UNIVERSITY OF NEW SOUTH WALES
SCHOOL OF MATHEMATICS AND STATISTICS

MATH3821 Statistical Modelling and Computing
Term Two 2019

Assignment Two

Given: 16 July 2019 **Due date: 4 August 2019**

INSTRUCTIONS: This assignment is to be done **collaboratively** by a group of **at most 3** students. You can find the other members of your group on Moodle. The same mark will be given for the report to each student within the group, unless I have good reasons to believe that somebody did not do anything.

1. You will need to produce and submit a report of your work in PDF format. This report will not contain more than 10 pages, excluding the Appendix that should contain your computing codes. The report is due 11:59 pm, Sunday 4th August. The first page of this PDF should be **this page**. Only one of the three students should submit the PDF file on Moodle, with the names of the other students in the group clearly indicated in the document. You will also bring on the day of your oral presentation, a printed copy of your report with your signatures and date on it (see below).

2. Each group will also be required to make a 9 minute presentation. The PDF slides are also due on 11:59 pm Sunday 4 August via Moodle. The presentation will take place either during your usual lecture times (Tuesday 6 August, 4 pm or Thursday 8, 11 am) or your enrolled tutorial times (Thursday 8 August, 12 pm, 1 pm or 2 pm). The precise day and time of your presentation (allocated randomly within your tutorial or within the lecture time) will be provided on Moodle. Each member of the group will be expected to make one part of the presentation (e.g., 3 minutes each). Since there are 31 groups, you will not be allowed to talk for more than 9 minutes (i.e., I will have to (politely) interrupt you). All students are expected to attend (and evaluate) the oral presentations of all other groups presenting in the same session as them - the lecture will be split into two sessions. Obviously, please don't be late.

I/We declare that this assessment item is my/our own work, except where acknowledged, and has not been submitted for academic credit elsewhere. I/We acknowledge that the assessor of this item may, for the purpose of assessing this item reproduce this assessment item and provide a copy to another member of the University; and/or communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the assessment item on its database for the purpose of future plagiarism checking). I/We certify that I/We have read and understood the University Rules in respect of Student Academic Misconduct.

Name Student No Signature Date

Derek Sun (z5160085), Hugo Perron (z5114824), Louis Ye (z5120556), Yujie He (z5161677)

MATH3821 Assignment 2

Derek Sun (z5160085), Hugo Perron (z5114824), Louis Ye (z5120556), Yujie He (z5161677)

Introduction

Our group has analyzed the 1986-1987 Baseball Data-set to answer the question:

- **Whether it is possible to determine whether a player is overpaid/underpaid through regression**

Whether it is possible to offer insight towards understanding:

- **How different measurements of performance each affect a player's salary**
- **Were baseball players paid based on their baseball players paid based on their performance?**

Salary and 1986 performance data from North American Major League Baseball Players, split by hitters and pitchers, as well as team data (containing the Average Team Salaries) .

Data Collection

Before any exploration into data, following steps have to be taken

- Retrieve data from the Statistical Computing website in the forms of csvs
- Import hitters, pitchers and teams csv respectively into R
- Apply changes (initial data has been revised)
- Remove NA entries from data-set
- Convert response 'Salary' to be same unit(thousands of dollars) across data sets

Hitters Data with 1987 Salary as the Response:

- Salary
- hitter's name
- #times at bat in 1986
- #hits in 1986
- #home runs in 1986
- #runs in 1986
- #runs batted in in 1986
- #walks in 1986
- #years in the major leagues
- #times at bat during his career
- #hits during his career
- #home runs during his career

- #runs during his career
- #runs batted in during his career
- #walks during his career
- player's league at the end of 1986
- player's division at the end of 1986
- player's team at the end of 1986
- player's position(s) in 1986
- #put outs in 1986
- #assists in 1986
- #errors in 1986
- player's league at the beginning of 1987
- player's team at the beginning of 1987

Pitcher Data, with 1987 Salary as the Response:

- Salary
- pitcher's name
- player's team at the end of in 1986
- player's league at the end of 1986
- #wins in 1986
- #losses in 1986
- earned run average in 1986
- #games in 1986
- #innings pitched in 1986
- #saves in 1986
- #years in the major leagues
- #wins during his career
- #losses during his career
- earned run average during his career
- #games during his career
- #innings pitched during his career
- #saves during his career
- player's league at the beginning of 1987
- player's team at the beginning of 1987

