# Stat151A HW5

*Jiying Zou*

*April 8, 2017*

# Introduction

In this project we will try to model and predict baseball players' salaries from a variety of related factors. The dataset we will be using contains information on many baseball players and their performance during 2012 as well as throughout their career. Some of these factors include performance stats such as number of hits, home runs, number of times at-bat etc., and others include number of years in the major leagues and player position. We will explore anomalies in the data and some appropriate transformations, as well as consider interactions between variables. In the end, we will use Mallows' Cp and BIC to produce a predictive model for our data.

```
library(corrplot)
library(car)
library(leaps)
#Load in dataset titled 'baseball'
load("~/Documents/stat151/HW/hw5/baseball2012.rda")
```

# Data Exploration/Feature Engineering

First, we want to prepare the data for better analysis by feature engineering on some of the variables. We will create new variables from existing variables that mirror the variables Fox creates in his analysis of a similar dataset (from section 22.1.2 of Fox's book *Applied Regression Analysis and Generalized Linear Models*).

```
#Feature engineer new variables

#2012 and career batting averages (hits/at-bats)
baseball$AVG <- baseball$H / baseball$AB
baseball$'Career AVG' <- baseball$CH / baseball$CAB

#2012 and career on base percentages (100 x [hits + walks]/[at-bats + walks])
baseball$OBP <- 100 * ((baseball$H + baseball$BB) / (baseball$AB + baseball$BB))
baseball$'Career OBP' <- 100 * ((baseball$CH + baseball$CBB) / (baseball$CAB + baseba
ll$CBB))

#Per-year statistics
baseball$'AB/year' <- baseball$CAB / baseball$years #At-bats per year
baseball$'H/year' <- baseball$CH / baseball$years #Hits per year
baseball$'HR/year' <- baseball$CHR / baseball$years #Home runs per year
baseball$'R/year' <- baseball$CR / baseball$years #Runs scored per year
baseball$'RBI/year' <- baseball$CRBI / baseball$years #Runs batted in per year

#Three player position dummy variables (middle infielders, catcher, center field)
baseball$MI <- (baseball$POS %in% c("2B", "SS")) - 0 #second base or shortstop
baseball$C <- (baseball$POS == "C") - 0
baseball$CF <-(baseball$POS == "CF") - 0

#Two dummy variables for years of major league experience (1 for 3-5 yrs, 1 for 6+ ye
ars)
baseball$'Arbitration eligible' <- (baseball$years %in% c(3:5)) - 0
baseball$'Free-agency eligible' <- (baseball$years >= 6) - 0
```

Since we are trying to predict player salary, we wouldn't want to consider cases without information about salary, so we remove these rows.

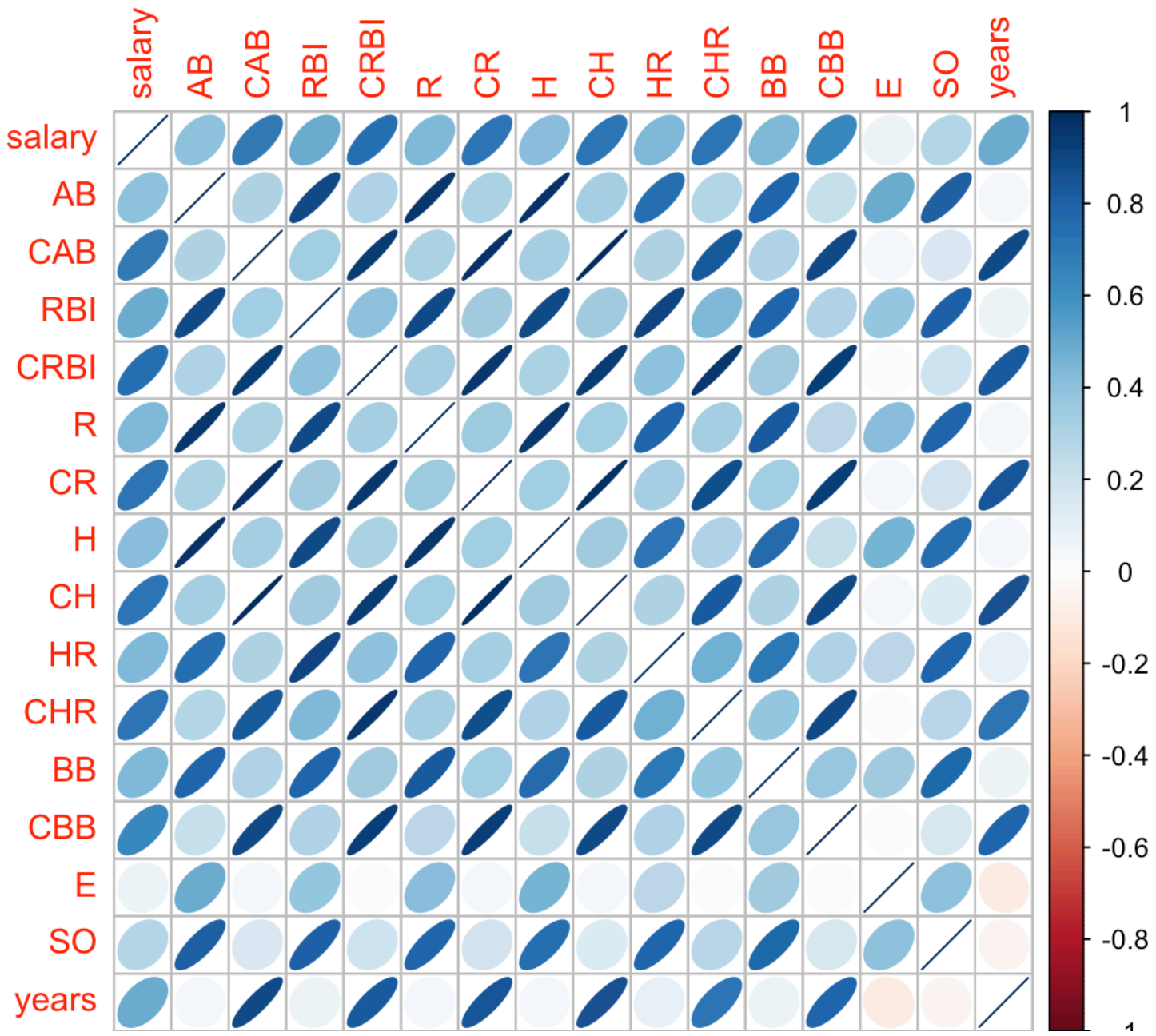
```
#Remove rows without salary data
baseball <- baseball[is.na(baseball$salary) == F,]
```

## Correlation Plot

First, to assess patterns of correlation, we can look at the correlation plot. Darker, skinnier blue ellipses indicate stronger, linear relationships. From this plot we can tell that:

- The variable `years` is strongly correlated with the career-related variables, and less correlated with 2012-specific variables
- Some variables, like the number of strikeouts in 2012 ( `SO` ), are positively correlated with 2012-related variables but weakly associated with overall career-related variables.
- The only slight negative association in this plot occurs between `years` and `E` or `SO` , which may be reasonable because as a player becomes more experienced, we'd expect less errors and strikeouts. However, this association is so slight that the scatterplot matrix is preferrable for examination.

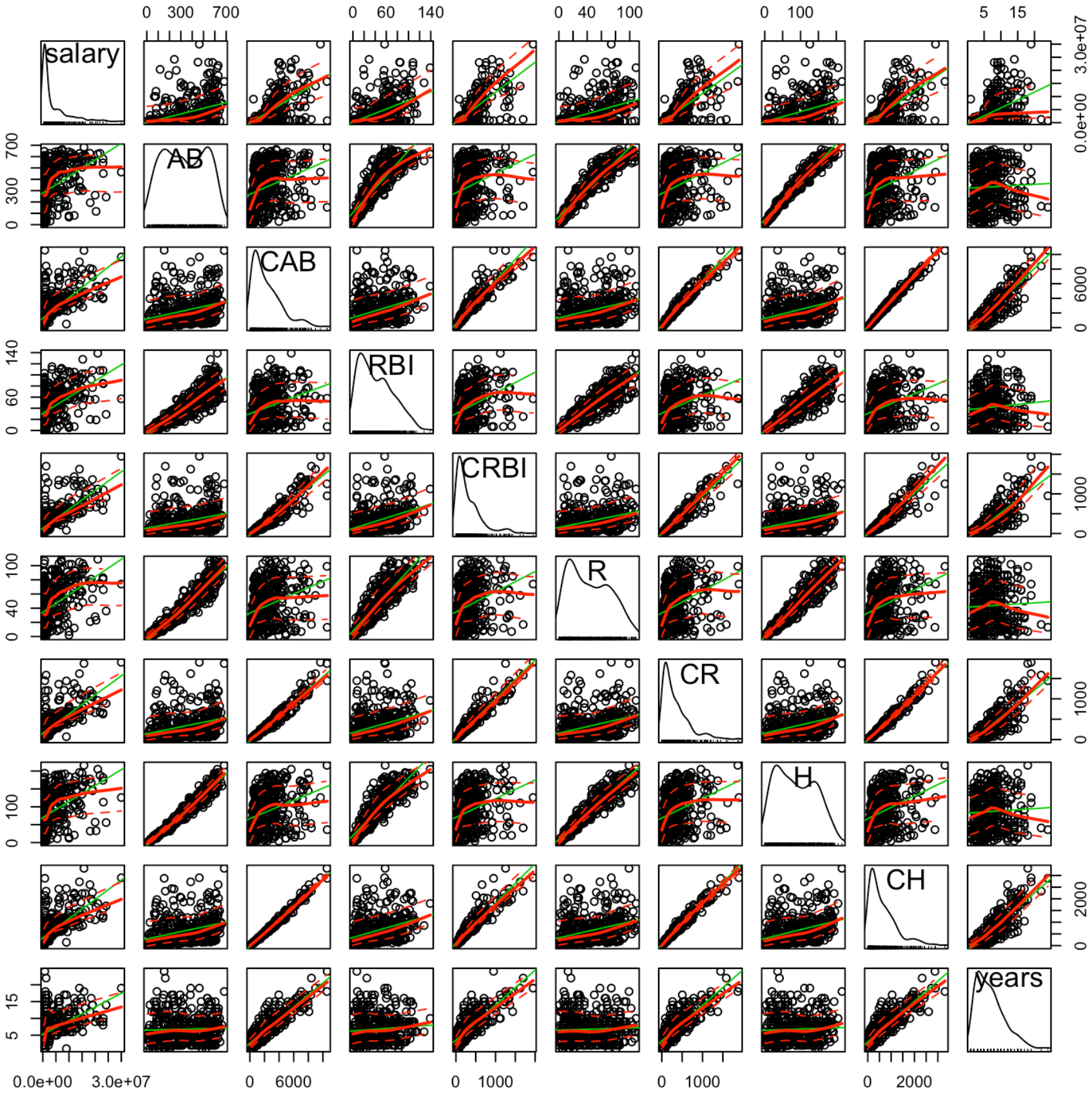
```
corrplot(cor(baseball[c("salary", "AB", "CAB", "RBI", "CRBI", "R", "CR", "H", "CH", "
HR", "CHR", "BB", "CBB", "E", "SO", "years")]), method = "ellipse")
```



