Spring 2020
15.071 Analytics Edge Section B
Homework 5

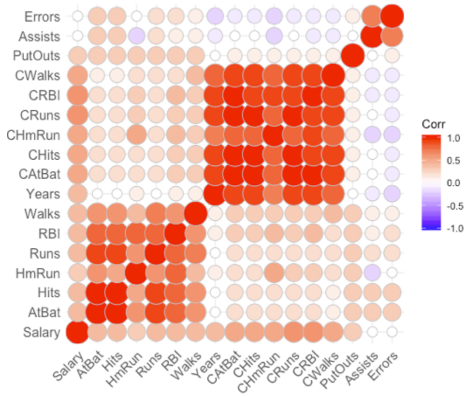
ByeongJo Kong
(kongb@mit.edu)

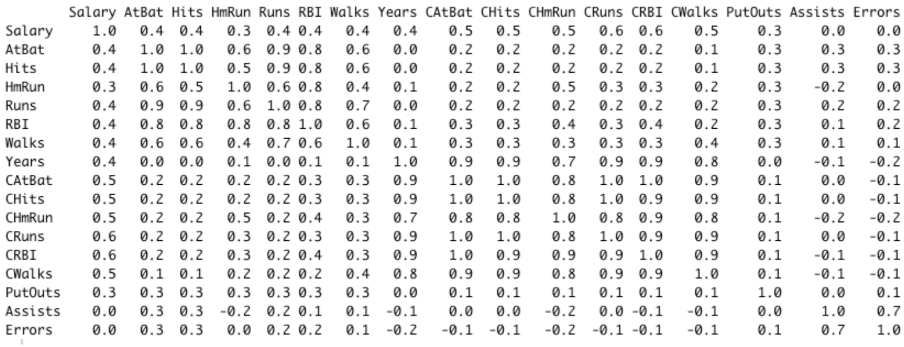
Problem 1 – Moneyball Analytics

a) Dataset

This paper aims to predict the salary of baseball players based on the dataset that contains information on 263 players from the Major League Baseball (MLB) in 1986. The first column reports the player names. The second column reports the player annual salaries (in $'000), which the paper aims to predict. The other variables report four sets of variables: offensive statistics during the season, offensive statistics over the player's career, defensive statistics, and team information.

- Name: the player's name
- Salary: the player's annual salary (in $'000)
- AtBat: Number of times at bat in the season
- Hits: Number of hits in the season
- HmRun: Number of home runs in the season
- Runs: Number of runs in the season
- RBI: Number runs enabled in the season
- Walks: Number of walks in the season
- Years: Number of years played in MLB
- CAtBat: Number of times at bat in the career
- CHits: Number of hits in the career
- CHmRun: Number of home runs in the career
- CRuns: Number of runs in the career
- CRBI: Number runs enabled in the career
- CWalks: Number of walks in the career
- PutOuts: Number of putouts in the season
- Assists: Number of assists in the season
- Errors: Number of errors in the season
- League: League in which team plays
- Division: League in which team plays
- NewLeague: League in which team plays next year

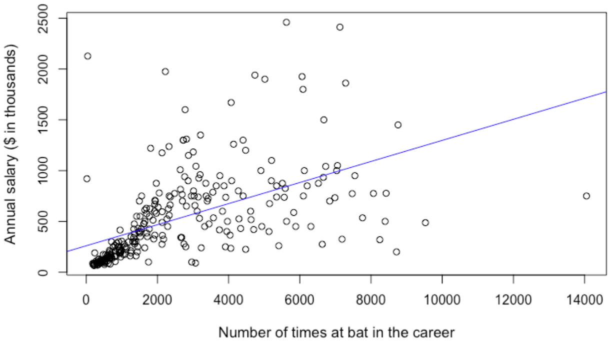


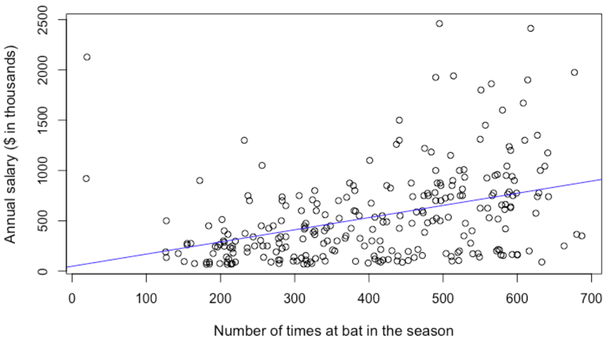


```
         Salary AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI CWalks PutOuts Assists Errors
Salary      1.0   0.4  0.4   0.3  0.4 0.4   0.4   0.4    0.5   0.5    0.5   0.6  0.6    0.5     0.3     0.0    0.0
AtBat       0.4   1.0  1.0   0.6  0.9 0.8   0.6   0.0    0.2   0.2    0.2   0.2  0.2    0.1     0.3     0.3    0.3
Hits        0.4   1.0  1.0   0.5  0.9 0.8   0.6   0.0    0.2   0.2    0.2   0.2  0.2    0.1     0.3     0.3    0.3
HmRun       0.3   0.6  0.5   1.0  0.6 0.8   0.4   0.1    0.2   0.2    0.5   0.3  0.3    0.2     0.3    -0.2    0.0
Runs        0.4   0.9  0.9   0.6  1.0 0.8   0.7   0.0    0.2   0.2    0.2   0.2  0.2    0.2     0.3     0.2    0.2
RBI         0.4   0.8  0.8   0.8  0.8 1.0   0.6   0.1    0.3   0.3    0.4   0.3  0.4    0.2     0.3     0.1    0.2
Walks       0.4   0.6  0.6   0.4  0.7 0.6   1.0   0.1    0.3   0.3    0.3   0.3  0.3    0.4     0.3     0.1    0.1
Years       0.4   0.0  0.0   0.1  0.0 0.1   0.1   1.0    0.9   0.9    0.7   0.9  0.9    0.8     0.0    -0.1   -0.2
CAtBat      0.5   0.2  0.2   0.2  0.2 0.3   0.3   0.9    1.0   1.0    0.8   1.0  1.0    0.9     0.1     0.0   -0.1
CHits       0.5   0.2  0.2   0.2  0.2 0.3   0.3   0.9    1.0   1.0    0.8   1.0  0.9    0.9     0.1     0.0   -0.1
CHmRun      0.5   0.2  0.2   0.5  0.2 0.4   0.3   0.7    0.8   0.8    1.0   0.8  0.9    0.8     0.1    -0.2   -0.2
CRuns       0.6   0.2  0.2   0.3  0.2 0.3   0.3   0.9    1.0   1.0    0.8   1.0  0.9    0.9     0.1     0.0   -0.1
CRBI        0.6   0.2  0.2   0.3  0.2 0.4   0.3   0.9    1.0   0.9    0.9   0.9  1.0    0.9     0.1    -0.1   -0.1
CWalks      0.5   0.1  0.1   0.2  0.2 0.2   0.4   0.8    0.9   0.9    0.8   0.9  0.9    1.0     0.1    -0.1   -0.1
PutOuts     0.3   0.3  0.3   0.3  0.3 0.3   0.3   0.0    0.1   0.1    0.1   0.1  0.1    0.1     1.0     0.0    0.1
Assists     0.0   0.3  0.3  -0.2  0.2 0.1   0.1  -0.1    0.0   0.0   -0.2   0.0 -0.1   -0.1     0.0     1.0    0.7
Errors      0.0   0.3  0.3   0.0  0.2 0.2   0.1  -0.2   -0.1  -0.1   -0.2  -0.1 -0.1   -0.1     0.1     0.7    1.0
```

