Data Analytics and Machine Learning PS4

Jiaming Huang, An Yang, Yichu Li, Hogun Kim

2019/4/24

Question 1: Automatic Stock Picking Algorithm

a. Download the data. The firm???level characteristics you will use are lnIssue, lnProf, lnInv, and lnME. For each of these four characteristics, create new, additional
characteristics as the squared value of the original characteristic. Name the new characteristics the same as the orignal, but with a "2" at the end. For instance, for lnProf, the squared value should be lnProf2. Further, create additional characteristics by multiplying each characteristic with lnME (except for lnME itself, which you already have squared). To name these, add _ME at the end. Thus, lnProf interacted with lnME is
named lnProf_ME. You should have now gone from 4 to 11 characteristics.

```
library(foreign)
```

```
## Warning: package 'foreign' was built under R version 3.4.4
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.4.4
```

```
setwd("/Users/jiaminghuang/Downloads")

loan <- as.data.table(read.dta("StockRetAcct_insample.dta"))
loan[,Excess_Ret := exp(lnAnnRet)-exp(lnRf)]
loan1 <- loan[,c("FirmID","year","Excess_Ret","lnIssue","lnProf","lnInv","lnME")]
loan1[,lnIssue2 := lnIssue^2]
loan1[,lnProf2 := lnProf^2]
loan1[,lnInv2 := lnInv^2]
loan1[,lnME2 := lnME^2]
loan1[,lnIssue_ME := lnIssue * lnME]
loan1[,lnProf_ME := lnProf * lnME]
loan1[,lnInv_ME := lnInv * lnME]
loan1
```

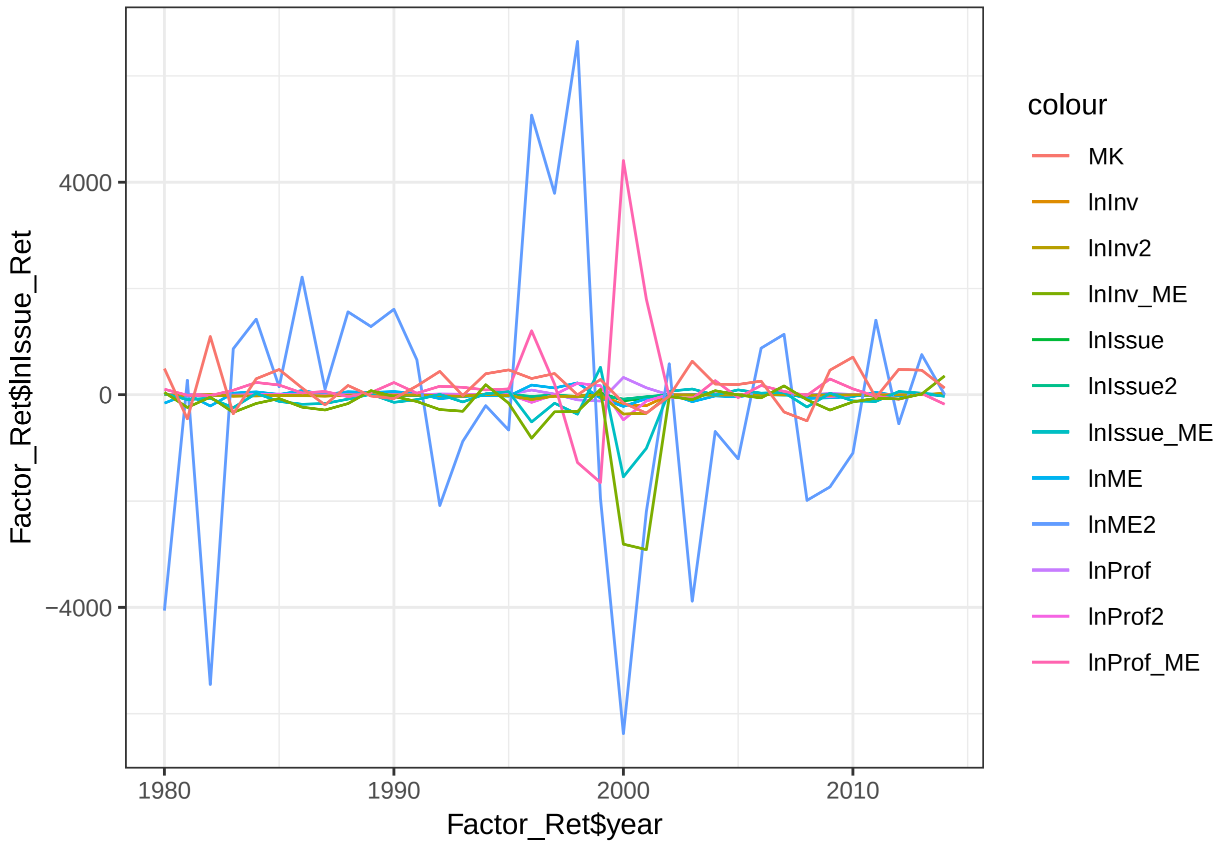
```
##        FirmID year  Excess_Ret     lnIssue      lnProf       lnInv
##     1:      6 1980  0.35639972  0.03134417  0.20176707  0.09362611
##     2:      6 1981 -0.39109730  0.04421350  0.21566088  0.08724214
##     3:      6 1982  0.06555250 -0.06819496  0.18408749  0.11166344
##     4:      6 1983  0.53803206 -0.07177968  0.16553123 -0.03311720
##     5:     10 1991 -0.46143337  0.11520413  0.23978782  0.30005118
##    ---
## 70752:  20314 2010  0.21933699          NA          NA          NA
## 70753:  20314 2011  0.07226863          NA -0.89174885  1.05899596
## 70754:  20314 2012  2.42904238  0.21500303 -1.26431298  0.61405981
## 70755:  20314 2013  1.23447438  0.26048917 -1.16386342  0.44577339
## 70756:  20314 2014  0.11629506  0.18348676  0.03606884  0.77437043
##             lnME     lnIssue2     lnProf2      lnInv2     lnME2 lnIssue_ME
```

```
##      1: 12.58147 0.0009824569 0.040709951 0.008765849 158.2934  0.3943558
##      2: 12.90800 0.0019548332 0.046509617 0.007611191 166.6164  0.5707076
##      3: 12.55777 0.0046505530 0.033888202 0.012468723 157.6977 -0.8563770
##      4: 12.56195 0.0051523219 0.027400589 0.001096749 157.8027 -0.9016930
##      5: 11.56583 0.0132719906 0.057498197 0.090030712 133.7685  1.3324315
##    ---
## 70752: 14.61343           NA          NA          NA 213.5523         NA
## 70753: 14.92373           NA 0.795216004 1.121472448 222.7178         NA
## 70754: 15.00809 0.0462263023 1.598487318 0.377069445 225.2426  3.2267838
## 70755: 16.38328 0.0678546055 1.354578062 0.198713918 268.4119  4.2676674
## 70756: 17.21366 0.0336673910 0.001300961 0.599649566 296.3099  3.1584779
##           lnProf_ME   lnInv_ME
##      1:   2.5385268  1.1779543
##      2:   2.7837499  1.1261212
##      3:   2.3117291  1.4022443
##      4:   2.0793957 -0.4160168
##      5:   2.7733454  3.4703413
##    ---
## 70752:          NA         NA
## 70753: -13.3082206 15.8041717
## 70754: -18.9749170  9.2158619
## 70755: -19.0679023  7.3032311
## 70756:   0.6208766 13.3297458
```