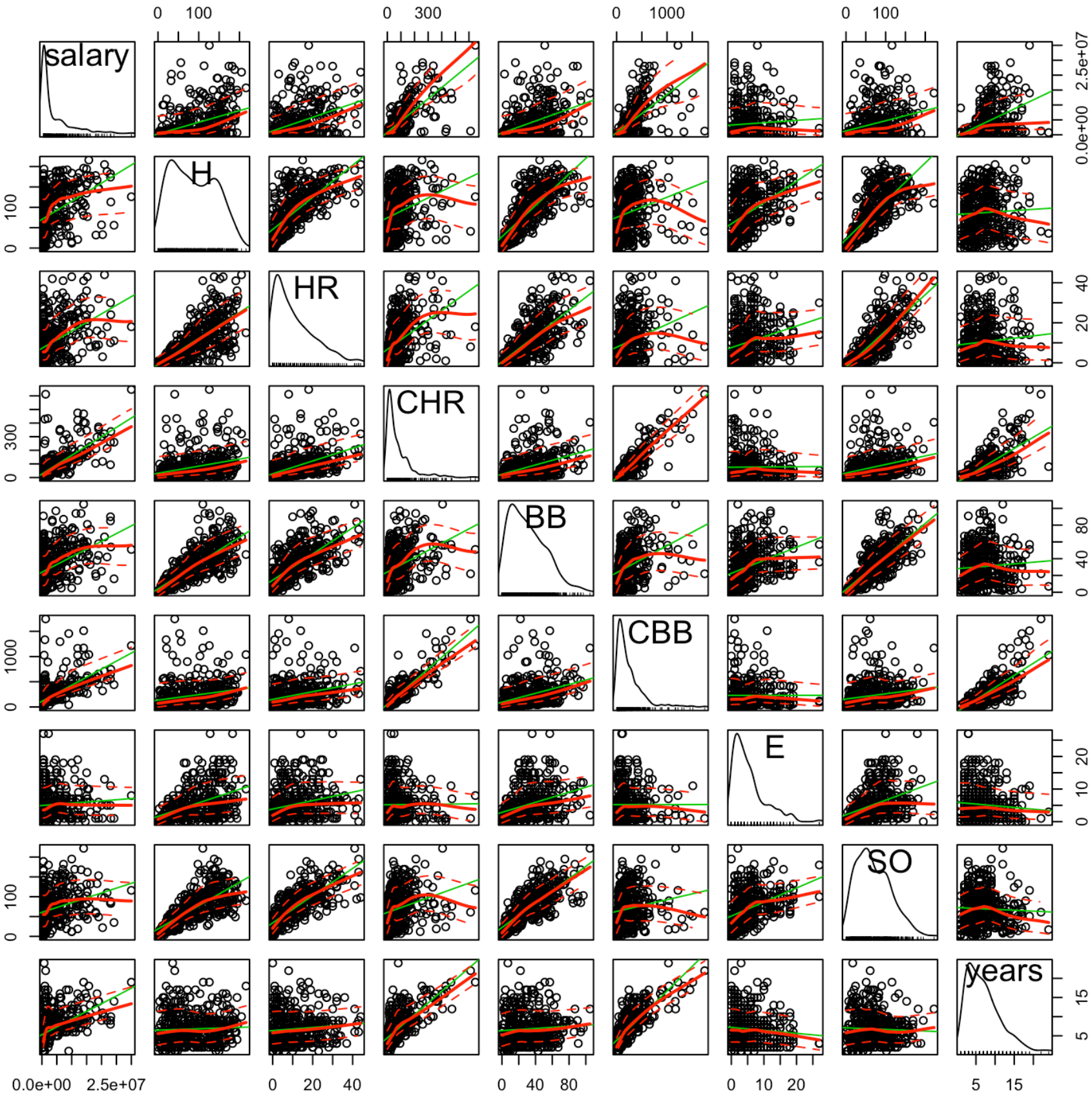

## Scatterplot Matrices

Next, let's use some scatterplot matrices to get more visual details about relationships between the response variable, `salary`, and judiciously selected explanatory variables. I have separately plotted the relationships between salary and previously feature engineered variables, since by design we expect some degree of collinearity from them with existing variables.

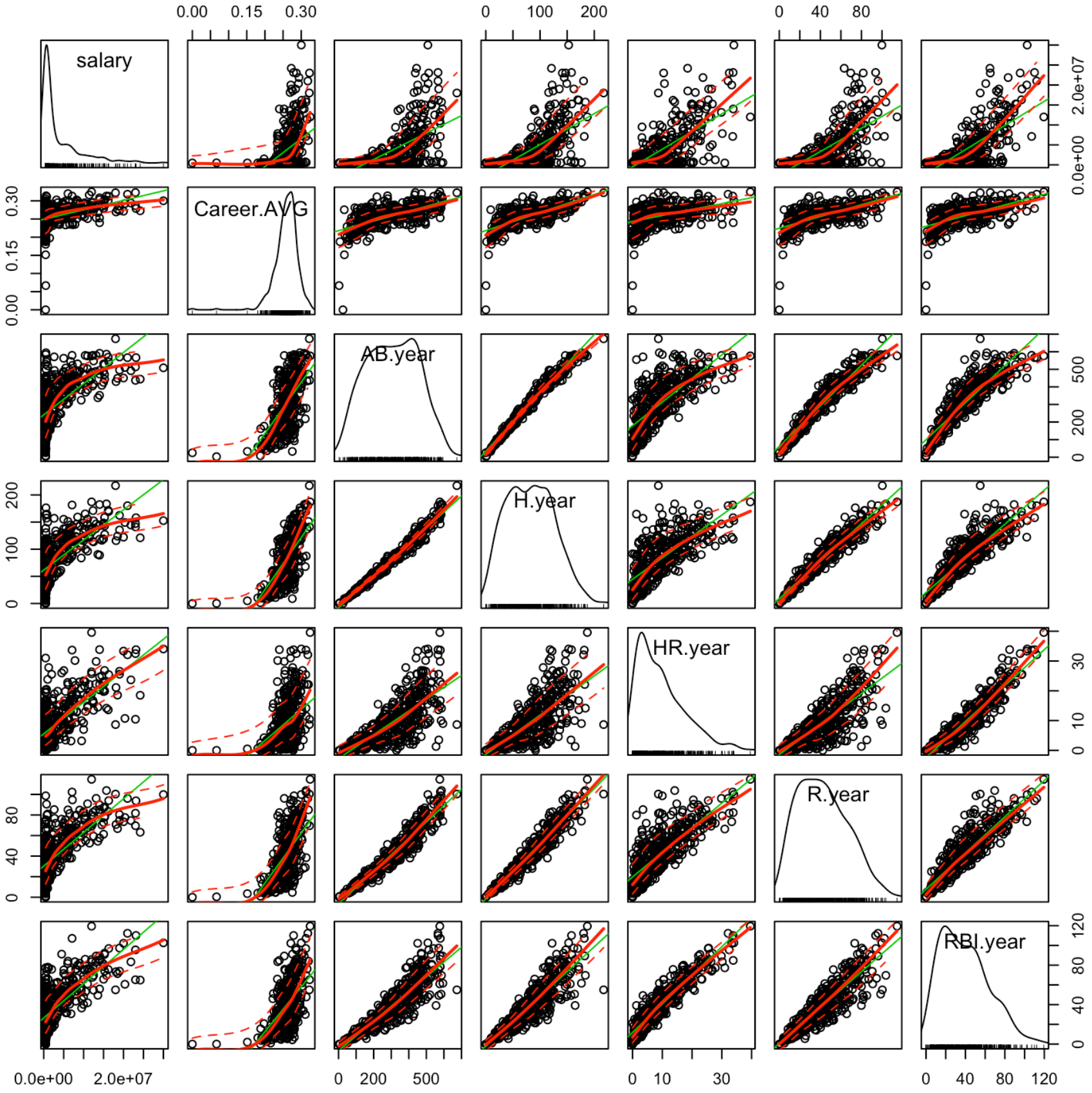
```
#Scatterplot matrix 1
scatterplotMatrix(baseball[c("salary", "AB", "CAB", "RBI", "CRBI", "R", "CR", "H", "C
H", "years")])
```




```
#Scatterplot matrix 2
scatterplotMatrix(baseball[c("salary", "H", "HR", "CHR", "BB", "CBB", "E", "SO", "yea
rs")])
```




```
#Scatterplot matrix for created variables
scatterplotMatrix(baseball[c("salary", "Career AVG", "AB/year", "H/year", "HR/year",
"R/year", "RBI/year")])
```




Confirming our previous speculation about `years` vs. `E` or `SO`, the corresponding plotted pairs show non-linear blobs of data points. In fact, `years` seems to have a curvilinear relationship with many other variables, indicating some need for a transformation on `years`. We will address this in a moment.

We also make the following observations from the first two pairs plots:

- Nonlinearity: Salary seems to be non-linearly correlated with many variables such as career at bats (`CAB`), career runs batted in (`CRBI`), career runs (`CR`), career hits (`CH`), career home runs (`CHR`), career walks (`CBB`), and years in major leagues (`years`). These plots show a somewhat "L"-shaped relationship, more obviously seen in `salary` vs. `E` or `salary` vs. `years`. Similar to `years`, this suggests that there may be two groups of observations inducing these curvilinear relationships – one

group where as their player stats increase, their salary increases, and another where no matter how their stats increase, their salary seems to stay the same! We may want to log transform `salary` to create a more linear relationship.

- Collinearity: Several of the career statistics (career home runs, career at-bats, career hits, etc.) are extremely collinear – as one career stat increases, others tend to follow. For example, career hits ( `CH` ) and career runs ( `CR` ) are strongly positively correlated, as is career at-bats ( `CAB` ) with career runs batted in ( `CRBI` ).
- Correlations: Performance stats in 2012 are generally not as strongly correlated with career performance stats as I had thought! In fact, many of the 2012 vs. career stats plots (e.g. between `RBI` and `CRBI` ) show almost vertical relationships, indicating near-independence of the two variables; there is a strong degree of randomness to perfomance in 2012 as compared to overall career stats.
- Influential points: There do seem to be some outliers or leverage points, but the density of points may cover up the details. However, there are two leverage points that consistently stick out from the rest of the data, best seen in number of errors's ( `E` ) relationship with number of hits ( `H` ). We will further examine this soon.

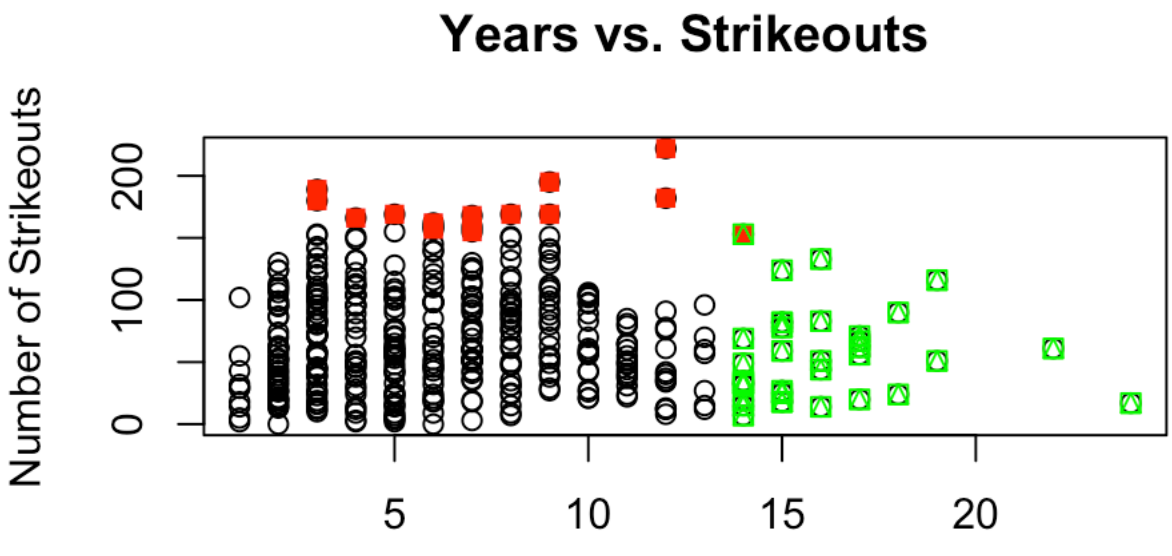
From the third correlation matrix (between salary and constructed variables), we further observe:

- Nonlinearity: Again, we see a curvilinear (seemingly exponential or log-like) relationship between `salary` and the created explanatory variables, encouraging yet again a log transformation of our y, `salary` .
- Correlations/collinearity: The number of at-bats per year ( `AB/year` ) vs. number of hits per year ( `H/year` ) shows the strongest positive, linear correlation in the pairs plot, though relationships of this type are seen all throughout this pairs plot. Collinearity between explanatory variables seems to be a significant problem in our modeling process!
- Influential points: There are indeed some influential points present, as seen in the left tail of `Career AVG` 's relationship with `AB/year` . These points skew the perception of `Career AVG` 's relationship with all other variables, and are far from the center of the "x"'s – thus likely have high leverage.

