project

Yiwen Shi

12/3/2019

Load the dataset & Doing some visualizations

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.1     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   1.0.0     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
load("baseball2012.rda")
summary(baseball)
```

```
##       ID                yearID        teamID             lgID          
##  Length:627         Min.   :2012   Length:627         Length:627        
##  Class :character   1st Qu.:2012   Class :character   Class :character  
##  Mode  :character   Median :2012   Mode  :character   Mode  :character  
##                     Mean   :2012                                        
##                     3rd Qu.:2012                                        
##                     Max.   :2012                                        
##                     NA's   :206                                         
##   nameFirst           nameLast             salary         
##  Length:627         Length:627         Min.   :  480000  
##  Class :character   Class :character   1st Qu.:  490600  
##  Mode  :character   Mode  :character   Median : 1100000  
##                                        Mean   : 3683130  
##                                        3rd Qu.: 5075000  
##                                        Max.   :30000000  
##                                        NA's   :206       
##      POS                 G.x               GS            InnOuts      
##  Length:627         Min.   :  1.00   Min.   :  0.00   Min.   :   0.0  
##  Class :character   1st Qu.: 24.00   1st Qu.: 18.00   1st Qu.: 482.5  
##  Mode  :character   Median : 64.00   Median : 50.00   Median :1367.0  
##                     Mean   : 71.36   Mean   : 62.01   Mean   :1659.5  
```

```
##                    3rd Qu.:117.00   3rd Qu.:103.50   3rd Qu.:2739.5
##                    Max.   :181.00   Max.   :161.00   Max.   :4374.0
##
##        PO                A                 E                DP
##  Min.   :   0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0
##  1st Qu.:  40.0   1st Qu.:  3.00   1st Qu.: 1.00   1st Qu.:  1.0
##  Median : 113.0   Median : 17.00   Median : 3.00   Median :  4.0
##  Mean   : 203.3   Mean   : 69.03   Mean   : 4.14   Mean   : 17.8
##  3rd Qu.: 257.0   3rd Qu.: 76.50   3rd Qu.: 6.00   3rd Qu.: 20.0
##  Max.   :1295.0   Max.   :529.00   Max.   :27.00   Max.   :135.0
##
##       G.y           G_batting          AB              R
##  Min.   :  1.00   Min.   : NA   Min.   :  1.0   Min.   :  0.00
##  1st Qu.: 34.00   1st Qu.: NA   1st Qu.: 81.0   1st Qu.:  8.00
##  Median : 77.00   Median : NA   Median :205.0   Median : 24.00
##  Mean   : 79.82   Mean   :NaN   Mean   :255.1   Mean   : 33.04
##  3rd Qu.:128.50   3rd Qu.: NA   3rd Qu.:422.0   3rd Qu.: 55.00
##  Max.   :162.00   Max.   : NA   Max.   :683.0   Max.   :129.00
##                   NA's   :627
##        H               X2B              X3B               HR
##  Min.   :  0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 18.00   1st Qu.: 3.00   1st Qu.: 0.000   1st Qu.: 1.000
##  Median : 49.00   Median :10.00   Median : 1.000   Median : 4.000
##  Mean   : 65.96   Mean   :13.01   Mean   : 1.463   Mean   : 7.812
##  3rd Qu.:108.00   3rd Qu.:20.00   3rd Qu.: 2.000   3rd Qu.:12.000
##  Max.   :216.00   Max.   :51.00   Max.   :15.000   Max.   :44.000
##
##       RBI               SB               CS               BB
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.000   Min.   :  0.00
##  1st Qu.:  7.00   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:  5.00
##  Median : 22.00   Median : 1.000   Median : 1.000   Median : 16.00
##  Mean   : 31.44   Mean   : 5.145   Mean   : 1.805   Mean   : 23.11
##  3rd Qu.: 52.50   3rd Qu.: 6.000   3rd Qu.: 3.000   3rd Qu.: 36.00
##  Max.   :139.00   Max.   :49.000   Max.   :13.000   Max.   :105.00
##
##        SO              IBB              HBP              SH
##  Min.   :  0.0   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 17.0   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 46.0   Median : 1.000   Median : 1.000   Median : 0.000
##  Mean   : 54.5   Mean   : 1.679   Mean   : 2.348   Mean   : 1.357
##  3rd Qu.: 84.0   3rd Qu.: 2.000   3rd Qu.: 4.000   3rd Qu.: 2.000
##  Max.   :222.0   Max.   :18.000   Max.   :17.000   Max.   :17.000
##
##        SF              GIDP             years            CAB
##  Min.   : 0.000   Min.   : 0.000   Min.   : 1.000   Min.   :    1
##  1st Qu.: 0.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.:  233
##  Median : 1.000   Median : 4.000   Median : 4.000   Median :  872
##  Mean   : 1.915   Mean   : 5.657   Mean   : 5.499   Mean   : 1681
##  3rd Qu.: 3.000   3rd Qu.: 9.000   3rd Qu.: 8.000   3rd Qu.: 2566
##  Max.   :12.000   Max.   :28.000   Max.   :24.000   Max.   :10586
```

```
##
##        CH              CHR               CR              CRBI
##  Min.   :   0.0   Min.   :  0.00   Min.   :   0.0   Min.   :   0.0
##  1st Qu.:  53.5   1st Qu.:  3.00   1st Qu.:  25.0   1st Qu.:  25.0
##  Median : 224.0   Median : 18.00   Median : 110.0   Median :  95.0
##  Mean   : 454.3   Mean   : 54.29   Mean   : 238.1   Mean   : 222.7
##  3rd Qu.: 689.5   3rd Qu.: 69.50   3rd Qu.: 356.5   3rd Qu.: 312.0
##  Max.   :3304.0   Max.   :647.00   Max.   :1898.0   Max.   :1950.0
##
##       CBB
##  Min.   :   0.0
##  1st Qu.:  16.0
##  Median :  74.0
##  Mean   : 165.4
##  3rd Qu.: 226.5
##  Max.   :1747.0
##
```

```
names(baseball)
```

```
##  [1] "ID"        "yearID"    "teamID"    "lgID"      "nameFirst"
##  [6] "nameLast"  "salary"    "POS"       "G.x"       "GS"
## [11] "InnOuts"   "PO"        "A"         "E"         "DP"
## [16] "G.y"       "G_batting" "AB"        "R"         "H"
## [21] "X2B"       "X3B"       "HR"        "RBI"       "SB"
## [26] "CS"        "BB"        "SO"        "IBB"       "HBP"
## [31] "SH"        "SF"        "GIDP"      "years"     "CAB"
## [36] "CH"        "CHR"       "CR"        "CRBI"      "CBB"
```

Here are some things you should do to explore the data and prepare for further analysis. You should do every step here but this is not the main part of the analysis and you will not be graded in detail. 1. Create the variables that Fox creates for his analysis. (Note that it is not possible to create DH from this data.) the player's 1986 and career batting average (AVG—i.e., number of hits divided by number of at-bats), (OBP = 100 · [hits+walks]/[at-bats+walks] at-bats, hits, home runs, runs scored, and runs batted per year

```
#add the average variable
AVG = baseball$CH/baseball$AB
baseball$AVG = AVG
#add obp
OBP = 100 * ((baseball$H + baseball$BB)/(baseball$AB + baseball$BB))
baseball$OBP = OBP
CHRperYear = baseball$CHR/baseball$years
baseball$CHRperYear = CHRperYear
CAB_avg = baseball$CAB/baseball$years
CH_avg = baseball$CH/baseball$years
CHR_avg = baseball$CHR/baseball$years
R_avg = baseball$R/baseball$years
RBI_avg=baseball$RBI/baseball$years
```

```
#baseball$CAB_avg = CAB_avg
#baseball$CH_avg = CH_avg
#baseball$CHR_avg = CHR_avg
#baseball$R_avg  = R_avg
#baseball$RBI_avg = RBI_avg
#add dummy variables
SS2BDummy = c()
for (i in 1:length(baseball$POS)) {
  if (baseball$POS[i] == "SS"|baseball$POS[i] == "2B") {
    SS2BDummy = c(SS2BDummy, 1)
  } else {
    SS2BDummy = c(SS2BDummy, 0)
  }
}
baseball$SS2BDummy = SS2BDummy
CenterDummy = c()
for (i in 1:length(baseball$POS)) {
  if (baseball$POS[i] == "C") {
    CenterDummy = c(CenterDummy, 1)
  } else {
    CenterDummy = c(CenterDummy, 0)
  }
}
baseball$CenterDummy = CenterDummy
CFDummy = c()
for (i in 1:length(baseball$POS)) {
  if (baseball$POS[i] == "CF") {
    CFDummy= c(CFDummy, 1)
  } else {
    CFDummy = c(CFDummy, 0)
  }
}
baseball$CFDummy = CFDummy
ThreeFive = c()
for (i in 1:length(baseball$years)) {
  if (baseball$years[i] >= 3 & baseball$years[i] <=5) {
    ThreeFive = c(ThreeFive, 1)
  } else {
    ThreeFive = c(ThreeFive, 0)
  }
}
#dummy vairables for experiences
baseball$ThreeFive = ThreeFive
SixLong = c()
for (i in 1:length(baseball$years)) {
  if (baseball$years[i] >= 6) {
    SixLong = c(SixLong, 1)
  } else {
    SixLong = c(SixLong, 0)
  }
```