(i) For each year in the sample, cross???sectionally demean each of the 11 characteristics. That is, for each characteristic and each year subtract the average value of that characteristic across stocks. Then add as final characteristic a column of 1's to the dataset. This effectively inserts an intercept in the relation between the MVE portfolio weight and the characteristics.
Next calculate the factor portfolio returns corresponding to each of these 12 characteristics, as explained at the end of the Topic 4 note. Note that the factor corresponding to the constant is simply an equal???weighted portfolio of all stocks (the "market"). The overall idea is that with this approach you have a market factor and long???short characteristics factors. We do not normalize characteristics to have unit variance, as (as an empirical observation) the magnitude of the spread in characteristics across stocks is informative for the portfolio weights.
Calculate and report the factor sample means and sample variance???covariance matrix for these 12 annual factor returns, as well as the factors' sample Sharpe ratios.

```
##demean process
loan1[,lnIssue := ( lnIssue - mean(lnIssue,na.rm = TRUE)), by = year]
loan1[,lnProf := ( lnProf - mean(lnProf,na.rm = TRUE)), by = year]
loan1[,lnInv := ( lnInv - mean(lnInv,na.rm = TRUE)), by = year]
loan1[,lnME := ( lnME - mean(lnME,na.rm = TRUE)), by = year]
loan1[,lnIssue2 := ( lnIssue2 - mean(lnIssue2,na.rm = TRUE)), by = year]
loan1[,lnProf2 := ( lnProf2 - mean(lnProf2,na.rm = TRUE)), by = year]
loan1[,lnInv2 := ( lnInv2 - mean(lnInv2,na.rm = TRUE)), by = year]
loan1[,lnME2 := ( lnME2 - mean(lnME2,na.rm = TRUE)), by = year]
loan1[,lnIssue_ME := ( lnIssue_ME - mean(lnIssue_ME,na.rm = TRUE)), by = year]
loan1[,lnProf_ME := ( lnProf_ME - mean(lnProf_ME,na.rm = TRUE)), by = year]
loan1[,lnInv_ME := ( lnInv_ME - mean(lnInv_ME,na.rm = TRUE)), by = year]
loan1[,Intercept:=1]
# calculate factor return
lnIssue_Ret = loan1[,.(lnIssue_Ret = sum(Excess_Ret*lnIssue,na.rm = TRUE)),by = year]
lnProf_Ret = loan1[,.(lnProf_Ret = sum(Excess_Ret*lnProf,na.rm = TRUE)),by = year]
```

```r
lnInv_Ret = loan1[,.(lnInv_Ret = sum(Excess_Ret*lnInv,na.rm = TRUE)),by = year]
lnME_Ret = loan1[,.(lnME_Ret = sum(Excess_Ret*lnME,na.rm = TRUE)),by = year]
lnIssue2_Ret = loan1[,.(lnIssue2_Ret = sum(Excess_Ret*lnIssue2,na.rm = TRUE)),by = year]
lnProf2_Ret = loan1[,.(lnProf2_Ret = sum(Excess_Ret*lnProf2,na.rm = TRUE)),by = year]
lnInv2_Ret = loan1[,.(lnInv2_Ret = sum(Excess_Ret*lnInv2,na.rm = TRUE)),by = year]
lnME2_Ret = loan1[,.(lnME2_Ret = sum(Excess_Ret*lnME2,na.rm = TRUE)),by = year]
lnIssue_ME_Ret = loan1[,.(lnIssue_ME_Ret = sum(Excess_Ret*lnIssue_ME,na.rm = TRUE)),by = year]
lnProf_ME_Ret = loan1[,.(lnProf_ME_Ret = sum(Excess_Ret*lnProf_ME,na.rm = TRUE)),by = year]
lnInv_ME_Ret = loan1[,.(lnInv_ME_Ret = sum(Excess_Ret*lnInv_ME,na.rm = TRUE)),by = year]
MK_Ret = loan1[,.(MK_Ret = sum(Excess_Ret*Intercept,na.rm = TRUE)),by = year]
Factor_Ret = merge(lnIssue_Ret,lnProf_Ret,by = "year")
Factor_Ret = merge(Factor_Ret,lnInv_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnME_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnIssue2_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnProf2_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnInv2_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnME2_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnIssue_ME_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnProf_ME_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,lnInv_ME_Ret, by = "year")
Factor_Ret = merge(Factor_Ret,MK_Ret, by = "year")
ggplot(Factor_Ret,aes(Factor_Ret$year))+
  geom_line(aes(y = Factor_Ret$lnIssue_Ret, col = "lnIssue"))+
  geom_line(aes(y = Factor_Ret$lnProf_Ret, col = "lnProf"))+
  geom_line(aes(y = Factor_Ret$lnInv_Ret, col = "lnInv"))+
  geom_line(aes(y = Factor_Ret$lnME_Ret, col = "lnME"))+
  geom_line(aes(y = Factor_Ret$lnIssue2_Ret, col = "lnIssue2"))+
  geom_line(aes(y = Factor_Ret$lnProf2_Ret, col = "lnProf2"))+
  geom_line(aes(y = Factor_Ret$lnInv2_Ret, col = "lnInv2"))+
  geom_line(aes(y = Factor_Ret$lnME2_Ret, col = "lnME2"))+
  geom_line(aes(y = Factor_Ret$lnIssue_ME_Ret, col = "lnIssue_ME"))+
  geom_line(aes(y = Factor_Ret$lnProf_ME_Ret, col = "lnProf_ME"))+
  geom_line(aes(y = Factor_Ret$lnInv_ME_Ret, col = "lnInv_ME"))+
  geom_line(aes(y = Factor_Ret$MK_Ret, col = "MK"))+
  theme_bw()
```




```r
Sample_Mean = apply(Factor_Ret[,2:13],MARGIN =  2, FUN = mean)
Cov = cov(Factor_Ret[,2:13])
SR = Sample_Mean/sqrt(diag(Cov))
Sample_Mean
```

```
##    lnIssue_Ret    lnProf_Ret     lnInv_Ret      lnME_Ret   lnIssue2_Ret
##      -8.907767     14.448370    -19.832221     -4.469544      -7.883374
##    lnProf2_Ret    lnInv2_Ret     lnME2_Ret lnIssue_ME_Ret lnProf_ME_Ret
##      -9.557348    -28.903515   -125.550004   -122.098391     188.716990
##   lnInv_ME_Ret        MK_Ret
##    -269.131282    176.069458
```

```r
Cov
```

```
##                lnIssue_Ret  lnProf_Ret  lnInv_Ret     lnME_Ret
## lnIssue_Ret       584.5430  -1369.1498   1010.505    208.62725
## lnProf_Ret      -1369.1498   4451.8878  -2501.044  -2292.40234
## lnInv_Ret        1010.5050  -2501.0437   2215.084    961.25905
## lnME_Ret          208.6272  -2292.4023    961.259   8438.99653
## lnIssue2_Ret      348.7129   -737.3853    632.710     92.90452
## lnProf2_Ret      1851.4490  -6454.4925   3301.827   3610.43007
## lnInv2_Ret       1798.8666  -4632.8950   3870.130   2135.45676
## lnME2_Ret        6272.8050 -67262.2478  29070.379 239571.03063
## lnIssue_ME_Ret   8159.7735 -19324.2152  14247.731   4001.29420
## lnProf_ME_Ret  -18640.2987  60242.9789 -33904.891 -29605.49119
## lnInv_ME_Ret    14647.2444 -36832.0378  31883.375  15756.48027
## MK_Ret           3180.9005  -4052.3242   5071.274  -8058.99331
##                lnIssue2_Ret lnProf2_Ret lnInv2_Ret    lnME2_Ret
```

```
## lnIssue_Ret       348.71294   1851.4490   1798.867    6272.805
## lnProf_Ret       -737.38527  -6454.4925  -4632.895  -67262.248
## lnInv_Ret         632.70999   3301.8273   3870.130   29070.379
## lnME_Ret           92.90452   3610.4301   2135.457  239571.031
## lnIssue2_Ret      271.96691    977.3913   1145.608    2991.992
## lnProf2_Ret       977.39127   9751.0286   6160.744  106327.184
## lnInv2_Ret       1145.60846   6160.7437   6947.559   63671.966
## lnME2_Ret        2991.99186 106327.1842  63671.966 6825870.842
## lnIssue_ME_Ret   4859.52613  26133.5053  25457.156  118455.413
## lnProf_ME_Ret  -10006.52065 -87270.4579 -62772.280 -870662.459
## lnInv_ME_Ret     9113.90652  48836.3567  56017.864  472514.913
## MK_Ret           2084.03573   5400.6552   9086.558 -212268.317
##                lnIssue_ME_Ret lnProf_ME_Ret lnInv_ME_Ret      MK_Ret
## lnIssue_Ret          8159.773     -18640.30    14647.244    3180.901
## lnProf_Ret         -19324.215      60242.98   -36832.038   -4052.324
## lnInv_Ret           14247.731     -33904.89    31883.375    5071.274
## lnME_Ret             4001.294     -29605.49    15756.480   -8058.993
## lnIssue2_Ret         4859.526     -10006.52     9113.907    2084.036
## lnProf2_Ret         26133.505     -87270.46    48836.357    5400.655
## lnInv2_Ret          25457.156     -62772.28    56017.864    9086.558
## lnME2_Ret          118455.413    -870662.46   472514.913 -212268.317
## lnIssue_ME_Ret     114166.371    -262950.59   206840.728   43510.896
## lnProf_ME_Ret     -262950.592     815691.20  -499162.398  -58455.207
## lnInv_ME_Ret       206840.728    -499162.40   459968.732   71918.981
## MK_Ret              43510.896     -58455.21    71918.981  124932.566
```

```
SR
```

```
##    lnIssue_Ret     lnProf_Ret      lnInv_Ret       lnME_Ret   lnIssue2_Ret
##    -0.36843479     0.21654430    -0.42138229    -0.04865391    -0.47802881
##    lnProf2_Ret     lnInv2_Ret      lnME2_Ret lnIssue_ME_Ret  lnProf_ME_Ret
##    -0.09678592    -0.34676444    -0.04805490    -0.36136064     0.20895276
##   lnInv_ME_Ret         MK_Ret
##    -0.39682591     0.49813401
```