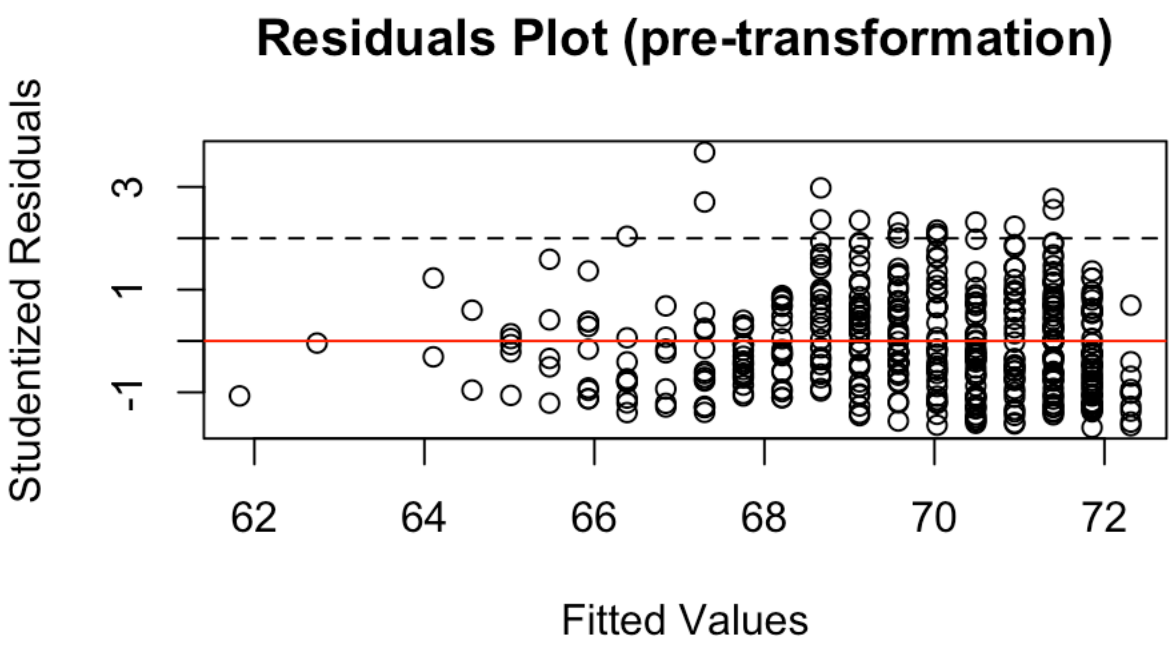
## Variable Transformations

The number of years a player has been in the major leagues ( `years` ) seems to consistently have a curvilinear relationship with other variables. Let's examine one of these relationships where the problem is most apparent, `years` vs. `SO` , to see if we can find an appropriate transformation.

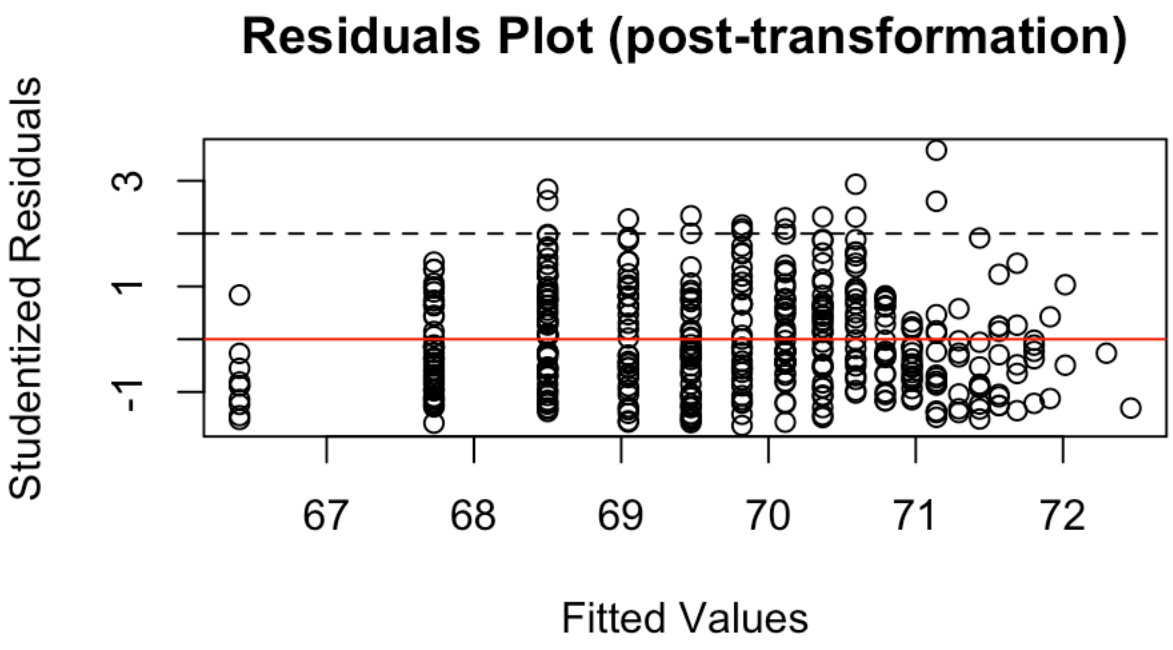
```
#Create model
expmod1 <- lm(SO ~ years, data = baseball)
exp_sturesid1 <- rstudent(expmod1) #studentized residuals

plot(baseball$SO ~ baseball$years, main = "Years vs. Strikeouts", xlab = "Years in Ma
jor Leagues", ylab = "Number of Strikeouts")
points(baseball[which(abs(exp_sturesid1) > 2), c("years", "SO")], col = "red", pch =
15) #red points -- outliers
points(baseball[which(hatvalues(expmod1) > 2*(2/nrow(baseball))), c("years", "SO")],
col = "green", pch = 14) #green points -- leverage points
```



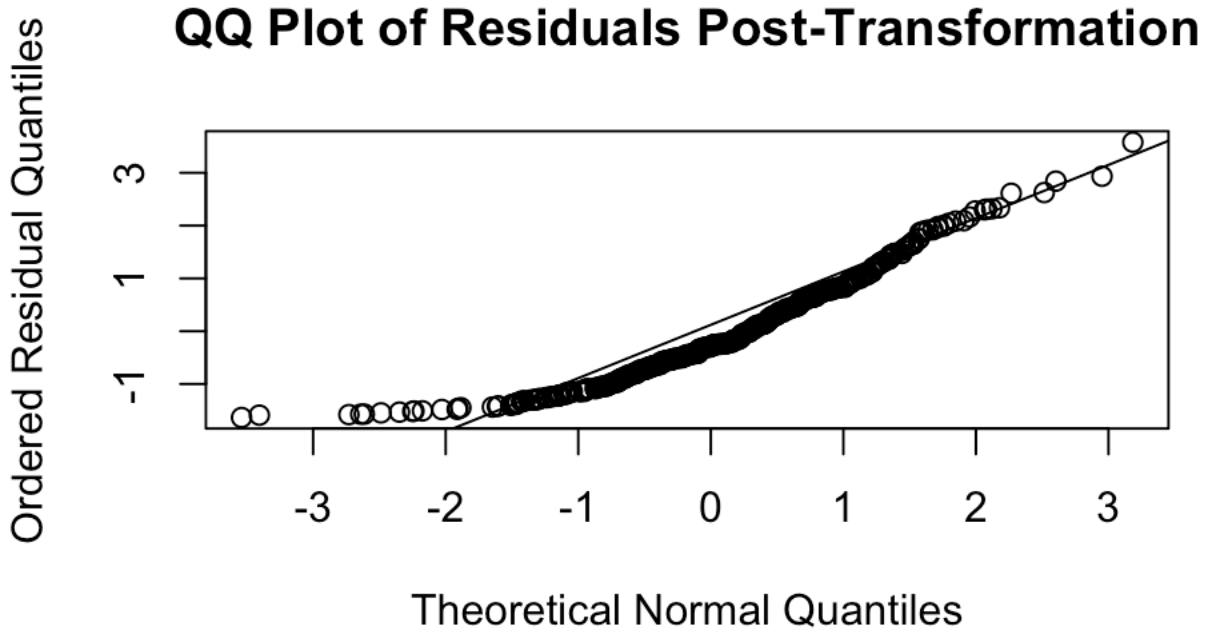

```
#Studentized residuals plot for relationship between H and BB, pre-transformation
plot(exp_sturesid1 ~ expmod1$fitted.values, main = "Residuals Plot (pre-transformatio
n)", xlab = "Fitted Values", ylab = "Studentized Residuals")
abline(h = 0, col = "red")
abline(h = 2, lty = 2)
```




```
#Residuals plot after years log transformation
expmod2 <- lm(SO ~ log(years), data = baseball)
exp_sturesid2 <- rstudent(expmod2)
plot(exp_sturesid2 ~ expmod2$fitted.values, main = "Residuals Plot (post-transformati
on)", xlab = "Fitted Values", ylab = "Studentized Residuals")
abline(h = 0, col = "red")
abline(h = 2, lty = 2)
```



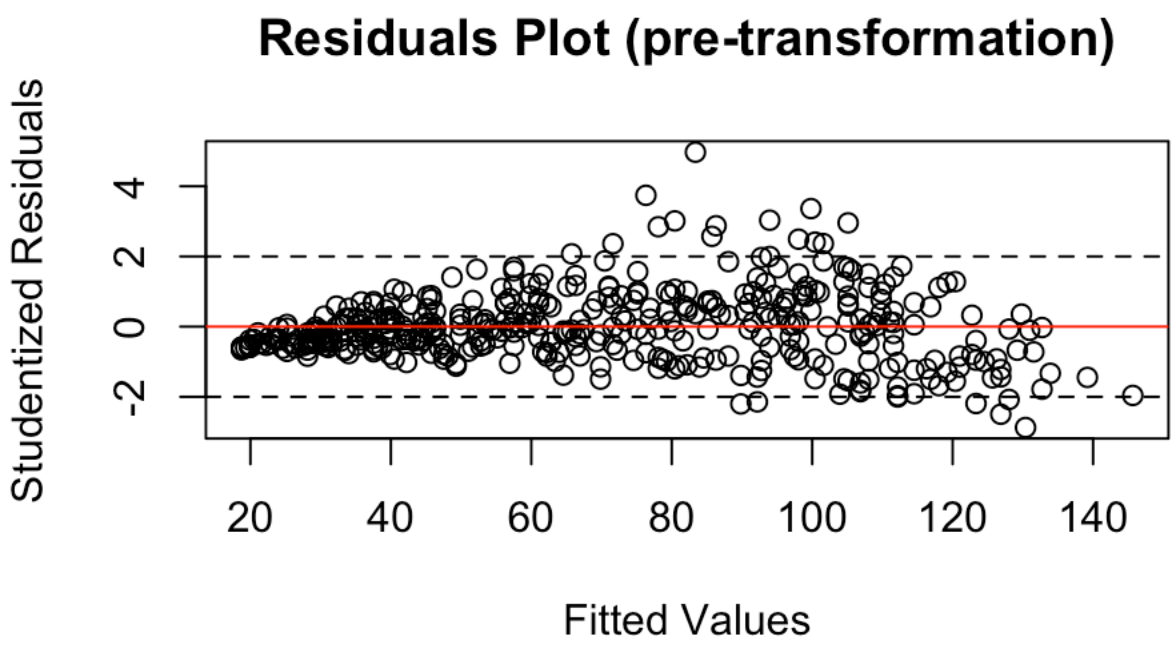

```
#Assess normality of resulting residuals
norms <- rnorm(length(expmod2$residuals), 0, 1)
qqplot(norms, exp_sturesid2, main = "QQ Plot of Residuals Post-Transformation", xlab
= "Theoretical Normal Quantiles", ylab = "Ordered Residual Quantiles")
qqline(norms)
```