```
}
baseball$SixLong = SixLong
```

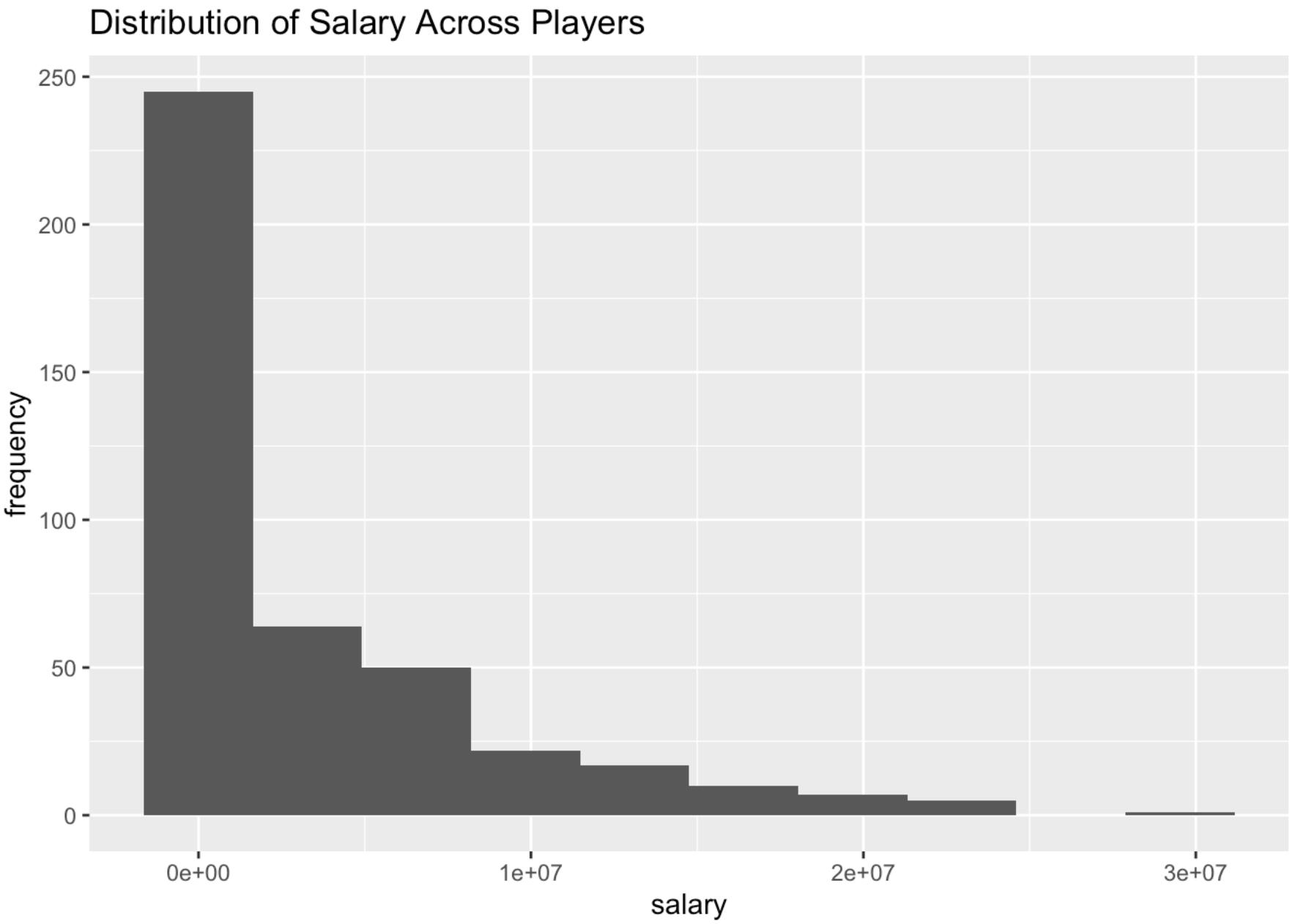
coded 1 for players who consistently played second base or shortstop (i.e., middle infielders, MI), catcher (C), center field (CF), or designated hitter (DH)

2. Make 2-4 plots to reveal interesting structure in the data. Label axes, consider using color and plotting characters to make the plots more informative.

a. The distribution of salary

```
library(ggplot2)
# I am interested in the distribution of salary across the dataset
#there are na values in the dataset, remove those
ggplot(baseball) + geom_histogram(aes(x=salary), bins=10) + xlab("salary") + ylab("fr
equency") +
  ggtitle("Distribution of Salary Across Players")
```

```
## Warning: Removed 206 rows containing non-finite values (stat_bin).
```

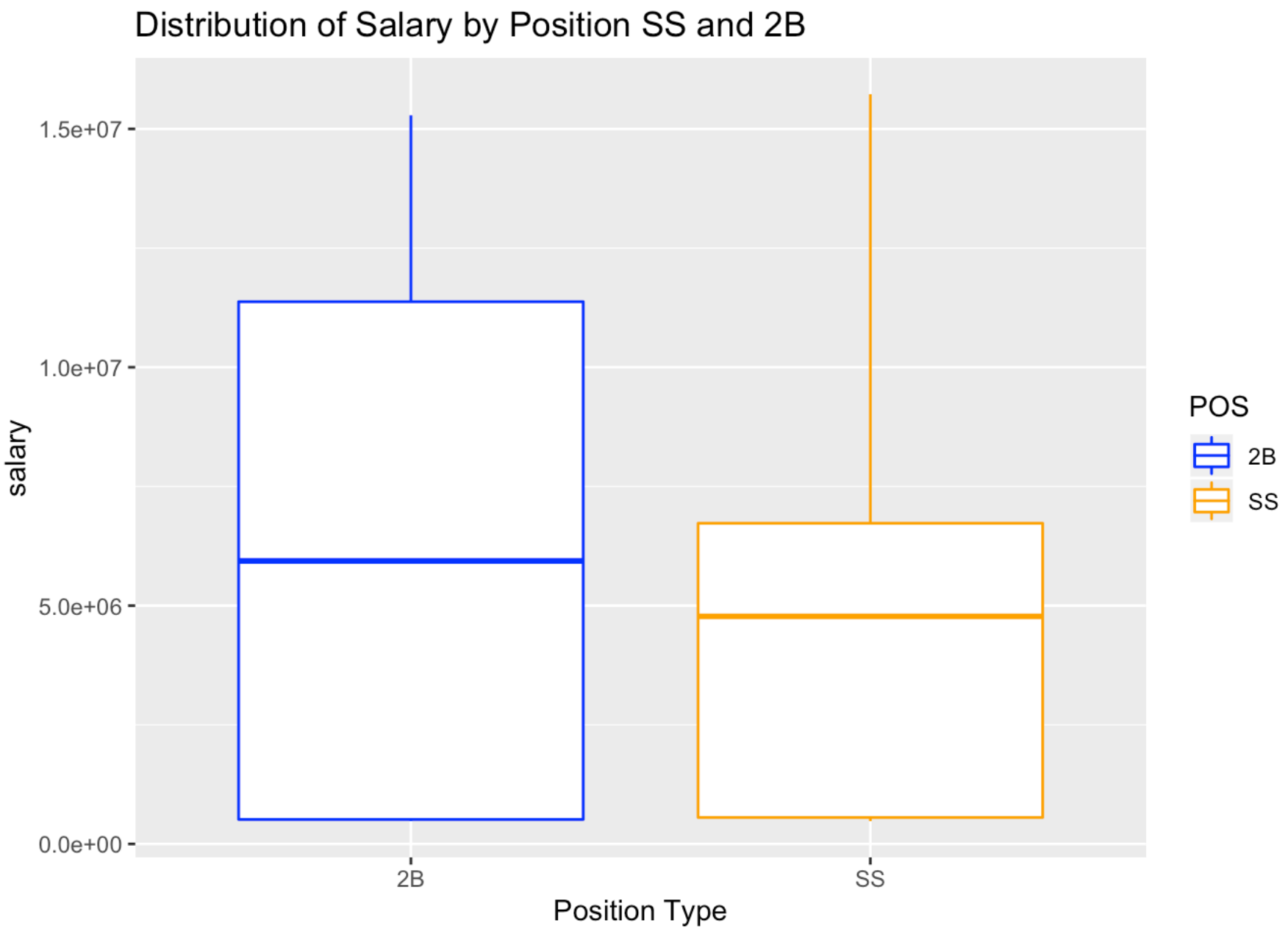


From this histogram of the salaries, we can see that the salary distribution is pretty skewed to the right. This is a tiny bin of very high salary to the right, separated from the other, it might be interesting to explore the reason for differences among players.

b. A boxplot of salary by positions, since Fox mentioned that he combined the dummy variables, it might be interesting to see if they have similar distributions of salary.