Team Data, with Average 1987 Team Salary as Response:

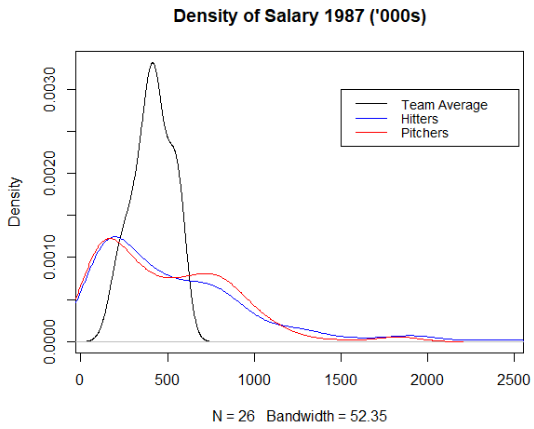
- Average Team Salary
- league
- division
- position in final league standings in 1986
- team
- #wins in 1986
- #losses in 1986
- attendance for home games in 1986
- attendance for away games in 1986

(all relevant R codes are attached in the appendix)

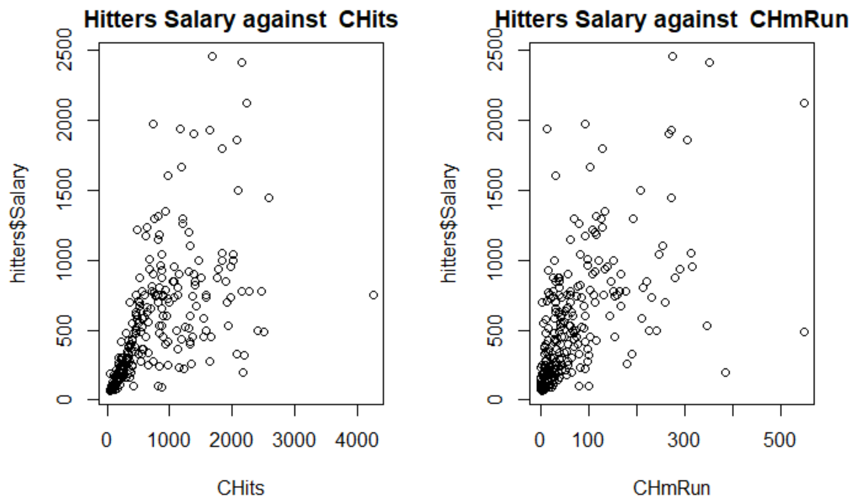
Exploratory Analysis

Since we have many predictors, we explored ways to simplify models by removing predictors where appropriate, through examining the Salary Density, looking at the relationship between variables, and examining multicollinearity.

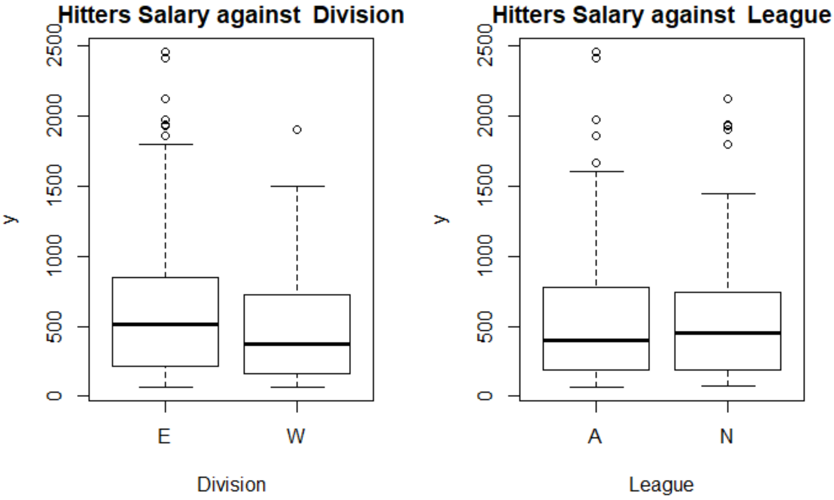
Observing the Salary density, we see that Pitchers and Hitters generally follow the same distribution and amount, while team salary seems to be symmetric.