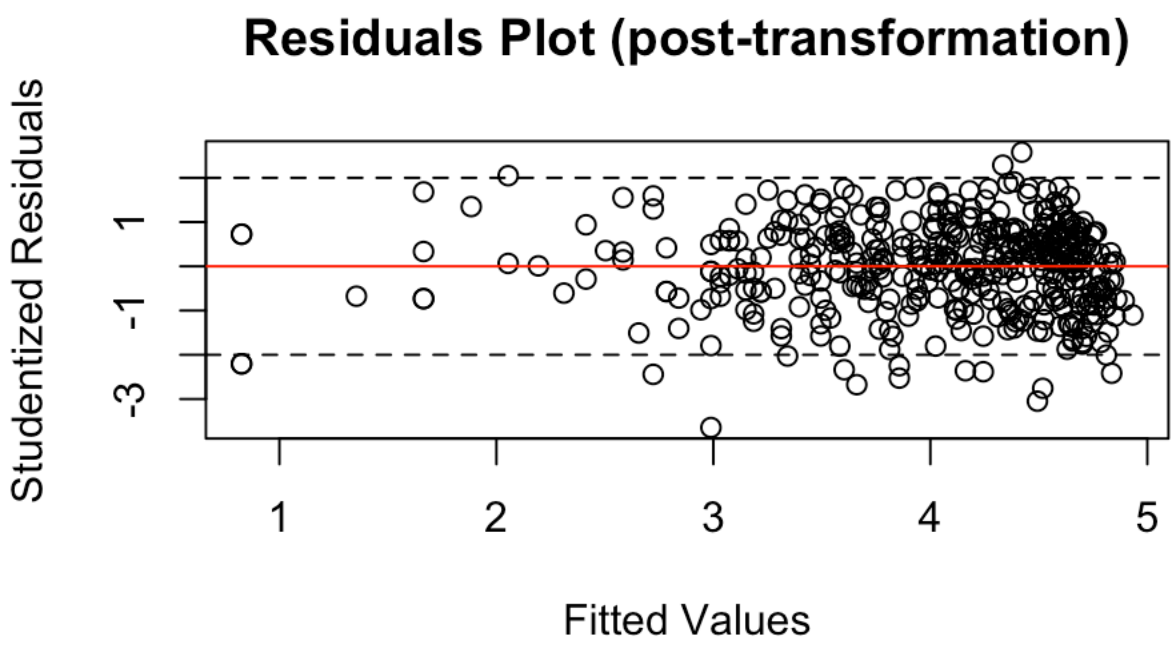

It turns out that what I thought was two groupings was indeed a bunch of leverage points (green) and outliers (red)! From the residuals plot we see that a more severe problem is non-constant variance. Taking the log transformation of `years` helped stabilize the variance, justifying our choice of transformation. From the QQ Plot we can see that the resulting residuals are slightly skewed compared to a normal distribution, but for the most part do match up with the normal distribution.

The variables `H`, `AB`, and `SO` also seem problematic – they seem to show consistent funneling in many residuals plots, and a transformation will likely stabilize variance. I will use the relationship between `H` and `SO` to carry out analysis.

```
#Studentized residuals plot for H vs SO
expmod3 <- lm(SO ~ H, data = baseball)
exp_sturesid3 <- rstudent(expmod3)
plot(exp_sturesid3 ~ expmod3$fitted.values, main = "Residuals Plot (pre-transformatio
n)", xlab = "Fitted Values", ylab = "Studentized Residuals")
abline(h = 0, col = "red")
abline(h = c(2, -2), lty = 2)
```




```
#Plot after double log transformation
expmod4 <- lm(log(SO+1) ~ log(H+1), data = baseball)
exp_sturesid4 <- rstudent(expmod4)
plot(exp_sturesid4 ~ expmod4$fitted.values, main = "Residuals Plot (post-transformati
on)", xlab = "Fitted Values", ylab = "Studentized Residuals")
abline(h = 0, col = "red")
abline(h = c(2, -2), lty = 2)
```




Although the double-log transformation clustered the data points towards higher fitted values, it does seem that logging `SO` and logging `H` are both variance-stabilizing transformations. This was the best transformation I got after fiddling with various combinations. Since `AB` has similar issues, we will also take its log in our analysis.

## Interaction Exploration

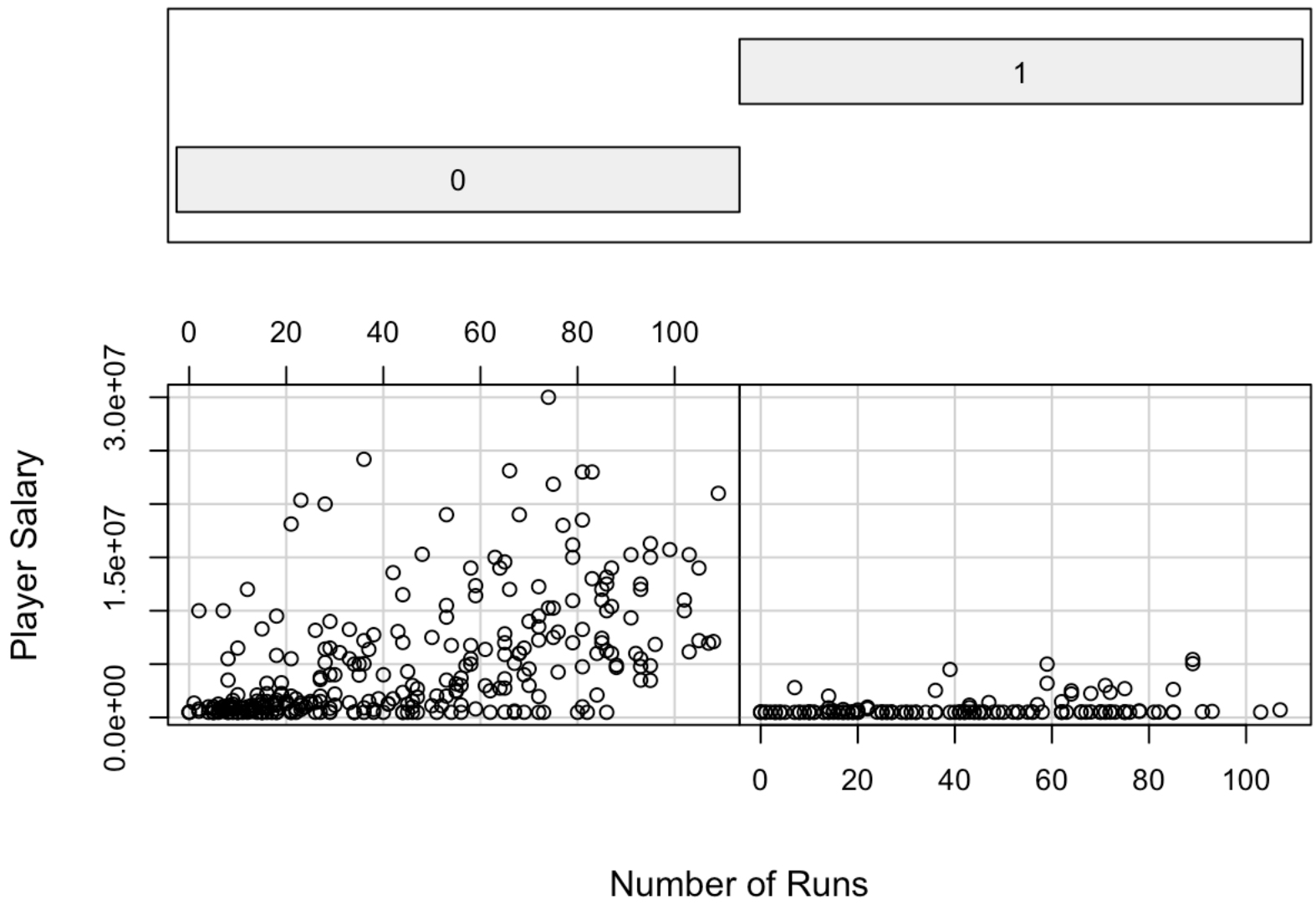
There may also be interaction between explanatory terms. For example, from the following conditional plots (coplots) we observe that a player's arbitration eligibility and free-agency eligibility affects the effect of the number of runs in 2012 on their salary.

Players who are not arbitration-eligible (major leagues career length < 3 years) have salaries ranging from comparatively very low (< $1 million) to around $30 million. Players who are arbitration-eligible (career length 3-5 years), however, seem to have drastically lower salaries in comparison. Their highest salaries are only around $8 million, with the majority very much below that!

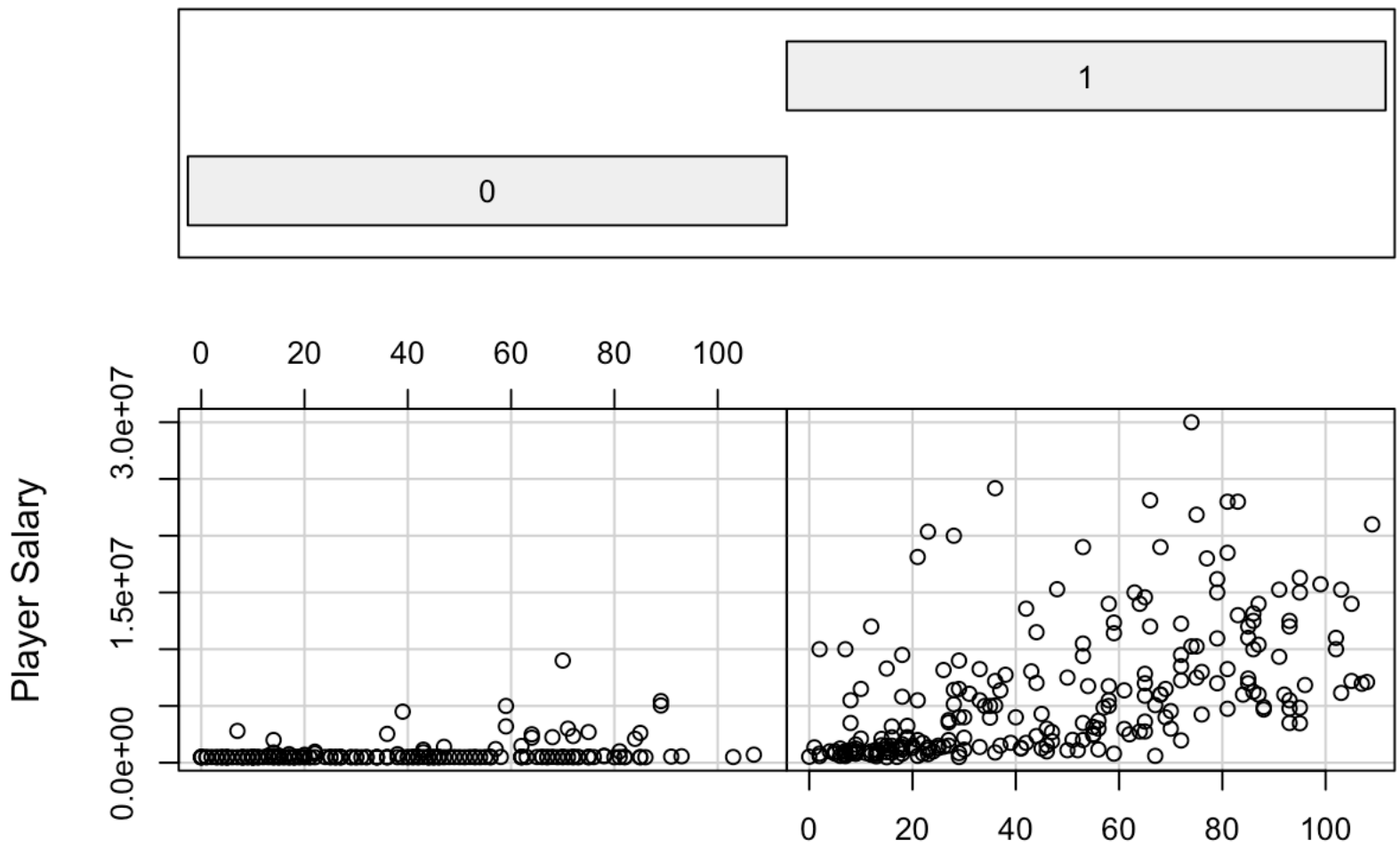
In comparison to players who are free-agency eligible, however, all the players previously mentioned seem to have low salaries! Free-agency eligible players (career length 6 years or more) have a much wider range of salaries, ranging from very low to almost $30 million!