(ii) Next, you are to use the Elastic Net procedure (alpha = 0.5 in glmnet) to estimate the b coefficients. Here, we cannot use the pre???programmed cross???validation procedure in cv.glmnet. The reason is that the right and left hand side variables depend on the sample. You are to run a cross???sectional regression of average returns to the factors on
the covariances of each factor with itself and the other factors (see slide 48 in Topic 4). A 5???fold cross???validation would tell you to first find the sample factor averages and sample factor covariance matrix in a 20???year subperiod, and then see how well the estimated b coefficients do in the 5???year out of sample period. In the out of sample
period, the average returns are the 5???year average factor returns for this period and the covariances are the 5???year covariances in this period. Thus, due to the combination of time series info (average returns and sample covariance matrix) and the cross??? sectional regression, our setting is a little more complicated than the standard
cv.glmnet code.
So, to be clear: first, find sample average factor returns and covariance matrix from 1980???1999. Estimate the Elastic Net using the glmnet procedure (use family = 'Gaussian', alpha = 0.5). This gives you a matrix of b coefficients as a function of lambda. For each of these sets of b coefficients (each vector of b's correspond to a particular lambda), calculate the mean squared error in the out of sample period 2000???2004. Now you have MSE as a function of lambda for one 5???year fold. Then repeat using as in???sample data the 1980???1984 and 1990???2004 period. The out of

sample data is then the 1985???1989 period. Get the MSEs as a function of lambda and save. Repeat until you have done all 5 folds. Then take the average MSE for each value
of lambda. Pick the lambda that gives the smallest average MSE. Finally, estimate the elastic net on the full 1980???2004 sample period. Pick the b???coefficient that
corresponds to the value of lambda you have chosen.