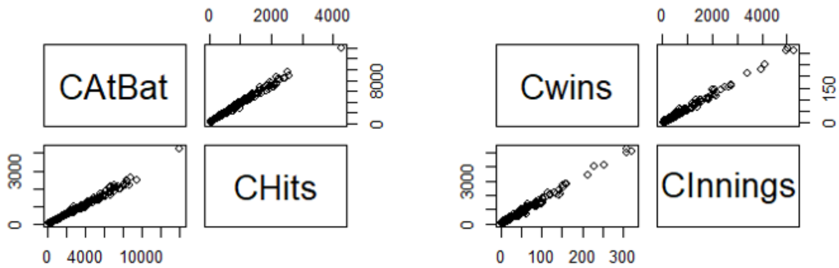
We also graph scatter-plots of relationships between the variables. Below is an extract from our plots, and we generally find that there is a clear positive relationship with many of the quantitative predictors. We also note that as variance gets larger, penalized regression might be better.



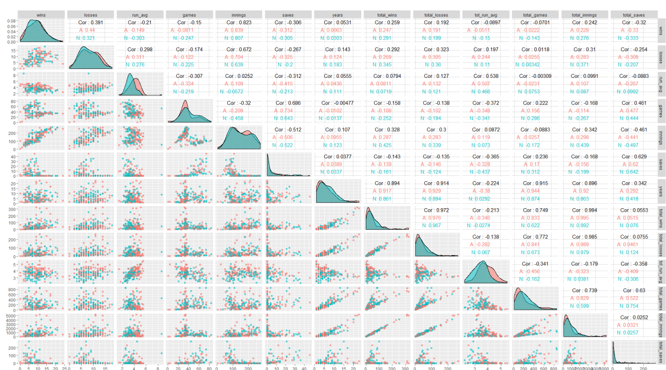
The qualitative variables of league and division also seem to significantly affect a player's salary:



We also examine multicollinearity by pair-plotting, and notice that some measures are highly correlated and positive proportional, such as **# of bats in career** and **# of hits in career** for Hitters, and **# of wins in career** and **# of innings in career** for Pitters.



We also note that Pitchers also displays the issue of heavily left-skewed data (in regards to "Saves"). We also note that there are some examples of outliers in the data, most obvious in the "run_avg". This stat also displays a different distribution when comparing the National and American leagues.



Model Choice and Fitting

Generalized Linear Model

We decided to begin model testing in R with all predictors available, and to cut predictors down in the process. In some of the later models we also added new predictors with interactions, such as $\frac{\#\text{Hits}}{\#\text{Number of Times at Bat}}$, and $\frac{\#\text{Wins}}{\#\text{Games}}$. Starting with all raw predictors and splitting the model into a 75% training set and 25% test set, and run an anova test to begin cutting down on predictors. We obtain the output for hitters and pitchers:

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: Salary
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        196   45077460
## AtBat      1 11395404       195   33682056 < 2.2e-16 ***
## Hits       1  2054491       194   31627564 4.174e-07 ***
## HmRun      1  1606825       193   30020739 7.618e-06 ***
## Runs       1    49213       192   29971526 0.4334682
## RBI        1   896389       191   29075137 0.0008292 ***
## Walks      1  2837552       190   26237585 2.721e-09 ***
## Years      1  5673343       189   20564242 < 2.2e-16 ***
## CAtBat     1  1809245       188   18754997 2.042e-06 ***
## CHits      1   235875       187   18519122 0.0863829 .
## CHmRun     1  2176300       186   16342822 1.902e-07 ***
## CRuns      1    51423       185   16291400 0.4233285
## CRBI       1    23431       184   16267968 0.5888734
## CWalks     1   196862       183   16071106 0.1172135
## League     1     9681       182   16061425 0.7282897
## Division   1   735661       181   15325764 0.0024587 **
## PutOuts    1  1031849       180   14293916 0.0003351 ***
## Assists    1    66330       179   14227586 0.3631717
## Errors     1    29419       178   14198167 0.5447827
## NewLeague  1       46       177   14198121 0.9808670
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Analysis of Deviance Table