Since both the arbitration-eligible and free-agency-eligible variables are related to career length ( `years` ), it would be wise in our further analysis to consider some sort of interaction between career length and number of runs.

```
coplot(baseball$'salary' ~ baseball$'R'|as.factor(baseball$`Arbitration eligible`), x
lab = "Number of Runs", ylab = "Player Salary")
```




```
coplot(baseball$'salary' ~ baseball$'R'|as.factor(baseball$`Free-agency eligible`), x
lab = "Number of Runs", ylab = "Player Salary")
```




### Preliminary Results

Some conclusions of our preliminary data analysis are that collinearity and non-linearity, as well as non-constant variance, are all problematic in our data. We have in this section decided to log transform player salary ( `salary` ), number of years in major leagues ( `years` ), number of hits ( `H` ), number of at-bats ( `AB` ), and number of strikeouts ( `SO` ). We have also realized that there may be interaction between career length ( `years` ) and number of runs ( `R` ) in their effects on salary.

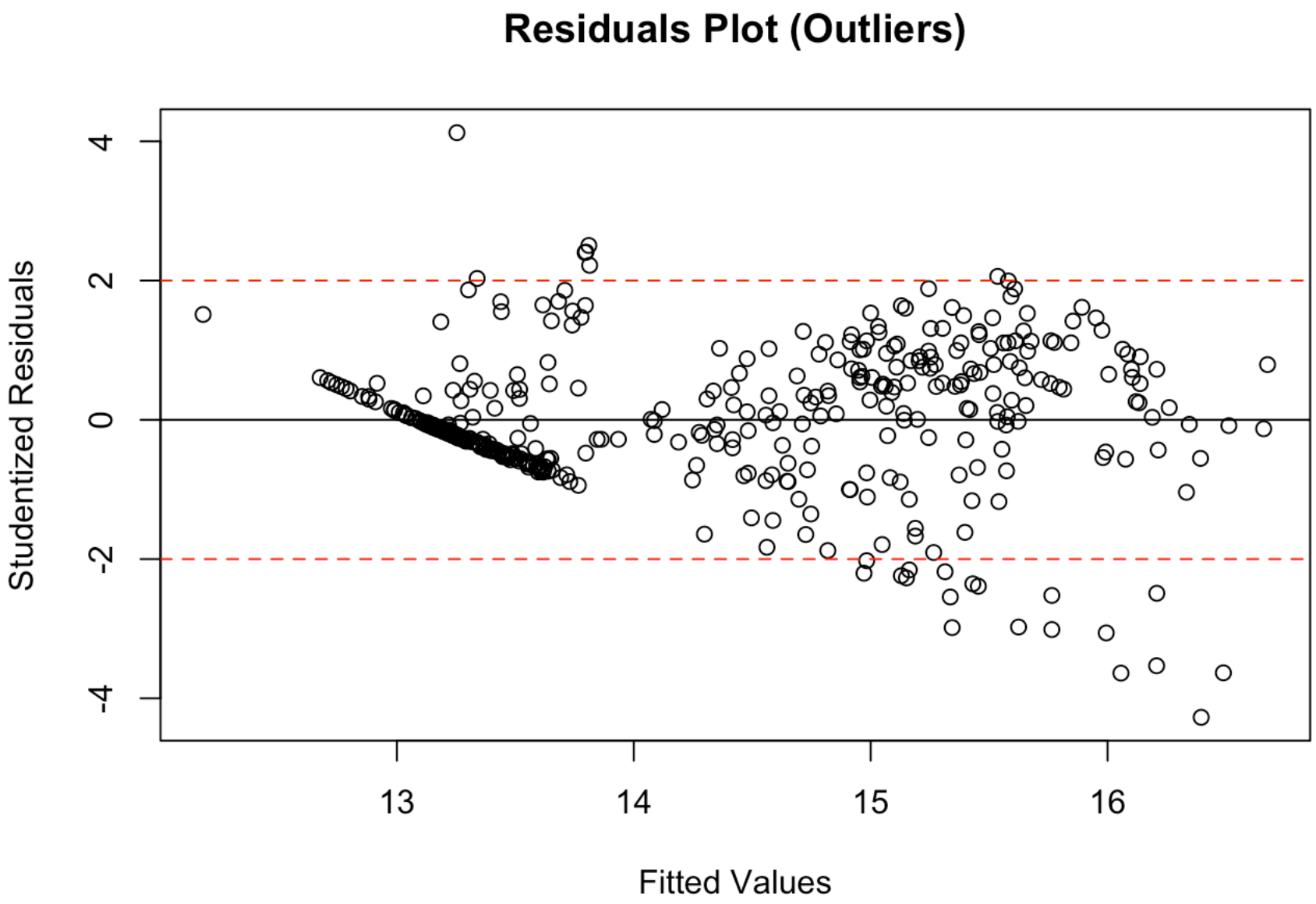
# Data Analysis

### Simple Models and Unusual Data

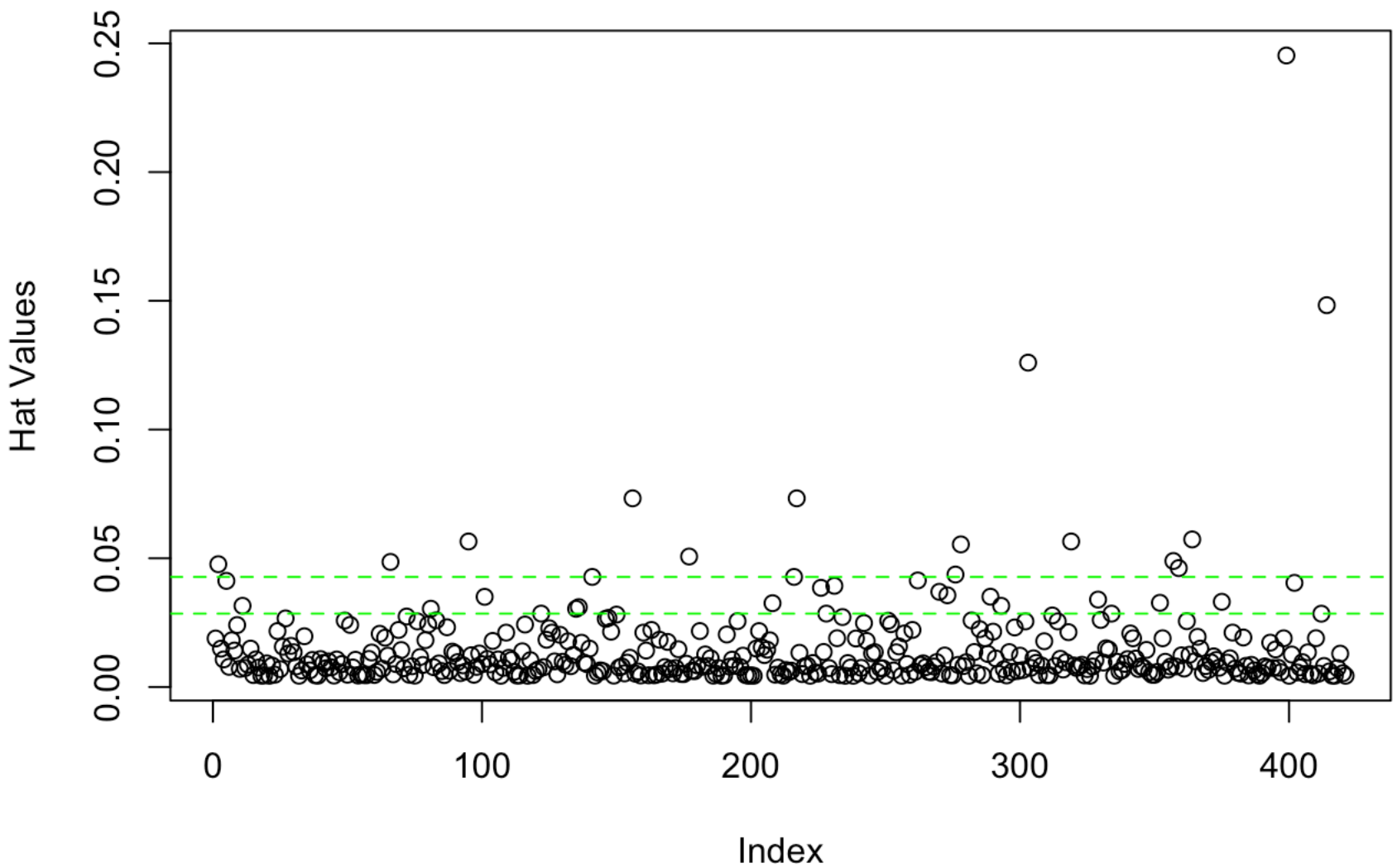
First, as instructed, let's fit a simple model predicting logged `salary` from the two length-of-career dummy variables we created, logged career runs, and the interaction between career length and career runs.

There seem to be a lot of outliers, leverage points, and influential points! From the combined plot involving residuals, hat values, and Cook's distance (circle size) we can tell that there is one data point that is not an outlier but is a major leverage and influential point. It is characterized by a large bubble and large hat value.