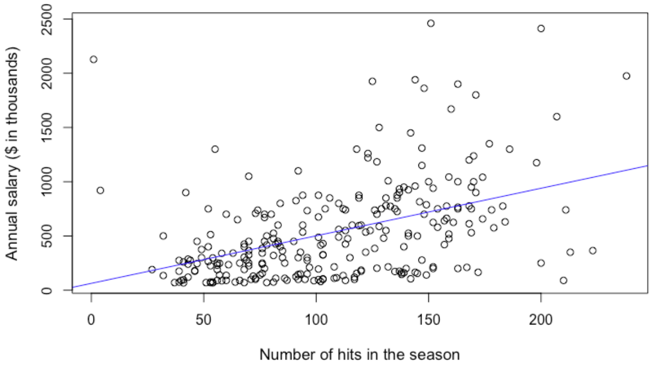
First, I extracted only the numerical predictors from the dataset to look at their correlations between each other. From the upper-left matrix, we can see that there are high correlations among the player's performance records, such as the numbers of walks, RBI, runs, homeruns, hits, and at bat; it is more prominent over the period of player's career than the season.

If we look at the predictors that are correlated with the salary, three most highly correlated ones are CAtBat (Number of times at bat in the career), AtBat (Number of times at bat in the season), and Hits (Number of hits in the season). As can be seen in the matrix above, we can learn from the plots below that there is a stronger correlation between the salary and the player's at bat record over the period of career than the season. This is presumably due to the larger sampling of observations over a longer period of time.





Likewise, the player's salary increases as the number of hits in the season increases.



R codes

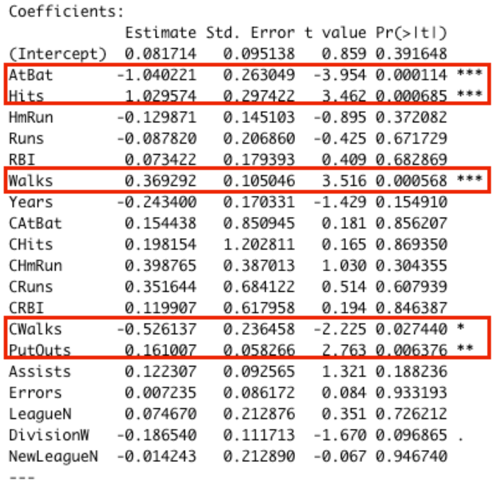
```
hitters_raw <- read.csv("Hitters.csv")
hitters_num <-hitters[,2:18]
ggcorrplot(round(cor(hitters_num),1), method="circle")
round(cor(hitters_num),1)

lm.mod <- lm(Salary ~., data = hitters_num)
summary(lm.mod)

#Plot Salary vs. CAtBat
plot(hitters_num$CAtBat,hitters_num$Salary,
    xlab='Number of times at bat in the career',
    ylab='Annual salary ($ in thousands)',cex.lab=1.1)
abline(lm(hitters_num$Salary ~ hitters_num$CAtBat), col="blue")

#Plot Salary vs. AtBat
```

```
plot(hitters_num$AtBat,hitters_num$Salary,
    xlab='Number of times at bat in the season',
    ylab='Annual salary ($ in thousands)',cex.lab=1.1)
abline(lm(hitters_num$Salary ~ hitters_num$AtBat), col="blue")

#Plot Salary vs. Hits
plot(hitters_num$Hits,hitters_num$Salary,
    xlab='Number of hits in the season',
    ylab='Annual salary ($ in thousands)',cex.lab=1.1)
abline(lm(hitters_num$Salary ~ hitters_num$Hits), col="blue")
```

b) i) Fit a linear regression model using training set

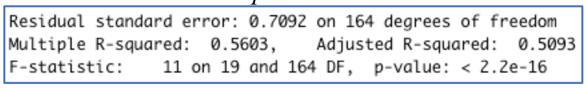
The data is normalized and split into a training set and a test set in a ratio of 7:3. The linear regression model was created using all the predictors of the training set. The summary of linear regression model indicates that variables with the high significance at the level of 95 % (p values < 0.05) are AtBat, Hits, Walks, CWalks, and PutOuts. These variables have the most impact on the salary variables.

