Stat 154 Problem Set Four

Jinze Gu SID:24968967

March 11, 2014

Problem One

(1)

In terms of data cleaning: I used the record that have exactly 1342 dates(without the abnormal date with hurricane) of records, so my total number of company is 422. Since there are NA values on the list of company with more than 1342 records, so I did not use them.
SP500 index is calculated by the price of companies that are decided by the committee of StandardPoor using a linear relationship. However, the actual coefficients(weights) of different stocks remain undecided. Under the background knowledge, we can replicate the SP500 index by regressing on the price of those companies. Since we want to build a sparse portfolio, we can use lasso to construct our sparse portfolio. In order to evaluate the quality of my model, firstly, I tested if I should use elastic model or pure lasso by using different alpha(0 < alpha <= 1) to conduct regression. It turns out that the model is better when alpha is around 0.7. So I chose alpha to be 0.7.Then, I did cross validation and pick out the best lambda with samllest MSE; then I use that lambda to construct sparse portfolio. It turns out that my prediction of SP 500 index is very close to the actual value, which is verified by figure "Best Lambda Fit" where the lines(my prediction) matches perfectly with points(the actual value of SP500 index). As the result, we constructed a sparse portfolio where we use only the price of 58 companies to predict the SP500 index.

(2)

In this case, I divided the time range of stock into intervals where each one contains 60 days of record(The last one has only 22), then I constructed a function that figures out the sparse portfolio by lasso, guarantees the quality of model by crossvalidation and returns the names as well as corresponding coefficients of companies in the portfolio. It turns out that our portfolio is not stable enough, for example, the number of companies included in the portfolio varies with time, which is illustrated in the figure with name "Number of Companies in Portfolio SP500 Index". Even if for the same company, we have different weights(coefficients) over time, which may increase the cost transaction when we actually put the model into application. As an example, please check plot "CME weights in replicating SP500 index", which indicates the change of coefficients of "C M E Group INC" in replicating SP500 index
In order to get a portfolio that changes little overtime, we can add penalty of $|\beta_t - \beta_s|$ for each coefficient over time. It indicates that we penalize the model with unstable coefficients. I believe it would help us to figure out the portfolio that changes little over time.

(3)

In this case, we can modify the penalty of lasso for a little bit. Since we don't want to have negative values, then we change the penalty of negative β to be infinity while leaving the penalty of positive β unchanged. The optimization problem (minimizing $\text{argmin} squarederror + lassopenalty$), if convex, would easily come up with the result with only positive βs.

(4)

Since the relationship between SP500 return and the price of every stock is no longer linear, then I expect that we may need to expand the number of stocks included in the portfolio in order to have a better prediction.

First Part

It turns out we do need a larger portfolio, specifically, I need a large portion of companies in my portfolio in order to track SP500 returns while I only need 58 companies to track the SP500 index.

Stability

In terms of the stability of sparse model of tracking SP500 returns every 60 day, our result is not stable although we do require more companies compared with tracking the index. By the plot "Number of Compnaies in Portfolio SP500 Return",we still have a fluctuating sparse portfolio. Possibly we need to change the penalty in order to really fix the problem.

(5)

Since we do not need to concern about transaction problems, we can use the price of all the companies on our list to predict SP500 index using linear regression. I divided my dataset into training and test dataset. It turns out the prediction is really good where the squared error is about 274.9754(which is the minimum compared with lasso, ridge and elastic net). Using the same training and testing dataset, I tried lasso, ridge regression and elasticnet model, but it turns out linear regression has the smallest test error. Thus, regardless of transaction cost, we can replicate SP500 index by linear regression.

Problem Two

On the hand written page attached.

Problem Three

For this problem, I used LDA, QDA, logistic model and logistic with lasso to analyze the data. I notice that the model could be better(we do not need all of the predictors to do classification) if we use stepwise or stagewise selection to select predictors carefully. However, we are comparing the quality of classification method, as long as we use the same model to all of the method, we can have a relative comparison. In order to compare the quality of classifiers, I used two ways:
1). split the data into training set and test set and calculate the traning error and test error for every method. Compare training and test error.
2). figure out the false positive and false negative index among different classifiers.
In conclusion:
In terms of training error and test error(check plot "Training and Test Error"), it seems that LDA outweighs the other three methods and Logisticlasso also have a good prediction behaviour. However, Logisticlasso have the best false positive error(check the plot "False Positive False Negative") while QDA has the best false negative error. Back to the practical situation of diagnosing cancer, the cost of diagnosing a noncancer patient as having cancer may be larger than diagnosing a cancer patient as not having cancer. Thus, if we want to elminate false positive error, we are better off with logistic lasso model. If we want to eliminate the total prediction error, we probably want to use LDA.