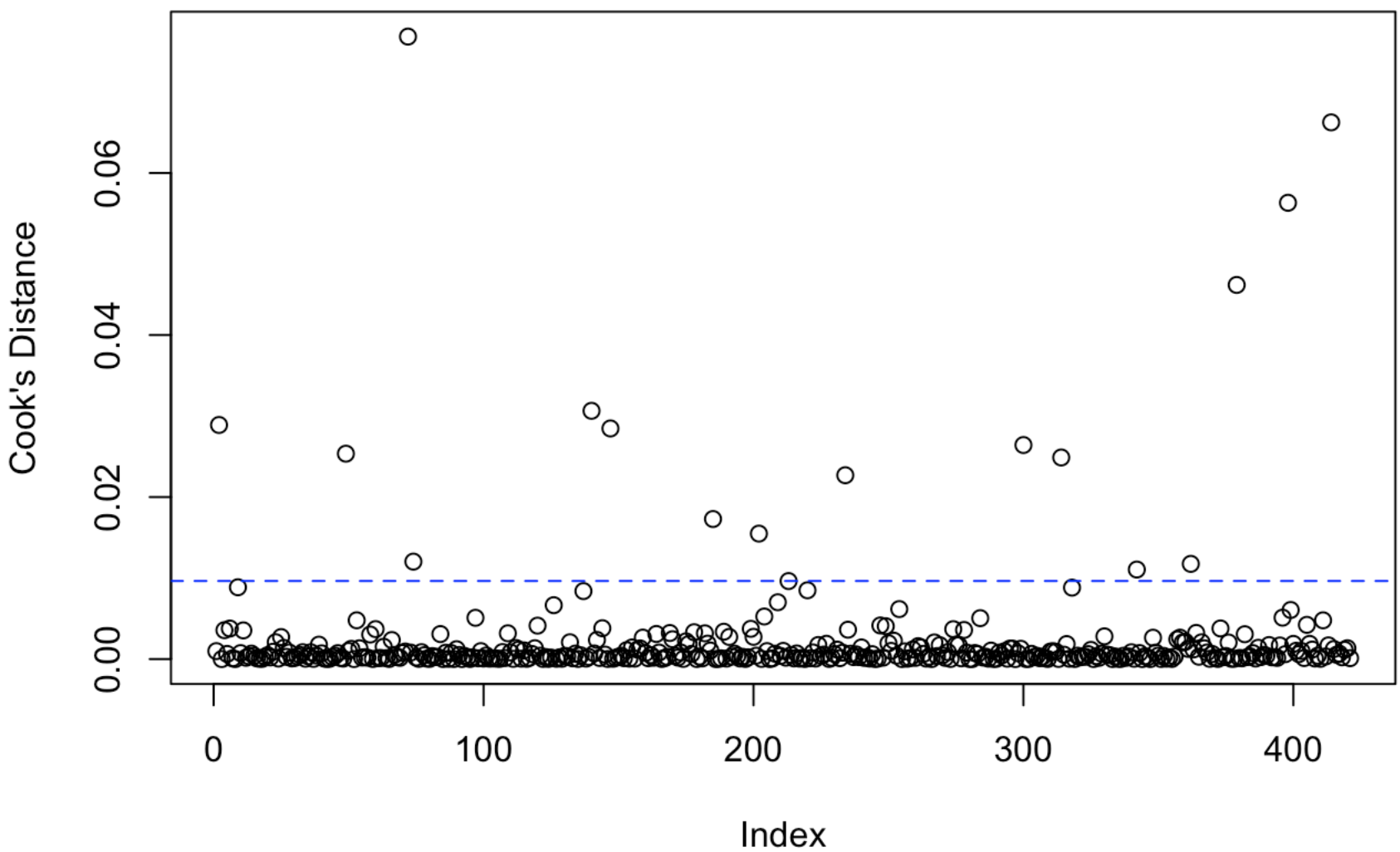
```
#Fit simple model
lm1 <- lm(log(salary) ~ `Arbitration eligible` + `Free-agency eligible` + log(CR + 1)
+ `Arbitration eligible`*log(CR + 1) + `Free-agency eligible`*log(CR + 1), data = bas
eball)

#Check for outliers -- studentized residuals plot
stu_resids <- rstudent(lm1)
plot(stu_resids ~ lm1$fitted.values, main = "Residuals Plot (Outliers)", xlab = "Fitt
ed Values", ylab = "Studentized Residuals")
#points(lm1$fitted.values[which(abs(stu_resids) > 2)], col = "red", pch = 15)
abline(h = 0)
abline(h = c(2, -2), lty = 2, col = "red")
```



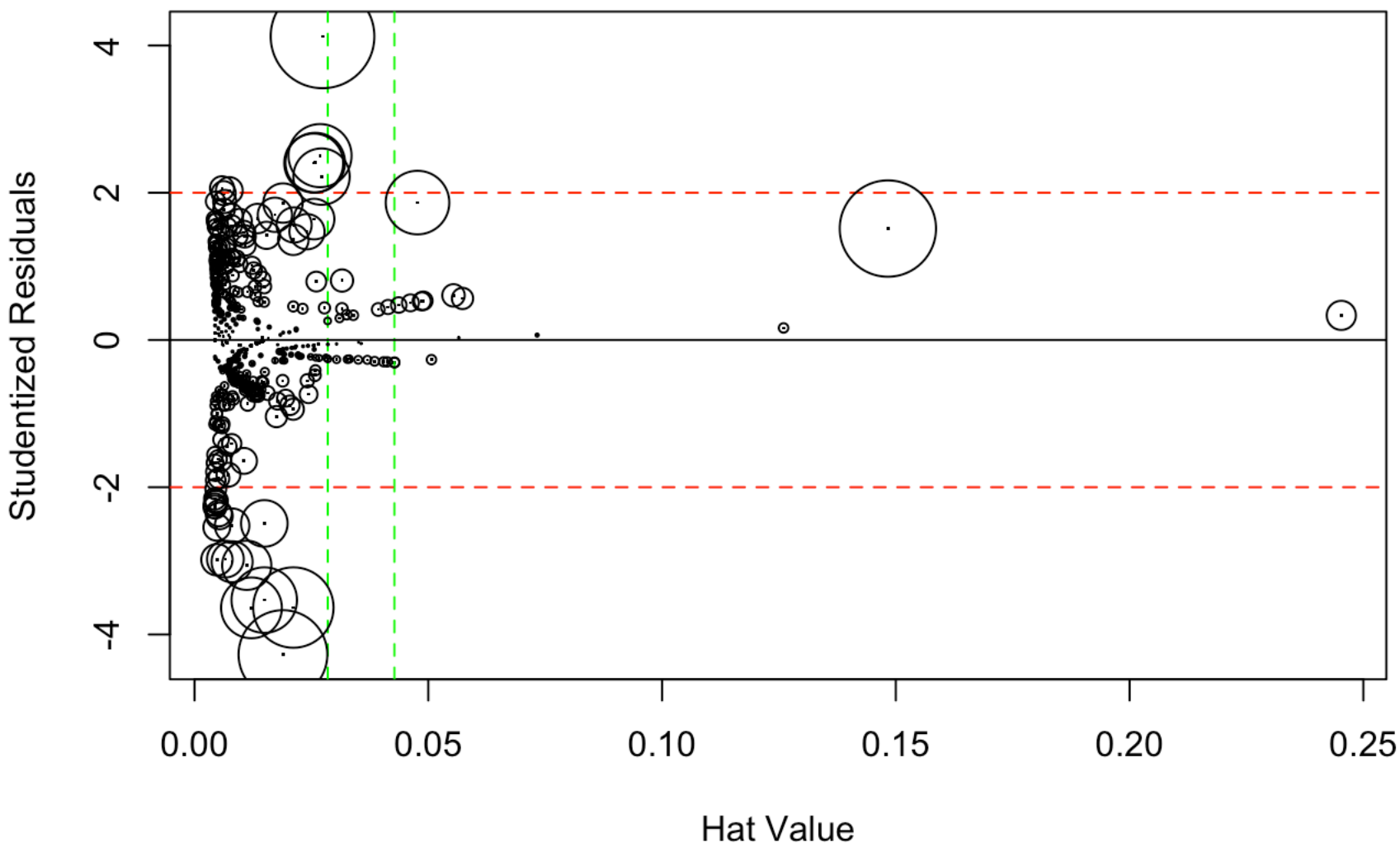

```
#Check for leverage points -- plot hat values
hats <- hatvalues(lm1)
dof <- nrow(baseball) - lm1$df.residual
hbar <- dof/nrow(baseball)
plot(hats, main = "Hat Values (Leverage)", ylab = "Hat Values")
abline(h = c(2*hbar, 3*hbar), lty = 2, col = "green")
```




```
#Check for influential points -- plot Cook's distance
cooks <- cooks.distance(lm1)
plot(cooks, main = "Cook's Distance (Influence)", ylab = "Cook's Distance")
abline(h = 4/lm1$df.residual, lty = 2, col = "blue")
```




```
#Combined plot
plot(stu_resids ~ hats, pch = ".", main = "Outliers, Leverage, Influence", xlab = "Ha
t Value", ylab = "Studentized Residuals")
abline(v = c(2*hbar, 3*hbar), lty = 2, col = "green")
abline(h = c(2,-2), lty = 2, col = "red")
abline(h = 0)
symbols(x = hats, y = stu_resids, circles = sqrt(cooks)/25, inches = F, add = T)
```




Purely out of curiosity, I discovered that a good amount of outliers and influential points are players with Hispanic last names, while almost none of the leverage point players are… for the purposes of this class, I will just leave this little discovery here.

```
#Outlier players
baseball$nameLast[which(abs(stu_resids) > 2)]
```

```
##  [1] "Bruce"    "Cairo"    "Cespedes" "Chavez"   "DeRosa"   "Giambi"
##  [7] "Gonzalez" "Harris"   "Ibanez"   "Izturis"  "Johnson"  "Jones"
## [13] "Kearns"   "Kennedy"  "Kotsay"   "Longoria" "Lopez"    "Mauer"
## [19] "Nady"     "Overbay"  "Pierre"   "Ramirez"  "Thome"    "Viciedo"
## [25] "Vizquel"  "Wilson"
```

```
#Leverage players
baseball$nameLast[which(hats > 2*hbar)]
```

```
##  [1] "Ackley"      "Altuve"      "Aoki"        "Carroll"     "Cozart"
##  [6] "De Jesus"    "Dirks"       "Flaherty"    "Galvis"      "Gamel"
## [11] "Goldschmidt" "Hague"       "Hosmer"      "Kawasaki"    "Kipnis"
## [16] "Komatsu"     "Lawrie"      "Liddi"       "Lobaton"     "Moore"
## [21] "Moustakas"   "Murphy"      "Navarro"     "Nickeas"     "Parrino"
## [26] "Pena"        "Pina"        "Recker"      "Robinson"    "Seager"
## [31] "Snyder"      "Sogard"      "Stewart"     "Thames"      "Vogt"
## [36] "Weeks"       "Wright"
```

```
#Influential players
baseball$nameLast[which(cooks > 4/lm1$df.residual)]
```

```
##  [1] "Ackley"   "Bruce"    "Cespedes" "Chavez"   "Giambi"   "Gonzalez"
##  [7] "Ibanez"   "Jones"    "Longoria" "Pierre"   "Ramirez"  "Sandoval"
## [13] "Span"     "Thome"    "Vizquel"  "Wright"
```

Next, we will fit a more complicated linear least-squares regression, incorporating transformations I came up with in our exploratory data analysis. This 35-variable model (including two interaction terms) explains away 81.08% of variation in the data (adjusted $R^2$). The interaction variable `Int1` is very insignificant, so we will remove it in our further analysis.

```
#Add in interaction terms as variables
baseball$Int1 <- baseball$`Arbitration eligible` * baseball$R
baseball$Int2 <- baseball$`Free-agency eligible` * baseball$R

#Linear least-squares model
lm2 <- lm(log(salary) ~ PO + A + E + log(AB) + R + log(H+1) + HR + RBI + BB + log(SO+
1) + IBB + HBP + log(years) + CAB + CH + CHR + CR + CRBI + CBB + MI + C + CF + AVG +
`Career AVG` + OBP + `Career OBP` + `AB/year` + `H/year` + `HR/year` + `R/year` + `RB
I/year` + `Arbitration eligible` + `Free-agency eligible` + `Int1` + `Int2`, data = b
aseball)

summary(lm2)
```

```
##
## Call:
## lm(formula = log(salary) ~ PO + A + E + log(AB) + R + log(H +
##     1) + HR + RBI + BB + log(SO + 1) + IBB + HBP + log(years) +
##     CAB + CH + CHR + CR + CRBI + CBB + MI + C + CF + AVG + `Career AVG` +
##     OBP + `Career OBP` + `AB/year` + `H/year` + `HR/year` + `R/year` +
##     `RBI/year` + `Arbitration eligible` + `Free-agency eligible` +
##     Int1 + Int2, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76514 -0.27614  0.00756  0.24149  1.88187
##
```

```
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.294e+01  5.750e-01  22.498  < 2e-16 ***
## PO                        6.564e-05  1.388e-04   0.473 0.636655
## A                        -4.650e-04  4.445e-04  -1.046 0.296152
## E                         9.416e-03  9.334e-03   1.009 0.313705
## log(AB)                   5.492e-02  2.281e-01   0.241 0.809892
## R                        -1.920e-02  5.244e-03  -3.661 0.000286 ***
## log(H + 1)                1.850e-01  2.733e-01   0.677 0.498871
## HR                       -1.511e-02  1.013e-02  -1.491 0.136759
## RBI                       5.369e-03  4.483e-03   1.198 0.231782
## BB                        2.555e-03  4.256e-03   0.600 0.548656
## log(SO + 1)              -2.916e-02  1.095e-01  -0.266 0.790046
## IBB                       4.855e-05  1.238e-02   0.004 0.996872
## HBP                      -8.531e-03  1.003e-02  -0.851 0.395523
## log(years)                1.882e-01  1.798e-01   1.046 0.296040
## CAB                       2.958e-04  3.085e-04   0.959 0.338309
## CH                        2.260e-04  1.432e-03   0.158 0.874644
## CHR                       2.382e-03  3.620e-03   0.658 0.510815
## CR                       -2.126e-03  1.558e-03  -1.365 0.173193
## CRBI                     -3.911e-04  1.740e-03  -0.225 0.822246
## CBB                      -9.750e-04  5.382e-04  -1.812 0.070808 .
## MI                        2.239e-01  1.299e-01   1.723 0.085657 .
## C                         9.503e-02  9.937e-02   0.956 0.339520
## CF                        4.899e-02  1.371e-01   0.357 0.721144
## AVG                      -1.829e+00  2.786e+00  -0.656 0.511972
## `Career AVG`             -9.724e-01  3.424e+00  -0.284 0.776563
## OBP                      -5.799e-03  2.104e-02  -0.276 0.783028
## `Career OBP`              2.220e-03  2.945e-02   0.075 0.939949
## `AB/year`                -6.429e-03  3.169e-03  -2.028 0.043201 *
## `H/year`                  4.113e-03  1.358e-02   0.303 0.762066
## `HR/year`                -1.439e-02  3.360e-02  -0.428 0.668615
## `R/year`                  4.655e-02  1.279e-02   3.641 0.000309 ***
## `RBI/year`                2.180e-02  1.557e-02   1.401 0.162161
## `Arbitration eligible` -1.883e-01  2.120e-01  -0.888 0.374956
## `Free-agency eligible` -9.778e-04  2.945e-01  -0.003 0.997353
## Int1                      1.929e-03  3.601e-03   0.536 0.592514
## Int2                      2.187e-02  3.748e-03   5.834 1.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5465 on 385 degrees of freedom
## Multiple R-squared:  0.8265, Adjusted R-squared:  0.8108
## F-statistic: 52.42 on 35 and 385 DF,  p-value: < 2.2e-16
```

