

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

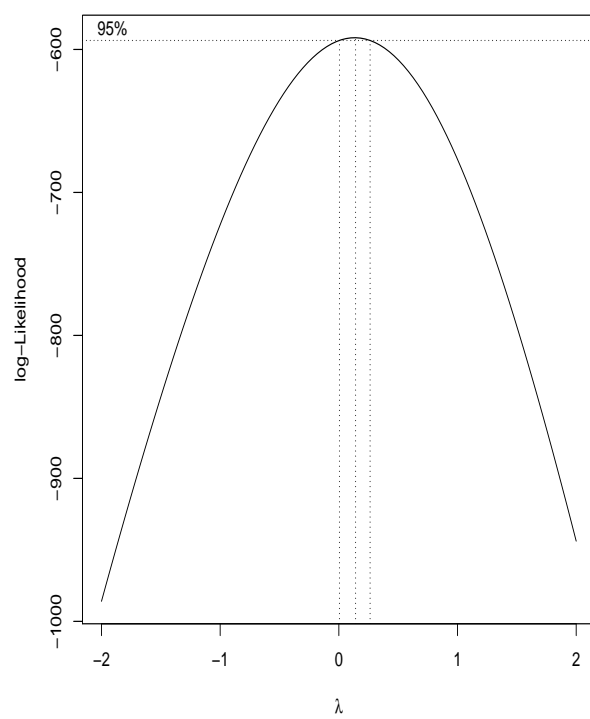
The data was taken in 1986 with 322 observations. There were 59 observations that were omitted because those cases had 'na' values. For those who do not know baseball statistics: AtBat is the number of times at bat, HmRun is the number of home runs, Runs is the number of runs, RBI is the number of runs 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