```
sub = baseball[baseball$POS == "SS"|baseball$POS == "2B",]
ggplot(sub) + geom_boxplot(aes(x=POS, y=salary, color = POS)) + scale_color_manual(va
lues=c("blue","orange")) + xlab("Position Type") + ggtitle("Distribution of Salary by
Position SS and 2B")
```

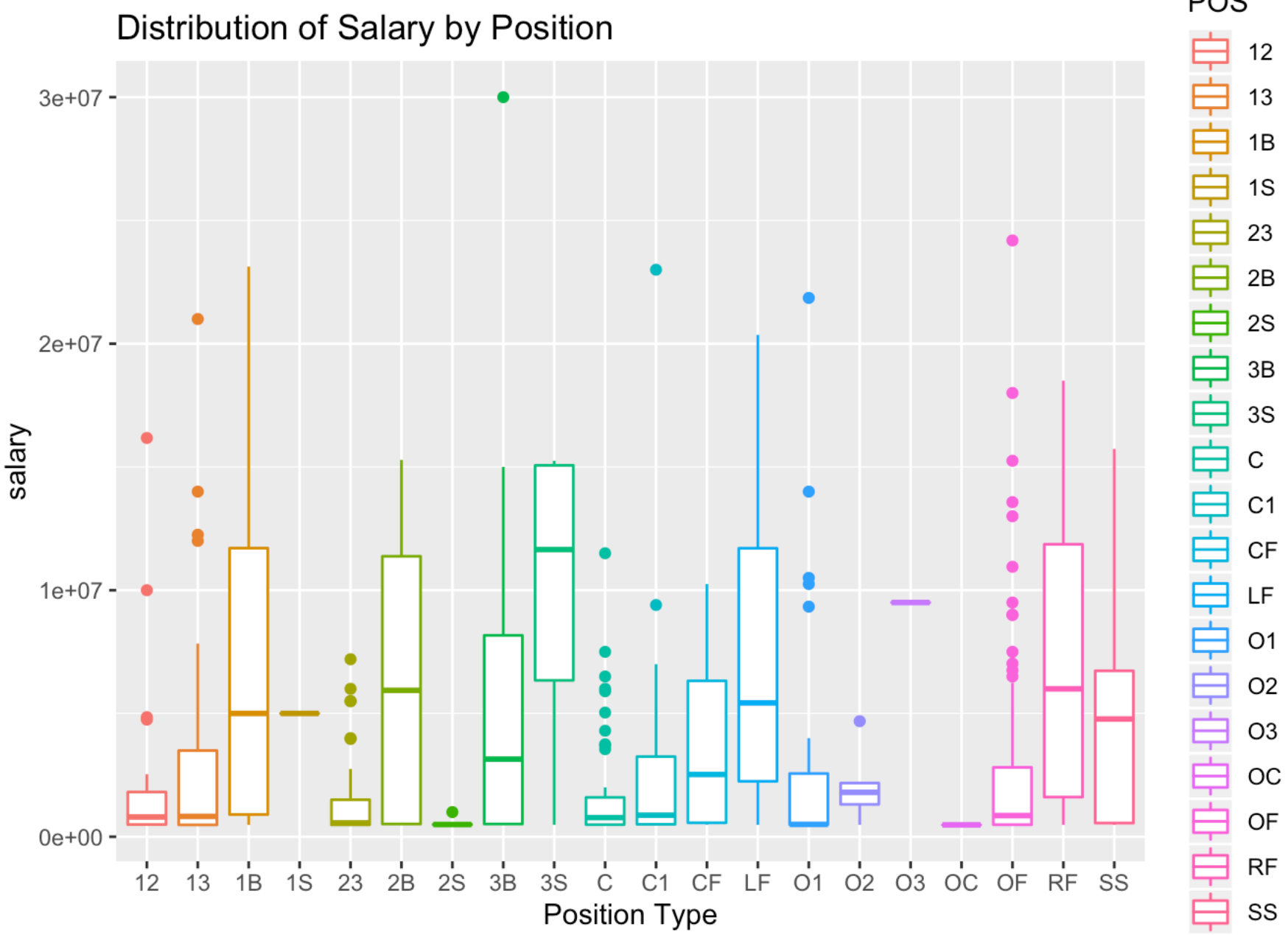
```
## Warning: Removed 16 rows containing non-finite values (stat_boxplot).
```



Their salary is different but not significantly different, so it is reasonaly to classify them together as MI, mentioned in the book.

```
#A boxplot of every position.
ggplot(baseball) + geom_boxplot(aes(x=POS, y=salary, color = POS)) +xlab("Position Ty
pe") + ggtitle("Distribution of Salary by Position")
```

```
## Warning: Removed 206 rows containing non-finite values (stat_boxplot).
```



Insteresting structure: It seems that the variance is very different among positions, and some positions were represented by a thick line, maybe the number of players in that position is very low.

c. Want to see if this dataset follows the intuition that more experiences lead to more money.

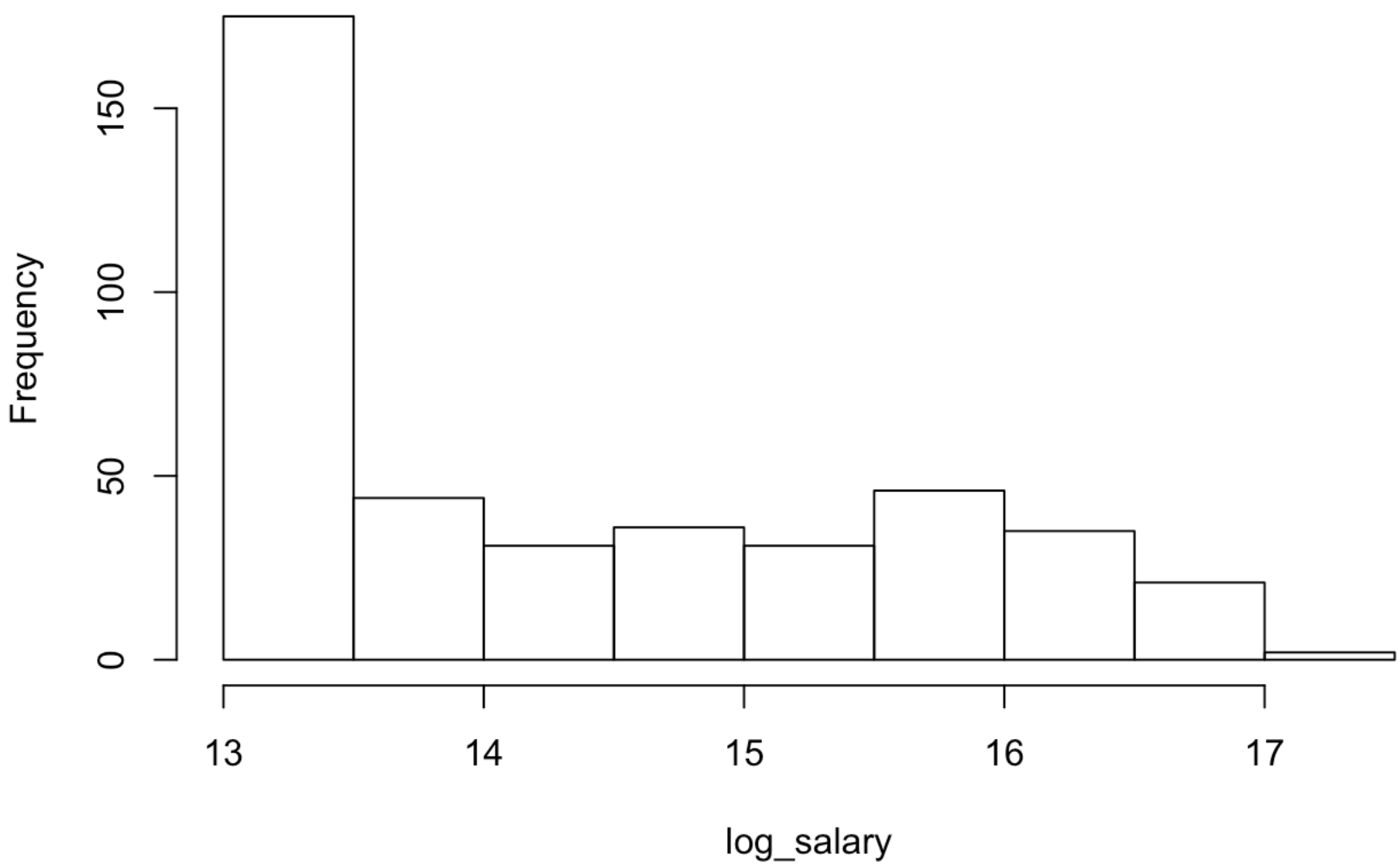
```
ggplot(baseball) + geom_point(aes(x=years, y=salary)) + facet_wrap(~POS) + xlab("Expe
rience") +
  ggtitle("Salary vs. Experience by Position")
```