Model: gaussian, link: identity

Response: Salary

Terms added sequentially (first to last)


            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                          175   24213875
League       1   583827       174   23630048  0.008399 **
Wins         1  1474161       173   22155887 2.814e-05 ***
Losses       1   210606       172   21945281  0.113428
RunAverage   1      384       171   21944897  0.946112
Games        1      377       170   21944520  0.946608
Innings      1   101463       169   21843057  0.271886
Saves        1  1445558       168   20397499 3.366e-05 ***
Cwins        1  4601136       167   15796363 1.373e-13 ***
Closses      1    28725       166   15767639  0.558809
CRunAverage  1   903832       165   14863807  0.001041 **
CGames       1   954030       164   13909776  0.000754 ***
CInnings     1   328290       163   13581486  0.048114 *
CSaves       1     1213       162   13580273  0.904380
NewLeague    1    48625       161   13531648  0.446883
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

At a 0.05 significance, we construct the glm model with the following predictors (giving $MSE_{hitters} = 100029.9$ and $MSE_{pitchers} = 60480.22$):

- AtBat
- Hits
- HmRun
- RBI
- Walks
- Years
- CAtBat
- CHmRun
- Division
- PutOuts

```
                Estimate  Std. Error
(Intercept) -29.60969135 94.68916289
AtBat        -1.37766195  0.64580660
Hits          7.17582224  2.17271067
HmRun         1.91919939  5.46732080
RBI          -1.88604373  2.58857571
Walks         3.69070555  1.32463172
Years        -4.76808107 13.24329833
CAtBat       -0.08776789  0.13762401
CHits         0.53552848  0.43553995
CHmRun        1.07086832  0.53723217
DivisionW   -59.31541854 43.43269813
PutOuts       0.35202796  0.08465343
```

```
             Estimate
(Intercept)  378.2980
LeagueN       66.9828
Wins          16.8755
Saves          0.8250
Cwins          3.0835
CRunAverage  -74.1340
CGames         0.6205
CInnings      -0.1061
```

Penalized Regression

From our earlier observation of inconsistent variance of the predictors against Salary, we use glmnet to fit a Penalized Regression Fit with Lasso. This gives $Hitter_{lasso}$ a λ of 13.70396, and $MSE_{lasso} = 80332.44$, and $Pitcher_{lasso}$ with a λ of 10.54306 and $MSE_{lasso} = 61908.06$.

```
12 x 1 sparse Matrix of class "dgCMatrix"
                     1
(Intercept) -84.4664127
(Intercept)   .
AtBat         .
Hits          2.5275481
HmRun         .
RBI           .
Walks         2.6855683
Years         .
CAtBat        0.0526253
CHmRun        1.1353218
DivisionW   -90.4915883
PutOuts       0.1909965
```

```
9 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept) 440.4432786
(Intercept)    .
LeagueN       44.1606079
Wins          14.7045672
Saves          .
CWins          1.2307638
CRunAverage  -79.0459742
CGames         0.5988865
CInnings       .
```

AIC Forward Selection of Model

We also used regsubsets (using AIC) to determine the best fitting model from the original set of 20 predictors. By extracting the model with minimum BIC and CP, as well as the model with maximum adjusted R^2, we found that the Hitters model with 6 predictors would be the best one in general (from this method), which has a MSE of 89638.63. For Pitchers, it is a model with 5 predictors, and a MSE of 58647.45.

Hitters:

```
##  (Intercept)        AtBat         Hits        Walks         CRBI
##  -62.7082652   -1.4012164    7.0983883    3.7541639    0.7240970
##    DivisionW      PutOuts
## -138.0145332    0.2813861
```

Pitchers:

```
(Intercept)      Losses     Innings       CWins CRunAverage      CSaves
 559.795204    5.276936    1.474735    2.108245 -131.575452    2.236095
```

Final Model

Choosing the model with smallest MSE, we use the Penalized Regressions for Hitter and AIC Selected Model for Pitcher:

$$\begin{aligned} Salary_{Hitter} = &-84 + 2.527Hits + 2.685Walks + 0.053CAtBat \\ &+ 1.135CHmRun - 90.491DivisionW + 0.1909965PutOuts \end{aligned}$$

$$\begin{aligned} Salary_{Pitcher} = &\ 559.795 + 5.28Losses + 1.474Innings + 2.108Cwins \\ &- 131.57CRunAverage + 2.236CGames \end{aligned}$$