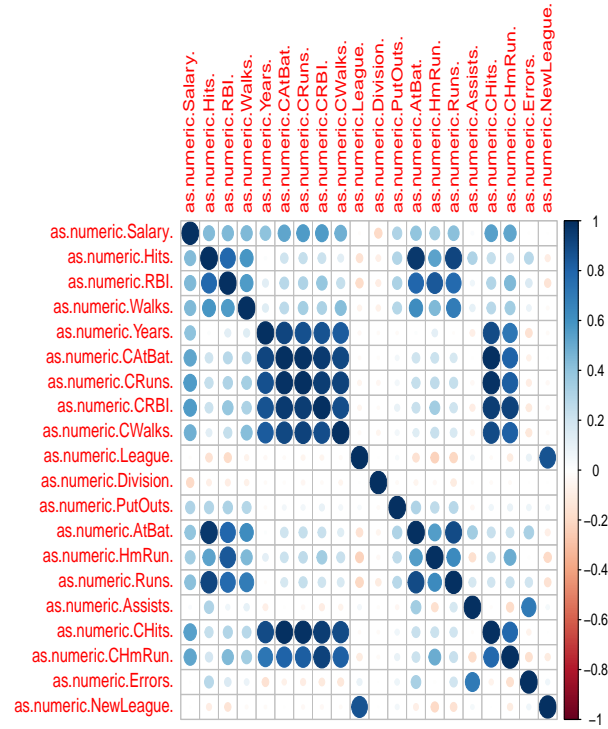
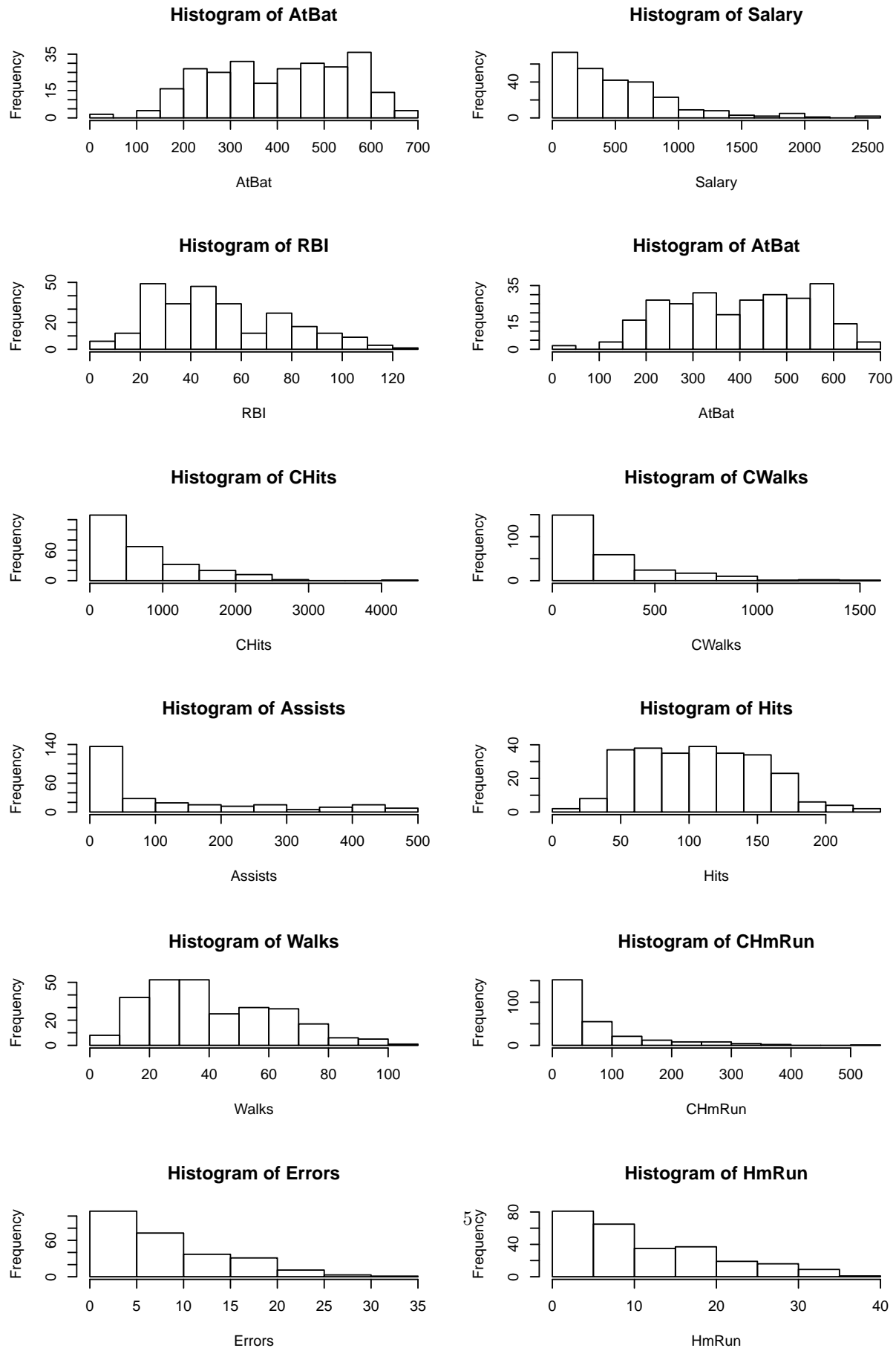