```
## Warning: Removed 206 rows containing missing values (geom_point).
```

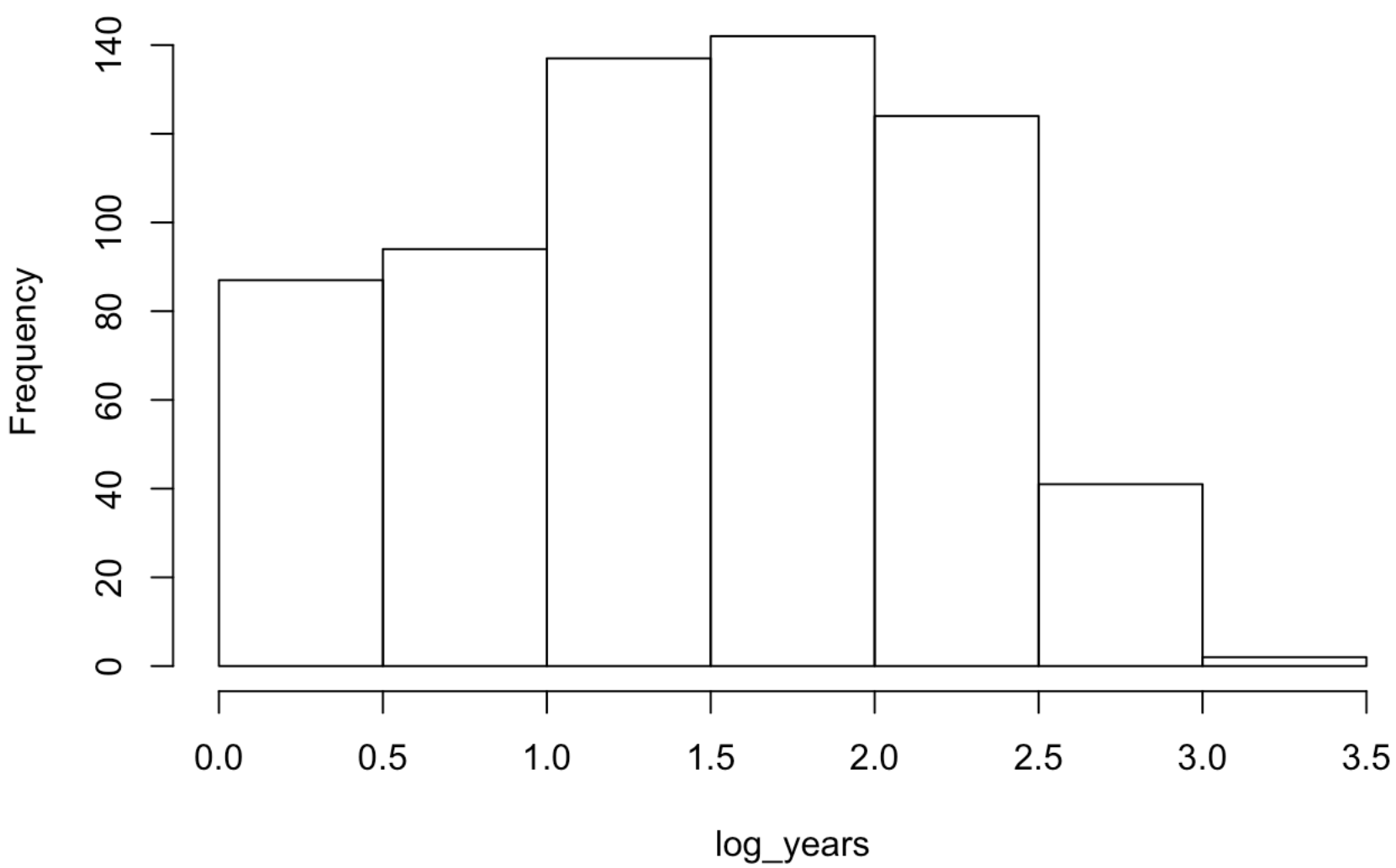
From this multiple-groups scatter plot, we cans see that most of them show a positive linear association between length of experience and salary (some are not significant, this is possibly due to the scale of y axis), although some plots may have points more sparsely distributed, some positions only have one point ploted, corresponding the the thick bar/line in the above boxplot.

3. Consider transformations to symmetrize distributions and stabilize variance. From the previous section we see the distribution of salary is very skewed, so a log transformation of the data would probably make it a little better. Also Fox suggests transforming years and CAB using log. They all seem more symmetric after the transformation

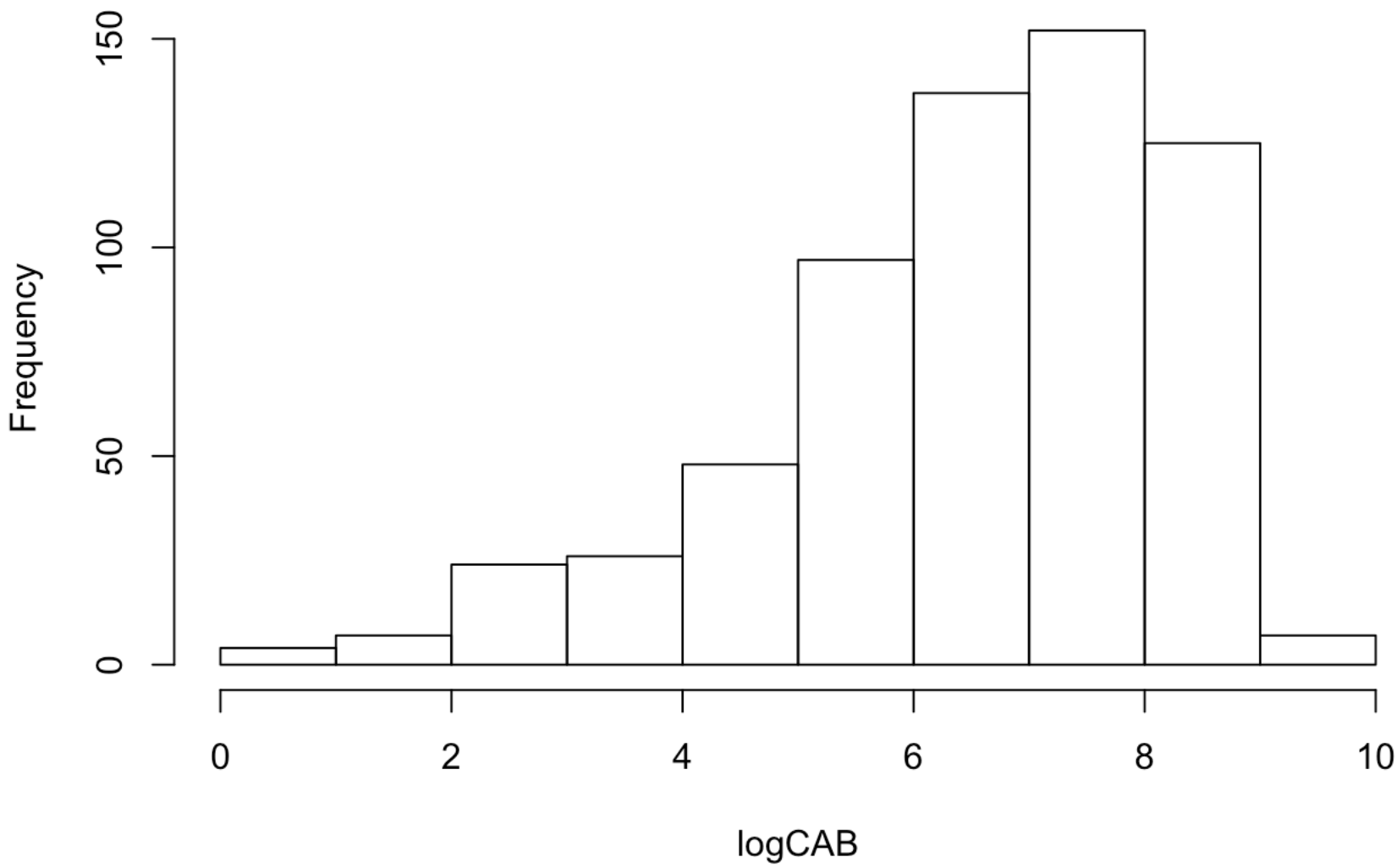
```
#log transformation, NA values excluded in plot
log_salary = log(baseball$salary)
#Fox also suggests log the years and CAB
log_years = log(baseball$years)
logCAB = log(baseball$CAB)
hist(log_salary)
```




```
hist(log_years)
```


Histogram of log_years
Frequency
0
20
40
60
80
100
140
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
log_years

```
hist(logCAB)
```



Stablize variance: log transformation would usually stablize variance.And sin the distribution is right skewed, a log would make it better. We could also standardize quantitative variables, avoiding different weights 4. Clean the data: remove records with no salary information.