## Assessing Collinearity

We can assess collinearity in this model by taking a look at the condition index = $\sqrt{\lambda_1/\lambda_j} \ \ \forall j \in [1, p]$ for each variable, derived using PCA.

```
#Transform years
baseball$logyears <- log(baseball$years)

#PCA
p_comp <- prcomp(scale(baseball[,c('PO', 'A', 'E', 'AB', 'R', 'H', 'HR', 'RBI', 'BB',
'SO', 'IBB', 'HBP', 'logyears', 'CAB', 'CH', 'CHR', 'CR', 'CRBI', 'CBB', 'MI', 'C', '
CF', 'AVG', 'Career AVG', 'OBP', 'Career OBP', 'AB/year', 'H/year', 'HR/year', 'R/yea
r', 'RBI/year', 'Arbitration eligible','Free-agency eligible', 'Int2')]))

#Condition index
cond_ind <- p_comp$sdev[1]/p_comp$sdev
cond_ind
```

```
##  [1]   1.000000   1.703845   2.595913   2.873945   3.229833   3.352639
##  [7]   3.964921   4.609562   4.665676   4.862949   5.215939   5.396005
## [13]   5.620017   5.759532   6.271155   7.371262   8.587163   9.635344
## [19]  11.267013  12.373016  14.399796  15.734839  19.365737  21.457137
## [25]  24.698652  26.005101  28.035908  29.873775  39.490328  42.207730
## [31]  69.813403  77.628778 118.044231 184.654709
```

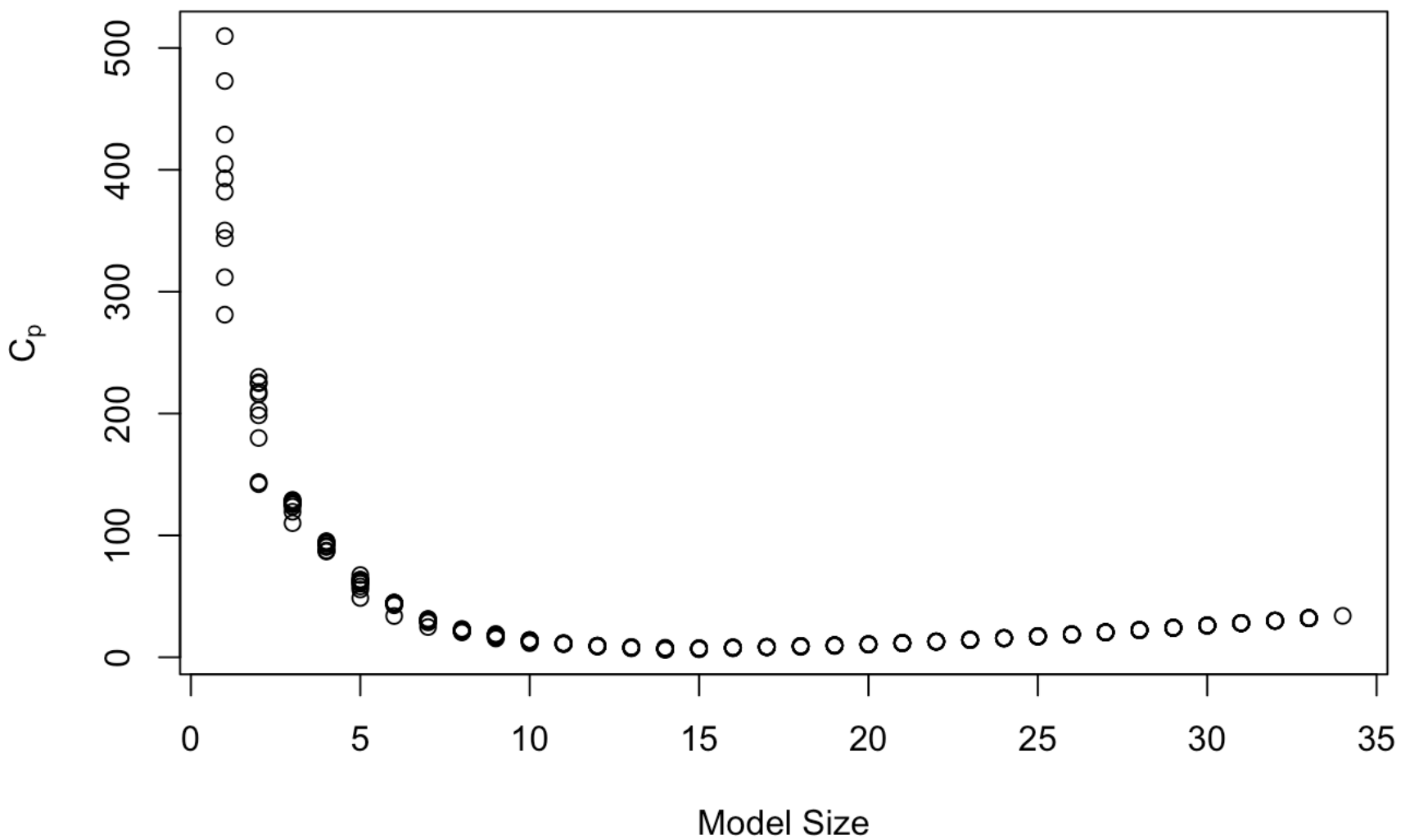
```
print(paste("Condition indexes > 10 indicate significant collinearity problems and an
instable regression coefficients. In our variables, there are", sum(cond_ind > 10), "
cases of bad collinearity."))
```

```
## [1] "Condition indexes > 10 indicate significant collinearity problems and an inst
able regression coefficients. In our variables, there are 16 cases of bad collinearit
y."
```

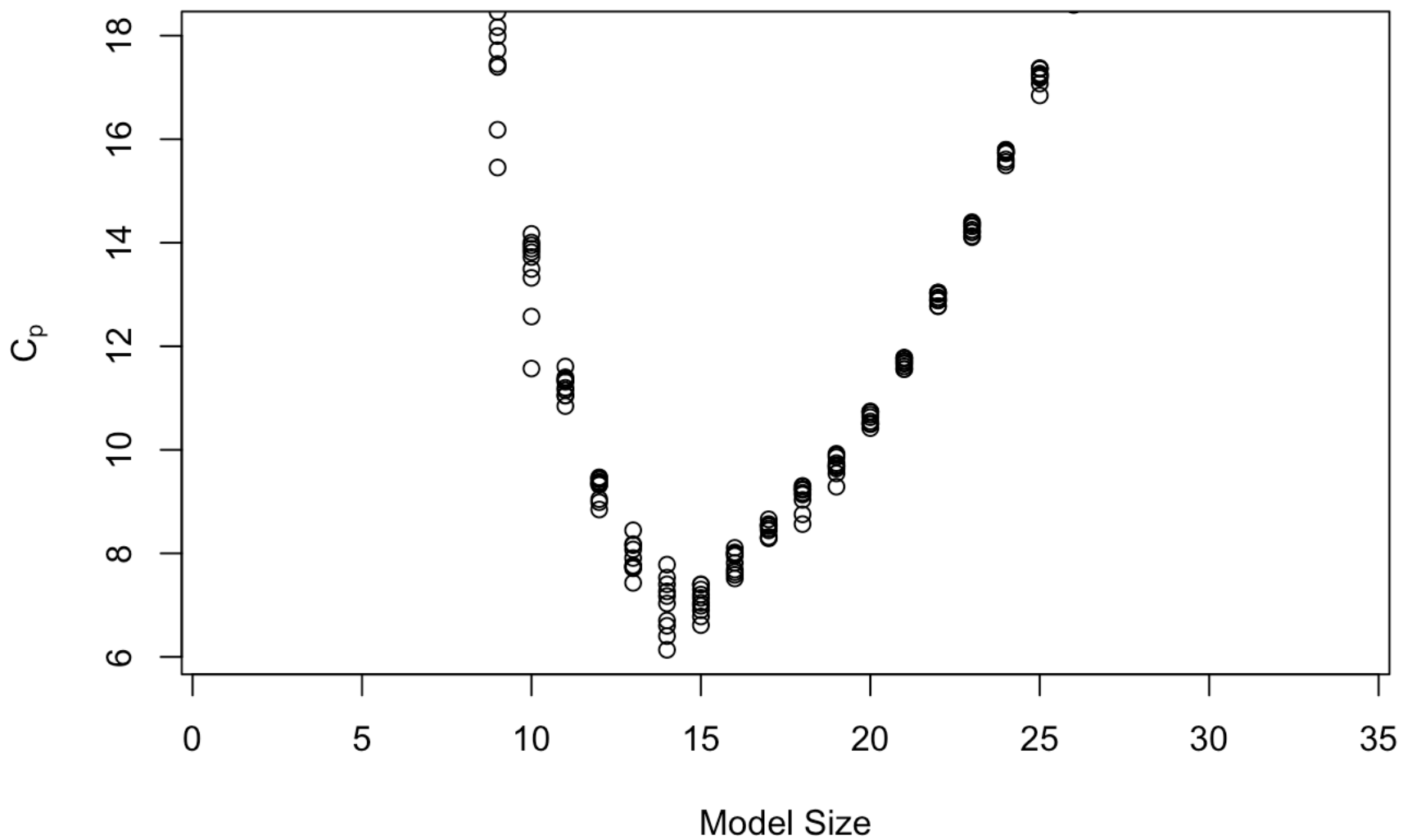
## Model Selection

To formally start the model selection process, I first use `leaps()` to identify the top 10 models of each size. I can then plot the Mallows' Cp value for various models of each size, and narrow in on the "bend" in the graph to see which model size yields the lowest Cp values. Models with 14-15 variables have the lowest Cp's!

```
#Top 10 models per model size
top10_persize <- leaps(x = baseball[,c('PO', 'A', 'E', 'AB', 'R', 'H', 'HR', 'RBI', '
BB', 'SO', 'IBB', 'HBP', 'logyears', 'CAB', 'CH', 'CHR', 'CR', 'CRBI', 'CBB', 'MI', '
C', 'CF', 'AVG', 'Career AVG', 'OBP', 'Career OBP', 'AB/year', 'H/year', 'HR/year', '
R/year', 'RBI/year', 'Arbitration eligible','Free-agency eligible', 'Int2')], y = bas
eball$salary, int = FALSE, strictly.compatible = FALSE)

#Cp plot
plot(top10_persize$size, top10_persize$Cp, xlab = "Model Size", ylab = expression(C[p
]), main = "Mallows' Cp for Various Model Sizes")
```




```
#Narrowed in Cp plot
plot(top10_persize$size, top10_persize$Cp, xlab = "Model Size", ylab = expression(C[p
]), ylim=c(min(top10_persize$Cp),median(top10_persize$Cp)), main = "Mallows' Cp (Zoom
ed In)")
```




```
#Regressors present in 14-16 variable models
colSums(top10_persize$which[131:150,])
```