Table 2: VIF of Regression Model without Interaction terms

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

```
1 library(ISLR)
2 library(MASS)
3 library(corrplot)
4 library(car)
5 library(KernSmooth)
6 library(leaps)
7 library(xtable)
8 library(foreach)
9 library(randomForest)
10 library(glmnet)
11 library(tree)
12 sum(is.na(Hitters))
13
14 Hitters2<-na.omit(Hitters)
15
16 head(Hitters2)
17 names(Hitters2)
18
19 attach(Hitters2)
20
21 hist(AtBat)
22 hist(Salary)
23 hist(RBI)
24 hist(AtBat)
25 hist(CHits)
26 hist(CWalks)
27 hist(Assists)
28 hist(Hits)
29 hist(Walks)
30 hist(CHmRun)
31 hist(Errors)
32 hist(HmRun)
33 hist(Years)
34 hist(CRuns)
35 hist(Runs)
36 hist(CAtBat)
37 hist(CRBI)
38 hist(PutOuts)
39
40
41 #####
42 #### Checking for Log Transformation #####
43 #####
44
45 bcl <- boxcox(Salary~., data = Hitters2)
46 bcl$x[bcl$y==max(bcl$y)]
47
48
49 lSalary <- log(Salary)
50
51 hist(lSalary, breaks = 20)
52 hist(Salary)
53
54 #####
55 ####LOWESS TO FIND FUNCTIONAL FORM OF VARIABLES#####
56 #####
57
58 plot(RBI,lSalary)
59 lines(lowess(RBI,lSalary), col="blue")
60
61 plot(AtBat,lSalary)
62 lines(lowess(AtBat,lSalary), col="blue")
63
```

```

64 plot(CWalks, lSalary)
65 lines(lowess(CWalks, lSalary), col="blue")
66
67 plot(log(CWalks), lSalary)
68 lines(lowess(log(CWalks), lSalary), col="blue")
69
70 plot(CHits, lSalary)
71 lines(lowess(CHits, lSalary), col="blue")
72
73 plot(log(CHits), lSalary)
74 lines(lowess(log(CHits), lSalary), col="blue")
75
76 plot(Assists, lSalary)
77 lines(lowess(Assists, lSalary), col="blue")
78
79 plot(Hits, lSalary)
80 lines(lowess(Hits, lSalary), col="blue")
81
82 plot(Walks, lSalary)
83 lines(lowess(Walks, lSalary), col="blue")
84
85 plot(CHmRun, lSalary)
86 lines(lowess(CHmRun, lSalary), col="blue")
87
88 plot(log(CHmRun), lSalary)
89 lines(lowess(log(CHmRun), lSalary), col="blue")
90
91 plot(Years, lSalary)
92 lines(lowess(Years, lSalary), col="blue")
93
94 plot(log(Years), lSalary)
95 lines(lowess(log(Years), lSalary), col="blue")
96
97 plot(CRuns, lSalary)
98 lines(lowess(CRuns, lSalary), col="blue")
99
100 plot(log(CRuns), lSalary)
101 lines(lowess(log(CRuns), lSalary), col="blue")
102
103 plot(Runs, lSalary)
104 lines(lowess(Runs, lSalary), col="blue")
105
106 plot(log(CAtBat), lSalary)
107 lines(lowess(log(CAtBat), lSalary), col="blue")
108
109 plot(CRBI, lSalary)
110 lines(lowess(CRBI, lSalary), col="blue")
111
112 plot(log(CRBI), lSalary)
113 lines(lowess(log(CRBI), lSalary), col="blue")
114
115 plot(PutOuts, lSalary)
116 lines(lowess(PutOuts, lSalary), col="blue")
117
118 plot(HmRun, lSalary)
119 lines(lowess(HmRun, lSalary), col="blue")
120
121 plot(Errors, lSalary)
122 lines(lowess(Errors, lSalary), col="blue")
123
124
125 boxplot(lSalary ~ League)
126 boxplot(lSalary ~ Division)
127
128 #Log transform below variables

```

```

129 lCWalks<-log(CWalks)
130 lCHits <-log(CHits)
131 lCRuns <-log(CRuns)
132 lCAtBat <-log(CAtBat)
133 lCRBI <- log(CRBI)
134 lYears <-log(Years)
135 lCHmRun <-log(CHmRun)
136
137 #####
138 ###CORRELATION MATRIX#####
139 #####
140
141 Hitters2vars = data.frame(as.numeric(Salary), as.numeric(Hits), as.numeric(RBI), as.numeric
(Walks),
142
as.numeric(Years), as.numeric(CAtBat), as.numeric(CRuns), as.
143
numeric(CRBI),
as.numeric(CWalks), as.numeric(League), as.numeric(Division), as.
144
numeric(PutOuts),
as.numeric(AtBat), as.numeric(HmRun), as.numeric(Runs), as.
145
numeric(Assists),
as.numeric(CHits), as.numeric(CHmRun), as.numeric(Errors), as.
146
numeric(NewLeague))
146 Hit2var = cor(Hitters2vars)
147 corrplot(Hit2var)
148
149 #####
150 #### Univariate Analysis #####
151 #####
152
153 fit.Years<-lm(lSalary~Years, data = Hitters2) #2e-16
154 summary(fit.Years)
155 fit.CAtBat<-lm(lSalary~CAtBat, data = Hitters2) #2e-16
156 summary(fit.CAtBat)
157 fit.CRuns<-lm(lSalary~CRuns, data = Hitters2) #2e-16
158 summary(fit.CRuns)
159 fit.CRBI<-lm(lSalary~CRBI, data = Hitters2) #2e-16
160 summary(fit.CRBI)
161 fit.CWalks<-lm(lSalary~CWalks, data = Hitters2) #2e-16
162 summary(fit.CWalks)
163 fit.CHits<-lm(lSalary~CHits, data = Hitters2) #2e-16
164 summary(fit.CHits)
165 fit.CHmRun<-lm(lSalary~CHmRun, data = Hitters2) #2e-16
166 summary(fit.CHmRun)
167 # Keep CHits, remove other variables #
168
169 fit.AtBat<-lm(lSalary~AtBat, data = Hitters2)
170 summary(fit.AtBat)
171 fit.Hits<-lm(lSalary~Hits, data = Hitters2)
172 summary(fit.Hits)
173 # Keep AtBat, remove Hits #
174
175 Hitters2.b.vars = data.frame(as.numeric(Salary), as.numeric(RBI), as.numeric(Walks),
176
as.numeric(League), as.numeric(Division), as.numeric(PutOuts),
177
as.numeric(AtBat), as.numeric(HmRun), as.numeric(Runs), as.
178
numeric(Assists),
as.numeric(Errors), as.numeric(NewLeague))
179 Hit2var.b = cor(Hitters2.b.vars)
180 corrplot(Hit2var.b)
181
182
183 #####
184 #### Model Building no/Interaction#####
185 #####
186
187