```
cleaned = baseball[!is.na(baseball$salary),]
nrow(cleaned)
```

```
## [1] 421
```

```
#cleaned is the table with NAs in salary removed
```

Data Analysis

1. Fit a simple model that predicts salary (logged) from the length of career (indicator for 3-5 and 6+ years), career runs (logged), allowing for an interaction between length of career and career runs.

```
#add log salary and log career runs to table, need to add 1 in CR since log0 is undef
ined
logSalary = log(cleaned$salary)
cleaned$logSalary = logSalary
logCR = log(cleaned$CR+1)
cleaned$logCR = logCR
model1 = lm(logSalary~years+logCR+years:logCR, cleaned)
summary(model1)
```

```
##
## Call:
## lm(formula = logSalary ~ years + logCR + years:logCR, data = cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.03151 -0.57915  0.03508  0.61142  2.35916
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.913748   0.286687  38.068  < 2e-16 ***
## years       -0.186232   0.066468  -2.802  0.00532 **
## logCR        0.655843   0.058952  11.125  < 2e-16 ***
## years:logCR  0.030594   0.009757   3.136  0.00184 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.82 on 417 degrees of freedom
## Multiple R-squared:  0.577,  Adjusted R-squared:  0.574
## F-statistic: 189.6 on 3 and 417 DF,  p-value: < 2.2e-16
```

From this model1, we see the interaction term and years are statistically significant, and intercept and logCR is very statistically significant, p-value of F test is also extremely small. Fo years and the interaction term, they are also significant although less so than the other two, the significance of the interaction term indicates the existence of a strong interaction between CR and length of experiences.

2. For this simple model, check for outliers, leverage points, and influential observations. (Note that Pete Rose was banned from baseball so his record will not be in the 2012 dataset.)

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
outlierTest(model1)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 599 -3.945652         9.3401e-05     0.039322
```

```
order(hatvalues(model1), decreasing = T)[1:10]
```

```
##  [1] 399 398 379 414 330 195 203 303 156 217
```

```
order(cooks.distance(model1), decreasing = T)[1:10]
```

```
##  [1] 398 399 379 414 140 303 185 202  72 300
```

Using the outlierTest function. we found one outlier,its unadjusted p-value is very small but with Bonferroni adjustment, the p value is timed by the 2*number of observations,but still for this data point it is significant, which means this is detected as an outlier. The index 599 corresponds to the id “vizquom01”, it corresponds to the 398th observation (599 is from the non-cleaned table).

For leverages, the hatvalues function outputs the hat values of all observations, and I order them in descending order, and here I extracted first 10 largest hat values, 398th observation has the second largest hat value, since it is the most significant outlier, it’s cook distance is probablity significantly bigger.

For influence I used the cook distance and got the first 10 largest, not surprisingly, 398th observation has the largest cook distance, being the most influential point.

3. Fit a linear least-squares regression of log salary on all the explanatory variables (Fox has 32 and you should have roughly that number). Notice that many of the explanatory variables are highly correlated.

I need to get rid of the non-numeric columns inclusing the first few, as well as separate the log salary column

```
y = cleaned$logSalary
data = cleaned[, 9:50] #take out the first 9 columns first, need to take out salary s
ince logged
data = data %>% select(-G_batting,  -CR) #G_batting is all NA, take out CR since ther
e is a log version
#Also can take out years, since we have the dummy variables in terms of experience;
#i did not include some of the variables generated in step 1, to avoid #potential hig
h correlation
data = data%>%select(-years)
#fit the rest variables
model2 = lm(logSalary~., data=data)
summary(model2)
```

```
##
## Call:
## lm(formula = logSalary ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82370 -0.35111  0.04968  0.33802  2.11146
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.2590743  0.2785923  47.593  < 2e-16 ***
## G.x          0.0038467  0.0045800   0.840 0.401499
## GS          -0.0066266  0.0111877  -0.592 0.553995
## InnOuts      0.0001584  0.0004920   0.322 0.747703
## PO           0.0001035  0.0002684   0.386 0.699983
## A            0.0001220  0.0008110   0.150 0.880544
## E            0.0001807  0.0106364   0.017 0.986453
## DP          -0.0033107  0.0025357  -1.306 0.192458
## G.y         -0.0133441  0.0033222  -4.017 7.11e-05 ***
## AB           0.0043850  0.0017307   2.534 0.011685 *
## R            0.0025572  0.0056434   0.453 0.650716
## H           -0.0076516  0.0052149  -1.467 0.143131
## X2B         -0.0055783  0.0070972  -0.786 0.432364
## X3B          0.0071079  0.0187405   0.379 0.704690
## HR          -0.0351251  0.0116177  -3.023 0.002668 **
## RBI          0.0122394  0.0054016   2.266 0.024017 *
## SB           0.0009059  0.0059622   0.152 0.879312
## CS          -0.0140923  0.0185306  -0.760 0.447431
## BB           0.0075864  0.0038323   1.980 0.048464 *
## SO          -0.0019215  0.0019354  -0.993 0.321414
## IBB          0.0117972  0.0136930   0.862 0.389476
## HBP          0.0075233  0.0111836   0.673 0.501537
## SH           0.0153087  0.0160753   0.952 0.341541
## SF           0.0065555  0.0206675   0.317 0.751273
## GIDP        -0.0048112  0.0093787  -0.513 0.608251
## CAB         -0.0002792  0.0002035  -1.372 0.170849
```

```
## CH            0.0017015  0.0007115   2.391 0.017273 *
## CHR          -0.0013620  0.0022425  -0.607 0.543967
## CRBI          0.0005341  0.0010428   0.512 0.608802
## CBB          -0.0008165  0.0004117  -1.983 0.048030 *
## AVG          -0.0380510  0.0108369  -3.511 0.000499 ***
## OBP          -0.0031172  0.0098266  -0.317 0.751250
## CHRperYear    0.0660203  0.0126006   5.239 2.67e-07 ***
## SS2BDummy     0.3134305  0.1407042   2.228 0.026490 *
## CenterDummy -0.1312835  0.1361999  -0.964 0.335704
## CFDummy       0.1320943  0.1459543   0.905 0.366015
## ThreeFive     0.0861373  0.1192145   0.723 0.470404
## SixLong       1.3033291  0.1637208   7.961 1.97e-14 ***
## logCR        -0.0570574  0.0899396  -0.634 0.526202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5917 on 382 degrees of freedom
## Multiple R-squared:  0.7983, Adjusted R-squared:  0.7782
## F-statistic: 39.77 on 38 and 382 DF,  p-value: < 2.2e-16
```

From the model, we see the intercept, # games played at any position, and the dummy variable indicating 6 years or more of experience are very sitatistically significant; AB, HR, RBI, CBB and CHR per year of experience are statistically significant, F test has a very small p value, indicating the slopes are not zero.