From the signs of coefficient values, we can learn that the number of times at bat in the season and the number of walks in the career are negatively correlated with the salary. Likewise, the numbers of hits, walks, putouts in the season are positively correlated with the salary.



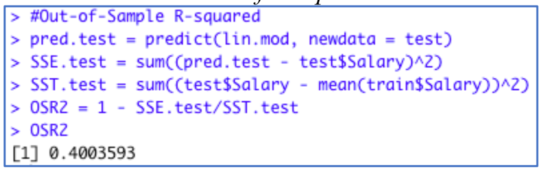
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.081714   0.095138   0.859 0.391648
AtBat       -1.040221   0.263049  -3.954 0.000114 ***
Hits         1.029574   0.297422   3.462 0.000685 ***
HmRun       -0.129871   0.145103  -0.895 0.372082
Runs        -0.087820   0.206860  -0.425 0.671729
RBI          0.073422   0.179393   0.409 0.682869
Walks        0.369292   0.105046   3.516 0.000568 ***
Years       -0.243400   0.170331  -1.429 0.154910
CAtBat       0.154438   0.850945   0.181 0.856207
CHits        0.198154   1.202811   0.165 0.869350
CHmRun       0.398765   0.387013   1.030 0.304355
CRuns        0.351644   0.684122   0.514 0.607939
CRBI         0.119907   0.617958   0.194 0.846387
CWalks      -0.526137   0.236458  -2.225 0.027440 *
PutOuts      0.161007   0.058266   2.763 0.006376 **
Assists      0.122307   0.092565   1.321 0.188236
Errors       0.007235   0.086172   0.084 0.933193
LeagueN      0.074670   0.212876   0.351 0.726212
DivisionW   -0.186540   0.111713  -1.670 0.096865 .
NewLeagueN  -0.014243   0.212890  -0.067 0.946740
---
```

In-sample R^2



```
Residual standard error: 0.7092 on 164 degrees of freedom
Multiple R-squared:  0.5603,    Adjusted R-squared:  0.5093
F-statistic:    11 on 19 and 164 DF,  p-value: < 2.2e-16
```

Out-of-sample R^2



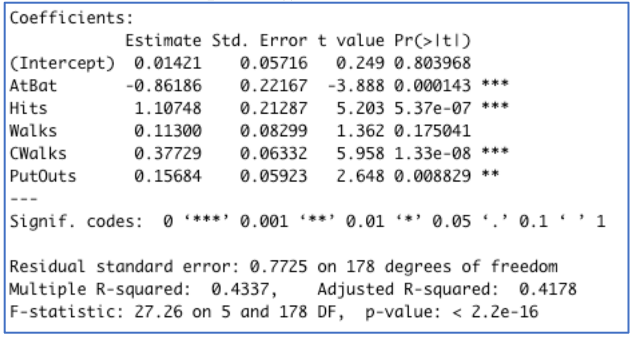
```
> #Out-of-Sample R-squared
> pred.test = predict(lin.mod, newdata = test)
> SSE.test = sum((pred.test - test$Salary)^2)
> SST.test = sum((test$Salary - mean(train$Salary))^2)
> OSR2 = 1 - SSE.test/SST.test
> OSR2
[1] 0.4003593
```

The in-sample and out-of-sample $R^2$are 0.5603 and 0.4004 respectively. Considering the large number of outliers observed in the visualization in **a)** and the value of residual standard error (0.7092) shown above, these outcomes are expected.

b) ii) Fit a restricted linear regression model selecting only the predictors with significance

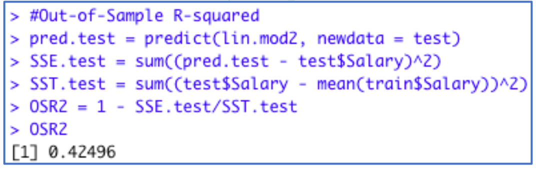
Five predictors with p value under 0.05 were selected to fit a linear regression model. The in-sample and out-of-sample R^2 are 0.4337 and 0.42496 respectively. The accuracy of in-sample prediction has worsened, whereas that of out-of-sample has improved slightly. These poor results are unexpected, considering only the predictors with high significance were selected to fit the model.

In-sample coefficients and R^2



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.01421    0.05716   0.249 0.803968    
AtBat       -0.86186    0.22167  -3.888 0.000143 ***
Hits         1.10748    0.21287   5.203 5.37e-07 ***
Walks        0.11300    0.08299   1.362 0.175041    
CWalks       0.37729    0.06332   5.958 1.33e-08 ***
PutOuts      0.15684    0.05923   2.648 0.008829 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7725 on 178 degrees of freedom
Multiple R-squared:  0.4337,	Adjusted R-squared:  0.4178 
F-statistic: 27.26 on 5 and 178 DF,  p-value: < 2.2e-16
```

Out-of-sample R^2



```
> #Out-of-Sample R-squared
> pred.test = predict(lin.mod2, newdata = test)
> SSE.test = sum((pred.test - test$Salary)^2)
> SST.test = sum((test$Salary - mean(train$Salary))^2)
> OSR2 = 1 - SSE.test/SST.test
> OSR2
[1] 0.42496
```

Only includes only those significant variables with stars.
Full -> remove insignificant variables -> remove multicollinear variable