```

```

188 fit.cor<-lm(lSalary ~ AtBat+HmRun+Runs+RBI+Walks+1CHits+
189             League+Division+PutOuts+Assists+Errors+NewLeague, data = Hitters2)
190 summary(fit.cor)
191
192 stepAIC(fit.cor, direction="both") ##AIC = -285
193
194
195 fit.cor.1<-lm(lSalary ~ AtBat + Runs + RBI + Walks + 1CHits +
196             League + Division + PutOuts + Assists + Errors, data = Hitters2)
197 summary(fit.cor.1)
198
199
200 #Remove Walks
201
202 fit.cor.2<-lm(lSalary ~ AtBat + Runs + RBI + 1CHits +
203             League + Division + PutOuts + Assists + Errors, data = Hitters2)
204 summary(fit.cor.2)
205
206 #Remove Assists
207
208 fit.cor.3<-lm(lSalary ~ AtBat + Runs + RBI + 1CHits +
209             League + Division + PutOuts + Errors, data = Hitters2)
210 summary(fit.cor.3)
211
212 #Remove Errors
213
214 fit.cor.4<-lm(lSalary ~ AtBat + Runs + RBI + 1CHits +
215             League + Division + PutOuts, data = Hitters2)
216 summary(fit.cor.4)
217
218 #Remove RBI
219
220 fit.cor.5<-lm(lSalary ~ AtBat + Runs + 1CHits +
221             League + Division + PutOuts, data = Hitters2)
222 summary(fit.cor.5)
223
224 #Remove AtBat
225
226 fit.cor.6<-lm(lSalary ~ Runs + 1CHits +
227             League + Division + PutOuts, data = Hitters2)
228 summary(fit.cor.6)
229
230 #Remove League
231
232 fit.cor.7<-lm(lSalary ~ Runs + 1CHits +
233             Division + PutOuts, data = Hitters2)
234 summary(fit.cor.7)
235
236 vif(fit.cor.7)
237
238 #####
239 ### Model Building w/interactions ###
240 #####
241
242 logtransforms = data.frame(AtBat, HmRun, RBI, Walks,
243                           1CHits, 1CRuns, League,
244                           Division, PutOuts, Assists, Errors, NewLeague)
245
246
247 fit.interaction<-lm(lSalary ~.*, data = logtransforms)
248 summary(fit.interaction)
249
250 stepAIC(fit.interaction, direction="both") ##AIC = -977.24
251
252 fit.with.interactions<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +