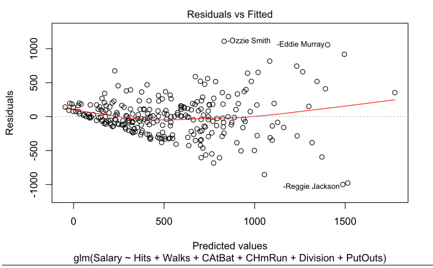
We also tested a model with predictor interactions $Hits/AtBat + HmRun/Runs + RBI/Walks + Years + CHmRun/CRuns + ICHits/CAtBat + PutOuts + Assists$ with the same approaches as above, but resulted in larger MSE's (code in Appendices). The best model there

($MSE = 115017.5$) was:

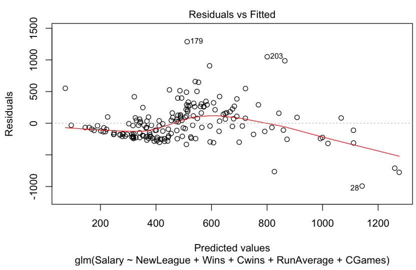
```
(Intercept)       -1931.8632282
(Intercept)              .
I(Hits/AtBat)       658.5426920
I(HmRun/Runs)      -527.9252006
I(RBI/Walks)        -75.9293042
Years                33.3146642
I(CHmRun/CRuns)    1237.5154155
I(CHits/CAtBat)    7187.5885363
PutOuts               0.3455984
Assists               0.3993428
```

Model Diagnostics

Residual Plots



The residuals plot for hitters is quite flat which apart from a few large outliers. This can happen in a sport like baseball where star players are extremely well paid.



We see a bit more shape in this plot for the pitchers. This illustrates that the residuals are not so random and the linear and the relationship is not entirely linear. This can also be attributed to the fact that there are some players earning a big salary and this is affecting the plot.

Bootstrapped confidence intervals

The std.errors for estimators presented in the summary only hold in the asymptotic case. An alternate way to generate std.errors for estimators (and any statistic in fact) is to use the bootstrap. This effectively generates new sets by sampling with replacement from the original. Assuming the original dataset is representative of the true distribution this should allow us to generate accurate confidence intervals. An attempt was made to implement bootstrap ourselves, however this failed on models using automatic variable selection. Ultimately a package implementing bootstrap for LASSO was used, and the corresponding confidence intervals were plotted. The broad range indicates relatively high uncertainty, however it could also be due to the LASSO regularizing certain Beta's to 0.

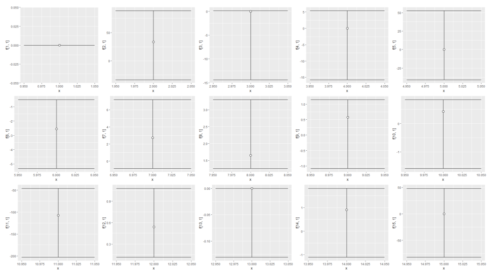
Toy example of bootstrap

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = pitchers2, statistic = coefficients, R = 1000, formula = salary ~
    years + tot_run_avg + total_innings)


