Data Analytics HW4

Jiaqi Li

April 26, 2019

Problem 1

```
options(warn = -1)
library(readxl)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(haven)
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
sample = read_dta("StockRetAcct_insample.dta") %>% as.data.table() %>% na.omit()
sample[,`:=` (ex_ret = exp(lnAnnRet)-exp(lnRf),lnIssue2 = lnIssue^2,
              lnProf2 = lnProf^2,lnInv2 = lnInv^2,lnME2 = lnME^2,
              lnIssue_ME = lnIssue*lnME,lnProf_ME = lnProf*lnME,
              lnInv_ME = lnInv*lnME)]

#########################################################################
#                               (i)                                     #
#########################################################################
final = sample[,`:=` (demean_lnIssue = lnIssue - mean(lnIssue,na.rm = T),
                      demean_lnIssue2 = lnIssue2 - mean(lnIssue2,na.rm = T),
                      demean_lnIssue_ME = lnIssue_ME - mean(lnIssue_ME,na.rm = T),
                      demean_lnProf = lnProf - mean(lnProf,na.rm = T),
                      demean_lnProf2 = lnProf2 - mean(lnProf2,na.rm = T),
```

```
                      demean_lnProf_ME = lnProf_ME - mean(lnProf_ME,na.rm = T),
                      demean_lnInv = lnInv - mean(lnInv,na.rm = T),
                      demean_lnInv2 = lnInv2 - mean(lnInv2,na.rm = T),
                      demean_lnInv_ME = lnInv_ME - mean(lnInv_ME,na.rm = T),
                      demean_lnME = lnME - mean(lnME,na.rm = T),
                      demean_lnME2 = lnME2 - mean(lnME2,na.rm = T),
                      intercept = 1),by = year]

final[,`:=` (ret_demean_LnIssue = sum(demean_lnIssue*ex_ret,na.rm=T),
             ret_demean_lnIssue2 = sum(demean_lnIssue2*ex_ret,na.rm=T),
             ret_demean_lnIssue_ME = sum(demean_lnIssue_ME*ex_ret,na.rm=T),
             ret_demean_lnProf = sum(demean_lnProf*ex_ret,na.rm=T),
             ret_demean_lnProf2 = sum(demean_lnProf2*ex_ret,na.rm=T),
             ret_demean_lnProf_ME = sum(demean_lnProf_ME*ex_ret,na.rm=T),
             ret_demean_lnInv = sum(demean_lnInv*ex_ret,na.rm=T),
             ret_demean_lnInv2 = sum(demean_lnInv2*ex_ret,na.rm=T),
             ret_demean_lnInv_ME = sum(demean_lnInv_ME*ex_ret,na.rm=T),
             ret_demean_lnME = sum(demean_lnME*ex_ret,na.rm=T),
             ret_demean_lnME2 = sum(demean_lnME2*ex_ret,na.rm=T),
             ret_all = sum(intercept*ex_ret,na.rm=T)),by = year]

M = final[,list(ret_demean_LnIssue,ret_demean_lnIssue2,ret_demean_lnIssue_ME,
                ret_demean_lnProf,ret_demean_lnProf2,ret_demean_lnProf_ME,
                ret_demean_lnInv,ret_demean_lnInv2,ret_demean_lnInv_ME,
                ret_demean_lnME,ret_demean_lnME2,ret_all)]
sample_mean = apply(M,2,function(x){mean(x,na.rm = T)})
vcov = var(M)
SR = sample_mean/diag(sqrt(vcov))
```