Appendix: Code for Problem One

```
setwd("/Volumes/æIJĹèČ¡åĞžæšą/Stat 154/HW_4")
sp500 <- read.csv("500sp.csv")
Index <- read.csv("track.csv")
# I used the percentage of return times 100 so that I don't have very small values
SPReturn <- (Index[672:2013, 2]/Index[671:2012, 2] - 1) * 100
Index <- Index[672:2013, ]
head(levels(factor(Index$Calendar.Date)))
```

```
## [1] "20070904" "20070905" "20070906" "20070907" "20070910" "20070911"
```

```
sp500$PRC <- abs(sp500$PRC)
sp500$date <- as.factor(sp500$date)
Compname <- as.matrix(levels(factor(sp500$COMNAM)))
# Notice that one date has an abscure record
head(tabulate(sp500$date))
```

```
## [1] 500 500 500 500 500 500
```

```
levels(sp500$date)[1301]   # The date without common record.
```

```
## [1] "20121029"
```

```
# We want to know if the SP500 index contains this date, and we need to kick it out
sp500$PRC[which(sp500$date == "20121029")]
```

```
## [1] NA NA NA NA NA
```

```
sp500 <- sp500[-which(sp500$date == "20121029")]
tabulate(sp500$TICKER)
```

```
##   [1] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342   62 1342 1342 1342 1342
##  [18] 1342  721 1342 1342 1342 1342 1342 1342 1342 1342 1342  945 1342 1342 1342 1342 1342
##  [35] 1342 1342  566  776 1342 1342 1342 1342 1342 1342  361 1342 1342 1342 1342 1342 1342
##  [52] 1342 1342 1342 1342 1342 1342 1342 1342  312 1342  688 1342 1342 1342 1342 1342 1342
##  [69] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1343 1342 1342 1342 1342 1342 2684 1342
##  [86] 1342 1342 1342 1342 1342  839  193 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [103] 1924 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [120] 1342 1342 1342 1342 1342 1342 1342 1342 1342  229 1342 1342 1342 1342 1342  787 1342
## [137] 1342 1342 1342 1342  280 1342 1342 1342 1342 1342 1342 1174 1342 1342 1342 1342 1342
## [154] 1342  258 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1084 1342
## [171] 1343 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [188] 1342 1342 1342 1342 1342 1030 1241 1342  706 1342 1342 1342 1113 1339 1342 1342 1342
## [205]   20 1342 1342 1342  971 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1096 1342
## [222] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [239] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342  310 1342
## [256] 1342 1342 1342 1342 1342 1342 1342  268 1074 1342 1342 1342 1342 1281 1342 1342 1342
## [273]  475 1342 1342 1342   61 1342 1342 1140 1342 2684 1342 1032 1342 1342 1342 1342 1342
## [290] 1342  177 1149 1342 1342 1342 1342  202 1342 1342  595 1342 1342 1342 1342 1342 1342
## [307] 1342 1342 1342 1342   61 1342  945 1342  979 2684 1342 1342  546 1342 1342 1342 1342
## [324]  841  377 1342 1342 1342 1343 1342  602 1342 1342 1342  119 1342 1042  583 1342 1342
## [341] 1342 1342 1342  636 1342 1342 1342 1343 1342 1342 1342 1342 1342  344 1342 1342 1342
## [358] 1342 1342 1342  333 1009  162 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [375] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1343
```

```
## [392] 1342   43 1199 1342 1344 1342 1342 1342 1342 1342 1342 1342  168 1342 1342 1342 1342
## [409] 1342  630 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342
## [426] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1135  654 1342 1342
## [443] 1342 1342 1342 1342 1342 1342 1342 2684 1342 1342 1342 1342 1342 1342 1342 2684 1324
## [460] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342  257 1342 1342 1342 1342 1342 1342
## [477] 1342 1343 1342 1342 1342 1342 1342 1342 1342 1342 1342 1206 1342 1342 1071  271 1342
## [494] 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342  416  926 1342 1342 1342 1128 1342
## [511]  484 1342 1342  250 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342 1342  292 1342
## [528] 1342 1342 1342
```

```
# The date with SP500 index recorded
date_index <- levels(factor(Index$Calendar.Date))
price <- matrix(NA, nrow = length(date_index), ncol = length(Compname))
# I will select the company that have exactly 1342 dates of record.
f <- function(compname) {
    if (length(sp500$PRC[sp500$COMNAM == compname]) == 1342) {
        return(sp500$PRC[which(sp500$COMNAM == compname)][1:1342])
    }
}
price <- matrix(unlist(sapply(Compname, f)), nrow = 1342)
sum(price <= 0)  # Make sure there are not weired price_value
```

```
## [1] 0
```

