Project 4 STAT 5474

Prince Appiah

10/19/2021

Read in Data

```
baseball <- read.table(file=
        "http://www.amstat.org/publications/jse/datasets/baseball.dat.txt",
        header = F, col.names=c("salary", "batting.avg", "OBP", "runs", "hits",
        "doubles", "triples", "homeruns", "RBI", "walks", "strike.outs",
        "stolen.bases", "errors", "free.agency.elig", "free.agent.91",
        "arb.elig", "arb.91", "name"))
head(baseball)
```

```
##   salary batting.avg   OBP runs hits doubles triples homeruns RBI walks
## 1   3300       0.272 0.302   69  153      21       4       31 104    22
## 2   2600       0.269 0.335   58  111      17       2       18  66    39
## 3   2500       0.249 0.337   54  115      15       1       17  73    63
## 4   2475       0.260 0.292   59  128      22       7       12  50    23
## 5   2313       0.273 0.346   87  169      28       5        8  58    70
## 6   2175       0.291 0.379  104  170      32       2       26 100    87
##   strike.outs stolen.bases errors free.agency.elig free.agent.91 arb.elig
## 1          80            4      3                1             0        0
## 2          69            0      3                1             1        0
## 3         116            6      5                1             0        0
## 4          64           21     21                0             0        1
## 5          53            3      8                0             0        1
## 6          89           22      4                1             0        0
##   arb.91              name
## 1      0 Andre Dawson
## 2      0 Steve Buchele
## 3      0 Kal Daniels
## 4      0 Shawon Dunston
## 5      0 Mark Grace
## 6      0 Ryne Sandberg
```

```
dim(baseball)
```

```
## [1] 337  18
```

Comments We printed the first 6 rows of the data set.Also the dimension of the baseball is 337 rows and 18 columns(that is 337 observations and 18 variables).

```r
bb92 <- read.csv(file="bb92-test.csv", header=T)
```

Comment Read in the data bb92 which will be used later for model deployment.

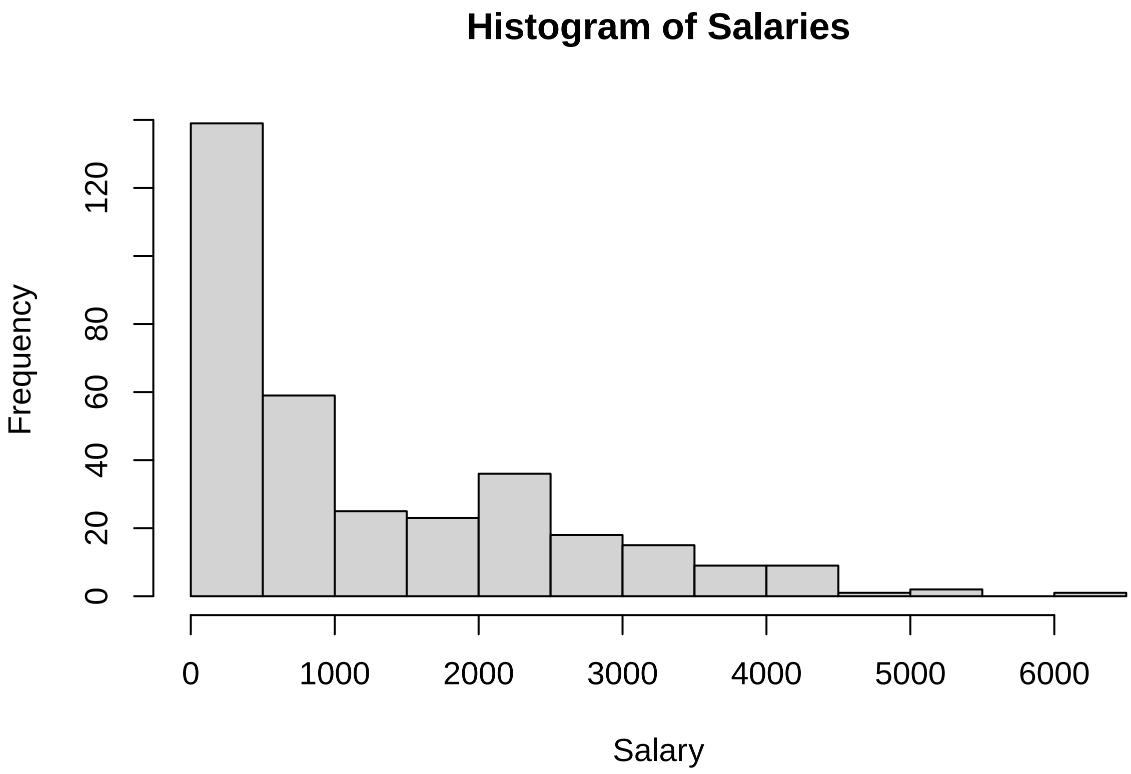
Linear Regression

Linear regression will be used to predict a hitter's salary based on his performance variables.

1. EDA

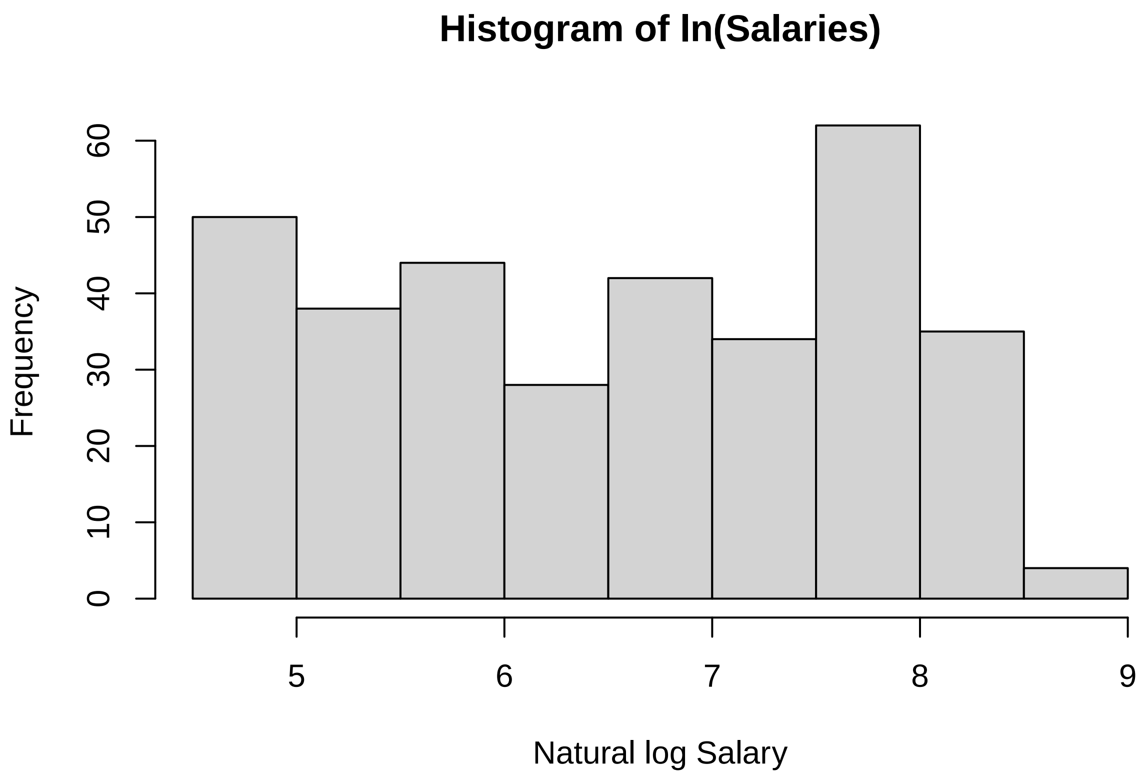
(a) Obtain the histograms of both salary and the logarithm (natural base) of salary and comment. Proceed with the log-transformed salary from this step on.

```r
# Histogram of log(salary)
hist(baseball$salary, main= "Histogram of Salaries", xlab= "Salary")
```



```r
baseball$logsalary <- log(baseball$salary) #transform salaries to natural log

hist(baseball$logsalary, main= "Histogram of ln(Salaries)",
     xlab= "Natural log Salary") #histogram of natural log Salaries
```



Comment We first observe that the histogram of the distribution of Salary is right skewed which follows an exponential distribution. After the log transform, we observe that the histogram for salary has been shifted to the right making our data appears more normally or uniformly distributed.