Bootstrap Statistics :
         original       bias    std. error
t1*  769.72751939 -3.67533097 172.1264214
t2*   32.41199660 -0.83584371  14.3781194
t3* -138.72596571  0.03893093  38.3719354
t4*    0.03856914  0.01250410   0.0953715
```

Confint for pitchers



Model Assessment

Prediction Assessment

The main way that we can assess the usefulness of this model is to compare each player's actual salary with their predicted salary using the model. With this data, we can calculate the difference in the two salaries and see if the model is over or undervaluing the players. Moreover, we can also calculate the deviation from the actual salary as a percentage to check the accuracy of the model. Another way to test the adequacy of this model at predicting salary would be to create a fictional player, average in every metric, and compare its predicted salary to that of the average salary in the league.

Limitations

Limitations identified in the making of this model:

- Missing data are removed instead of imputed, the number of such entries are low and we assumed would not have significant impact on the result
- The data provided from 1986 does not provide statistics on the players to the same level of depth that data available today does

- We are using the assumption that a player's salary is purely based on their performance on the field, however in reality this may not be the only factor, albeit the most important.
- In the 80's the rules around salaries weren't as regulated as they are now. This meant that teams in bigger markets such as New York and Los Angeles were able to pay their players a higher average salary due to the fact that the franchise made more money than other franchises.
- Rookie contracts and minimum salaries are both examples of where salary may not be a true representative of a player's value. A minimum salary might overstate a player's worth, whereas a rookie contract could understate a player's worth.
- In addition, baseball is a team game and this can result in some of non-player-specific statistics to be distorted. For example, a bad player may still have a high win-probability if the rest of his team is good. This idea of isolating a player's impact from his team essentially created the field of sabermetrics, which was still in its infancy at the time of this dataset.

Conclusion

Out of the models used, penalized regression with Lasso seems to be the best choice with lowest MSE. We managed to access the extended list of fields and simplify the model, leaving 6 predictors for hitters and 5 predictors for pitchers.

For hitters, number of hits, walks, bats in career, home runs in careers, division and putouts are used in predicting salaries while for pitchers, the league, number of wins, earned run average and number of games.

Some further suggestions for future investigation

- Regression on teams data and combine that with hitters and pitchers
- Log transformation of some variables like salary as there is skewed pattern
- More data would be extremely useful, especially salary across different years which allows tracking pattern over time
- Can perhaps use summary salary statistics for MBL players to create a relativity table across years to predict salary at present (attached in APPENDIX)

Appendix

Hoffman, M. (2014) Analysis of Salary for Major League Baseball Players https://library.ndsu.edu/ir/bitstrear

Wiseman, F., Chatterjee, S. (1997) Major League Baseball Player Salaries: Bringing Realism into Introductory Statistics Courses, *The American Statistician*, 51(4): 350-352
Lackritz, J.R. (1990) Salary Evaluation for Professional Baseball Players, *The American Statistician*, 44(1): 4-8

0.0.1 Code snippets

```
library(MASS)
library(ISLR)
library(ggplot2)
library(GGally)
library(boot)

Hitters=na.omit(hitters)
with(hitters,sum(is.na(Salary)))
Pitchers=na.omit(pitchers)   \\remove NA's from data
Teams=na.omit(teams)

head(Hitters)
head(Pitchers)
head(Teams) \\view head of data
\\Code not repeated for different datasets.

\\Pairwise plots + Density + Correlation coefficient
ggpairs(hitters2[, c(1:13, 18:21)], aes(colour = hitters2$league, alpha = 0.4))

\\ Density plots
require(graphics)
x_min = min(min(hitters$Salary),min(teams$AverageSalary/1000),min(hitters$Salar
x_max = max(max(hitters$Salary),max(teams$AverageSalary/1000),max(hitters$Salar
plot(density.default(teams$AverageSalary/1000), main = "Density of Salary 1987
lines(density.default(hitters$Salary), col = "blue")
lines(density.default(pitchers$Salary), col = "red")
legend(1500, 0.003, legend=c("Team Average", "Hitters", "Pitchers"),
       col=c("black", "blue", "red"), lty=1:1, cex=0.9)

\\Scatterplot
par(mfrow = c(1,2))
plot(hitters[,9], hitters$Salary, xlab = colnames(hitters)[9], main = paste("H
plot(pitchers[,15], pitchers$Salary, xlab = colnames(pitchers)[15], main = past


\\Scatterplot 2
par(mfrow = c(1,2))
plot(hitters[,14], hitters$Salary, xlab = colnames(hitters)[14], main = paste("
```

```
plot(pitchers[,3], pitchers$Salary, xlab = colnames(pitchers)[3], main = paste(


par(mfrow = c(1,2))
pairs(hitters[,c("CAtBat", "CHits")])
pairs(pitchers[,c("Cwins", "CInnings")])