R codes

```
#Normalize and split data
pp <- preProcess(hitters_raw, method=c("center", "scale"))
Hitters <- predict(pp, hitters_raw)
set.seed(15071)
train.obs <- sort(sample(seq_len(nrow(Hitters)), 0.7*nrow(Hitters)))
train <- Hitters[train.obs,2:21]
test <- Hitters[-train.obs,2:21]

#Fit a Linear Regression and predict the test set
lin.mod <- lm(train$Salary ~ ., data = train)
pred.train = predict(lin.mod, newdata = train)
summary(lin.mod)

#Out-of-Sample R-squared
pred.test = predict(lin.mod, newdata = test)
SSE.test = sum((pred.test - test$Salary)^2)
SST.test = sum((test$Salary - mean(train$Salary))^2)
OSR2 = 1 - SSE.test/SST.test

#Fit a restricted Linear Regression with variables with significance
head(train)
lin.mod2 <- lm(Salary ~ AtBat+Hits+Walks+CWalks+PutOuts,
            data = train)
pred.train = predict(lin.mod2, newdata = train)
summary(lin.mod2)

#Out-of-Sample R-squared
```

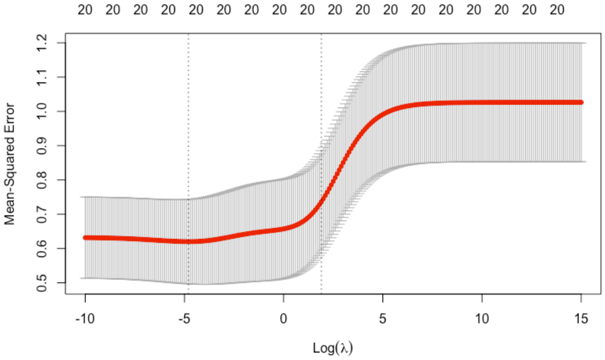
```
pred.test = predict(lin.mod2, newdata = test)
SSE.test = sum((pred.test - test$Salary)^2)
SST.test = sum((test$Salary - mean(train$Salary))^2)
OSR2 = 1 - SSE.test/SST.test
```

c) Regularization

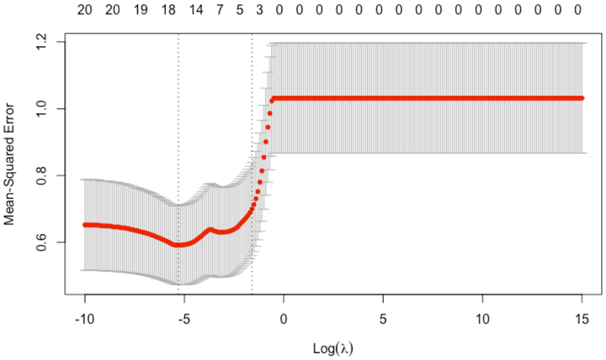
i) Train Ridge regression and LASSO models with 10-fold cross-validation

Plots of cross-validated Mean Squared Error as a function of λ:

Ridge Regression



LASSO

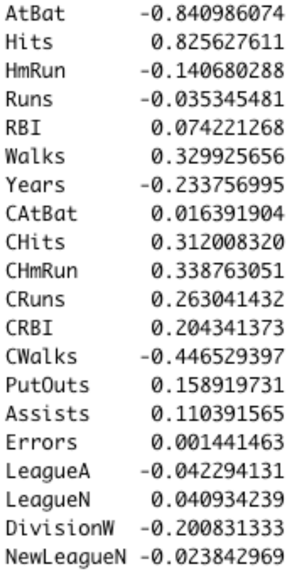


The value of λ that minimizes the Mean Squared Error for Ridge Regression and LASSO are 0.00552 and 0.00274 respectively.

ii) Retrain Ridge Regression and LASSO with the selected λ values

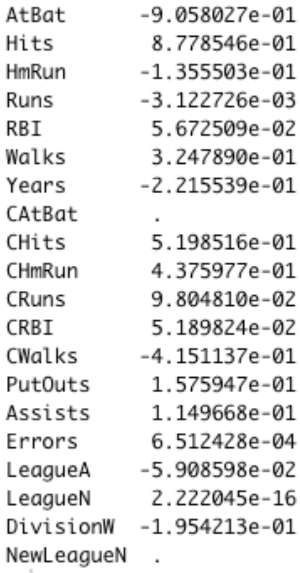
The coefficients of ridge regression and LASSO are as shown below.

Ridge Regression



```
AtBat       -0.840986074
Hits         0.825627611
HmRun       -0.140680288
Runs        -0.035345481
RBI          0.074221268
Walks        0.329925656
Years       -0.233756995
CAtBat       0.016391904
CHits        0.312008320
CHmRun       0.338763051
CRuns        0.263041432
CRBI         0.204341373
CWalks      -0.446529397
PutOuts      0.158919731
Assists      0.110391565
Errors       0.001441463
LeagueA     -0.042294131
LeagueN      0.040934239
DivisionW   -0.200831333
NewLeagueN  -0.023842969
```

LASSO



```
AtBat       -9.058027e-01
Hits         8.778546e-01
HmRun       -1.355503e-01
Runs        -3.122726e-03
RBI          5.672509e-02
Walks        3.247890e-01
Years       -2.215539e-01
CAtBat       .
CHits        5.198516e-01
CHmRun       4.375977e-01
CRuns        9.804810e-02
CRBI         5.189824e-02
CWalks      -4.151137e-01
PutOuts      1.575947e-01
Assists      1.149668e-01
Errors       6.512428e-04
LeagueA     -5.908598e-02
LeagueN      2.222045e-16
DivisionW   -1.954213e-01
NewLeagueN   .
```

The signs of coefficients remain the same as the ones in part b) i); AtBat, HmRun, Runs, Years, CWalks, LeagueA, DivisionW are still negative. As for LASSO, CAtBat and NewLeagueN shrank to zero, whereas in the ridge regression only few variables, such as CAtBat, showed a significant reduction in their coefficient values.