The followings are sample mean, variance covariance matrix, and the sharpe ratio:

```
sample_mean
```

```
##    ret_demean_LnIssue   ret_demean_lnIssue2 ret_demean_lnIssue_ME
##             -8.884411             -7.990101           -123.140321
##     ret_demean_lnProf    ret_demean_lnProf2  ret_demean_lnProf_ME
##             10.569521             -9.063130            138.892903
##      ret_demean_lnInv     ret_demean_lnInv2   ret_demean_lnInv_ME
##            -13.556299            -18.265288           -189.170446
##       ret_demean_lnME      ret_demean_lnME2              ret_all
##             -8.549582           -256.795107            155.566331
```

```
vcov
```

```
##                       ret_demean_LnIssue ret_demean_lnIssue2
## ret_demean_LnIssue              579.4779            321.7846
## ret_demean_lnIssue2             321.7846            244.8080
## ret_demean_lnIssue_ME          8137.6441           4504.0039
## ret_demean_lnProf              -922.9955           -440.5659
## ret_demean_lnProf2             1270.4419            602.0173
## ret_demean_lnProf_ME         -12385.5433          -5888.0566
## ret_demean_lnInv               559.6739            328.9810
## ret_demean_lnInv2              853.8482            514.4303
## ret_demean_lnInv_ME           8392.1733           4893.3573
```

```
## ret_demean_lnME                533.4284           289.7847
## ret_demean_lnME2             16055.7750          8812.6437
## ret_all                       1236.2184           885.5326
##                       ret_demean_lnIssue_ME ret_demean_lnProf
## ret_demean_LnIssue                 8137.644         -922.9955
## ret_demean_lnIssue2                4504.004         -440.5659
## ret_demean_lnIssue_ME            114504.129       -13055.5516
## ret_demean_lnProf                -13055.552         1908.8584
## ret_demean_lnProf2                18006.439        -2729.7635
## ret_demean_lnProf_ME            -175038.819        25446.8636
## ret_demean_lnInv                   7897.680         -823.8128
## ret_demean_lnInv2                 12083.767        -1318.6946
## ret_demean_lnInv_ME              118742.439       -12717.7706
## ret_demean_lnME                    8519.104        -1720.6135
## ret_demean_lnME2                 255348.101       -51641.1836
## ret_all                           17210.042         -185.1245
##                       ret_demean_lnProf2 ret_demean_lnProf_ME
## ret_demean_LnIssue            1270.4419           -12385.543
## ret_demean_lnIssue2            602.0173            -5888.057
## ret_demean_lnIssue_ME        18006.4391          -175038.819
## ret_demean_lnProf            -2729.7635            25446.864
## ret_demean_lnProf2            4016.0759           -36313.156
## ret_demean_lnProf_ME        -36313.1558           339590.130
## ret_demean_lnInv              1076.2506           -11034.991
## ret_demean_lnInv2             1740.2316           -17631.410
## ret_demean_lnInv_ME          16801.5444          -170041.581
## ret_demean_lnME               2737.0788           -21593.017
## ret_demean_lnME2             82015.7718          -649790.274
## ret_all                         91.6165            -3888.264
##                       ret_demean_lnInv ret_demean_lnInv2
## ret_demean_LnIssue            559.6739          853.8482
## ret_demean_lnIssue2           328.9810          514.4303
## ret_demean_lnIssue_ME        7897.6797        12083.7669
## ret_demean_lnProf            -823.8128        -1318.6946
## ret_demean_lnProf2           1076.2506         1740.2316
## ret_demean_lnProf_ME       -11034.9907       -17631.4097
## ret_demean_lnInv             786.9997         1152.2820
## ret_demean_lnInv2           1152.2820         1745.3910
## ret_demean_lnInv_ME        11520.8751        17001.3662
## ret_demean_lnME              662.6408         1213.8310
## ret_demean_lnME2           20195.5446        36700.5943
## ret_all                     1325.2838         2208.0682
##                       ret_demean_lnInv_ME ret_demean_lnME ret_demean_lnME2
## ret_demean_LnIssue               8392.173        533.4284        16055.775
## ret_demean_lnIssue2              4893.357        289.7847         8812.644
## ret_demean_lnIssue_ME          118742.439       8519.1039       255348.101
## ret_demean_lnProf              -12717.771      -1720.6135       -51641.184
## ret_demean_lnProf2              16801.544       2737.0788        82015.772
## ret_demean_lnProf_ME          -170041.581     -21593.0172      -649790.274
## ret_demean_lnInv               11520.875        662.6408        20195.545
## ret_demean_lnInv2              17001.366       1213.8310        36700.594
## ret_demean_lnInv_ME            169585.848      11813.0286       357049.082
## ret_demean_lnME                 11813.029       7994.3474       232107.731
## ret_demean_lnME2               357049.082     232107.7315      6757796.843
```

```
## ret_all                          18898.577      -3472.4200       -94845.324
##                            ret_all
## ret_demean_LnIssue       1236.2184
## ret_demean_lnIssue2       885.5326
## ret_demean_lnIssue_ME   17210.0418
## ret_demean_lnProf        -185.1245
## ret_demean_lnProf2         91.6165
## ret_demean_lnProf_ME    -3888.2644
## ret_demean_lnInv         1325.2838
## ret_demean_lnInv2        2208.0682
## ret_demean_lnInv_ME     18898.5766
## ret_demean_lnME         -3472.4200
## ret_demean_lnME2       -94845.3242
## ret_all                 65318.3880
```

```
SR
```

```
##    ret_demean_LnIssue   ret_demean_lnIssue2 ret_demean_lnIssue_ME
##           -0.36907122           -0.51066899           -0.36390641
##     ret_demean_lnProf    ret_demean_lnProf2  ret_demean_lnProf_ME
##            0.24191815           -0.14301358            0.23834307
##      ret_demean_lnInv     ret_demean_lnInv2   ret_demean_lnInv_ME
##           -0.48322999           -0.43720001           -0.45936562
##       ret_demean_lnME      ret_demean_lnME2               ret_all
##           -0.09562102           -0.09878345            0.60869240
```

```
#########################################################################
#                                 (ii)                                  #
#########################################################################
K_fold = list(c(1980:1999),c(1980:1984,1990:2004),c(1980:1989,1995:2004),
              c(1980:1994,2000:2004),c(1985:2004))
out_sample = list(c(2000:2004),c(1985:1999),c(1990:1994),c(1995:1999),c(1980:1984))
M = final[,list(ret_demean_LnIssue,ret_demean_lnIssue2,ret_demean_lnIssue_ME,
                ret_demean_lnProf,ret_demean_lnProf2,ret_demean_lnProf_ME,
                ret_demean_lnInv,ret_demean_lnInv2,ret_demean_lnInv_ME,
                ret_demean_lnME,ret_demean_lnME2,ret_all,year)]
alpha = 0.5
setkey(M,year)
test = M[.(1980:2004)]
MSE = c()
lambda = c()
for(i in 1:5){
  temp = as.data.frame(test[.(unlist(K_fold[i]))])[,-13]
  temp_mean = apply(temp,2,function(x){mean(x,na.rm = T)})
  temp_M = cov(temp)
  lambda_test = glmnet(temp_M,temp_mean,family = "gaussian",alpha = alpha,standardize = T)$lambda
  lambda = c(lambda,lambda_test)
}
lambda = seq(1.1,10,0.1)
for(i in 1:5){
  temp = as.data.frame(test[.(unlist(K_fold[i]))])[,-13]
  temp_mean = apply(temp,2,function(x){mean(x,na.rm = T)})
  temp_M = var(temp)
```

```
  Lamb = glmnet(temp_M,temp_mean,family = "gaussian",alpha = alpha,standardize = T,lambda = lambda)
  out_temp = as.data.frame(test[.(unlist(out_sample[i]))])[,-13]
  out_vcov = var(out_temp,na.rm = T)
  try = predict(Lamb,out_vcov,s=lambda,type = "response")
  #out_return = (out_vcov)%*%try[-1,]
  MSE = rbind(MSE,colMeans((temp_mean-try)^2))
}
compare = colMeans(MSE)
Lamb_pos = which(compare == min(compare))
lambda = Lamb$lambda[Lamb_pos]
test_in = as.data.frame(test[,-13])
b = glmnet(var(test_in),colMeans(test_in),family = "gaussian",alpha = alpha,standardize = T,lambda = la
```