\\Interaction Plots with multiple models
hitters.glm<-glm(Salary~I(Hits/AtBat) + I(HmRun/Runs)+I(RBI/Walks)+Years+I(CHmI
                 ,data=hitters)
par(mfrow=c(2,2))
plot(hitters.glm)
set.seed(3)
train <- sample(nrow(hitters), floor(0.75 * nrow(hitters)))
test <- (1:nrow(hitters))[-train]
\\AIC
regfit <- regsubsets(Salary ~ I(Hits/AtBat) + I(HmRun/Runs)+I(RBI/Walks)+Years+
                    , data = hitters[train,], nvmax =8, method = "forward")
regfit_summary <- summary(regfit)
regfit_summary
coef(hitters_lasso, s=best_cvlambda)
which.min(regfit_summary$bic)
which.min(regfit_summary$cp)
which.max(regfit_summary$adjr2)
test_mat = model.matrix (Salary ~ I(Hits/AtBat) + I(HmRun/Runs)+I(RBI/Walks)+Y
                         , data = hitters[test,])
val_errors = rep(NA,19)

# Iterates over each size i
for(i in 1:8){

    # Extract the vector of predictors in the best fit model on i predictors
    coefi = coef(regfit, id = i)

    # Make predictions using matrix multiplication of the test matirx and the c
    pred = test_mat[,names(coefi)]%*%coefi

    # Calculate the MSE
    val_errors[i] = mean((hitters[test,]$Salary-pred)^2)
}



    \\ Bootstrap
    glmboot <- function(formula, data, indices) {
        d <- data[indices,] # allows boot to select sample
        fit <- glm(formula, data=d)
        return(fit$coefficients)
    }
```

```
    boot(glmboot, formula=Salary~I(Hits/AtBat)+I(HmRun/Runs)+I(RBI/Walks)+Years
    include(HDCI)
    c <- bootLasso(pitchers_X, as.matrix(pitchers2$salary), type.boot="residual

    \\plotting code excluded due to extreme length (grid.arrange cannot take li

    \\Pitchers
    pitchers.glm<-glm(Salary~League+Wins+Saves+Cwins+CRunAverage+CGames+CInning
    summary(pitchers.glm)
    par(mfrow=c(2,2))
    plot(pitchers.glm)
    set.seed(2)
    train <- sample(nrow(pitchers), floor(0.75 * nrow(pitchers)))
    test <- (1:nrow(pitchers))[-train]

    \\Pitcher Lasso
    preds <- predict(pitchers.glm, newdata = pitchers[test,])
    MSE <- mean((pitchers[test, "Salary"] - preds)^2)
    pitchers.lasso <- glmnet(model.matrix(Salary ~ League+Wins+Saves+Cwins+CRu
                                         ), as.matrix(pitchers[,"Salary"]), alpha
    pitchers_X <- model.matrix(Salary ~ League+Wins+Saves+Cwins+CRunAverage+CGa
    pitchers_lassocv <- cv.glmnet(pitchers_X, as.matrix(pitchers[,"Salary"]), a
    plot(pitchers_lassocv)

    best_cvlambda <- pitchers_lassocv$lambda.min
    best_cvlambda
    preds.lasso <- predict(pitchers.lasso, s = best_cvlambda,  pitchers_X[test,
    MSE.lasso <- mean((pitchers[test, "Salary"] - preds.lasso)^2)
    MSE.lasso

    coef(pitchers.lasso, s=best_cvlambda)

    \\Pitcher AIC
    regfit <- regsubsets(Salary ~ League+Wins+Losses+RunAverage+Games+Innings+S
                       , data = pitchers[train,], nvmax =14, method = "forward")
    regfit_summary <- summary(regfit)
    regfit_summary
    which.min(regfit_summary$bic)
    which.min(regfit_summary$cp)
    which.max(regfit_summary$adjr2)

    # Finding MSE
    test_mat = model.matrix (Salary ~     League+Wins+Losses+RunAverage+Games+I
                               , data = pitchers[test,])
    val_errors = rep(NA,14)

    # Iterates over each size i
    for(i in 1:14){

    # Extract the vector of predictors in the best fit model on i predictors
```

```
    coefi = coef(regfit , id = i)

    # Make predictions using matrix multiplication of the test matirx and the c
    pred = test_mat[,names(coefi)]%*%coefi

    # Calculate the MSE
    val_errors[i] = mean((pitchers[test,]$Salary-pred)^2)
}
min <- which.min(val_errors)

    
```