Model	**In-sample R^2**	**Out-of-sample R^2**
Linear regression	0.5603	0.4004
Ridge regression (λ = 0.00552)	0.5581	0.4266
LASSO (λ = 0.00274)	0.5586	0.4181

The table above shows the performance comparison of each model based on the *in-sample* and *out-of-sample R^2*. Among the three models, the ridge regression shows the worst performance in the in-sample prediction, but the best performance in the out-of-sample prediction. The LASSO model had a slightly better results than the ridge regression in the in-sample prediction, but worse performance in the out-of-sample prediction. This is caused by the overfitting, which can be alleviated by selecting the higher value of *λ*.

iii) The LASSO model often results in many coefficients set to zero, and it occurred in my model as well. In the coefficient data in part ii), the coefficient values of CAtBat and NewLeagueN are set to zero during the process of cross-validation of *λ*.

R codes

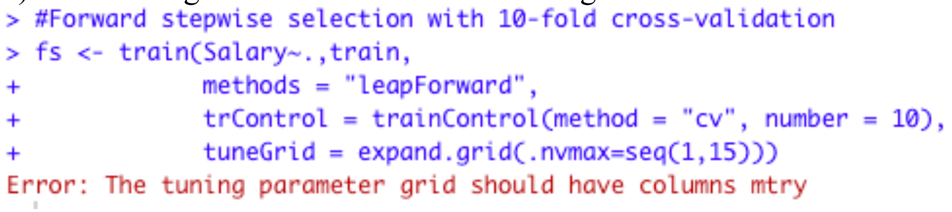
```
### Run Ridge Regression and LASSO in the train Set
x.train=model.matrix(Salary~.-1,data=train)
y.train=train$Salary
x.test=model.matrix(Salary~.-1,data=test)
y.test=test$Salary

all.lambdas <- c(exp(seq(15, -10, -.1)))
cv.ridge=cv.glmnet(x.train,y.train,alpha=0,lambda=all.lambdas, nfold=10)
cv.lasso=cv.glmnet(x.train,y.train,alpha=1,lambda=all.lambdas, nfold=10)

plot(cv.ridge)
plot(cv.lasso)
cv.ridge$lambda.min
cv.lasso$lambda.min

### Prediction on the train and test sets
# Re-train ridge regression and LASSO models on full training set.
ridge.final <- glmnet(x.train, y.train, alpha=0, lambda=cv.ridge$lambda.min)
lasso.final <- glmnet(x.train, y.train, alpha=1, lambda=cv.lasso$lambda.min)

ridge.final$beta
lasso.final$beta
```

```
pred.train.ridge <- predict(ridge.final, x.train)
pred.train.lasso <- predict(lasso.final, x.train)

R2.ridge <- 1-sum((pred.train.ridge-train$Salary)^2)/sum((mean(train$Salary)-train$Salary)^2)
R2.lasso <- 1-sum((pred.train.lasso-train$Salary)^2)/sum((mean(train$Salary)-train$Salary)^2)

pred.test.ridge <- predict(ridge.final, x.test)
pred.test.lasso <- predict(lasso.final, x.test)

OSR2.ridge <- 1-sum((pred.test.ridge-test$Salary)^2)/sum((mean(train$Salary)-test$Salary)^2)
OSR2.lasso <- 1-sum((pred.test.lasso-test$Salary)^2)/sum((mean(train$Salary)-test$Salary)^2)
```

d) i) Retrain LASSO model with reselected value of λ

The value of λ that led to the best LASSO model consisting of at most nvars nonzero coefficients is **0.082085**. The in-sample R^2 is 0.4563 and out-of-sample R^2 is 0.514. With the newly selected value of λ, the out-of-sample prediction improved significantly compared to the one conducted by the LASSO model in part c) iii). The six predictors selected by the model are Hits, Walks, CHits, CRBI, PutOuts, and DivisionsW.

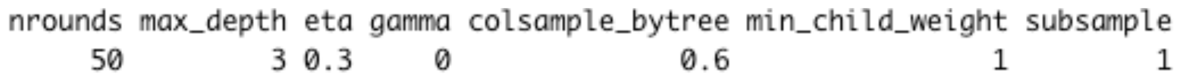
ii) I could not get the result due to the error I got as below.



```
> #Forward stepwise selection with 10-fold cross-validation
> fs <- train(Salary~.,train,
+             methods = "leapForward",
+             trControl = trainControl(method = "cv", number = 10),
+             tuneGrid = expand.grid(.nvmax=seq(1,15)))
Error: The tuning parameter grid should have columns mtry
```

e) Use XGBoost framework

Below are the best values for the hyperparameters that was cross-validated with the XGBoost model.



```
nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
     50         3 0.3     0              0.6                1         1
```

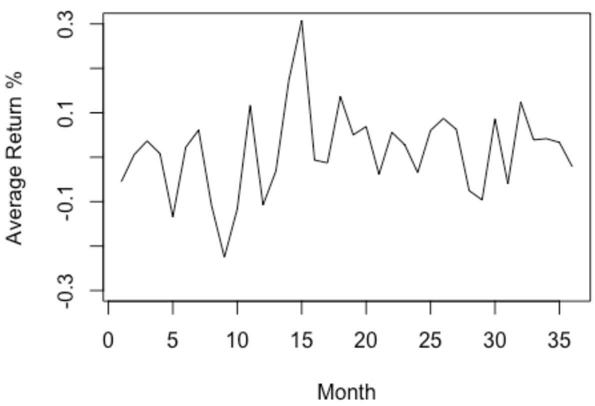
With this model, the in-sample R^2 is 0.9946 and out-of-sample R^2 is 0.63057. This result is a significant improvement comparing to the ones obtained through the previous models; linear regression, ridge regression, and LASSO.