(b) Inspect the data and answer these questions: Are there any missing data? Among all the predictors, how many of them are continuous, integer counts, and categorical, respectively?

```
#install.packages("questionr")
library(questionr)
freq.na(baseball)
```

```
##                   missing %
## salary                  0 0
## batting.avg             0 0
## OBP                     0 0
## runs                    0 0
## hits                    0 0
## doubles                 0 0
## triples                 0 0
## homeruns                0 0
## RBI                     0 0
## walks                   0 0
## strike.outs             0 0
## stolen.bases            0 0
## errors                  0 0
## free.agency.elig        0 0
## free.agent.91           0 0
## arb.elig                0 0
## arb.91                  0 0
## name                   0 0
## logsalary              0 0
```

```
str(baseball)
```

```
## ’data.frame’:    337 obs. of  19 variables:
##  $ salary          : int  3300 2600 2500 2475 2313 2175 600 460 240 200 ...
##  $ batting.avg     : num  0.272 0.269 0.249 0.26 0.273 0.291 0.258 0.228 0.25 0.203 ...
##  $ OBP             : num  0.302 0.335 0.337 0.292 0.346 0.379 0.37 0.279 0.327 0.24 ...
##  $ runs            : int  69 58 54 59 87 104 34 16 40 39 ...
##  $ hits            : int  153 111 115 128 169 170 86 38 61 64 ...
##  $ doubles         : int  21 17 15 22 28 32 14 7 11 10 ...
##  $ triples         : int  4 2 1 7 5 2 1 2 0 1 ...
##  $ homeruns        : int  31 18 17 12 8 26 14 3 1 10 ...
##  $ RBI             : int  104 66 73 50 58 100 38 21 18 33 ...
##  $ walks           : int  22 39 63 23 70 87 15 11 24 14 ...
##  $ strike.outs     : int  80 69 116 64 53 89 45 32 26 96 ...
##  $ stolen.bases    : int  4 0 6 21 3 22 0 2 14 13 ...
##  $ errors          : int  3 3 5 21 8 4 10 3 2 6 ...
##  $ free.agency.elig: int  1 1 1 0 0 1 1 0 0 0 ...
##  $ free.agent.91   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ arb.elig        : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ arb.91          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ name            : chr  "Andre Dawson     " "Steve Buchele    " "Kal Daniels      " "Shawon Dunston
##  $ logsalary       : num  8.1 7.86 7.82 7.81 7.75 ...
```

Comment Clearly, we see that there are no missing values in our data set. Also, we observe that there are 3 variables that are continuous counts, 11 variables that are integer counts and 4 variables that are categorical counts. Moreover, the variable "name" is chr which can be classified as categorical variable.

2. Linear Regression with Variable Selection

(a) Partition the data randomly into two sets: the training data D and the test data D with a ratio of about 2:1.

```
set.seed(150)
ratio <- 2/3
rem.sal.nam <- baseball[, -c(1,18)] # we are now using the variable "logsalary" so salary is irrelevant
names(rem.sal.nam)
```

```
##  [1] "batting.avg"      "OBP"              "runs"             "hits"
##  [5] "doubles"          "triples"          "homeruns"         "RBI"
##  [9] "walks"            "strike.outs"      "stolen.bases"     "errors"
## [13] "free.agency.elig" "free.agent.91"    "arb.elig"         "arb.91"
## [17] "logsalary"
```

```
dt <- sort(sample(nrow(rem.sal.nam), nrow(rem.sal.nam)*ratio))
D <-rem.sal.nam[dt,] # training data
D0 <- rem.sal.nam[-dt,] # test data
dim(D)
```

```
## [1] 224  17
```

```
dim(D0)
```

```
## [1] 113  17
```

Comment The data set has been partitioned into two sets with 2/3 being the training data with dimension 224 observations and 17 variables and 1/3 being the test data with dimension 113 observations and 17 variables.

(b) Using the training data D, apply three variable selection methods of your choice and identify your 'best' models accordingly.

Full Model

```
formula0 <- logsalary ~ batting.avg + OBP + runs+ hits + doubles + triples + homeruns + RBI + walks + s
y <- D[, all.vars(formula0)[1]]
X <- model.matrix(as.formula(formula0),D)
X <- as.data.frame(scale(X, center = TRUE, scale = TRUE)) # Standardize X
y <- scale(y, center = TRUE, scale = FALSE) # At least center y
dat <- as.data.frame(cbind(X, y))
fit.full <- lm(formula0, data=D)
BIC(fit.full)
```

```
## [1] 651.1518
```

```
summary(fit.full)
```

```
##
## Call:
## lm(formula = formula0, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8648 -0.3685  0.0984  0.5535  4.2764
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## batting.avg       3.7501624  4.4432558   0.844 0.399632
## OBP              13.2376159  3.5969349   3.680 0.000297 ***
## runs              0.0128500  0.0099848   1.287 0.199542
## hits             -0.0071170  0.0054440  -1.307 0.192546
## doubles           0.0046210  0.0133000   0.347 0.728610
## triples          -0.0179946  0.0372294  -0.483 0.629360
## homeruns         -0.0337686  0.0199278  -1.695 0.091657 .
## RBI               0.0178886  0.0085291   2.097 0.037170 *
## walks            -0.0205905  0.0073739  -2.792 0.005721 **
## strike.outs       0.0085117  0.0030703   2.772 0.006073 **
## stolen.bases      0.0004241  0.0079888   0.053 0.957709
## errors           -0.0022978  0.0119069  -0.193 0.847161
## free.agency.elig  1.9422531  0.1765085  11.004  < 2e-16 ***
## free.agent.91    -0.2691886  0.2083324  -1.292 0.197753
## arb.elig          1.7152330  0.1841113   9.316  < 2e-16 ***
## arb.91           -0.0348620  0.3956364  -0.088 0.929869
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8748 on 208 degrees of freedom
## Multiple R-squared:  0.9841, Adjusted R-squared:  0.9829
## F-statistic: 803.9 on 16 and 208 DF,  p-value: < 2.2e-16
```

Comment We observe from the output that the full model is found to be statistically significant with given the F-Values and p-values and with an R-Squared of 0.983. Six variables were found to be statistically significant with p-values less than 0.5. Also, the BIC is 651 which means the complexity of full model has quite increased.

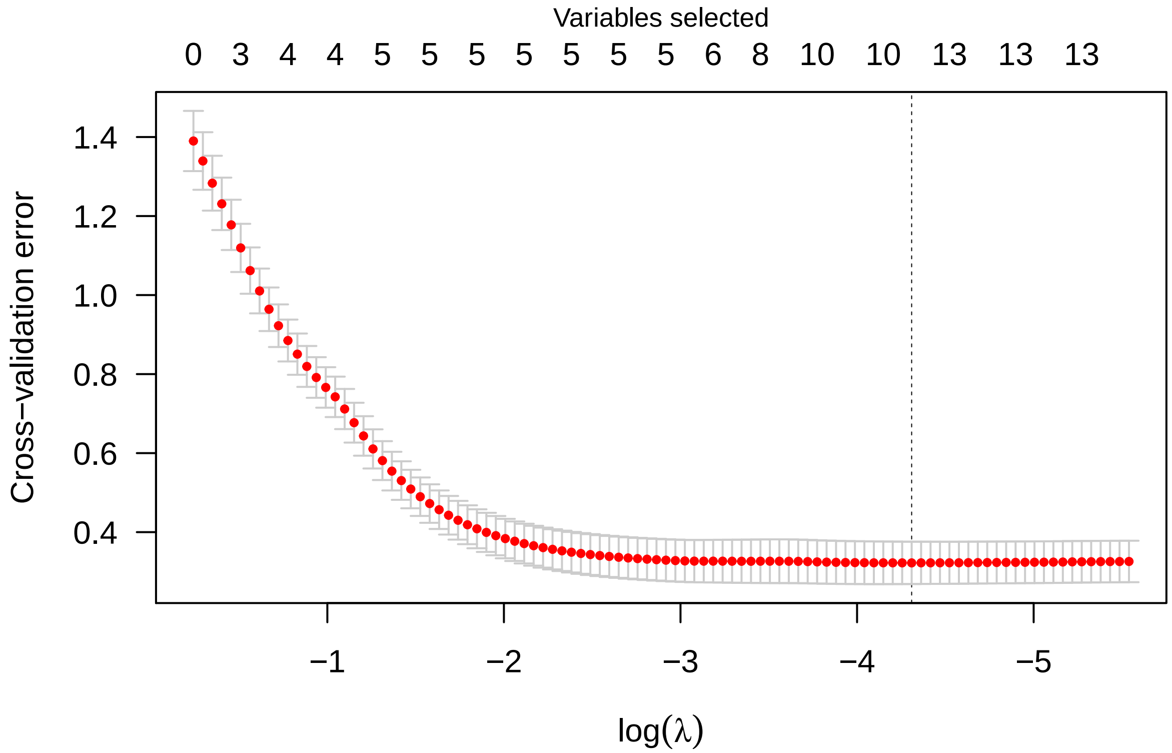
Method 1: LASSO

```
set.seed(150)
library(glmnet)
```

