters=tree(Salary~ . ,Hitters2 ,subset=train)
summary(tree.Hitters)

plot(tree.Hitters)
text(tree.Hitters ,pretty=0)

bag.Hitters=randomForest(Salary~ .,data=Hitters2,subset=train,
                         mtry=19,importance =TRUE)
bag.Hitters

yhat.bag = predict(bag.Hitters ,newdata=Hitters2[-train ,])
plot(yhat.bag, Hitters.test)
abline(0,1)
mean((yhat.bag-Hitters.test)^2)


rf.Hitters=randomForest(Salary~ .,data=Hitters2,subset=train,
                        mtry=7,importance =TRUE)
yhat.rf = predict(rf.Hitters ,newdata=Hitters2[-train ,])

importance(rf.Hitters)
```

```

varImpPlot (rf.Hitters)

############################
#### Finding best MSE #####
############################


#create a new dataframe with hitters for the logsalary

Hitters3<-data.frame(lSalary, Hits, RBI, Walks,
                       Years, CAtBat, CRuns, CRBI,
                       CWalks, League, Division, PutOuts,
                       AtBat, HmRun, Runs, Assists,
                       CHits, CHmRun, Errors, NewLeague)

# Generate training and testing sets

set.seed(1)
train = sample(1:nrow(Hitters3), nrow(Hitters3)/2)
Hitters.train = Hitters3[train,]
Hitters.test = Hitters3[-train,]

# Perform regression
fit_lm = lm(lSalary~., data=Hitters.train)
pred_lm = predict(fit_lm, Hitters.test) # predicted test set
lm_MSE = mean((pred_lm - Hitters.test$lSalary)^2) # MSE ~ 0.378902

# Bagging (bootstrap aggregration) a regression model fit
set.seed(1)
iterations = 1000; n = nrow(Hitters.train)
predictions = foreach(m=1:iterations,.combine=cbind) %do% {
  # sample with replacement (bootstrap)
  training_positions = sample(nrow(Hitters.train), size=n, replace=TRUE)
  lm_fit = lm(lSalary ~ ., data=Hitters.train[training_positions,])
  predict(lm_fit, newdata=Hitters.test)
}

pred_bag<-rowMeans(predictions)
bag_MSE = sum((Hitters.test$lSalary-pred_bag)^2)/n # MSE ~ 0.3672615
plot(pred_bag)
# Bagging regression without bootstrap
# randomly subset training data rather than bootstrap
set.seed(1)
bagging_lm = function(training, testing, length_divisor=4, iterations=1000)
{
  predictions<-foreach(m=1:iterations,.combine=cbind) %do% {
    training_positions = sample(nrow(training), size=floor((nrow(training)/length_divisor))
    )
    train_pos = 1:nrow(training) %in% training_positions
    # FUNCTION NOT AUTOMATED: must name response in following 'lm' call
    lm_fit = lm(lSalary ~ ., data=training[train_pos,])
    predict(lm_fit,newdata=testing)
  }
  rowMeans(predictions)
}
bagreg_pred = bagging_lm(Hitters.train, Hitters.test)
bagreg_MSE = sum((Hitters.test$lSalary-bagreg_pred)^2)/n # MSE ~ 0.3568651
# Results
results = cbind(lm_MSE, bag_MSE, bagreg_MSE)
colnames(results) = c("Regression", "Bagging", "No Bootstrap")
results
```

Listing 1: Baseball Salary