Problem 2 – Clustering Stock Returns

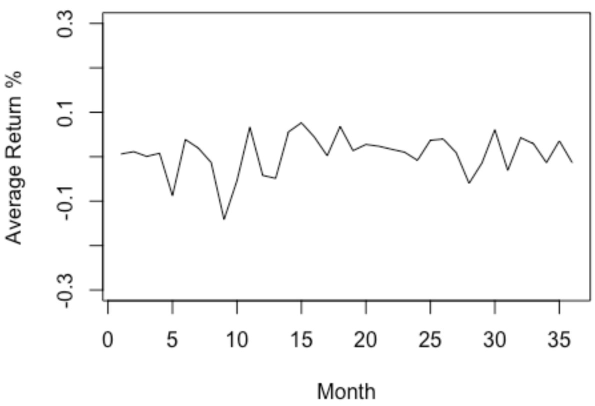
a) The number of companies in each industry sector

Industry Sector	Number of Companies
Consumer Discretionary	69
Consumer Staples	32
Energy	38
Financials	78
Health Care	44
Industrials	55
Information Technology	56
Materials	28
Telecommunications Services	5
Utilities	28

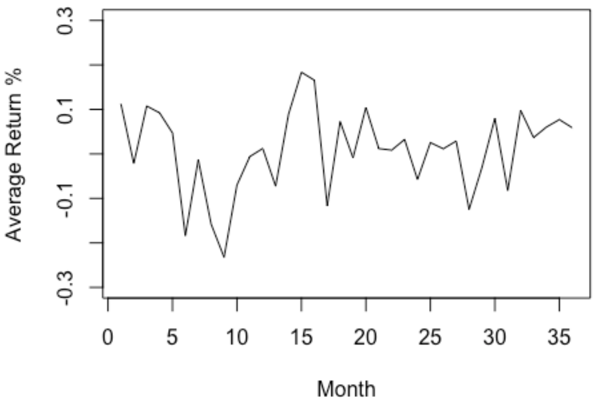
Consumer Discretionary



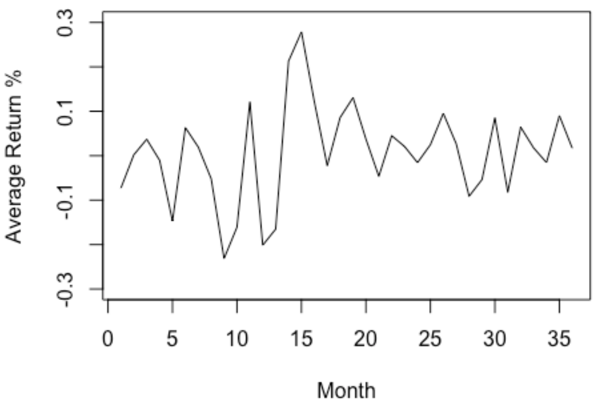
Consumer Staples

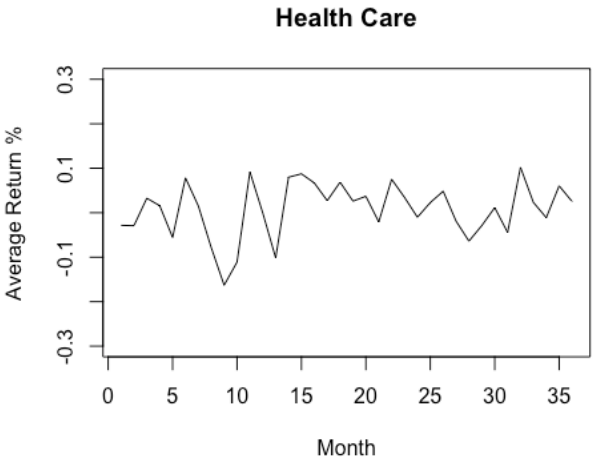


Energy

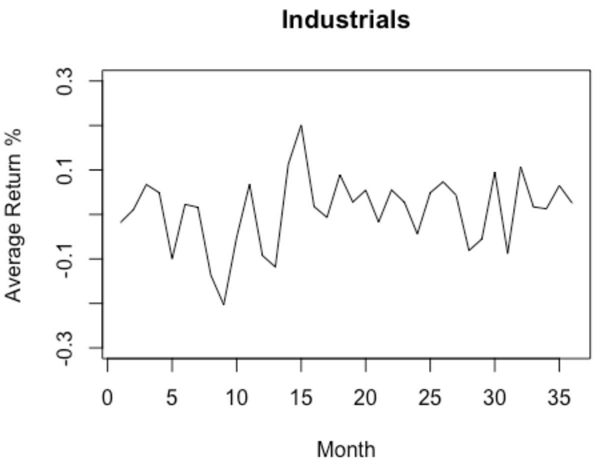


Financials

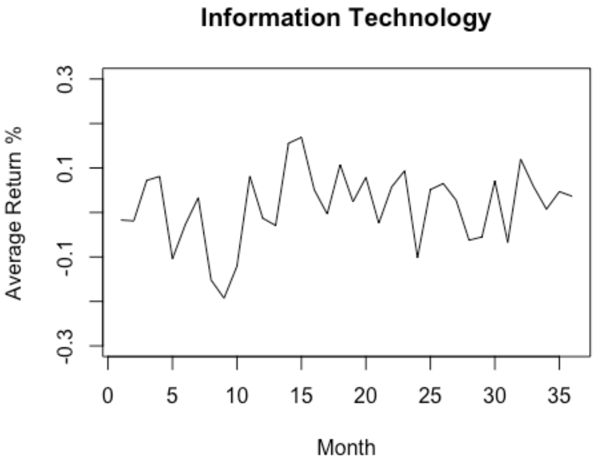




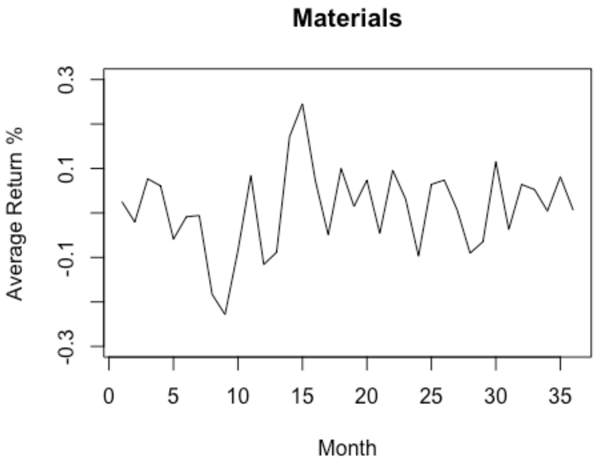
Health Care



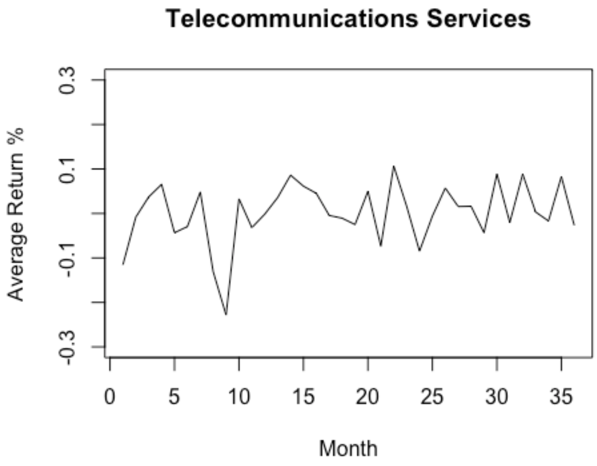