```

```

253 ICRuns + League + Division + PutOuts + Assists + Errors +
254 NewLeague + AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:
    League +
255 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
256 Walks:ICRuns + Walks:Division + 1CHits:ICRuns + 1CHits:League +
257 1CHits:Division + 1CHits:Assists + 1CHits:Errors + 1CHits:
    NewLeague +
258 ICRuns:League + ICRuns:Division + ICRuns:NewLeague + League:
    Division +
259 League:NewLeague + Division:PutOuts + Division:Assists +
260 PutOuts:Assists + League:PutOuts, data = logtransforms)
261 summary(fit.with.interactions)
262
263 #Remove Walks:Division
264
265 fit.with.interactions1<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
266 ICRuns + League + Division + PutOuts + Assists + Errors +
267 NewLeague + AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:
    League +
268 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
269 Walks:ICRuns + 1CHits:ICRuns + 1CHits:League +
270 1CHits:Division + 1CHits:Assists + 1CHits:Errors + 1CHits:
    NewLeague +
271 ICRuns:League + ICRuns:Division + ICRuns:NewLeague + League:
    Division +
272 League:NewLeague + Division:PutOuts + Division:Assists +
273 PutOuts:Assists + League:PutOuts, data = logtransforms)
274 summary(fit.with.interactions1)
275
276 #Remove PutOut:Division
277
278 fit.with.interactions2<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
279 ICRuns + League + Division + PutOuts + Assists + Errors +
280 NewLeague + AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:
    League +
281 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
282 Walks:ICRuns + 1CHits:ICRuns + 1CHits:League +
283 1CHits:Division + 1CHits:Assists + 1CHits:Errors + 1CHits:
    NewLeague +
284 ICRuns:League + ICRuns:Division + ICRuns:NewLeague + League:
    Division +
285 League:NewLeague + Division:Assists +
286 PutOuts:Assists + League:PutOuts, data = logtransforms)
287 summary(fit.with.interactions2)
288
289
290 #Remove CHits:Division
291
292 fit.with.interactions3<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
293 ICRuns + League + Division + PutOuts + Assists + Errors +
294 NewLeague + AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:
    League +
295 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
296 Walks:ICRuns + 1CHits:ICRuns + 1CHits:League +
297 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
298 ICRuns:League + ICRuns:Division + ICRuns:NewLeague + League:
    Division +
299 League:NewLeague + Division:Assists +
300 PutOuts:Assists + League:PutOuts, data = logtransforms)
301 summary(fit.with.interactions3)
302
303 #Remove CRuns:Division
304
305 fit.with.interactions4<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
306 ICRuns + League + Division + PutOuts + Assists + Errors +

```

```

307 NewLeague + AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:
    League +
308 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
309 Walks:1CRuns + 1CHits:1CRuns + 1CHits:League +
310 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
311 1CRuns:League + 1CRuns:NewLeague + League:Division +
312 League:NewLeague + Division:Assists +
313 PutOuts:Assists + League:PutOuts, data = logtransforms)
314 summary(fit.with.interactions4)
315
316 #Remove NewLeague
317
318 fit.with.interactions5<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
319 1CRuns + League + Division + PutOuts + Assists + Errors +
320 AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:League +
321 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
322 Walks:1CRuns + 1CHits:1CRuns + 1CHits:League +
323 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
324 1CRuns:League + 1CRuns:NewLeague + League:Division +
325 League:NewLeague + Division:Assists +
326 PutOuts:Assists + League:PutOuts, data = logtransforms)
327 summary(fit.with.interactions5)
328
329 #Remove Division
330
331 fit.with.interactions6<-lm(lSalary ~ AtBat + HmRun + RBI + Walks + 1CHits +
332 1CRuns + League + PutOuts + Assists + Errors +
333 AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:League +
334 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
335 Walks:1CRuns + 1CHits:1CRuns + 1CHits:League +
336 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
337 1CRuns:League + 1CRuns:NewLeague + League:Division +
338 League:NewLeague + Division:Assists +
339 PutOuts:Assists + League:PutOuts, data = logtransforms)
340 summary(fit.with.interactions6)
341
342 #Remove CHits
343
344 fit.with.interactions7<-lm(lSalary ~ AtBat + HmRun + RBI + Walks +
345 1CRuns + League + PutOuts + Assists + Errors +
346 AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:League +
347 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
348 Walks:1CRuns + 1CHits:1CRuns + 1CHits:League +
349 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
350 1CRuns:League + 1CRuns:NewLeague + League:Division +
351 League:NewLeague + Division:Assists +
352 PutOuts:Assists + League:PutOuts, data = logtransforms)
353 summary(fit.with.interactions7)
354
355 #Remove League: Division
356
357 fit.with.interactions8<-lm(lSalary ~ AtBat + HmRun + RBI + Walks +
358 1CRuns + League + PutOuts + Assists + Errors +
359 AtBat:Walks + AtBat:1CHits + HmRun:1CHits + HmRun:League +
360 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:1CHits +
361 Walks:1CRuns + 1CHits:1CRuns + 1CHits:League +
362 1CHits:Assists + 1CHits:Errors + 1CHits:NewLeague +
363 1CRuns:League + 1CRuns:NewLeague +
364 League:NewLeague + Division:Assists +
365 PutOuts:Assists + League:PutOuts, data = logtransforms)
366 summary(fit.with.interactions8)
367
368 #Remove Walks
369 fit.with.interactions9<-lm(lSalary ~ AtBat + HmRun + RBI +
370 1CRuns + League + PutOuts + Assists + Errors +