4. Find the 10 best models for each model size using the regsubsets() function (leaps package).

```
library(leaps)
#design matrix X
X = data %>%select(-logSalary)
regSubset = regsubsets(X, y, nbest = 10, nvmax=25)
subset = summary(regSubset)
#subset is the result of the best 10 models, the variable with "*" indicates that it
is selected in the corresponding model, did not print to save space.
```

5. Use BIC to select the best 5 models from the previous step (of varying sizes, but some may be the same size). Rank these models by BIC. We want lowest values of BIC, so I used order function to get the index ot first five smallest models

```
as.vector(order(subset$bic)[1:5])
```

```
## [1] 81 91 92 71 51
```

```
subset$which[as.vector(order(subset$bic)[1:5]),]
```

```
##    (Intercept)   G.x    GS InnOuts    PO     A     E    DP  G.y   AB     R
## 9         TRUE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
## 10        TRUE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
## 10        TRUE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
## 8         TRUE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
## 6         TRUE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE
##        H   X2B   X3B    HR   RBI    SB    CS    BB    SO   IBB   HBP    SH
## 9  FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 10 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 10 FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 8  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 6  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##       SF  GIDP   CAB   CH   CHR  CRBI   CBB  AVG   OBP CHRperYear
## 9  FALSE FALSE FALSE TRUE FALSE FALSE  TRUE TRUE FALSE       TRUE
## 10 FALSE FALSE FALSE TRUE FALSE FALSE  TRUE TRUE FALSE       TRUE
## 10 FALSE FALSE  TRUE TRUE FALSE FALSE  TRUE TRUE FALSE       TRUE
## 8  FALSE FALSE FALSE TRUE FALSE FALSE  TRUE TRUE FALSE       TRUE
## 6  FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE       TRUE
##    SS2BDummy CenterDummy CFDummy ThreeFive SixLong logCR
## 9      FALSE       FALSE   FALSE     FALSE    TRUE FALSE
## 10     FALSE       FALSE   FALSE     FALSE    TRUE FALSE
## 10     FALSE       FALSE   FALSE     FALSE    TRUE FALSE
## 8      FALSE       FALSE   FALSE     FALSE    TRUE FALSE
## 6      FALSE       FALSE   FALSE     FALSE    TRUE FALSE
```

In order, these five models are the ones with best BIC

- First model Index 81 (the best one) has a size of 9 (excluding intercept): #Games in any position, # At bats, # Home runs, # Walks, # Career hits, # Career Walks, #Career Home Runs per year of experience, # Career batting AVG and dummy variable indicating experience longer than six years.
- Second model Index 91 has a size of 10 (excluding intercept): #Games in any position, # At bats, # Home runs, # Walks, # Career hits, # Career Walks,#Runs Batted In, #Career Home Runs per year of experience, # Career batting AVGand dummy variable indicating experience longer than six years.
- Third model index 92 has a size of 10 (excluding intercept): #Games in any position, # At bats, # Walks, # Career hits, # Career Walks, #Career Home Runs per year of experience,# Career batting AVG, and dummy variable indicating experience longer than six years..
- Fourth model index 71 has a size of 8 (excluding intercept): #Games in any position, # At bats, # Walks, # Career hits, # Career Walks, # Career batting AVG, #Career Home Runs per year of experience and dummy variable indicating experience longer than six years.
- Fifth model Index 51 has a size of 6 (excluding intercept): #Games in any position, # At bats, # Career hits, #Career Home Runs per year of experience,# Career batting AVG, and dummy variable indicating experience longer than six years, log of Career Runs.

Conclusion: All of the models include the experience dummy variable, which means length of experience plays a significant role and has a positive association with salary. Number of games played in any postion is also strongly associated, and there might be correlation between experience and number of games since longer

experience should usually result in more games. Home runs also are important, especially CHR averaged per year. Career batting average are also true for every model, so averaging over the years might represent the variable better than non-avaraged.

6. Use cross validation (formula (22.2) on page 673 of Fox) to also rank these 5 models. Rather than doing the leave one out method, do 10-fold cross validation.

```
#randomly permute indices and then break into subsets for CV
permutation <- sample(1:nrow(data))
folds <- c(rep(1:10, each = nrow(data)/10),10)
design = model.matrix(logSalary~., data)
avg_test_MSE <- rep(0,5)
#loop over models of each size
for(i in 1:5){
  modelIndex = order(subset$bic)[i]
  test_MSE <- rep(0,10)
  #loop over folds
    for(j in 1:10){
      #identify training and test sets
      idx_train <- permutation[folds != j]
      idx_test <- permutation[folds == j]
      #extract which variables to use from regfit
      vars <- as.vector(subset$which[modelIndex,])
      X_best_subset <- design[,vars]
      mod <- lm(data$logSalary ~ X_best_subset-1, subset = idx_train)
      X_test <- X_best_subset[idx_test,]
      test_predictions <- X_test %*% as.matrix(coef(mod))
      test_MSE[j] <- mean((data$logSalary[idx_test] - test_predictions)^2)
    }
  avg_test_MSE[i] <- mean(test_MSE)
}
avg_test_MSE
```

```
## [1] 0.3573567 0.3584094 0.3573392 0.3653530 0.3732628
```

```
test_errors = data.frame(modelIndex=as.vector(order(subset$bic)[1:5]), CVerrors =avg_
test_MSE )
test_errors = test_errors[order(test_errors$CVerrors),]
test_errors
```

```
##   modelIndex  CVerrors
## 3         92 0.3573392
## 1         81 0.3573567
## 2         91 0.3584094
## 4         71 0.3653530
## 5         51 0.3732628
```

In the above code chunk, I used cross validation procedure introduced in the lab section and creat an data frame containing the model index ($which in summary of reg subsets) and the corresponding errors. I rank them by order the data frame in ascending order of error. The lowest error model appears to be #91, corresponding to the model of size 9 with: #Games in any position, # At bats, # Home runs, # Walks, # Career hits, # Career Walks,#Runs Batted In, #Career Home Runs per year of experience, # Career batting AVG and dummy variable indicating experience longer than six years. The full order is model 91, 92, 81, 71, 51.

7. Use LASSO to fit the model instead, using all the same variables (roughly 32) as before. Produce a plot like the left panel of Fig 6.13 on page 229 of ISLR. Choose a suitable lambda and further analyze this fit. Which variables end up with non-zero coefficients? Compare to the CV MSE you calculated in step 6 for the 5 good models you chose before. How did LASSO do compared to these?