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

```
## Loaded glmnet 4.1-2
```

```
library(ncvreg)
formula.LASSO<- cv.ncvreg(X=X,y=y, nfolds=10, family="gaussian",
    penalty="lasso", lambda.min=.005, nlambda=100, eps=.001, max.iter=1000)
plot(formula.LASSO)
```



```r
names(formula.LASSO)
```

```
## [1] "cve"        "cvse"       "fold"       "lambda"     "fit"
## [6] "min"        "lambda.min" "null.dev"   "Bias"
```

```r
beta.hat <- coef(formula.LASSO)  # THE LASSO COEFFICIENTS WITH MINIMUM CV ERROR
cutoff <- 0.0001
terms <- names(beta.hat)[abs(beta.hat) > cutoff]
formula.LASS <- as.formula(paste(c("logsalary ~ ", terms[-1]), collapse=" + "))
fit.L1 <- lm(formula.LASS, data = D)
summary(fit.L1)
```

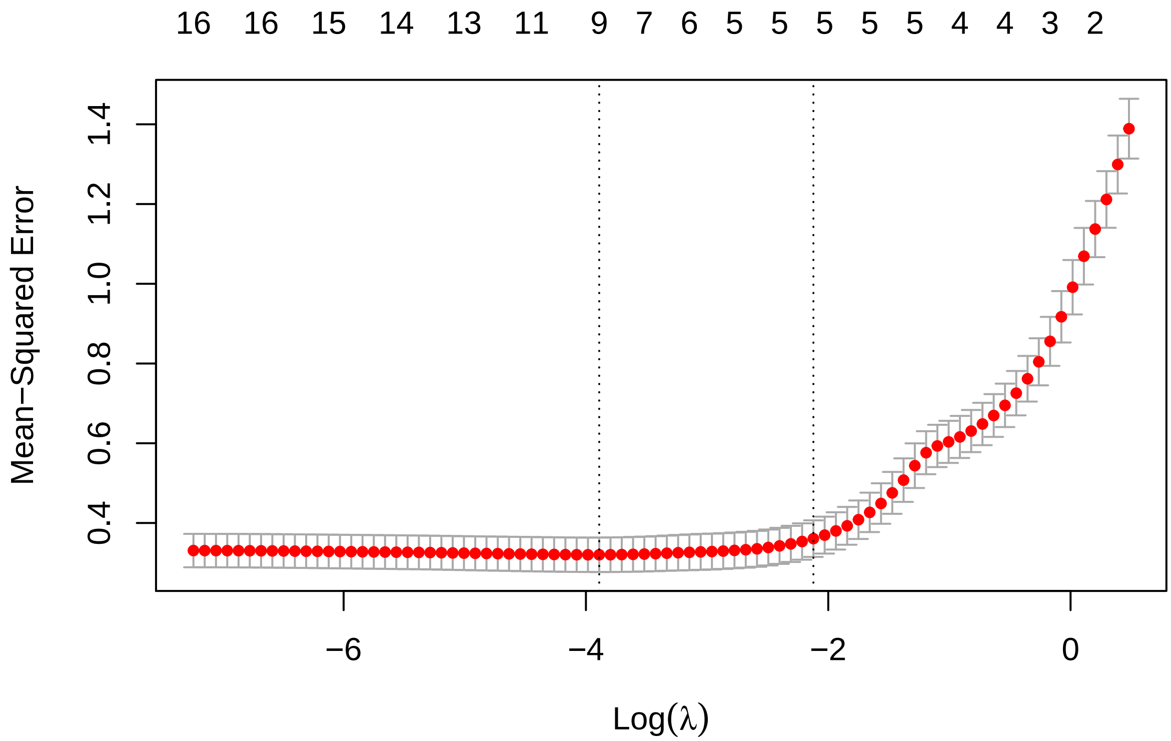
```
##
## Call:
## lm(formula = formula.LASS, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37372 -0.31528 -0.02907  0.35740  1.33806
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.0478401  0.0856630  58.927  < 2e-16 ***
## runs              0.0088369  0.0035751   2.472 0.014225 *
## hits              0.0009294  0.0021618   0.430 0.667684
## RBI               0.0102313  0.0028812   3.551 0.000472 ***
## strike.outs      -0.0045481  0.0017034  -2.670 0.008168 **
## errors           -0.0086052  0.0073986  -1.163 0.246095
## free.agency.elig  1.5935528  0.1010925  15.763  < 2e-16 ***
## free.agent.91    -0.2221174  0.1274167  -1.743 0.082728 .
## arb.elig          1.3958810  0.1138242  12.263  < 2e-16 ***
## arb.91            0.0953615  0.2415324   0.395 0.693370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5436 on 214 degrees of freedom
## Multiple R-squared:  0.7953, Adjusted R-squared:  0.7867
## F-statistic:  92.4 on 9 and 214 DF,  p-value: < 2.2e-16
```

Comment The graph indicates that as the variable selected increases, the cross-validation error decreases (getting close to zero). The dash line indicates the value for which the model has the lowest cross-validation mean squared error. Also, we observe that the variables runs,RBI,strike.outs,free.agency.elig and arb.elig are statistically significant as their p-values are less than 0.05. Also, looking at the overall p-value=2e-16 < 0.05, we can conclude that there is some form of relationship between the logsalary and the independent variables. That is at least one of the coefficients of our independent variables is not zero.

Method 2: ADAPTIVE LASSO

```r
set.seed(150)
library(MESS)
library(glmnet)
wt <- adaptive.weights(x=X, y=y, weight.method="univariate")
```

```
cv.ALASSO <- cv.glmnet(x=as.matrix(X), y=y, family="gaussian", alpha=1, nlambda=100,
                       penalty.factor=as.numeric(wt$weights), standardize=FALSE)
plot(cv.ALASSO)
```



```
beta.hat.alasso <- coef(cv.ALASSO, s="lambda.1se")
cutoff <- 0
terms <- names(X)[abs(as.vector(beta.hat.alasso[-1])) > cutoff]
formula.ALASSO <- as.formula(paste(c("logsalary ~ ", terms),
                                   collapse=" + "))
fit.ALASSO <- lm(formula.ALASSO, data =D)
summary(fit.ALASSO)
```

```
## 
## Call:
## lm(formula = formula.ALASSO, data = D)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.90995 -0.28712 -0.03074  0.38619  1.21698 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      4.919193   0.078272  62.847  < 2e-16 ***
## runs             0.007730   0.003563   2.170  0.03111 *  
## hits             0.001176   0.002102   0.560  0.57630    
## RBI              0.006497   0.002473   2.627  0.00922 ** 
## free.agency.elig 1.543823   0.092397  16.709  < 2e-16 ***
## arb.elig         1.401052   0.110993  12.623  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5552 on 218 degrees of freedom
## Multiple R-squared:  0.7825, Adjusted R-squared:  0.7775
## F-statistic: 156.8 on 5 and 218 DF,  p-value: < 2.2e-16
```

Comment The graph indicates that as the variable selected increases, the mean-squared error decreases (getting close to zero). The dash line to the left indicates the value for which the model has lowest cross-validation mean squared error while the dash line to the right indicates indicates 1 standard error from the minimum mean squared error. Also, we observe that the variables runs,RBI,free.agency.elig and arb.elig are statistically significant as their p-values are less than 0.05. Also, looking at the overall p-value=2e-16 < 0.05, we can conclude that there is a relationship between the logsalary and the independent variables. That is at least one of the coefficients of our independent variables is not zero.