```

```

371 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
372 HmRun:NewLeague + RBI:League + RBI:Errors + Walks:lCHits +
373 Walks:ICRuns + lCHits:ICRuns + lCHits:League +
374 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
375 lCRuns:League + lCRuns:NewLeague +
376 League:NewLeague + Division:Assists +
377 PutOuts:Assists + League:PutOuts, data = logtransforms)
378 summary(fit.with.interactions9)
379
380 #Remove Walks:Hits
381 fit.with.interactions10<-lm(lSalary ~ AtBat + HmRun + RBI +
382 lCRuns + League + PutOuts + Assists + Errors +
383 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
384 HmRun:NewLeague + RBI:League + RBI:Errors +
385 Walks:ICRuns + lCHits:ICRuns + lCHits:League +
386 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
387 lCRuns:League + lCRuns:NewLeague +
388 League:NewLeague + Division:Assists +
389 PutOuts:Assists + League:PutOuts, data = logtransforms)
390 summary(fit.with.interactions10)
391
392 #Remove CRuns:Walks
393 fit.with.interactions11<-lm(lSalary ~ AtBat + HmRun + RBI +
394 lCRuns + League + PutOuts + Assists + Errors +
395 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
396 HmRun:NewLeague + RBI:League + RBI:Errors +
397 lCHits:ICRuns + lCHits:League +
398 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
399 lCRuns:League + lCRuns:NewLeague +
400 League:NewLeague + Division:Assists +
401 PutOuts:Assists + League:PutOuts, data = logtransforms)
402 summary(fit.with.interactions11)
403
404 #Remove HmRun:NewLeague
405 fit.with.interactions12<-lm(lSalary ~ AtBat + HmRun + RBI +
406 lCRuns + League + PutOuts + Assists + Errors +
407 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
408 RBI:League + RBI:Errors +
409 lCHits:ICRuns + lCHits:League +
410 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
411 lCRuns:League + lCRuns:NewLeague +
412 League:NewLeague + Division:Assists +
413 PutOuts:Assists + League:PutOuts, data = logtransforms)
414 summary(fit.with.interactions12)
415
416 #Remove League
417 fit.with.interactions13<-lm(lSalary ~ AtBat + HmRun + RBI +
418 lCRuns + PutOuts + Assists + Errors +
419 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
420 RBI:League + RBI:Errors +
421 lCHits:ICRuns + lCHits:League +
422 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
423 lCRuns:League + lCRuns:NewLeague +
424 League:NewLeague + Division:Assists +
425 PutOuts:Assists + League:PutOuts, data = logtransforms)
426 summary(fit.with.interactions13)
427
428 #Remove League:NewLeague
429 fit.with.interactions14<-lm(lSalary ~ AtBat + HmRun + RBI +
430 lCRuns + PutOuts + Assists + Errors +
431 AtBat:Walks + AtBat:lCHits + HmRun:lCHits + HmRun:League +
432 RBI:League + RBI:Errors +
433 lCHits:ICRuns + lCHits:League +
434 lCHits:Assists + lCHits:Errors + lCHits:NewLeague +
435 lCRuns:League + lCRuns:NewLeague +