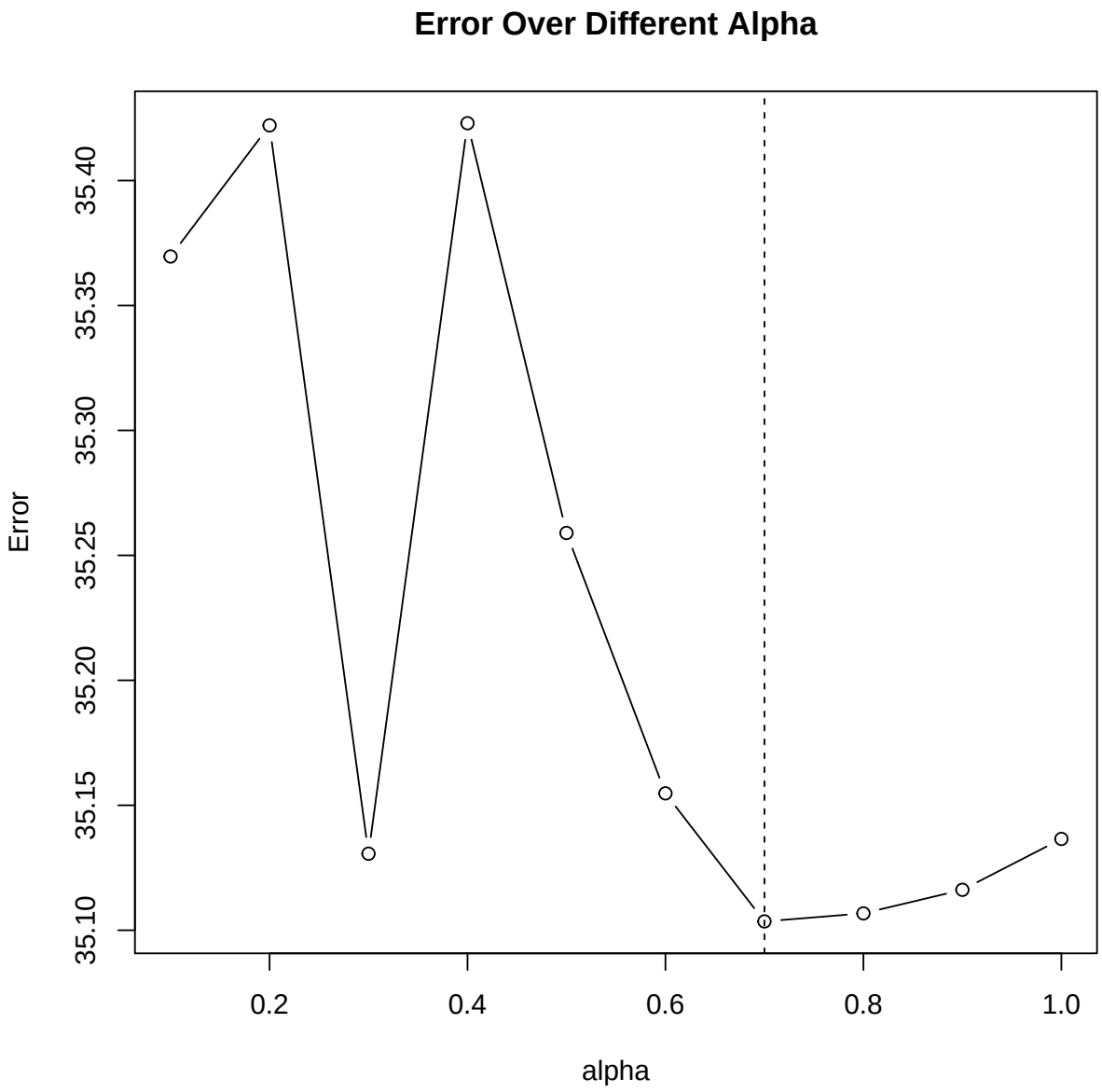
```
names <- levels(sp500$COMNAM)[which(tabulate(sp500$COMNAM) == 1342)]
colnames(price) <- names
# Now we want to check if the date of stock price match the date on which there are records
# of SP500 value, it turns out they match.
sum(levels(factor(sp500$date))[-1301] != date_index)
```

```
## [1] 0
```