Method 3 (Stepwise Regression)

```r
set.seed(150)
library(MASS)
fit.step <- stepAIC(fit.full, direction = "backward", k=log(nrow(D)))
```

```
## Start:  AIC=10.06
## logsalary ~ batting.avg + OBP + runs + hits + doubles + triples +
##     homeruns + RBI + walks + strike.outs + stolen.bases + errors +
##     free.agency.elig + free.agent.91 + arb.elig + arb.91 - 1
##
##                    Df Sum of Sq    RSS     AIC
## - stolen.bases      1     0.002 159.18   4.647
## - arb.91            1     0.006 159.18   4.652
## - errors            1     0.028 159.20   4.684
## - doubles           1     0.092 159.26   4.774
## - triples           1     0.179 159.35   4.896
## - batting.avg       1     0.545 159.72   5.410
## - runs              1     1.267 160.44   6.421
## - free.agent.91     1     1.278 160.45   6.435
## - hits              1     1.308 160.48   6.477
## - homeruns          1     2.197 161.37   7.715
## - RBI               1     3.366 162.54   9.332
## <none>                          159.17  10.056
## - strike.outs       1     5.881 165.05  12.771
## - walks             1     5.967 165.14  12.887
## - OBP               1    10.365 169.54  18.775
## - arb.elig          1    66.419 225.59  82.761
## - free.agency.elig  1    92.659 251.83 107.408
##
## Step:  AIC=4.65
## logsalary ~ batting.avg + OBP + runs + hits + doubles + triples +
##     homeruns + RBI + walks + strike.outs + errors + free.agency.elig +
##     free.agent.91 + arb.elig + arb.91 - 1
##
##                    Df Sum of Sq    RSS     AIC
## - arb.91            1     0.007 159.18  -0.755
## - errors            1     0.028 159.20  -0.725
## - doubles           1     0.090 159.27  -0.637
## - triples           1     0.182 159.36  -0.509
## - batting.avg       1     0.544 159.72  -0.001
## - free.agent.91     1     1.280 160.46   1.030
## - hits              1     1.321 160.50   1.087
## - runs              1     1.849 161.02   1.822
## - homeruns          1     2.291 161.47   2.437
## - RBI               1     3.412 162.59   3.987
## <none>                          159.18   4.647
## - walks             1     6.248 165.42   7.860
## - strike.outs       1     6.335 165.51   7.977
## - OBP               1    10.374 169.55  13.378
## - arb.elig          1    66.474 225.65  77.406
## - free.agency.elig  1    92.665 251.84 102.004
## 
## Step:  AIC=-0.76
## logsalary ~ batting.avg + OBP + runs + hits + doubles + triples + 
##     homeruns + RBI + walks + strike.outs + errors + free.agency.elig + 
##     free.agent.91 + arb.elig - 1
## 
##                    Df Sum of Sq    RSS    AIC
## - errors            1     0.028 159.21 -6.128
## - doubles           1     0.087 159.27 -6.045
## - triples           1     0.177 159.36 -5.919
## - batting.avg       1     0.549 159.73 -5.395
## - free.agent.91     1     1.284 160.47 -4.368
## - hits              1     1.321 160.50 -4.315
## - runs              1     1.843 161.03 -3.588
## - homeruns          1     2.288 161.47 -2.971
## - RBI               1     3.415 162.60 -1.412
## <none>                          159.18 -0.755
## - walks             1     6.249 165.43  2.458
## - strike.outs       1     6.346 165.53  2.590
## - OBP               1    10.368 169.55  7.967
## - arb.elig          1    70.644 229.82 76.102
## - free.agency.elig  1    93.009 252.19 96.904
## 
## Step:  AIC=-6.13
## logsalary ~ batting.avg + OBP + runs + hits + doubles + triples + 
##     homeruns + RBI + walks + strike.outs + free.agency.elig + 
##     free.agent.91 + arb.elig - 1
## 
##                    Df Sum of Sq    RSS     AIC
## - doubles           1     0.086 159.30 -11.418
## - triples           1     0.174 159.38 -11.295
## - batting.avg       1     0.547 159.76 -10.772
## - free.agent.91     1     1.333 160.54  -9.672
## - hits              1     1.388 160.60  -9.595
## - runs              1     1.845 161.06  -8.958
## - homeruns          1     2.270 161.48  -8.368
## - RBI               1     3.454 162.66  -6.732
## <none>                          159.21  -6.128
## - walks             1     6.232 165.44  -2.939
## - strike.outs       1     6.472 165.68  -2.614
## - OBP               1    10.363 169.57   2.586
## - arb.elig          1    72.060 231.27  72.093
## - free.agency.elig  1    94.357 253.57  92.711
##
## Step:  AIC=-11.42
## logsalary ~ batting.avg + OBP + runs + hits + triples + homeruns +
##     RBI + walks + strike.outs + free.agency.elig + free.agent.91 +
##     arb.elig - 1
##
##                    Df Sum of Sq    RSS     AIC
## - triples           1     0.240 159.54 -16.492
## - batting.avg       1     0.521 159.82 -16.099
## - free.agent.91     1     1.305 160.60 -15.002
## - hits              1     1.331 160.63 -14.966
## - runs              1     1.896 161.19 -14.180
## - homeruns          1     2.274 161.57 -13.655
## - RBI               1     3.804 163.10 -11.543
## <none>                          159.30 -11.418
## - strike.outs       1     6.386 165.68  -8.025
## - walks             1     6.591 165.89  -7.749
## - OBP               1    10.562 169.86  -2.450
## - arb.elig          1    72.008 231.30  66.715
## - free.agency.elig  1    94.923 254.22  87.875
##
## Step:  AIC=-16.49
## logsalary ~ batting.avg + OBP + runs + hits + homeruns + RBI +
##     walks + strike.outs + free.agency.elig + free.agent.91 +
##     arb.elig - 1
##
##                    Df Sum of Sq    RSS     AIC
## - batting.avg       1     0.574 160.11 -21.099
## - hits              1     1.315 160.85 -20.065
## - free.agent.91     1     1.348 160.88 -20.020
## - runs              1     1.660 161.20 -19.585
## - homeruns          1     2.033 161.57 -19.067
## - RBI               1     3.648 163.18 -16.839
## <none>                          159.54 -16.492
## - strike.outs       1     6.159 165.69 -13.419
## - walks             1     6.380 165.92 -13.120
## - OBP               1    10.415 169.95  -7.738
## - arb.elig          1    71.961 231.50  61.490
## - free.agency.elig  1    95.232 254.77  82.947
##
## Step:  AIC=-21.1
## logsalary ~ OBP + runs + hits + homeruns + RBI + walks + strike.outs +
##     free.agency.elig + free.agent.91 + arb.elig - 1
##
##                    Df Sum of Sq     RSS    AIC
## - hits              1      0.84  160.96 -25.33
## - free.agent.91     1      1.27  161.39 -24.73
## - runs              1      1.63  161.74 -24.24
## - homeruns          1      1.99  162.10 -23.75
## - RBI               1      3.73  163.84 -21.35
## <none>                           160.11 -21.10
## - strike.outs       1      5.71  165.82 -18.66
## - walks             1     17.09  177.20  -3.79
## - arb.elig          1     71.43  231.54  56.12
## - free.agency.elig  1     95.34  255.46  78.14
## - OBP               1    908.60 1068.71 398.72
##
## Step:  AIC=-25.33
## logsalary ~ OBP + runs + homeruns + RBI + walks + strike.outs +
##     free.agency.elig + free.agent.91 + arb.elig - 1
##
##                    Df Sum of Sq     RSS    AIC
## - runs              1      0.80  161.75 -29.64
## - homeruns          1      1.14  162.10 -29.16
## - free.agent.91     1      1.17  162.13 -29.12
## - RBI               1      3.19  164.15 -26.34
## <none>                           160.96 -25.33
## - strike.outs       1      5.24  166.19 -23.57
## - walks             1     16.74  177.70  -8.58
## - arb.elig          1     70.94  231.89  51.05
## - free.agency.elig  1     96.52  257.48  74.49
## - OBP               1    936.67 1097.63 399.29
##
## Step:  AIC=-29.64
## logsalary ~ OBP + homeruns + RBI + walks + strike.outs + free.agency.elig +
##     free.agent.91 + arb.elig - 1
##
##                    Df Sum of Sq     RSS    AIC
## - free.agent.91     1      1.25  163.00 -33.32
## - homeruns          1      1.59  163.34 -32.86
## <none>                           161.75 -29.64
## - strike.outs       1      5.04  166.79 -28.18
## - RBI               1      7.30  169.05 -25.17
## - walks             1     19.00  180.75 -10.18
## - arb.elig          1     71.64  233.40  47.09
## - free.agency.elig  1     96.15  257.91  69.45
## - OBP               1    972.75 1134.51 401.28
##
## Step:  AIC=-33.32
## logsalary ~ OBP + homeruns + RBI + walks + strike.outs + free.agency.elig +
##     arb.elig - 1
##
##                    Df Sum of Sq     RSS    AIC
## - homeruns          1      1.20  164.20 -37.09
## <none>                           163.00 -33.32
## - strike.outs       1      4.95  167.95 -32.03
## - RBI               1      6.71  169.71 -29.70
## - walks             1     18.13  181.13 -15.11
## - arb.elig          1     71.29  234.29  42.53
## - free.agency.elig  1    108.01  271.02  75.15
## - OBP               1    971.80 1134.80 395.92
##
## Step:  AIC=-37.09
## logsalary ~ OBP + RBI + walks + strike.outs + free.agency.elig +
##     arb.elig - 1
##
##                    Df Sum of Sq     RSS    AIC
## - strike.outs       1      3.95  168.15 -37.18
## <none>                           164.20 -37.09
## - RBI               1      6.79  170.99 -33.43
## - walks             1     17.16  181.36 -20.24
## - arb.elig          1     74.38  238.58  41.19
## - free.agency.elig  1    106.90  271.10  69.81
## - OBP               1   1215.77 1379.98 434.33
##
## Step:  AIC=-37.18
## logsalary ~ OBP + RBI + walks + free.agency.elig + arb.elig - 
##     1
##
##                    Df Sum of Sq     RSS    AIC
## <none>                           168.15 -37.18
## - walks             1     14.23  182.39 -24.39
## - RBI               1     19.87  188.02 -17.57
## - arb.elig          1     72.42  240.57  37.64
## - free.agency.elig  1    103.68  271.83  65.00
## - OBP               1   1377.66 1545.81 454.34
```

```
fit.step$anova
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## logsalary ~ batting.avg + OBP + runs + hits + doubles + triples + 
##     homeruns + RBI + walks + strike.outs + stolen.bases + errors + 
##     free.agency.elig + free.agent.91 + arb.elig + arb.91 - 1
## 
## Final Model:
## logsalary ~ OBP + RBI + walks + free.agency.elig + arb.elig - 
##     1
## 
## 
##               Step Df    Deviance Resid. Df Resid. Dev         AIC
## 1                                       208   159.1730  10.0556968
## 2   - stolen.bases  1 0.002157112       209   159.1751   4.6470863
## 3         - arb.91  1 0.006566307       210   159.1817  -0.7553194
## 4         - errors  1 0.027707165       211   159.2094  -6.1279794
## 5        - doubles  1 0.086321505       212   159.2957 -11.4182082
## 6        - triples  1 0.240390338       213   159.5361 -16.4920746
## 7    - batting.avg  1 0.574378193       214   160.1105 -21.0987012
## 8           - hits  1 0.844838887       215   160.9553 -25.3314964
## 9           - runs  1 0.796675826       216   161.7520 -29.6371511
## 10 - free.agent.91  1 1.249754228       217   163.0017 -33.3247442
## 11      - homeruns  1 1.200804317       218   164.2026 -37.0922714
## 12   - strike.outs  1 3.948216054       219   168.1508 -37.1816178
```

```
summary(fit.step)
```

```
## 
## Call:
```

```
## lm(formula = logsalary ~ OBP + RBI + walks + free.agency.elig +
##     arb.elig - 1, data = D)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6027 -0.3661  0.1212  0.5644  4.2522
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## OBP              16.759969   0.395667  42.359  < 2e-16 ***
## RBI               0.014643   0.002878   5.087 7.81e-07 ***
## walks            -0.015864   0.003684  -4.306 2.51e-05 ***
## free.agency.elig  1.724804   0.148430  11.620  < 2e-16 ***
## arb.elig          1.655395   0.170447   9.712  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8762 on 219 degrees of freedom
## Multiple R-squared:  0.9832, Adjusted R-squared:  0.9828
## F-statistic:  2561 on 5 and 219 DF,  p-value: < 2.2e-16
```