```

```

436             Division:Assists +
437             PutOuts:Assists + League:PutOuts, data = logtransforms)
438 summary(fit.with.interactions14)
439
440 vif(fit.with.interactions14)
441
442 #####
443 ##### MODEL DIAGNOSTICS no Interactions #####
444 #####
445
446 plot(predict(fit.cor.7), rstudent(fit.cor.7), ylab="Studentized Residuals", xlab="Predicted
447 ")
448 identify(predict(fit.cor.7), rstudent(fit.cor.7), labels=row.names(Hitters2)) # 'escape to
449 finish '
450 predict(fit.cor.7)[rstudent(fit.cor.7)==min(rstudent(fit.cor.7))])
451
452 sresid <- studres(fit.cor.7)
453 hist(sresid, freq=FALSE, xlab = "Residuals", main="Distribution of Studentized Residuals")
454 xfit<-seq(min(sresid),max(sresid),length=40)
455 yfit<-dnorm(xfit)
456 lines(xfit, yfit, col = "blue")
457
458 qqPlot(fit.cor.7, main="QQ Plot", ylab="Studentized Residuals")
459
460 cutoff <- 4/((nrow(set2)-length(fit.cor.7$coefficients)-2))
461 plot(fit.cor.7, which=4, cook.levels=cutoff) # influence Plot
462
463
464 influencePlot(fit.cor.7, id.method="identify",
465             main="Influence Plot", sub="Circle size is proportional to Cook's Distance")
466
467 varif = vif(fit.cor.7)
468 varif
469
470
471
472
473 #####
474 ##### MODEL DIAGNOSTICS w/Interactions #####
475 #####
476
477 plot(predict(fit.with.interactions14), rstudent(fit.with.interactions14), ylab="Studentized
478 Residuals", xlab="Predicted")
479 identify(predict(fit.with.interactions14), rstudent(fit.with.interactions14), labels=row.
480 names(Hitters2)) # 'escape to finish '
481 predict(fit.with.interactions14)[rstudent(fit.with.interactions14)==min(rstudent(fit.with.
482 interactions14))])
483
484 sresid <- studres(fit.with.interactions14)
485 hist(sresid, freq=FALSE, xlab = "Residuals", main="Distribution of Studentized Residuals")
486 xfit<-seq(min(sresid),max(sresid),length=40)
487 yfit<-dnorm(xfit)
488 lines(xfit, yfit, col = "blue")
489
490 qqPlot(fit.with.interactions14, main="QQ Plot", ylab="Studentized Residuals")
491
492 cutoff <- 4/((nrow(set2)-length(fit.with.interactions14$coefficients)-2))
493 plot(fit.cor.7, which=4, cook.levels=cutoff) # influence Plot
494
495

```

```

496 influencePlot(fit.with.interactions14, id.method="identify",
497               main="Influence Plot", sub="Circle size is propotional to Cook's Distance")
498 vif(fit.with.interactions14)
499 #####
500 ##### Lasso #####
501 #####
502
503 x=model.matrix(Salary~., Hitters2)[-1]
504 y=Salary
505
506 grid=10^seq(10,-2,length=100)
507 lasso.mod = glmnet(x, Salary, alpha=1, lambda=grid) # alpha=1 is L1 norm, lasso penalty
508 plot(lasso.mod)
509 lasso.coef = predict(lasso.mod,type="coefficients")
510 lasso.coef = predict(lasso.mod,type="coefficients", s=0) # s is penalty parameter lambda
511 summary(lm(Salary~., Hitters2)) # same coef as above lasso
512
513 set.seed(1)
514 train=sample(1:nrow(x), nrow(x)/2) # split data in half
515 test=(-train)
516 y.test=y[test]
517 lasso.mod=glmnet(x[train,], y[train], alpha=1, lambda=grid)
518 plot(lasso.mod)
519 set.seed(1)
520 # minimizes squared-error loss (prediction error)
521 # run lasso on training set
522
523 test=(-train)
524 y.test=y[test]
525 cv.out=cv.glmnet(x[train,], y[train], alpha=1)
526
527 plot(cv.out)
528 bestlam=cv.out$lambda.min #16.78016
529 lasso.pred=predict(lasso.mod, s=bestlam, newx=x[test,])
530 mean((lasso.pred-y.test)^2) # prediction error for best lambda
531 out=glmnet(x, y, alpha=1, lambda=grid) # run lasso on whole data set
532 lasso.coef=predict(out, type="coefficients", s=bestlam)
533 lasso.coef
534 lasso.coef[lasso.coef!=0]
535
536 mean((lasso.pred-y.test)^2) #100743.4
537
538 set.seed(1)
539 train = sample(1:nrow(Hitters2), nrow(Hitters2)/2)
540 tree.Hitters=tree(Salary~., Hitters2, subset=train)
541 summary(tree.Hitters)
542
543 plot(tree.Hitters)
544 text(tree.Hitters, pretty=0)
545
546 bag.Hitters=randomForest(Salary~., data=Hitters2, subset=train,
547                          mtry=19, importance =TRUE)
548 bag.Hitters
549
550 yhat.bag = predict(bag.Hitters, newdata=Hitters2[-train,])
551 plot(yhat.bag, Hitters.test)
552 abline(0,1)
553 mean((yhat.bag-Hitters.test)^2)
554
555
556 rf.Hitters=randomForest(Salary~., data=Hitters2, subset=train,
557                          mtry=7, importance =TRUE)
558 yhat.rf = predict(rf.Hitters, newdata=Hitters2[-train,])
559
560 importance(rf.Hitters)