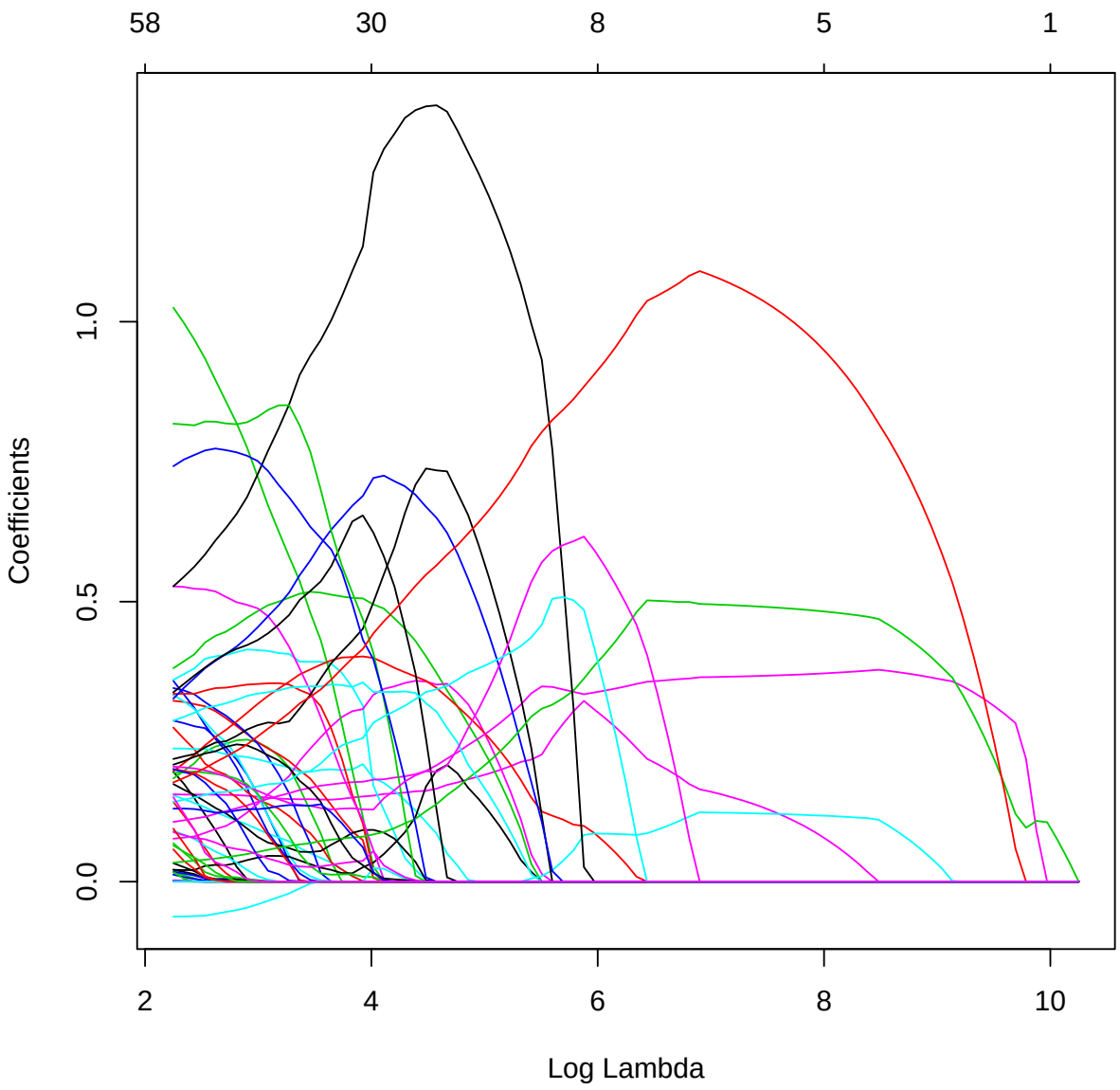
```
# 1)
spindex <- as.matrix(Index$SP.500.Level)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.0.2
## Loading required package: Matrix
## Loading required package: lattice
## Loaded glmnet 1.9-5
```