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.4

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.3

## Loaded glmnet 2.0-16
```

```
Factor_Ret1 = Factor_Ret
Factor_Ret = Factor_Ret[between(year,1980,2004)]
five_years = c(1980,1985,1990,1995,2000)
MSEs = NULL
count = 1
for (five_year in five_years) {
  test_sample = Factor_Ret[year >= five_year & year <= five_year+4]
  train_sample = Factor_Ret[!(year >= five_year & year <= five_year+4)]
  train_mean = apply(train_sample[,2:13],MARGIN =  2, FUN = mean)
  train_cov = cov(train_sample[,2:13])
  test_mean = apply(test_sample[,2:13],MARGIN =  2, FUN = mean)
  test_cov = cov(test_sample[,2:13])
  out = glmnet(x = train_cov,y = train_mean,family = "gaussian",alpha = 0.5,lambda = seq(300,0,-.1),inte
  lambda = out$lambda
  beta = as.matrix(out$beta)
  out = predict(out,newx = test_cov)
  out_sample = data.table(
    lambda = lambda,
    lnIssue_Ret = out[1,],
    lnProf_Ret = out[2,],
    lnInv_Ret = out[3,],
    lnME_Ret = out[4,],
    lnIssue2_Ret = out[5,],
    lnProf2_Ret = out[6,],
    lnInv2_Ret = out[7,],
    lnME2_Ret = out[8,],
    lnIssue_ME_Ret = out[9,],
    lnProf_ME_Ret = out[10,],
    lnInv_ME_Ret = out[11,],
    MK_Ret = out[12,]
  )
  test_pred = as.matrix(out_sample)
  MSE = (test_mean["lnIssue_Ret"]-test_pred[,"lnIssue_Ret"])^2+(test_mean["lnProf_Ret"]-test_pred[,"lnPr
           (test_mean["lnInv_Ret"]-test_pred[,"lnInv_Ret"])^2+(test_mean["lnME_Ret"]-test_pred[,"lnME_
           (test_mean["lnIssue2_Ret"]-test_pred[,"lnIssue2_Ret"])^2+(test_mean["lnProf2_Ret"]-test_pre
           (test_mean["lnInv2_Ret"]-test_pred[,"lnInv2_Ret"])^2+(test_mean["lnME2_Ret"]-test_pred[,"ln
           (test_mean["lnIssue_ME_Ret"]-test_pred[,"lnIssue_ME_Ret"])^2+(test_mean["lnProf_ME_Ret"]-te
           (test_mean["lnInv_ME_Ret"]-test_pred[,"lnInv_ME_Ret"])^2+(test_mean["MK_Ret"]-test_pred[,"M
```

```r
  MSE = MSE/12
  DT = data.table(
    lambda = lambda,
    MSE = MSE
  )
  colnames(DT) <- c("lambda",five_year)
  if (count == 1){
    MSEs = DT
  }
  else{
    MSEs = merge(MSEs,DT,by = "lambda")
  }
  count = count + 1
}
MSE_min = c(min(MSEs$`1980`,na.rm = TRUE),min(MSEs$`1985`,na.rm = TRUE),min(MSEs$`1990`,na.rm = TRUE),mi

lambda_min = c(MSEs[MSEs$`1980` == min(MSEs$`1980`,na.rm = TRUE),]$lambda,MSEs[MSEs$`1985` == min(MSEs$

lambda_fin = lambda_min[which.min(MSE_min)]

whole_mean = apply(Factor_Ret[,2:13],MARGIN =  2, FUN = mean)
whole_cov = cov(Factor_Ret[,2:13])
out1 = glmnet(x = whole_cov,y = whole_mean,family = "gaussian",alpha = 0.5,intercept = FALSE,lambda = se
beta = out1$beta[,2962]
beta
```

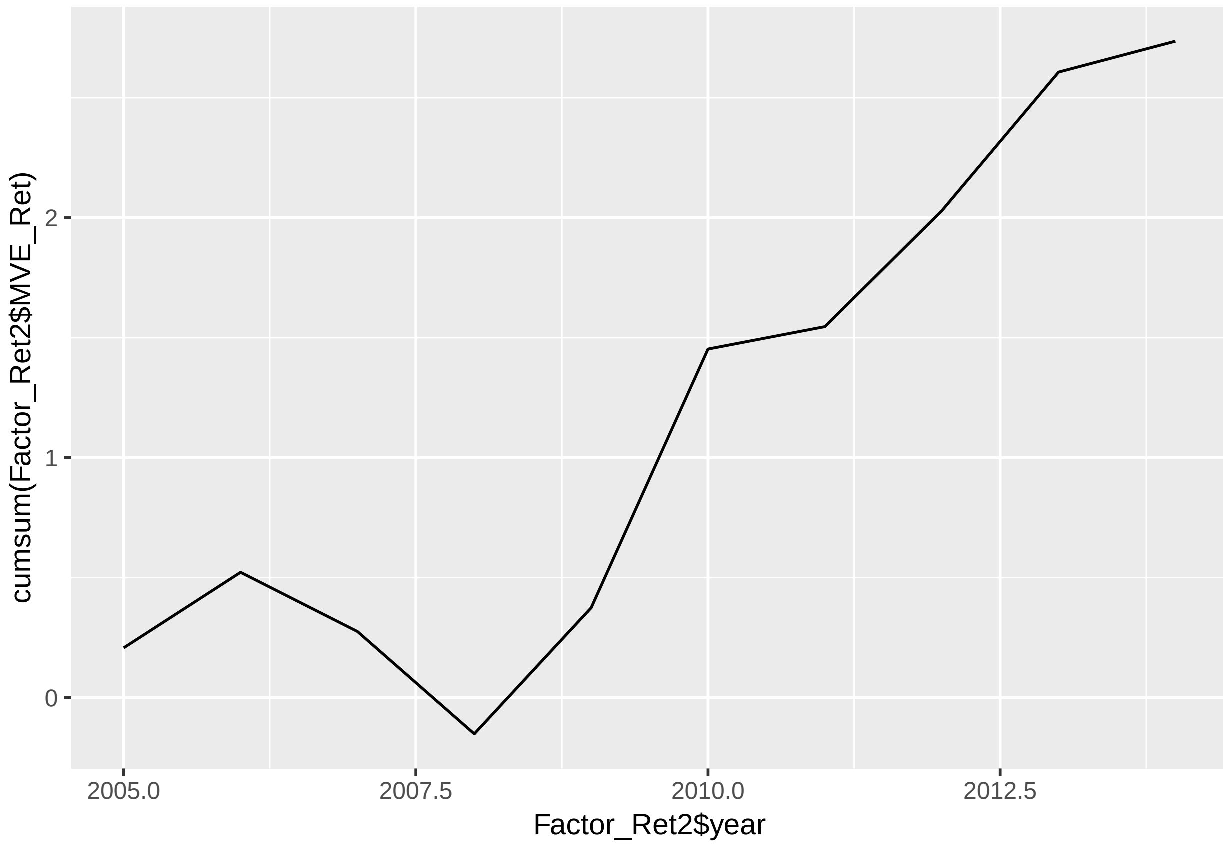
```
##    lnIssue_Ret     lnProf_Ret      lnInv_Ret       lnME_Ret   lnIssue2_Ret
##  -4.435128e-04   0.000000e+00  -2.741984e-03   6.350011e-04  -2.571694e-02
##    lnProf2_Ret     lnInv2_Ret      lnME2_Ret lnIssue_ME_Ret  lnProf_ME_Ret
##   6.153037e-04   0.000000e+00   2.484632e-05   0.000000e+00   0.000000e+00
##   lnInv_ME_Ret         MK_Ret
##  -7.579993e-05   1.244093e-03
```