Industrials



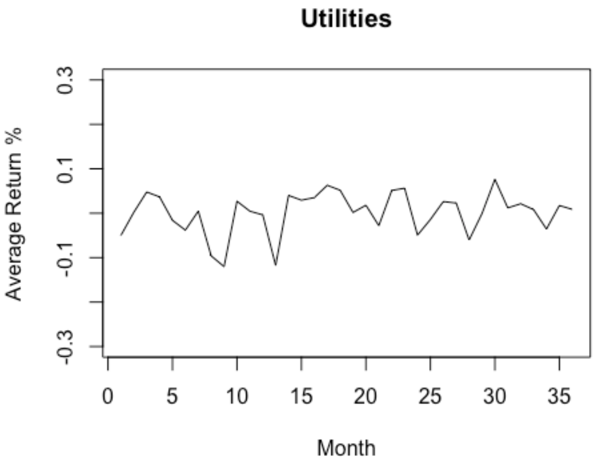
Information Technology



Materials



Telecommunications Services



Utilities

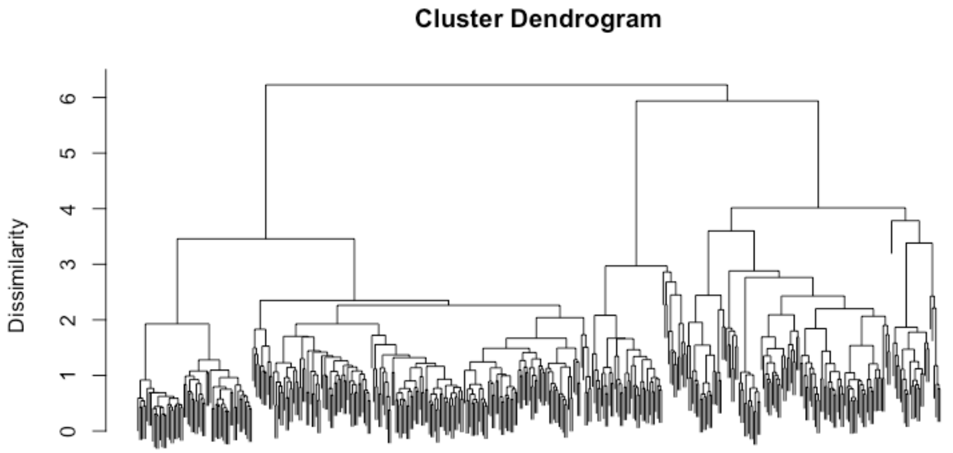
You can see above the plots of the average stock return between 01/2008 and 12/2010. The industries that had the worst average return in September 2008 are Consumer Discretionary, Energy, Financial, Industrials, IT, Materials, and Telecommunications Services. On the other hand, industries such as Consumer Staples, Healthcare, Utilities are relatively less volatile than the other industries, which is because they are the basic necessities that people need to continue to consume despite the economic recession.