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

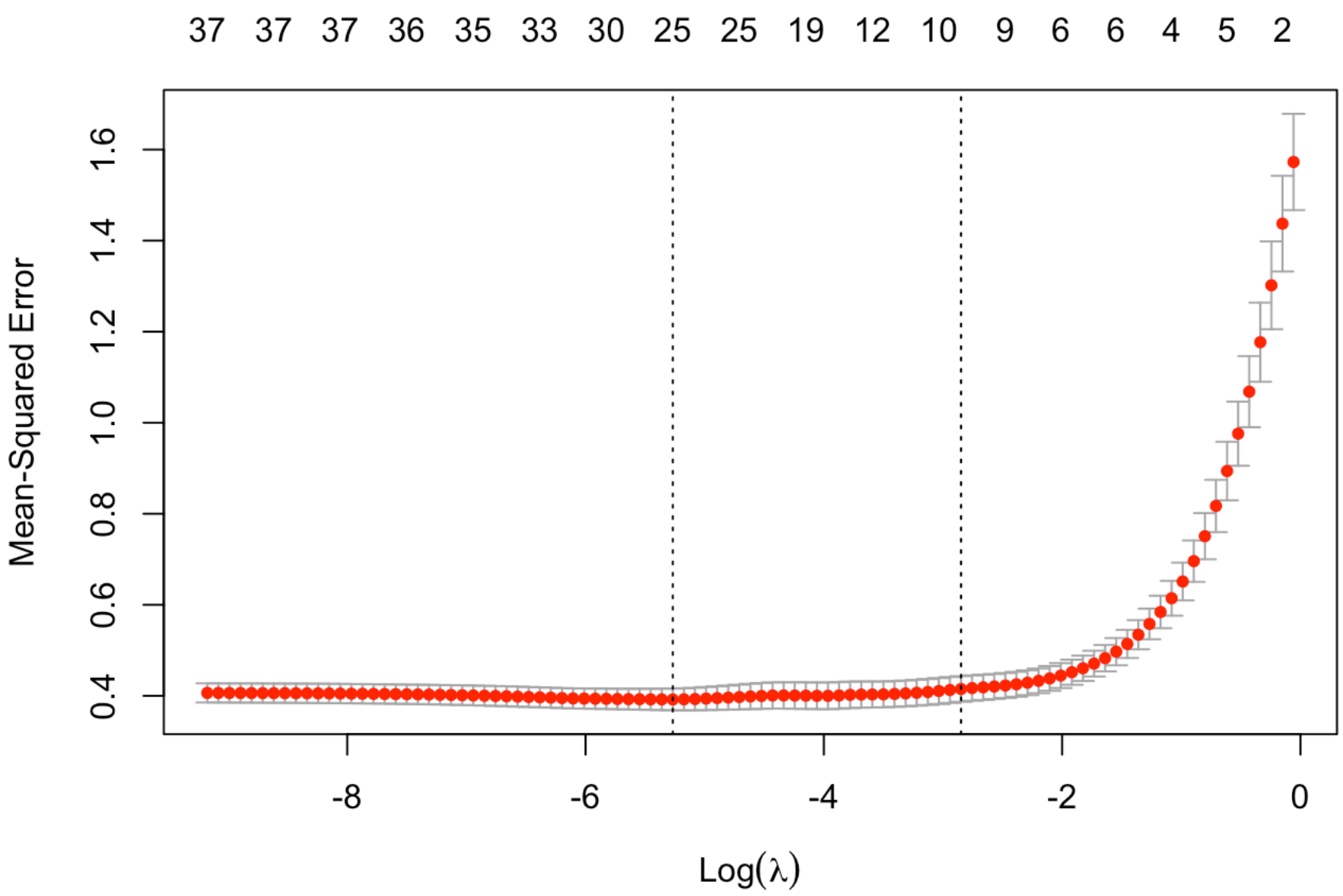
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 3.0-1
```

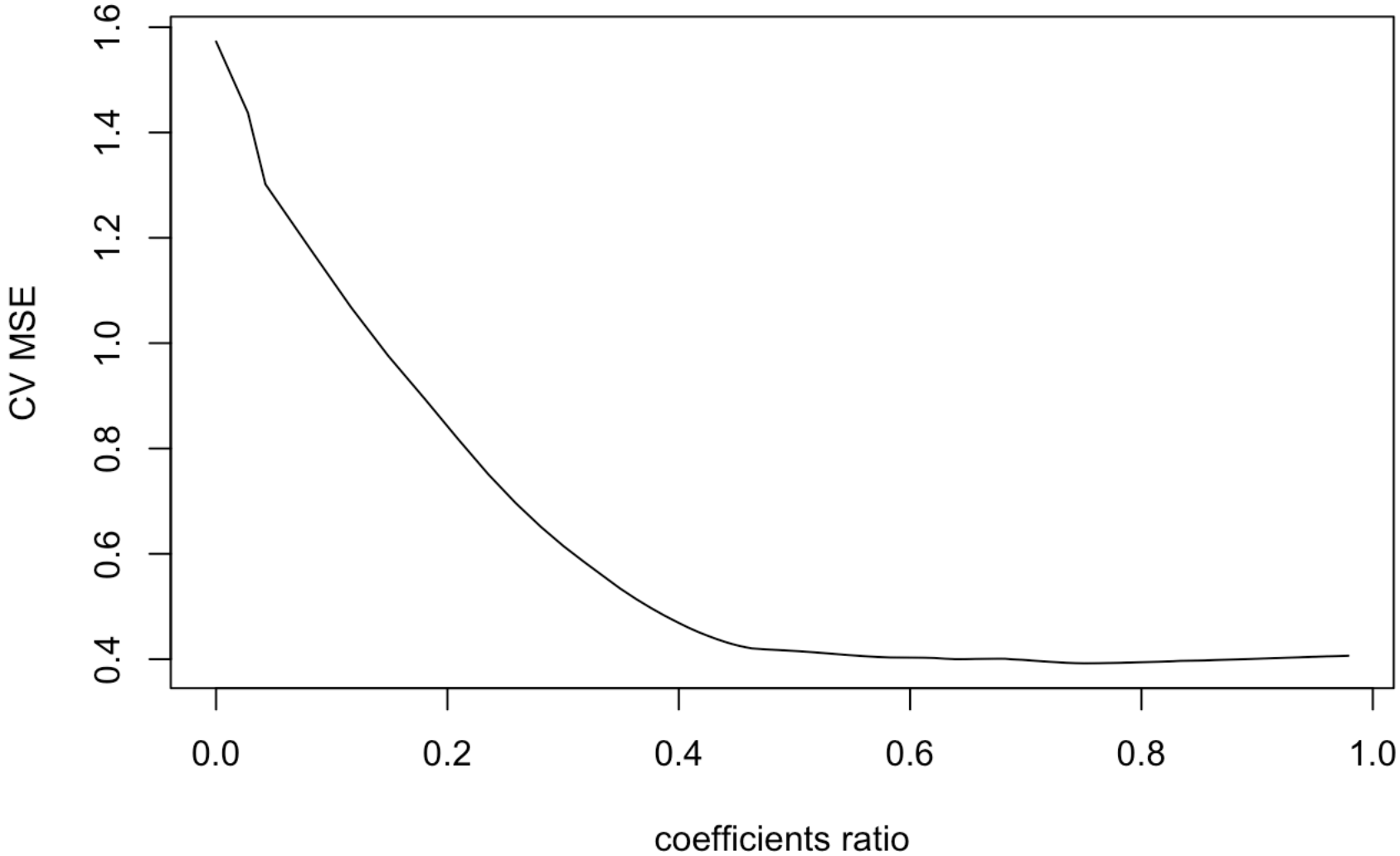
```
X_matrix = design[,-1]
y_vector = data$logSalary
nrow(X_matrix) ==length(y_vector)
```