Comment We observe from the output that AIC reduces after each iteration. This selection criteria produce a model with 5 variables (OBP ,RBI,walks, free.agency.elig, arb.elig) all been statistically significant) as each p-value is less than 0.05.

(c) Report the essential steps and/or key quantities involved in the variable selection procedure that you choose.

i) LASSO: The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so the method apply a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero.

ii) Adaptive LASSO: Adaptive LASSO selection is a modification of LASSO selection. In adaptive LASSO selection, weights are applied to each of the parameters in forming the LASSO constraint.Adaptive LASSO enjoys the oracle properties; namely, it performs as well as if the true underlying model were given in advance.

iii) Stepwise regression is a combination of the forward and backward selection techniques. Stepwise regression is a modification of the forward selection so that after each step in which a variable was added, all candidate variables in the model are checked to see if their significance has been reduced below the specified tolerance level. If a nonsignificant variable is found, it is removed from the model.Stepwise regression requires two significance levels: one for adding variables and one for removing variables. The cutoff probability for adding variables should be less than the cutoff probability for removing variables so that the procedure does not get into an infinite loop.

(d) Output the necessary fitting results for each 'best' model, e.g., in particular, selected variables and their corresponding slope parameter estimates.

```r
# Outputting  the best fit for the LASSO selections method.
fit1<- lm(logsalary~ runs  + hits + RBI + strike.outs + errors + free.agency.elig + free.agent.91 + arb
summary(fit1)
```

```
##
## Call:
## lm(formula = logsalary ~ runs + hits + RBI + strike.outs + errors +
##     free.agency.elig + free.agent.91 + arb.elig + arb.91, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37372 -0.31528 -0.02907  0.35740  1.33806
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.0478401  0.0856630  58.927  < 2e-16 ***
## runs              0.0088369  0.0035751   2.472 0.014225 *
## hits              0.0009294  0.0021618   0.430 0.667684
## RBI               0.0102313  0.0028812   3.551 0.000472 ***
## strike.outs      -0.0045481  0.0017034  -2.670 0.008168 **
## errors           -0.0086052  0.0073986  -1.163 0.246095
## free.agency.elig  1.5935528  0.1010925  15.763  < 2e-16 ***
## free.agent.91    -0.2221174  0.1274167  -1.743 0.082728 .
## arb.elig          1.3958810  0.1138242  12.263  < 2e-16 ***
## arb.91            0.0953615  0.2415324   0.395 0.693370
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5436 on 214 degrees of freedom
## Multiple R-squared:  0.7953, Adjusted R-squared:  0.7867
## F-statistic:  92.4 on 9 and 214 DF,  p-value: < 2.2e-16
```

Comment The OLS model of the best fits for LASSO varibles is found to be statistically significant given the F-Values and p-values from the output with an R-Squared of 0.795 which is not different from the initial LASSO fit R-squared of 0.795.*

```
#Outputting  the best fit for the Adaptive LASSO selections method.
fit2<- lm(logsalary ~ runs  + hits + RBI + free.agency.elig + arb.elig, data=D )
summary(fit2)
```

```
##
## Call:
## lm(formula = logsalary ~ runs + hits + RBI + free.agency.elig +
##     arb.elig, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.90995 -0.28712 -0.03074  0.38619  1.21698
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.919193   0.078272  62.847  < 2e-16 ***
## runs             0.007730   0.003563   2.170  0.03111 *
## hits             0.001176   0.002102   0.560  0.57630
## RBI              0.006497   0.002473   2.627  0.00922 **
## free.agency.elig 1.543823   0.092397  16.709  < 2e-16 ***
## arb.elig         1.401052   0.110993  12.623  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5552 on 218 degrees of freedom
## Multiple R-squared:  0.7825, Adjusted R-squared:  0.7775
## F-statistic: 156.8 on 5 and 218 DF,  p-value: < 2.2e-16
```

Comment The OLS model of the best fits for ALASSO varibles is found to be statistically significant given the F-Values and P-values from the output with an R-Squared of 0.782 which is not significantly different from the initial ALASSO fit R-squared of 0.782

```
# Outputting  the best fit for the stepwise selections method.
fit3<-lm(formula = logsalary ~ OBP + RBI + walks + free.agency.elig + arb.elig,
         data = D)
summary(fit3)
```

```
##
## Call:
## lm(formula = logsalary ~ OBP + RBI + walks + free.agency.elig +
##     arb.elig, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74852 -0.32022 -0.03694  0.40524  1.20924
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.073105   0.293842  17.265  < 2e-16 ***
## OBP              -0.297981   1.021089  -0.292    0.771
## RBI               0.012198   0.001880   6.487 5.79e-10 ***
## walks             0.004222   0.002667   1.583    0.115
## free.agency.elig  1.550116   0.097220  15.944  < 2e-16 ***
## arb.elig          1.436759   0.111754  12.856  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5708 on 218 degrees of freedom
## Multiple R-squared:  0.7701, Adjusted R-squared:  0.7648
## F-statistic:   146 on 5 and 218 DF,  p-value: < 2.2e-16
```

Comment The OLS model of the best fits for Stepwise variable is found to be statistically significant given the F-Values and p-values from the output with an R-Squared of 0.77 which is significantly different from the initial Stepwise fit R-squared of 0.983.

