# 46-926 Homework 4, Part I

*Jingyi Guo, Pittsburgh Campus*

*2/11/2017*

```r
library(quantmod)
load("/Users/apple/retmat.Robj")
stockdata = getSymbols("MSFT", auto.assign=FALSE, from="2016-1-1", to="2016-12-31")
msft = dailyReturn(stockdata[,6])
msft = as.numeric(msft)
library(glmnet)
```

## 1 Lasso Procedure

```r
retmat = as.matrix(retmat)
glmnetout = glmnet(retmat, msft)
print(glmnetout)
```

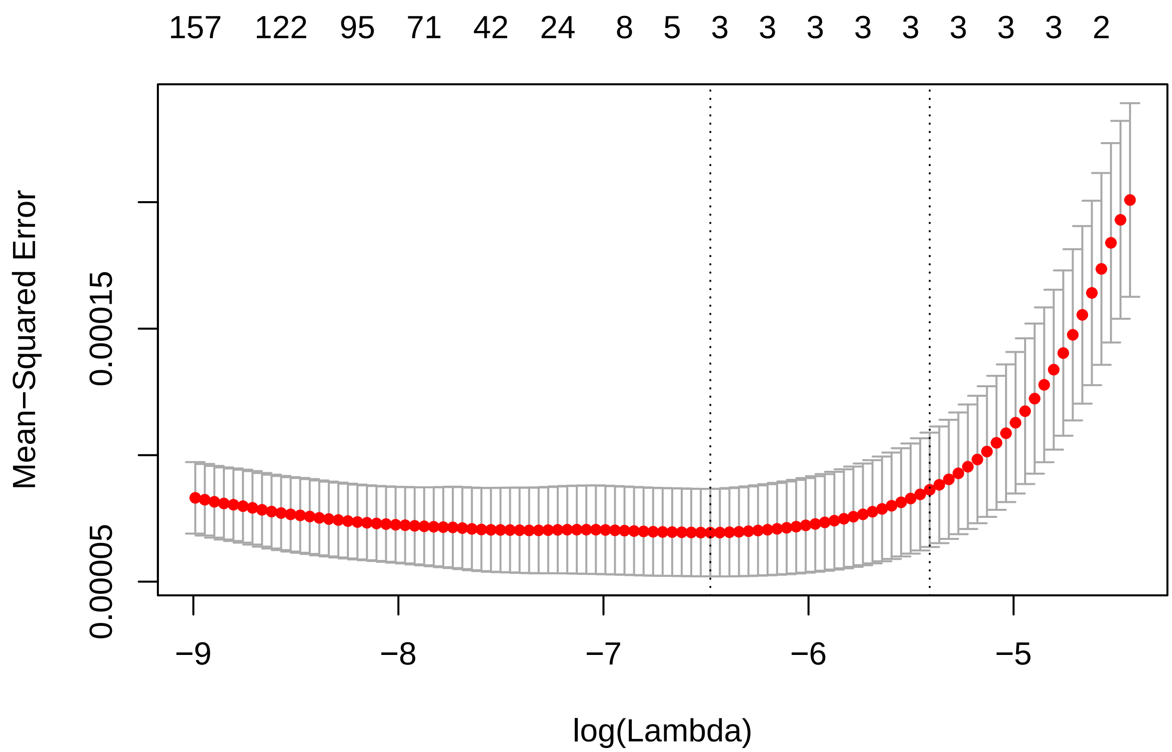
```
##
## Call:  glmnet(x = retmat, y = msft)
##
##         Df    %Dev    Lambda
##   [1,]   0 0.00000 0.0118900
##   [2,]   2 0.06192 0.0113500
##   [3,]   2 0.11840 0.0108300
##   [4,]   2 0.16990 0.0103400
##   [5,]   2 0.21680 0.0098720
##   [6,]   2 0.25950 0.0094230
##   [7,]   2 0.29850 0.0089950
##   [8,]   2 0.33390 0.0085860
##   [9,]   3 0.36640 0.0081960
##  [10,]   3 0.39590 0.0078230
##  [11,]   3 0.42290 0.0074680
##  [12,]   3 0.44740 0.0071280
##  [13,]   3 0.46980 0.0068040
##  [14,]   3 0.49020 0.0064950
##  [15,]   3 0.50880 0.0062000
##  [16,]   3 0.52570 0.0059180
##  [17,]   3 0.54110 0.0056490
##  [18,]   3 0.55520 0.0053920
##  [19,]   3 0.56800 0.0051470
##  [20,]   3 0.57960 0.0049130
##  [21,]   3 0.59030 0.0046900
##  [22,]   3 0.59990 0.0044770
##  [23,]   3 0.60880 0.0042730
##  [24,]   3 0.61680 0.0040790
##  [25,]   3 0.62410 0.0038940
##  [26,]   3 0.63080 0.0037170
##  [27,]   3 0.63690 0.0035480
##  [28,]   3 0.64240 0.0033860
##  [29,]   3 0.64750 0.0032330
```

```
## [30,]   3 0.65210 0.0030860
## [31,]   3 0.65630 0.0029450
## [32,]   3 0.66010 0.0028120
## [33,]   3 0.66350 0.0026840
## [34,]   3 0.66670 0.0025620
## [35,]   3 0.66960 0.0024450
## [36,]   3 0.67220 0.0023340
## [37,]   3 0.67460 0.0022280
## [38,]   3 0.67680 0.0021270
## [39,]   3 0.67880 0.0020300
## [40,]   3 0.68060 0.0019380
## [41,]   3 0.68230 0.0018500
## [42,]   3 0.68380 0.0017660
## [43,]   3 0.68510 0.0016850
## [44,]   3 0.68640 0.0016090
## [45,]   4 0.68760 0.0015360
## [46,]   4 0.68890 0.0014660
## [47,]   5 0.69040 0.0013990
## [48,]   5 0.69200 0.0013360
## [49,]   5 0.69330 0.0012750
## [50,]   6 0.69470 0.0012170
## [51,]   