```
## [1] TRUE
```

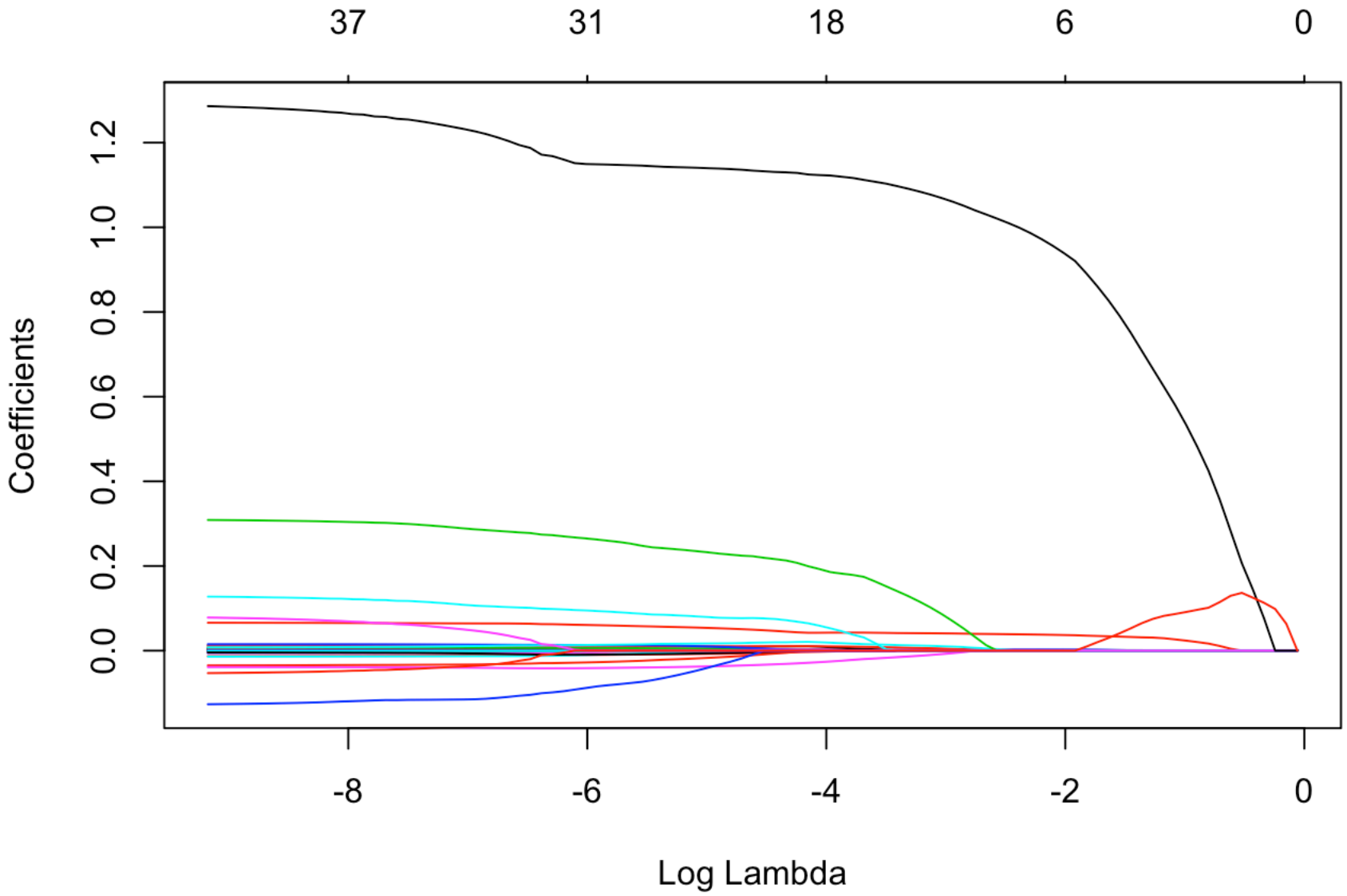
```
lasso.mod <- glmnet(x = X_matrix, y = y_vector, alpha = 1)
cv.lasso.mod <- cv.glmnet(x = X_matrix, y = y_vector, alpha = 1, nfolds = 10)
#The first plot was the well-know MSE vs. log lambda plot, after performing #cross va
lidaton a list of errors were stores and were used as y values, #with the log lambda
values, from the graph we can see that the minimum MSE #occured at approximately log
lambda value between -5 and -6 .
plot(cv.lasso.mod)
```



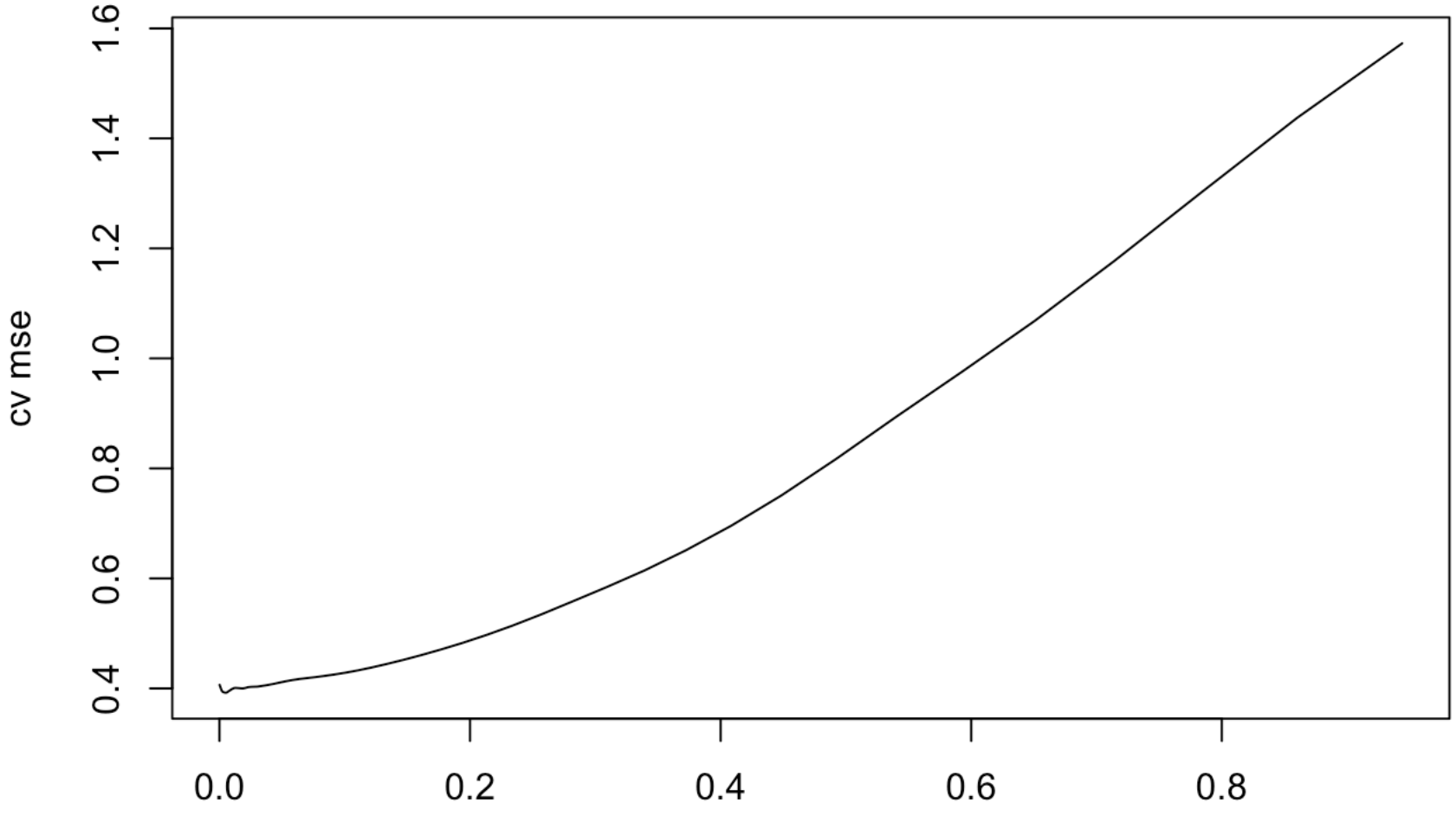

```
#For the second plot, it is supposed to look like Fig 6.13 on page 229 of #ISLR, I go
t a vector of the ratio of the magnitude of coefficients, and #plot them agains the c
ross validation mse in order (put mse and ratio in a #dataframe, sort in increasing o
rder. The mse decreases as the ratio #increases.
lasso_coef1 = c()
for (i in 1:length(lasso.mod$lambda)) {
  lasso_coef1 = c(lasso_coef1, sum(abs(coef(lasso.mod)[,i][2:39])))
}
orig1 = sum(abs(model2$coefficients[2:39]))
temp1 = data.frame(coef_ratio = lasso_coef1/orig1, cv_mse = cv.lasso.mod$cvm)
temp1 = temp1[order(temp1$coef_ratio),]
plot(temp1$coef_ratio, temp1$cv_mse, type = "l",ylab="CV MSE", xlab="coefficients rat
io")
```




```
#this third plot is like the right figure of the page, showing the changes in coeffic
ients as the lambda increases, al of them converge to zero at the end, the lowest mse
appears at ratio around 0.96
plot(lasso.mod, xvar = "lambda")
```




```
temp2 = data.frame(lamb= cv.lasso.mod$lambda, cv_mse = cv.lasso.mod$cvm)
temp2 = temp2[order(temp2$lamb),]
#this is a plot of MSE againse lambda, no log scale.
plot(temp2$lamb, temp2$cv_mse, xlab="lambda", ylab="cv mse", type="l")
```




```
lambdaMin = cv.lasso.mod$lambda.min
lambdaSe = cv.lasso.mod$lambda.1se
lambdaMin
```

```
## [1] 0.005156078
```

```
log(lambdaMin)
```

```
## [1] -5.267579
```

We use lambda.min as the “best” lambda, and further analyze its fit, its log is -5.36 corresponding to the cv red plot between the range of -5 and -6.

```
best_coef = coef(lasso.mod)[,which(lasso.mod$lambda==lambdaMin)]
best_coef
```

```
##   (Intercept)           G.x            GS       InnOuts            PO
##  1.317764e+01  0.000000e+00  0.000000e+00  9.591518e-05  0.000000e+00
##             A             E            DP           G.y            AB
##  0.000000e+00 -1.223563e-03 -1.784613e-03 -8.365238e-03  1.010878e-03
##             R             H           X2B           X3B            HR
##  4.668772e-04  0.000000e+00  0.000000e+00  7.502873e-03 -2.050401e-02
##           RBI            SB            CS            BB            SO
##  5.987629e-03  0.000000e+00 -4.254014e-03  6.209377e-03 -8.437263e-04
##           IBB           HBP            SH            SF          GIDP
##  1.086896e-02  5.484766e-03  1.200707e-02  1.619803e-02  0.000000e+00
##           CAB            CH           CHR          CRBI           CBB
##  0.000000e+00  7.476410e-04  0.000000e+00  0.000000e+00 -5.239702e-04
##           AVG           OBP    CHRperYear    SS2BDummy   CenterDummy
## -3.876587e-02 -7.499336e-03  5.587443e-02  2.397428e-01 -5.916760e-02
##       CFDummy     ThreeFive       SixLong         logCR
##  8.363732e-02  0.000000e+00  1.142116e+00  0.000000e+00
```

```
cv.lasso.mod$cvm[which(lasso.mod$lambda==lambdaMin)]
```

```
## [1] 0.3921301
```

```
test_errors$CVerrors
```

```
## [1] 0.3573392 0.3573567 0.3584094 0.3653530 0.3732628
```

This lasso model makes a few coefficients zero but still more than half non-zero, compared with the original linear model. Here I compared the minimum CV mse generated by this lambda and the five average CV errors from a previous step, surprisingly, all the CV errors from the previous step is smaller than the lasso error, by around 0.2, therefore, lasso in this case does help more than other method with variable selection, but is not robust as best subset regression plus cross validation. Regsubsets perform selection on each size separately, but LASSO performs the regression on full model, might be the reason why LASSO did not outperform regsubsets.

Variables ending with non-zero coefficients: InnOuts, E, DP, G.y, AB, RX2B, X3B, HR, RBI, CS BB, SO, IBB HBP,SH, SF, CHCBB, AVG, OBP, CHR/YEAR, MI Dummy, Center Dummy, CF Dummy and SixLong. Actually a lot more variables are nonzero than the ones found by regression subsets.