(e) Apply your ‘best’ models to the test data D0 Output the sum of squared prediction error (SSPE). Let’s consider the one yielding the minimum SSPE as the final model.

```
# LASSO fit with test data
fit1.D0 <- lm(logsalary~ runs + hits + RBI + strike.outs + errors + free.agency.elig + free.agent.91 +
pred1.D0 <- predict(fit1.D0, newdata = D0)
# Adaptive LASSO fit with test data
fit2.D0 <- lm(logsalary ~ runs + hits + RBI + free.agency.elig + arb.elig, data=D0)
```

```
pred2.D0<-predict(fit2.D0, newdata = D0)
# Stepwise fit with test data
fit3.D0 <- lm(logsalary ~ OBP + RBI + walks + free.agency.elig + arb.elig, data=D0)
pred3.D0 <- predict(fit3.D0, newdata= D0)
```

```
# Estimating the SSPE
MSE.LASSO <- sum((D0$logsalary-pred1.D0)**2) # sum of square error
MSE.ALASSO <- sum((D0$logsalary-pred2.D0)**2) # sum of square error
MSE.STEP <- sum((D0$logsalary-pred3.D0)**2) # sum of square error
c(MSE.LASSO, MSE.ALASSO, MSE.STEP) # print sum of square errors
```

```
## [1] 27.93581 30.67009 31.14704
```

Comment We see that least SSPE is 27.9 which is the LASSO model. Thus, the final model is the LASSO model.

3. Final Model

Refit your final model using the entire data, i.e., DuD. Call it fit.final. Provide the output from your final model with summary(fit.final). Interpret the results.

```
fit.final = fit.L1
 summary(fit.final)
```

```
##
## Call:
## lm(formula = formula.LASS, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37372 -0.31528 -0.02907  0.35740  1.33806
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.0478401  0.0856630  58.927  < 2e-16 ***
## runs              0.0088369  0.0035751   2.472 0.014225 *
## hits              0.0009294  0.0021618   0.430 0.667684
## RBI               0.0102313  0.0028812   3.551 0.000472 ***
## strike.outs      -0.0045481  0.0017034  -2.670 0.008168 **
## errors           -0.0086052  0.0073986  -1.163 0.246095
## free.agency.elig  1.5935528  0.1010925  15.763  < 2e-16 ***
## free.agent.91    -0.2221174  0.1274167  -1.743 0.082728 .
## arb.elig          1.3958810  0.1138242  12.263  < 2e-16 ***
## arb.91            0.0953615  0.2415324   0.395 0.693370
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5436 on 214 degrees of freedom
## Multiple R-squared:  0.7953, Adjusted R-squared:  0.7867
## F-statistic:  92.4 on 9 and 214 DF,  p-value: < 2.2e-16
```

Comment We see that we have an adjusted R-squared of 0.787 which is relatively strong. Hence, it appears our chosen model fits the data well. We also see that almost all the independent variables are significant with the exception of hits, errors and arb.91. Moreover, we see from the estimated coefficients and intercepts that the expected value or average salary is 5.047840.

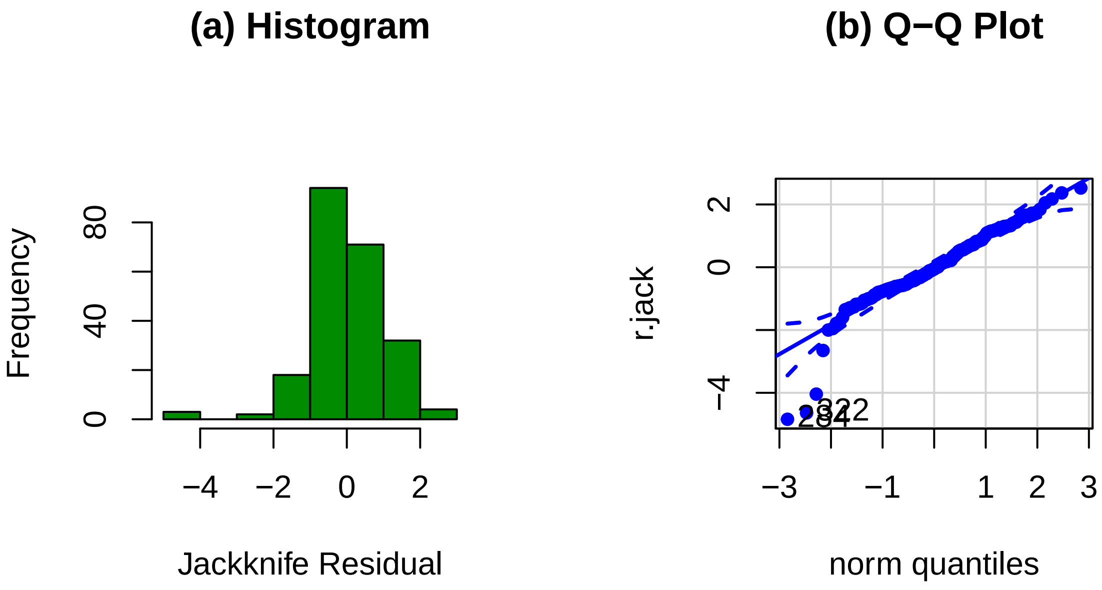
4.Model diagnostics

(a) Normality

```
r.jack <- rstudent(fit.final)
par(mfrow=c(1,2),mar=c(8,4,8,4))
# The fisrt plot: Histogram
hist(r.jack, xlab="Jackknife Residual", col="green4",
    main="(a) Histogram")
# install.packages("car")
library(car)
```

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

```
qqPlot(r.jack, pch=19, cex=.8, col="blue", main="(b) Q-Q Plot")
```



```
## 284 322
## 189 213
```

```
# THE SHAPIRO-WILKS NORMALITY TEST: A LARGE P-VALUE WOULD JUSTIFY NORMALITY
shapiro.test(r.jack)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  r.jack
## W = 0.9355, p-value = 2.274e-08
```

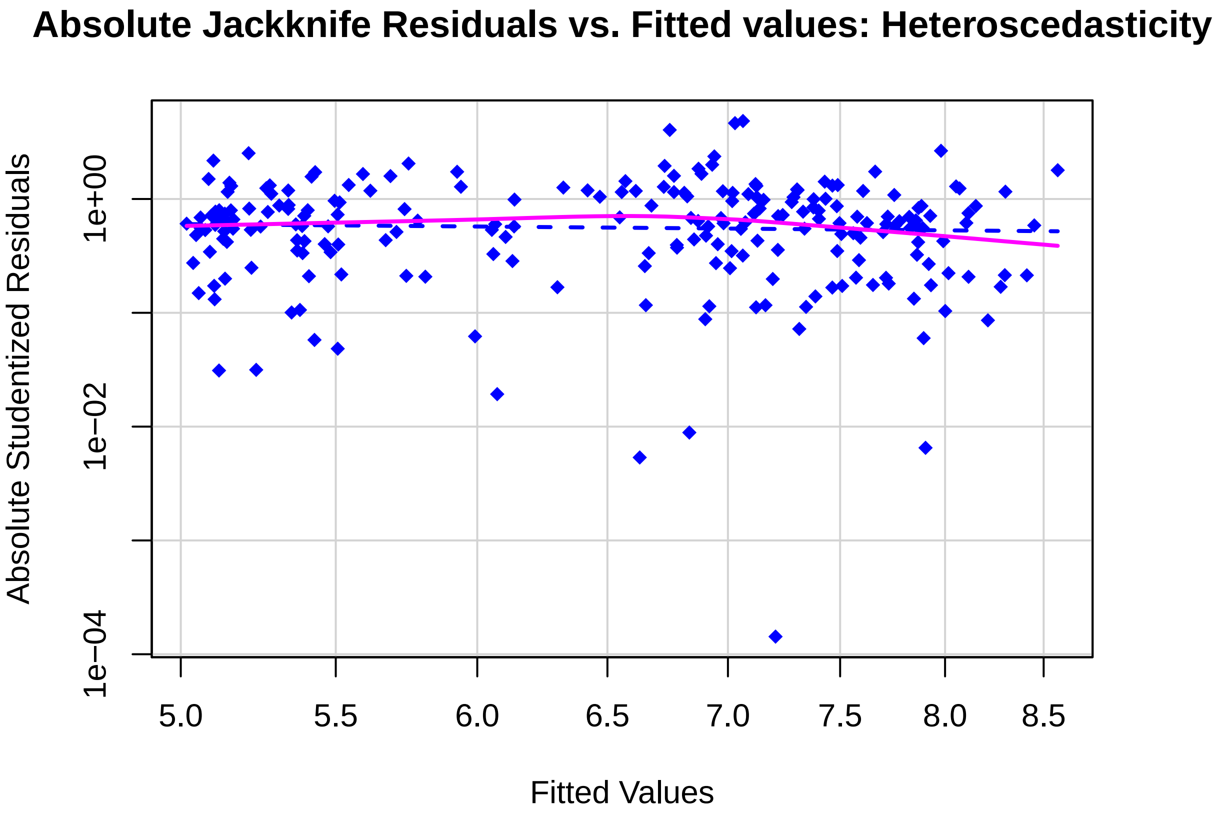
Comment The jackknife residuals based on the Histogram appear to be normally distributed with possible few outliers. Whereas on the the QQ Plot the residuals tend to stay on the line, just a couple outliers in particular observations 28, 34 and 22. Moreover, the output for Shapiro-Wilk normality test gave a p value = 2e-08 less than the level of significance at 0.05, hence we do not have normality.