```

```

561
562 varImpPlot (rf.Hitters)
563
564 #####
565 ##### Finding best MSE #####
566 #####
567
568
569 #create a new dataframe with hitters for the logsalary
570
571 Hitters3<-data.frame(lSalary, Hits, RBI, Walks,
572                      Years, CAtBat, CRuns, CRBI,
573                      CWalks, League, Division, PutOuts,
574                      AtBat, HmRun, Runs, Assists,
575                      CHits, CHmRun, Errors, NewLeague)
576
577 # Generate training and testing sets
578
579 set.seed(1)
580 train = sample(1:nrow(Hitters3), nrow(Hitters3)/2)
581 Hitters.train = Hitters3[train,]
582 Hitters.test = Hitters3[-train,]
583
584 # Perform regression
585 fit_lm = lm(lSalary ~ ., data=Hitters.train)
586 pred_lm = predict(fit_lm, Hitters.test) # predicted test set
587 lm_MSE = mean((pred_lm - Hitters.test$lSalary)^2) # MSE ~ 0.378902
588
589 # Bagging (bootstrap aggregation) a regression model fit
590 set.seed(1)
591 iterations = 1000; n = nrow(Hitters.train)
592 predictions = foreach(m=1:iterations, .combine=cbind) %do% {
593   # sample with replacement (bootstrap)
594   training_positions = sample(nrow(Hitters.train), size=n, replace=TRUE)
595   lm_fit = lm(lSalary ~ ., data=Hitters.train[training_positions,])
596   predict(lm_fit, newdata=Hitters.test)
597 }
598
599 pred_bag<-rowMeans(predictions)
600 bag_MSE = sum((Hitters.test$lSalary-pred_bag)^2)/n # MSE ~ 0.3672615
601 plot(pred_bag)
602 # Bagging regression without bootstrap
603 # randomly subset training data rather than bootstrap
604 set.seed(1)
605 bagging_lm = function(training, testing, length_divisor=4, iterations=1000)
606 {
607   predictions<-foreach(m=1:iterations, .combine=cbind) %do% {
608     training_positions = sample(nrow(training), size=floor((nrow(training)/length_divisor))
609     )
610     train_pos = 1:nrow(training) %in% training_positions
611     # FUNCTION NOT AUTOMATED: must name response in following 'lm' call
612     lm_fit = lm(lSalary ~ ., data=training[train_pos,])
613     predict(lm_fit, newdata=testing)
614   }
615   rowMeans(predictions)
616 }
617 bagreg_pred = bagging_lm(Hitters.train, Hitters.test)
618 bagreg_MSE = sum((Hitters.test$lSalary-bagreg_pred)^2)/n # MSE ~ 0.3568651
619 # Results
620 results = cbind(lm_MSE, bag_MSE, bagreg_MSE)
621 colnames(results) = c("Regression", "Bagging", "No Bootstrap")
622 results

```

Listing 1: Baseball Salary