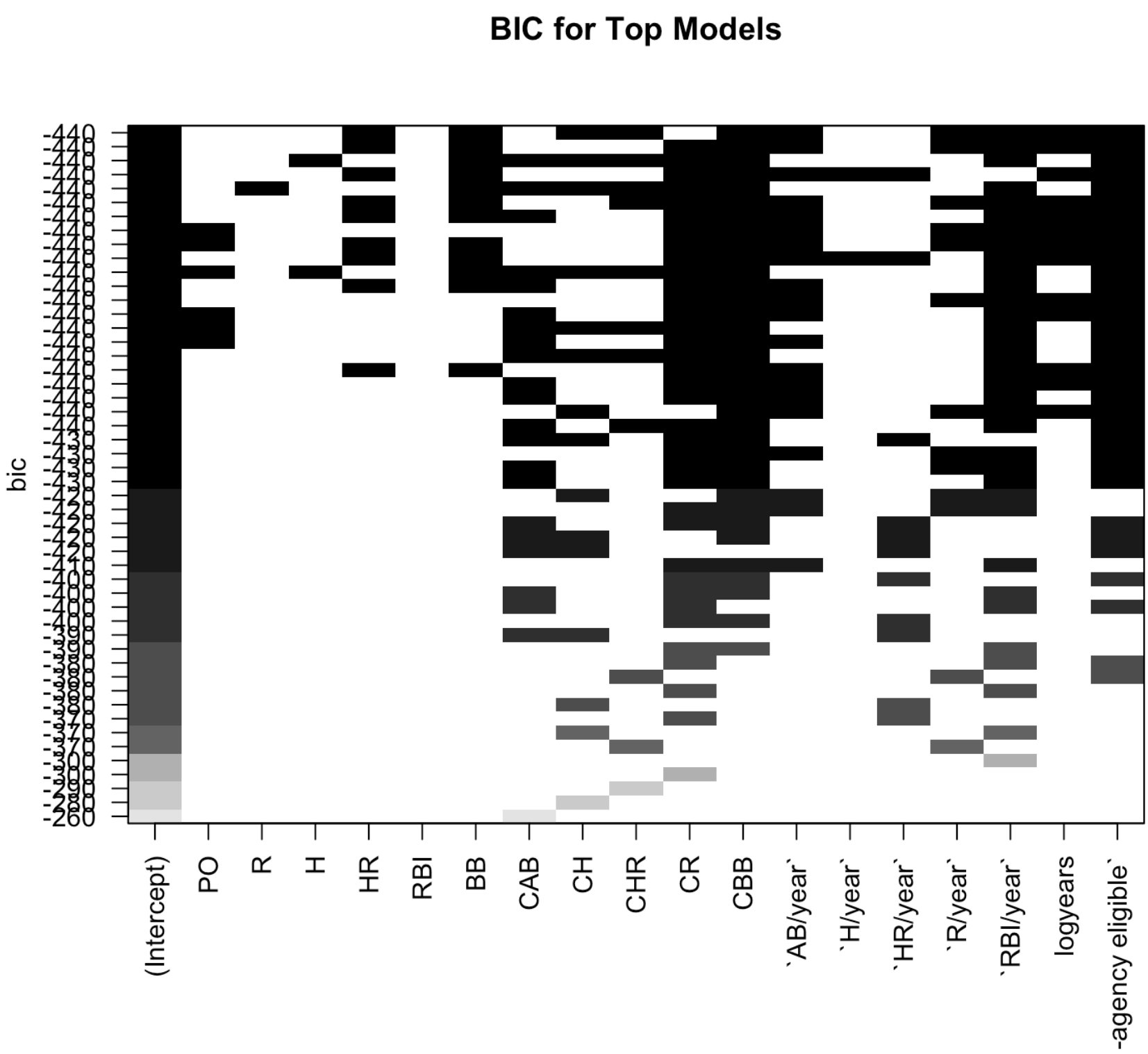
```
##                  PO                   A                   E
##                  20                   0                   0
##                  AB                   R                   H
##                   0                  20                   6
##                  HR                 RBI                  BB
##                  20                  18                  20
##                  SO                 IBB                 HBP
##                   2                   0                   0
##            logyears                 CAB                  CH
##                   2                  20                  20
##                 CHR                  CR                CRBI
##                  17                  11                   3
##                 CBB                  MI                   C
##                  20                   0                   0
##                  CF                 AVG          Career AVG
##                   0                   0                   0
##                 OBP          Career OBP            AB/year
##                   0                   0                   9
##              H/year              HR/year              R/year
##                   5                  11                  20
##           RBI/year Arbitration eligible Free-agency eligible
##                   9                   0                  17
##                Int2
##                  20
```

```
#Subset the dataset
small <- baseball[ ,c('salary', 'PO', 'R', 'H', 'HR', 'RBI', 'BB', 'CAB', 'CH', 'CHR'
, 'CR', 'CBB', 'AB/year', 'H/year', 'HR/year', 'R/year', 'RBI/year', 'logyears', 'Fre
e-agency eligible')]
```

The last table shows which of the regressors I'm considering are included in the 14-15 variable models. Since these seem to be the most important regressors, according to this method, I subsetted the dataset using these columns (that appear > 5 times) and perform further model selection on this basis. Although log(years) only shows up twice, I will include it as to lower my chances of violating the principle of marginality later on.

In summary, Mallows's Cp helped us narrow down the number of variables to consider; we now only have 19 regressor variables as opposed to 35 before. Next, we can use BIC as criteria to choose some of the top models. It turns out that most models produced while including `Int2` violate the principle of marginality, since `logyears` was not included in many of those models; thus I decided to remove `Int2` and rerun the process. From the BIC plot we find the best models near the top, and can tell from the presence of black boxes which variables are present. The final chart serves a similar purpose, containing the ordered top 12 models (ranked by lowest BIC), variables included, and BIC value for each model.

```
#Top five models of size up to 10 variables
subsets <- regsubsets(salary ~ ., data = small, nbest = 5, nvmax = 10, method = "exha
ustive")

#BIC plot
plot(subsets, scale = "bic", main = "BIC for Top Models")
```




```
#Top 12 models
indexes <- rank(summary(subsets)$bic) #indexes of ordered BIC, lowest -> highest
best <- summary(subsets)$which[indexes,] #reorder models based on BIC
top12 <- best[1:12,] #take top 12 best models

#Models, included variables, and BIC value
cbind(top12 - 0, BIC = sort(summary(subsets)$bic)[1:12])
```

```
##    (Intercept) PO R H HR RBI BB CAB CH CHR CR CBB `AB/year` `H/year`
## 10           1  0 0 0  1   0  1   0  1   1  0   1          1         0
## 10           1  0 0 0  1   0  1   0  0   1  1   1          1         0
## 10           1  1 0 0  1   0  1   0  0   0  1   1          1         0
## 10           1  0 0 0  1   0  1   0  0   0  1   1          1         1
## 10           1  1 0 1  0   0  1   1  1   1  1   1          0         0
## 9            1  0 0 0  1   0  1   0  0   0  1   1          1         0
## 9            1  0 0 1  0   0  1   1  1   1  1   1          0         0
## 9            1  0 0 0  1   0  1   0  0   0  1   1          1         1
## 9            1  0 1 0  0   0  1   1  1   1  1   1          0         0
## 9            1  0 0 0  1   0  1   1  0   0  1   1          1         0
## 8            1  1 0 0  0   0  0   0  0   0  1   1          1         0
## 8            1  0 0 0  1   0  1   1  0   0  1   1          1         0
##    `HR/year` `R/year` `RBI/year` logyears `Free-agency eligible`       BIC
## 10         0        1          1        1                      1 -444.8510
## 10         0        1          1        1                      1 -443.4295
## 10         0        1          1        1                      1 -442.9845
## 10         1        0          1        1                      1 -442.6342
## 10         0        0          1        0                      1 -442.6103
## 9          0        1          1        1                      1 -442.3230
## 9          0        0          1        0                      1 -442.0815
## 9          1        0          0        1                      1 -441.7948
## 9          0        0          1        0                      1 -441.3378
## 9          0        0          1        1                      1 -441.0109
## 8          0        1          1        1                      1 -440.9008
## 8          0        0          1        0                      1 -440.8074
```

## Final Selection

If the goal was prediction, I would favor a more complicated model. For model interpretability, I would favor a simpler model. I have decided to compromise by selecting the 9-variable model with lowest BIC, or the sixth row in our chart. **This model predicts player salary based on the number of home runs, walks, career runs, career walks, at bats per year, runs per year, runs batted in per year, logged career length ( `logyears` ), and free-agency eligibility.** A quick examination shows that all of these coefficients are, fortunately, significant, and this model explains away about 70% of variation in our data.

```
final_lm <- lm(salary ~ HR + BB + CR + CBB + `AB/year` + `R/year` + `RBI/year` + `log
years` + `Free-agency eligible`, data = baseball)
summary(final_lm)
```

```
##
## Call:
## lm(formula = salary ~ HR + BB + CR + CBB + `AB/year` + `R/year` +
##     `RBI/year` + logyears + `Free-agency eligible`, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9967041 -1279394    83481  1131042 13455580
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1158147     654359   1.770 0.077486 .
## HR                         -83499      25896  -3.224 0.001363 **
## BB                          39974      11036   3.622 0.000329 ***
## CR                          15645       1665   9.399  < 2e-16 ***
## CBB                        -13594       1803  -7.540 3.04e-13 ***
## `AB/year`                  -28790       3949  -7.290 1.60e-12 ***
## `R/year`                    84422      26315   3.208 0.001441 **
## `RBI/year`                 171856      18101   9.494  < 2e-16 ***
## logyears                 -1679110     471718  -3.560 0.000415 ***
## `Free-agency eligible`    2750801     502230   5.477 7.54e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2815000 on 411 degrees of freedom
## Multiple R-squared:  0.6978, Adjusted R-squared:  0.6912
## F-statistic: 105.5 on 9 and 411 DF,  p-value: < 2.2e-16
```

# Conclusion

In this project, we have built a predictive and descriptive model of baseball player's salaries based on their performance stats. We have identified unusual data points, found appropriate variable transformations, and considered interactions between terms. In our final model, the number of walks, career runs, runs per year, runs batted in per year, and free-agency eligibility all contributed positively towards player salary. Negative contributors included (sometimes counterintuitively) the number of home runs, career walks, at bats per year, and logged years in major leagues. This 9-variable model explained away around 70% of the variation in our data, which is an incredible amount considering that our original 35-variable model had explained away only around 81% of it!

# Credits

A special thanks to Professor Nolan's class lecture slides for inspiration on many plotting and model selection functions, and examples of how to use certain functions that are similarly implemented here sporadically (e.g. in `scatterplotMatrix()`, `corrplot()`, `leaps()`, etc.).