(b) Homoscedasticity

```
ncvTest(fit.final)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.1820798, Df = 1, p = 0.66959
```

```
# A LARGE P-VALUE (>0.05) JUSTIFIES EQUAL VARIANCE
# Plot Absolute Jackknife Residuals vs. Fitted values
# Power Box-Cox Transformation on the response Y is suggested
par(mfrow=c(1,1),mar=c(4, 4, 4, 4))
spreadLevelPlot(fit.final, pch=18, cex=0.5, col="blue",
    main="Absolute Jackknife Residuals vs. Fitted values: Heteroscedasticity")
```



```
##
## Suggested power transformation:  1.269442
```

Comment We used the Breusch-Pagan Test to check for non-constant error variance. We had a p- value=0.7 greater than the level of significance at 0.05, hence we can assume equal variance. The line on the graph is relatively horizontal or flat which shows equal variance. We also see from the graph that the residuals are spread equally along the ranges of predictors.Hence, we conclude that the heteroscedasticity is not present.

(c) Independence

```
durbinWatsonTest(fit.final)
```

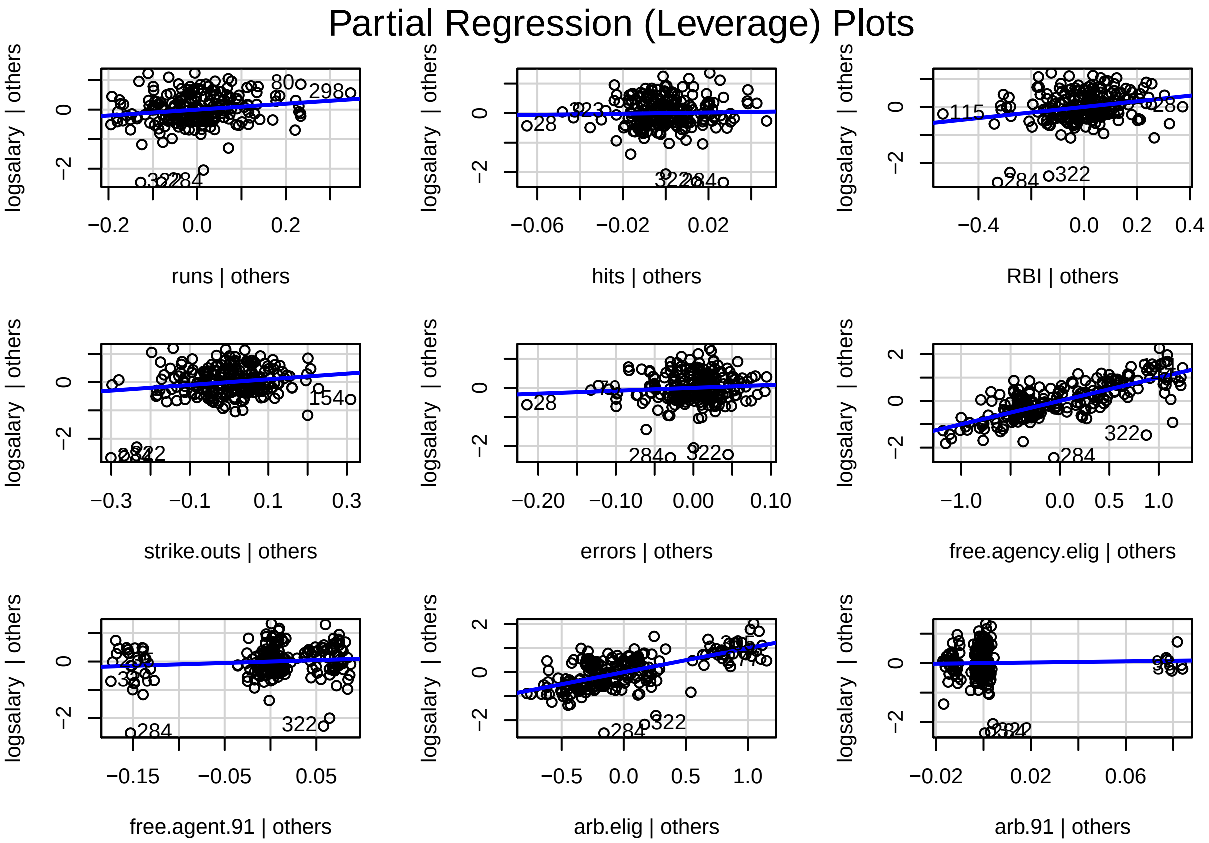
```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.1487853      1.696708   0.014
##  Alternative hypothesis: rho != 0
```

```
# LARGE P-VALUE (>0.05) JUSTIFIES INDEPENDENCE
```

Comment We have p-value=0.024 $<$ 0.05, hence we cannot assume independence.

(d) Linearity

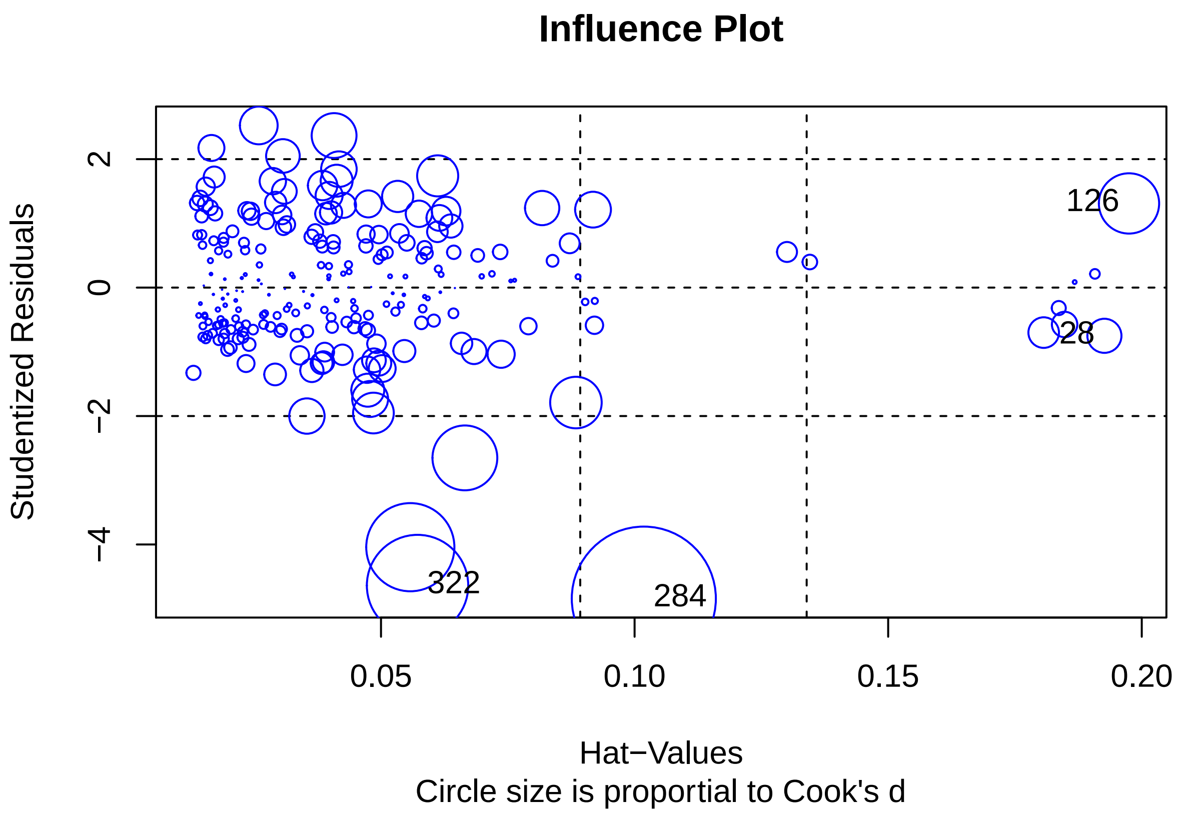
```
# leverage plots or partial regression plot
leveragePlots(fit.final, main="Partial Regression (Leverage) Plots")
```