```r
out1$lambda[2962]
```

```
## [1] 3.9
```

(iii) With the final b???vector in hand, calculate the out???of???sample average return, standard deviation, and Sharpe ratio for the corresponding estimated “ex ante” MVE portfolio with return b’F_t in the period 2005???2014.

```r
Factor_Ret2 = Factor_Ret1[between(year,2005,2014)]
Factor_Ret2[,MVE_Ret := beta["lnIssue_Ret"]*lnIssue_Ret+beta["lnProf_Ret"]*lnProf_Ret+beta["lnInv_Ret"]*
              beta["lnIssue2_Ret"]*lnIssue2_Ret+beta["lnProf2_Ret"]*lnProf2_Ret+beta["lnInv2_Ret"]*lnInv
              beta["lnIssue_ME_Ret"]*lnIssue_ME_Ret+beta["lnProf_ME_Ret"]*lnProf_ME_Ret+beta["lnInv_ME_R

qplot(x = Factor_Ret2$year,y = cumsum(Factor_Ret2$MVE_Ret),geom = "line")
```



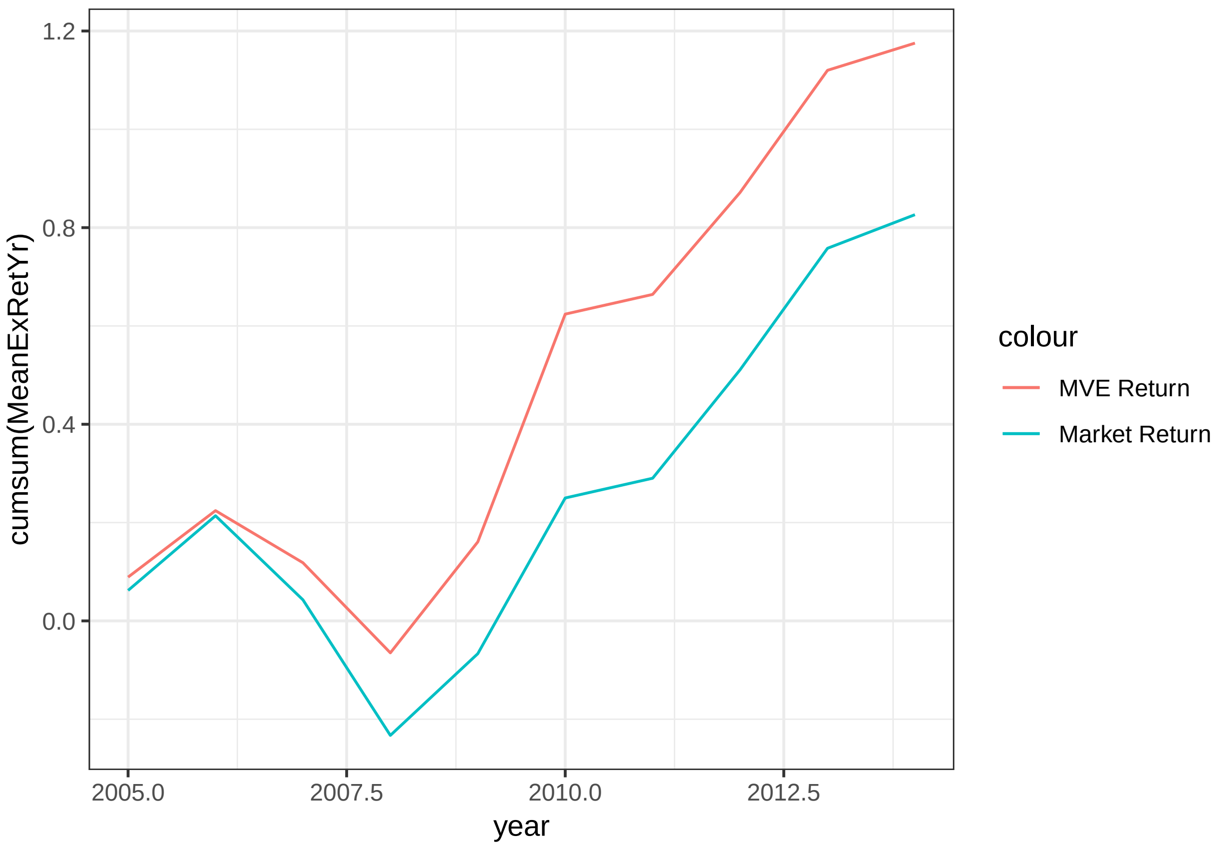

```
average_ret = mean(Factor_Ret2$MVE_Ret)
std_ret = sd(Factor_Ret2$MVE_Ret)
sr_ret = average_ret/std_ret
stat = data.table(
  Mean = average_ret,
  STD = std_ret,
  SR = sr_ret
)
stat
```

```
##         Mean       STD        SR
## 1: 0.2735894 0.4305522 0.6354384
```

(iv) Plot the cumulative return on this portfolio relative to that on the market (get market return using the value???weights in the sample, MEwt) over the 2005???2014 period, where you normalize the "MVE" portfolio's standard deviation to be the same as the market over this period. Compare. Note that one should really redo the estimation each year to get proper out of sample results that would mimic what you would do in the real world. Also, you could experiment in the in???sample cross???validation with different values for alpha to see what works best.