R codes

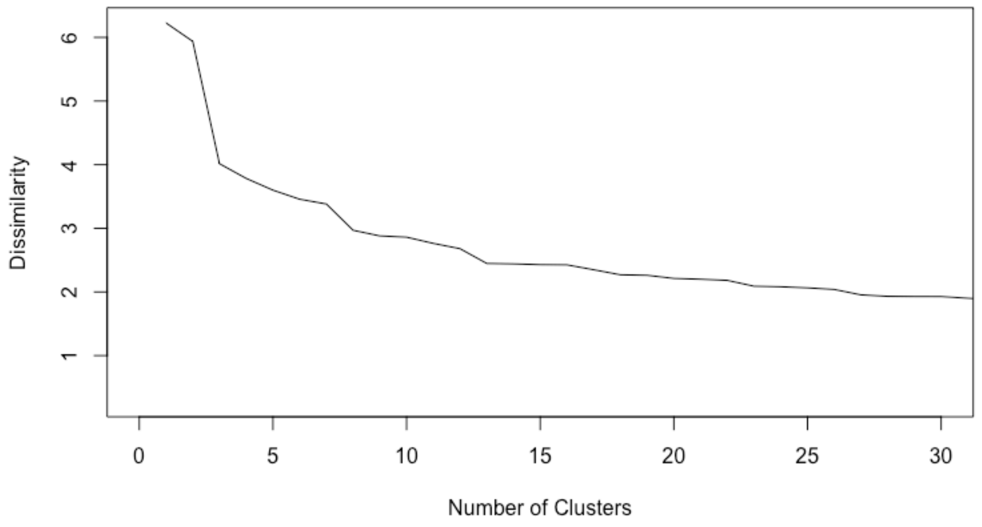
```
grep("avg200801",colnames(data))
grep("avg201012",colnames(data))

summary(data$Industry)
industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[1,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Consumer Discretionary")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[2,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Consumer Staples")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[3,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Energy")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[4,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Financials")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[5,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Health Care")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[6,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Industrials")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[7,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Information Technology")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[8,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
```

```
title("Materials")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[9,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Telecommunications Services")

industry = aggregate(.~Industry, data=data, mean)[-2]
plot(t(industry[10,][25:60]), type="l", ylim=c(-0.3, 0.3),
    xlab = "Month", ylab="Average Return %")
title("Utilities")
```

b) Cluster the stocks according to their monthly returns



Scree plot:



The dissimilarity decreases drastically between the number of clusters 3 and 7. From the dendrogram, in order to yield a final cluster of 3, a dissimilarity equal or less than 4 are agglomerated, meaning they are still too big dissimilarity. Therefore, 7 seems to be the more reasonable number of clusters.

R codes

```
d = dist(returns)
mod.hclust = hclust(d, method="ward.D2")
plot(mod.hclust, labels=F, xlab="", ylab="Dissimilarity", sub="")

dissim.hc = data.frame(k=seq_along(mod.hclust$height), dissimilarity=rev(mod.hclust$height))
plot(dissim.hc$k, dissim.hc$dissimilarity, type="l", xlim=c(0,30),
    xlab="Number of Clusters", ylab="Dissimilarity")
```

c) Use the cutree function to cut the dendrogram tree

The table below shows the number of companies in each cluster and industry.

	Number of Companies in each Cluster						
Industry Sector	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Consumer Discretionary	29	22	1	3		4	10
Consumer Staples	12	3	16			1	
Energy	3	2		33			
Financials	18	32	2		1	16	9
Health Care	27		15				2
Industrials	37	12	1	4			1
Information Technology	38	17		1			
Materials	16	1		7			4
Telecommunications Services	2		2	1			
Utilities	1		25	2			
	183	89	62	51	1	21	26

The average returns by cluster in October 2008 and in March 2009 is as shown below.

	Cluster						
	1	**2**	**3**	**4**	**5**	**6**	**7**
October 2008	-0.172	-0.276	-0.089	-0.260	-0.493	-0.112	-0.356
March 2009	0.113	0.195	0.051	0.108	0.941	0.241	0.219

Looking at the two tables above:

- Cluster 1 has the average returns
- Cluster 2 tends to have a high risk of poor returns during the recession and gaining a good return during the economic recovery

- Cluster 3's return is very stable which includes industries that provide basic necessities to people
- Cluster 4 is highly concentrated on the performance of Energy industry
- Cluster 5 is solely relied on the performance of Financial sector
- Cluster 6 is highly relying on the Financial sector
- Cluster 7 is including industries that are very dependent on the economic situation

R codes

```
h.clusters = cutree(mod.hclust, 7)
h.cluster.result = data.frame(h.clusters)
data.h = data.frame(data$Industry)
data.h$cluster = as.factor(h.cluster.result$h.clusters)
result = aggregate(returns, by=list(h.clusters), mean) %>% select(-Group.1)
table(h.clusters)
```

d) Cluster the data using the k-means clustering algorithm

The table below shows the number of companies in each cluster and industry using the k-mean clustering algorithm

	Number of Companies in each Cluster						
Industry Sector	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Consumer Discretionary	18	29	12	7	3		
Consumer Staples	1	7		24			
Energy	1	2		2		33	
Financials	6	26	17	9	18		2
Health Care		25	2	17			
Industrials	8	37	3	4		3	
Information Technology	24	30		2			
Materials	3	10	4	5		6	
Telecommunications Services				4		1	
Utilities		1		27			
	61	167	38	101	21	43	2

From the table, we can see that many of the companies now shifted to Cluster 2 and 4, and some to 5 and 6 as well. Cluster 1, 3 and 7 showed a huge decrease in their numbers. Looking at the industries, it is noticeable that to the change is not as drastic as the cluster.

As shown below, many clusters match up, which are indicated as zeros.

		***k*-means Clustering**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Hierarchical Clustering	**1**	13	135	0	34	0	1	0

	2	44	26	14	3	2	0	0
	3	0	0	0	62	0	0	0
	4	3	5	0	1	0	42	0
	5	0	0	0	0	0	0	1
	6	0	1	0	1	19	0	0
	7	1	0	24	0	0	0	1

R codes

```
km = kmeans(returns, centers=7, iter.max=100)
km.clusters = km$cluster
km.cluster.result = data.frame(km.clusters)
data.km = data.frame(data$Industry)
data.km$cluster = as.factor(km.cluster.result$km.clusters)
data.km
names(km)
km.centroids = km$centers
km$tot.withinss
km.size = km$size
table(h.clusters, km.clusters)
```

e) Short paragraph to an investor

We build a model using the monthly average returns of 433 stocks among the S&P500 from March 2006 to February 2016. Based on the model, there are 7 final clusters that includes stocks from different industries that show similar patterns in return. Referring to this model, investors can select the industries or stocks they want to invest in, depending on their preferences to aggressive or conservative investments.