Comment We observe some form of clustering from the plots of the predictors free.agent.91,arb.elig and arb.91 whereas the other predictors appear linear if the outliers are ignored.

(e) Outlier Detection

```
influencePlot(fit.final, id.method="identify",
    col="blue",
    main="Influence Plot",
    sub="Circle size is proportial to Cook’s d")
```



```
##         StudRes        Hat      CookD
## 28  -0.7508387 0.19263560 0.01347865
## 126  1.3106358 0.19748418 0.04212972
## 284 -4.8434316 0.10182067 0.24067859
## 322 -4.6395277 0.05720475 0.11917536
```

Comment From the plot we have few observations that are outliers.In particular the observations 322,284,28 and 126 are outliers. Observations 28 and 126 have more potential to influence the model so it would be best to remove them. We observe that the best range to work with is [-2,2] on the vertical axis(studentized residuals) and [0,0.8] on the horizontal axis(Hat-Values).

(f) Multicollinearity

```
# CONDITION NUMBER to get matrix x
# WITHOUT INTERCEPT
kappa(lm(logsalary~ runs + hits + RBI + strike.outs + errors + free.agency.elig + free.agent.91 + arb.el
```

```
## [1] 646.2765
```

```
vif(fit.final)
```

```
##             runs             hits              RBI      strike.outs
##         8.408450         9.674817         5.749452         2.611966
##           errors free.agency.elig    free.agent.91         arb.elig
##         1.362771         1.862112         1.304676         1.523478
##           arb.91
##         1.152865
```

Comment The condition number is 646 > 100(the threshold given in the question).Hence, multicollinearity could be present. According to the result from VIF we see that all of our variables are less than 10, so we can conclude that there is no multicollinearity in our model.

5.Model Deployment

```
# predicting the data
pred <- predict(fit.final, bb92, interval="prediction")
# taking exponential of the predicted values
dat.plot <- data.frame(player=1:NROW(bb92), exp(pred))
names(dat.plot)
```

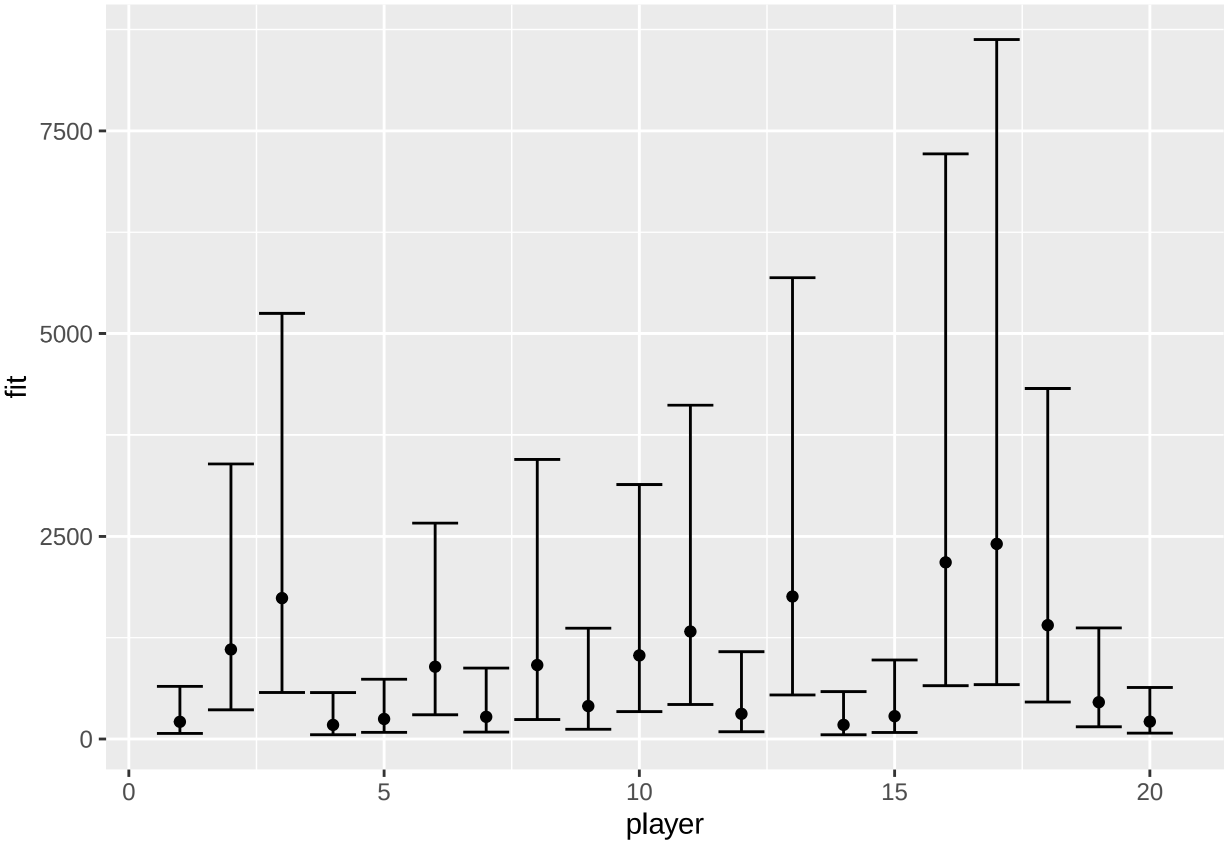
```
## [1] "player" "fit"    "lwr"    "upr"
```

```
dat.plot
```

```
##    player       fit       lwr       upr
## 1       1  212.9294  69.72982  650.2086
## 2       2 1104.6685 359.74910 3392.0656
## 3       3 1737.8128 575.11638 5251.0996
## 4       4  174.2623  52.91347  573.9056
## 5       5  247.6487  83.13297  737.7324
## 6       6  892.0858 298.81536 2663.2399
## 7       7  273.9419  85.76630  874.9844
## 8       8  912.4219 241.29997 3450.1193
## 9       9  406.6829 120.95785 1367.3442
## 10     10 1031.3648 338.94538 3138.3032
## 11     11 1325.9282 426.89714 4118.2885
## 12     12  311.1550  89.92570 1076.6379
## 13     13 1757.7131 543.13389 5688.3861
## 14     14  175.0597  52.34007  585.5151
## 15     15  282.0266  81.61276  974.5906
## 16     16 2178.6919 657.76541 7216.4002
## 17     17 2406.8278 671.52625 8626.3497
## 18     18 1404.0265 456.20805 4321.0339
## 19     19  454.2504 150.66239 1369.5746
## 20     20  215.1080  72.65206  636.8912
```

Comment The output above displays the log transform of the predicted values and their corresponding confidence intervals. We see that all the predicted values lies within each corresponding confidence interval.

```
# producing the error plot
library(ggplot2)
ggplot(dat.plot, aes(x=player, y=fit)) +
geom_errorbar(aes(ymin=lwr, ymax=upr)) + geom_point()
```



Comment We apply our final model to predict the log-salary for the new data set in the file bb92-test.csv, which contains the performance data only for 20 players, as well as the prediction intervals. From the plot the distance between the upper and lower portion of the bar or line represents the variability. We can see that almost all the players have a relatively good fitting, however for players 16 and 17 this model could not be a really good fit for them to predict their salary. Player 20 has good fitting since the variability is small.