```
setwd("/Users/jiaminghuang/Downloads")
stockdata <- as.data.table(read.dta("StockRetAcct_insample.dta"))
stockdata[,ExRet:=exp(lnAnnRet) - exp(lnRf)]
setkey(stockdata, year)
vwretd <- stockdata[, list(MeanExRetYr = weighted.mean(ExRet, MEwt, na.rm = T)), by = year]
vwretd = vwretd[between(year,2005,2014)]
multi = sd(vwretd$MeanExRetYr)/std_ret
```

```
result4 = merge(vwretd,Factor_Ret2[,list(year,MVE_Ret)],by = "year")
result4$MVE_Ret = multi*result4$MVE_Ret
ggplot(data = result4,aes(year))+
  geom_line(aes(y = cumsum(MeanExRetYr),col="Market Return"))+
  geom_line(aes(y = cumsum(MVE_Ret),col = "MVE Return"))+
  theme_bw()
```

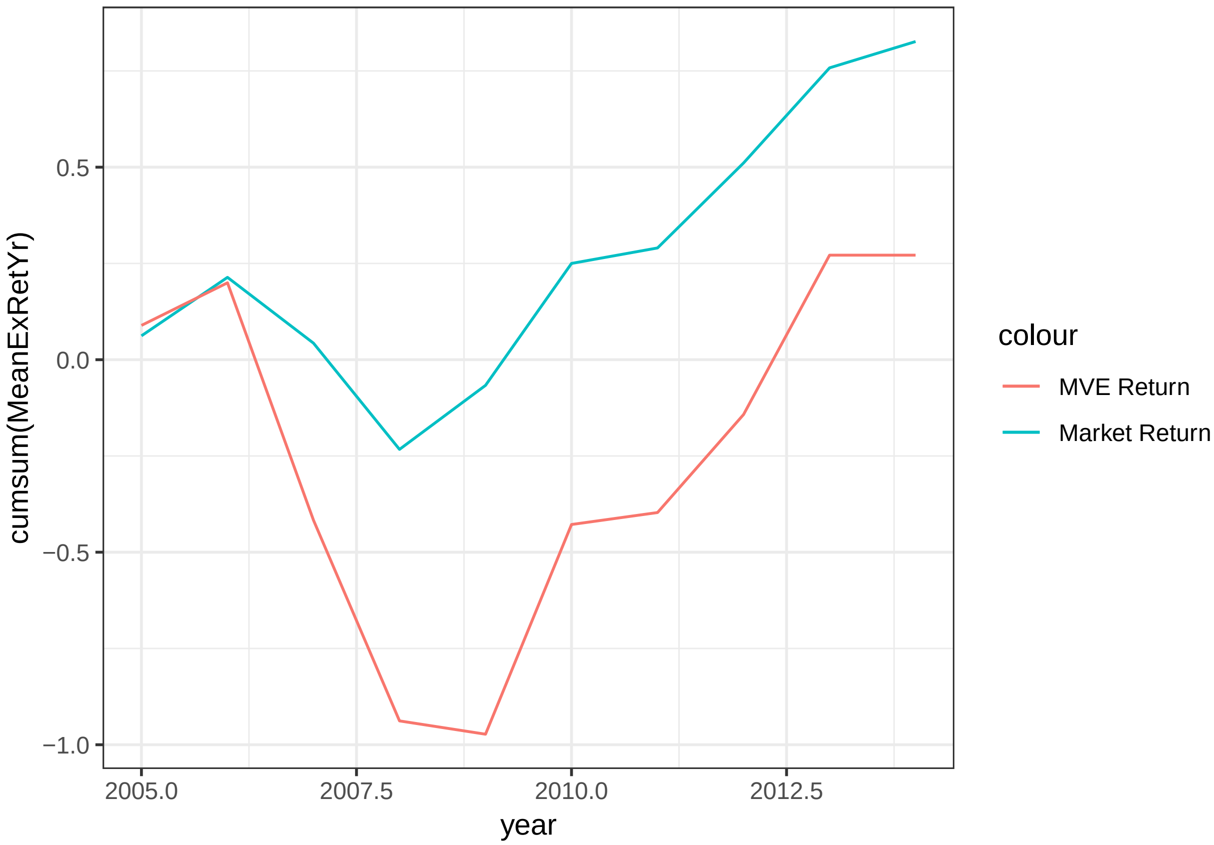


The graph above is the result we redo the estimation every twenty years and hold this bate estimation. We can see the MVE portfolio is better than market.

```
test_year = vwretd$year
for (y in test_year) {
  Factor_test = Factor_Ret1[year == y]
  Factor_train = Factor_Ret1[between(year,y-25,y-1)]
  five_years = c(y-25,y-20,y-15,y-10,y-5)
  MSEs = NULL
  count = 1
  for (five_year in five_years) {
    test_sample = Factor_train[year >= five_year & year <= five_year+4]
    train_sample = Factor_train[!(year >= five_year & year <= five_year+4)]
    train_mean = apply(train_sample[,2:13],MARGIN =  2, FUN = mean)
    train_cov = cov(train_sample[,2:13])
    test_mean = apply(test_sample[,2:13],MARGIN =  2, FUN = mean)
    test_cov = cov(test_sample[,2:13])
    out = glmnet(x = train_cov,y = train_mean,family = "gaussian",alpha = 0.5,lambda
                 = seq(300,0,-.1),intercept = FALSE)
    lambda = out$lambda
```