The following is the set of coeffcients (b's) for each factor.

```
b
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## ret_demean_LnIssue     .
## ret_demean_lnIssue2   -1.155924e-02
## ret_demean_lnIssue_ME  .
## ret_demean_lnProf     -1.197819e-03
## ret_demean_lnProf2     1.907649e-03
## ret_demean_lnProf_ME  -7.513213e-05
## ret_demean_lnInv      -1.559012e-02
## ret_demean_lnInv2     -3.019773e-03
## ret_demean_lnInv_ME   -4.368480e-04
## ret_demean_lnME        1.218199e-03
## ret_demean_lnME2       4.256260e-05
## ret_all                3.695704e-03
```

```
##########################################################################
#                                (iii)                                   #
##########################################################################
test = M[.(2005:2014)][,year := NULL]
test_ret = as.vector(b)%*%t(as.matrix(test))
MeanRet = mean(test_ret)
StdRet = sd(test_ret)
SRRet = MeanRet/StdRet
```

The followings are the mean return, the standard deviation, and the Sharpe Ratio of the Mean Variance efficient porfolio:

```
MeanRet
```

```
## [1] 0.6327074
```

```
StdRet
```

```
## [1] 1.245661
```

```
SRRet
```

```
## [1] 0.5079289
```

```
#########################################################################
#                                 (iiii)                                #
#########################################################################
setkey(sample,year)
sample[,MktRet := MEwt*ex_ret]
iiii = c(sample[.(2005:2014)]$MktRet,test_ret/length(test_ret))
iiii = cbind(iiii,c(rep("Market",length(sample[.(2005:2014)]$MktRet)),
                    rep("Predict",length(test_ret))))
iiii = as.data.table(iiii)
iiii[,Ret := cumsum(iiii),by = V2]
time = c(1:(length(sample[.(2005:2014)]$year))*2)
iiii = cbind(iiii,time)
colnames(iiii) = c("Return","Porfolio Type","Cumulative Return","Time")
qplot(Time,`Cumulative Return`,geom ="line", data = iiii,
      color = `Porfolio Type`,main = "Machine Learning based Trading") +
  theme_bw()
```