7 0.69770 0.0011620
## [52,]   7 0.70050 0.0011090
## [53,]   8 0.70320 0.0010590
## [54,]   8 0.70580 0.0010100
## [55,]  10 0.70830 0.0009645
## [56,]  15 0.71200 0.0009206
## [57,]  15 0.71680 0.0008788
## [58,]  16 0.72120 0.0008389
## [59,]  18 0.72590 0.0008007
## [60,]  21 0.73120 0.0007643
## [61,]  24 0.73730 0.0007296
## [62,]  26 0.74430 0.0006964
## [63,]  28 0.75100 0.0006648
## [64,]  32 0.75770 0.0006346
## [65,]  33 0.76440 0.0006057
## [66,]  34 0.77060 0.0005782
## [67,]  39 0.77670 0.0005519
## [68,]  42 0.78280 0.0005268
## [69,]  45 0.78910 0.0005029
## [70,]  47 0.79540 0.0004800
## [71,]  49 0.80130 0.0004582
## [72,]  54 0.80700 0.0004374
## [73,]  58 0.81300 0.0004175
## [74,]  63 0.81950 0.0003985
## [75,]  71 0.82600 0.0003804
## [76,]  74 0.83250 0.0003631
## [77,]  80 0.83890 0.0003466
## [78,]  85 0.84530 0.0003309
## [79,]  90 0.85210 0.0003158
## [80,]  90 0.85870 0.0003015
## [81,]  91 0.86460 0.0002878
## [82,]  95 0.87010 0.0002747
## [83,]  99 0.87520 0.0002622
```

```
##  [84,] 101 0.88010 0.0002503
##  [85,] 106 0.88510 0.0002389
##  [86,] 109 0.88950 0.0002281
##  [87,] 112 0.89390 0.0002177
##  [88,] 115 0.89830 0.0002078
##  [89,] 119 0.90240 0.0001983
##  [90,] 122 0.90630 0.0001893
##  [91,] 124 0.91000 0.0001807
##  [92,] 132 0.91380 0.0001725
##  [93,] 135 0.91780 0.0001647
##  [94,] 141 0.92180 0.0001572
##  [95,] 144 0.92580 0.0001500
##  [96,] 148 0.92960 0.0001432
##  [97,] 151 0.93330 0.0001367
##  [98,] 153 0.93680 0.0001305
##  [99,] 157 0.94010 0.0001246
## [100,] 160 0.94340 0.0001189
```

## 2 Five-Fold Cross Validation

```r
cvglmout = cv.glmnet(retmat, msft)
plot(cvglmout)
```




## 3 Lambda Selection

```r
optlambda=cvglmout$lambda.1se
print(optlambda)
```

```
## [1] 0.004476724
```

I would use

$$\lambda = 0.03716655$$

in this case.

# 4 Selected ETF

```
optlambdapos=which(cvglmout$glmnet.fit$lambda==optlambda)
glmnetout$beta[glmnetout$beta[,optlambdapos]!=0,optlambdapos]
```

```
##         IYW        TECS         XLK
##  0.17730140 -0.07376143  0.35970019
```

Therefore ITY, TECS, XLK are included in the final model.