```
  beta = as.matrix(out$beta)
  out = predict(out,newx = test_cov)
  out_sample = data.table(
    lambda = lambda,
    lnIssue_Ret = out[1,],
    lnProf_Ret = out[2,],
    lnInv_Ret = out[3,],
    lnME_Ret = out[4,],
    lnIssue2_Ret = out[5,],
    lnProf2_Ret = out[6,],
    lnInv2_Ret = out[7,],
    lnME2_Ret = out[8,],
    lnIssue_ME_Ret = out[9,],
    lnProf_ME_Ret = out[10,],
    lnInv_ME_Ret = out[11,],
    MK_Ret = out[12,]
  )
  test_pred = as.matrix(out_sample)
  MSE = (test_mean["lnIssue_Ret"]-test_pred[,"lnIssue_Ret"])^2+(test_mean["lnProf_Ret"]-test_pred[,"ln
         (test_mean["lnInv_Ret"]-test_pred[,"lnInv_Ret"])^2+(test_mean["lnME_Ret"]-test_pred[,"lnME_
         (test_mean["lnIssue2_Ret"]-test_pred[,"lnIssue2_Ret"])^2+(test_mean["lnProf2_Ret"]-test_pre
         (test_mean["lnInv2_Ret"]-test_pred[,"lnInv2_Ret"])^2+(test_mean["lnME2_Ret"]-test_pred[,"ln
         (test_mean["lnIssue_ME_Ret"]-test_pred[,"lnIssue_ME_Ret"])^2+(test_mean["lnProf_ME_Ret"]-te
         (test_mean["lnInv_ME_Ret"]-test_pred[,"lnInv_ME_Ret"])^2+(test_mean["MK_Ret"]-test_pred[,"M

  MSE = MSE/12
  DT = data.table(
    lambda = lambda,
    MSE = MSE
  )
  colnames(DT) <- c("lambda",five_year)
  if (count == 1){
    MSEs = DT
  }
  else{
    MSEs = merge(MSEs,DT,by = "lambda")
  }
  count = count + 1
}
colnames(MSEs) = c("lambda","1980","1985","1990","1995","2000")
MSE_min = c(min(MSEs$`1980`,na.rm = TRUE),min(MSEs$`1985`,na.rm =TRUE),min(MSEs$`1990`,na.rm = TRUE),m

lambda_min = c(MSEs[which.min(MSEs$`1980`),]$lambda,MSEs[which.min(MSEs$`1985`),]$lambda,MSEs[which.mi

lambda_fin = lambda_min[which.min(MSE_min)]

whole_mean = apply(Factor_train[,2:13],MARGIN =  2, FUN = mean)
whole_cov = cov(Factor_train[,2:13])
out1 = glmnet(x = whole_cov,y = whole_mean,family = "gaussian",alpha = 0.5,intercept = FALSE,lambda =
beta = out1$beta[,(300-lambda_fin)*10+1]
Factor_Ret2[year == y,MVE_Ret:=beta["lnIssue_Ret"]*lnIssue_Ret+beta["lnProf_Ret"]*lnProf_Ret+beta["lnI
            beta["lnIssue2_Ret"]*lnIssue2_Ret+beta["lnProf2_Ret"]*lnProf2_Ret+beta["lnInv2_Ret"]*lnInv
            beta["lnIssue_ME_Ret"]*lnIssue_ME_Ret+beta["lnProf_ME_Ret"]*lnProf_ME_Ret+beta["lnInv_ME_R
```

```
}
result4 = merge(vwretd,Factor_Ret2[,list(year,MVE_Ret)],by = "year")
result4$MVE_Ret = multi*result4$MVE_Ret
ggplot(data = result4,aes(year))+
  geom_line(aes(y = cumsum(MeanExRetYr),col="Market Return"))+
  geom_line(aes(y = cumsum(MVE_Ret),col = "MVE Return"))+
  theme_bw()
```



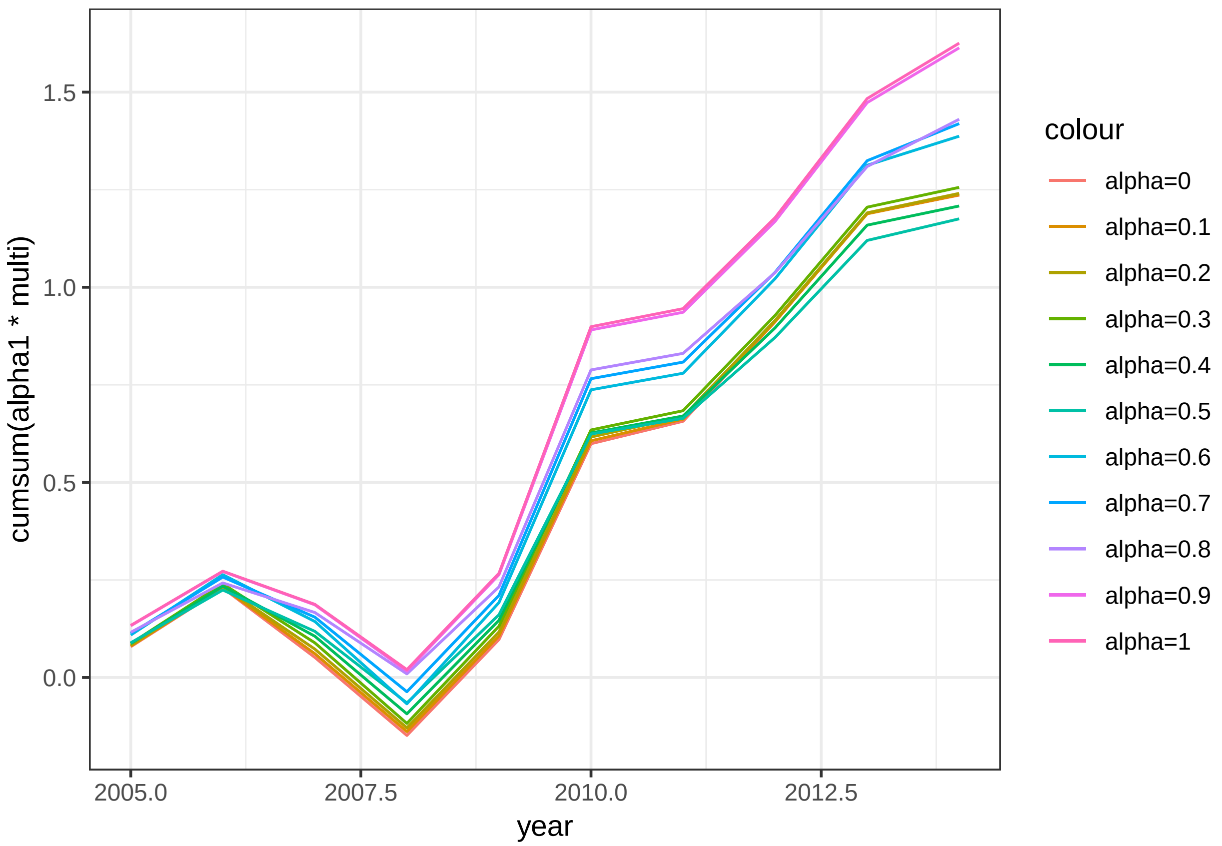
The graph above is the result of redoing the estimation every year. We can see that if we redo the estimation each year, the MVE portfolio is not as good as market.

```
alphas = seq(0,1,0.1)
resultn = NULL
for (alpha in alphas) {
  five_years = c(1980,1985,1990,1995,2000)
MSEs = NULL
count = 1
for (five_year in five_years) {
  test_sample = Factor_Ret[year >= five_year & year <= five_year+4]
  train_sample = Factor_Ret[!(year >= five_year & year <= five_year+4)]
  train_mean = apply(train_sample[,2:13],MARGIN =  2, FUN = mean)
  train_cov = cov(train_sample[,2:13])
  test_mean = apply(test_sample[,2:13],MARGIN =  2, FUN = mean)
  test_cov = cov(test_sample[,2:13])
  out = glmnet(x = train_cov,y = train_mean,family = "gaussian",alpha = alpha,lambda = seq(300,0,-.1),i
  lambda = out$lambda
  beta = as.matrix(out$beta)
```

```
  out = predict(out,newx = test_cov)
  out_sample = data.table(
    lambda = lambda,
    lnIssue_Ret = out[1,],
    lnProf_Ret = out[2,],
    lnInv_Ret = out[3,],
    lnME_Ret = out[4,],
    lnIssue2_Ret = out[5,],
    lnProf2_Ret = out[6,],
    lnInv2_Ret = out[7,],
    lnME2_Ret = out[8,],
    lnIssue_ME_Ret = out[9,],
    lnProf_ME_Ret = out[10,],
    lnInv_ME_Ret = out[11,],
    MK_Ret = out[12,]
  )
  test_pred = as.matrix(out_sample)
  MSE = (test_mean["lnIssue_Ret"]-test_pred[,"lnIssue_Ret"])^2+(test_mean["lnProf_Ret"]-test_pred[,"lnPr
           (test_mean["lnInv_Ret"]-test_pred[,"lnInv_Ret"])^2+(test_mean["lnME_Ret"]-test_pred[,"lnME_
           (test_mean["lnIssue2_Ret"]-test_pred[,"lnIssue2_Ret"])^2+(test_mean["lnProf2_Ret"]-test_pre
           (test_mean["lnInv2_Ret"]-test_pred[,"lnInv2_Ret"])^2+(test_mean["lnME2_Ret"]-test_pred[,"ln
           (test_mean["lnIssue_ME_Ret"]-test_pred[,"lnIssue_ME_Ret"])^2+(test_mean["lnProf_ME_Ret"]-te
           (test_mean["lnInv_ME_Ret"]-test_pred[,"lnInv_ME_Ret"])^2+(test_mean["MK_Ret"]-test_pred[,"M

  MSE = MSE/12
  DT = data.table(
    lambda = lambda,
    MSE = MSE
  )
  colnames(DT) <- c("lambda",five_year)
  if (count == 1){
    MSEs = DT
  }
  else{
    MSEs = merge(MSEs,DT,by = "lambda")
  }
  count = count + 1
}
MSE_min = c(min(MSEs$`1980`,na.rm = TRUE),min(MSEs$`1985`,na.rm = TRUE),min(MSEs$`1990`,na.rm = TRUE),mi

  lambda_min = c(MSEs[which.min(MSEs$`1980`),]$lambda,MSEs[which.min(MSEs$`1985`),]$lambda,MSEs[which.mi

lambda_fin = lambda_min[which.min(MSE_min)]

whole_mean = apply(Factor_Ret[,2:13],MARGIN =  2, FUN = mean)
whole_cov = cov(Factor_Ret[,2:13])
out1 = glmnet(x = whole_cov,y = whole_mean,family = "gaussian",alpha = alpha,intercept = FALSE,lambda =
beta = out1$beta[,(300-lambda_fin)*10+1]
Factor_Ret2[,MVE_Ret:=beta["lnIssue_Ret"]*lnIssue_Ret+beta["lnProf_Ret"]*lnProf_Ret+beta["lnInv_Ret"]*ln
              beta["lnIssue2_Ret"]*lnIssue2_Ret+beta["lnProf2_Ret"]*lnProf2_Ret+beta["lnInv2_Ret"]*lnInv
              beta["lnIssue_ME_Ret"]*lnIssue_ME_Ret+beta["lnProf_ME_Ret"]*lnProf_ME_Ret+beta["lnInv_ME_R

temp = Factor_Ret2[,list(year,MVE_Ret)]
```

```r
colnames(temp) = c("year",alpha)

if(is.null(resultn)){
  resultn = temp
}
else{
  resultn = merge(resultn,temp, by = "year")
}
}
```

```r
colnames(resultn) = c("year","alpha1","alpha2","alpha3","alpha4","alpha5","alpha6"
                      ,"alpha7","alpha8","alpha9","alpha10","alpha11")
ggplot(data = resultn,aes(year))+
  geom_line(aes(y = cumsum(alpha1*multi),col = "alpha=0"))+
  geom_line(aes(y = cumsum(alpha2*multi),col = "alpha=0.1"))+
  geom_line(aes(y = cumsum(alpha3*multi),col = "alpha=0.2"))+
  geom_line(aes(y = cumsum(alpha4*multi),col = "alpha=0.3"))+
  geom_line(aes(y = cumsum(alpha5*multi),col = "alpha=0.4"))+
  geom_line(aes(y = cumsum(alpha6*multi),col = "alpha=0.5"))+
  geom_line(aes(y = cumsum(alpha7*multi),col = "alpha=0.6"))+
  geom_line(aes(y = cumsum(alpha8*multi),col = "alpha=0.7"))+
  geom_line(aes(y = cumsum(alpha9*multi),col = "alpha=0.8"))+
  geom_line(aes(y = cumsum(alpha10*multi),col = "alpha=0.9"))+
  geom_line(aes(y = cumsum(alpha11*multi),col = "alpha=1"))+
  theme_bw()+xlim(2005,2014)
```



From the graph above, we know that the out of sample estimation will be better with larger alpha.