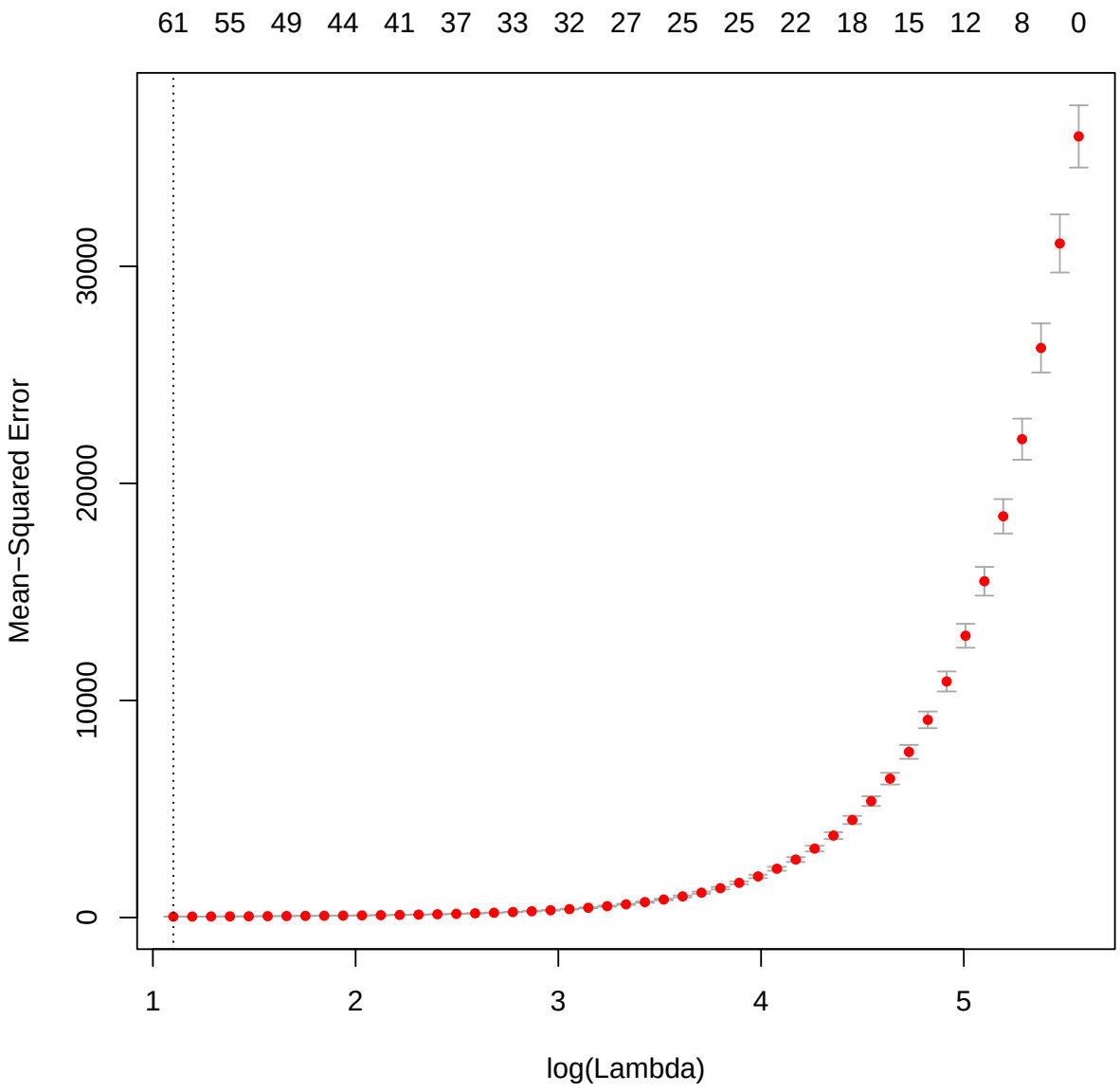
```
test_alpha <- function(a, y) {
    glmnet_fit <- glmnet(price, y, family = c("gaussian"), standardize = FALSE, nlambda = 100,
        alpha = a)
    cv_glmnet <- cv.glmnet(price, y, family = "gaussian", alpha = a, nfolds = 10)
    predict_glmnet <- predict(glmnet_fit, newx = price, s = cv_glmnet$lambda.min, type = "link")
    return(sum((predict_glmnet - y)^2)/length(y))
}
alpha <- seq(0.1, 1, by = 0.1)
alpha_test_error <- matrix(NA, length(alpha))
for (i in alpha) {
    alpha_test_error[i * 10, ] <- test_alpha(i, spindex)
}
plot(x = alpha, y = alpha_test_error, main = "Error Over Different Alpha", type = "b", xlab = "alpha",
    ylab = "Error")
abline(v = alpha[which.min(alpha_test_error)], lty = 2)
```



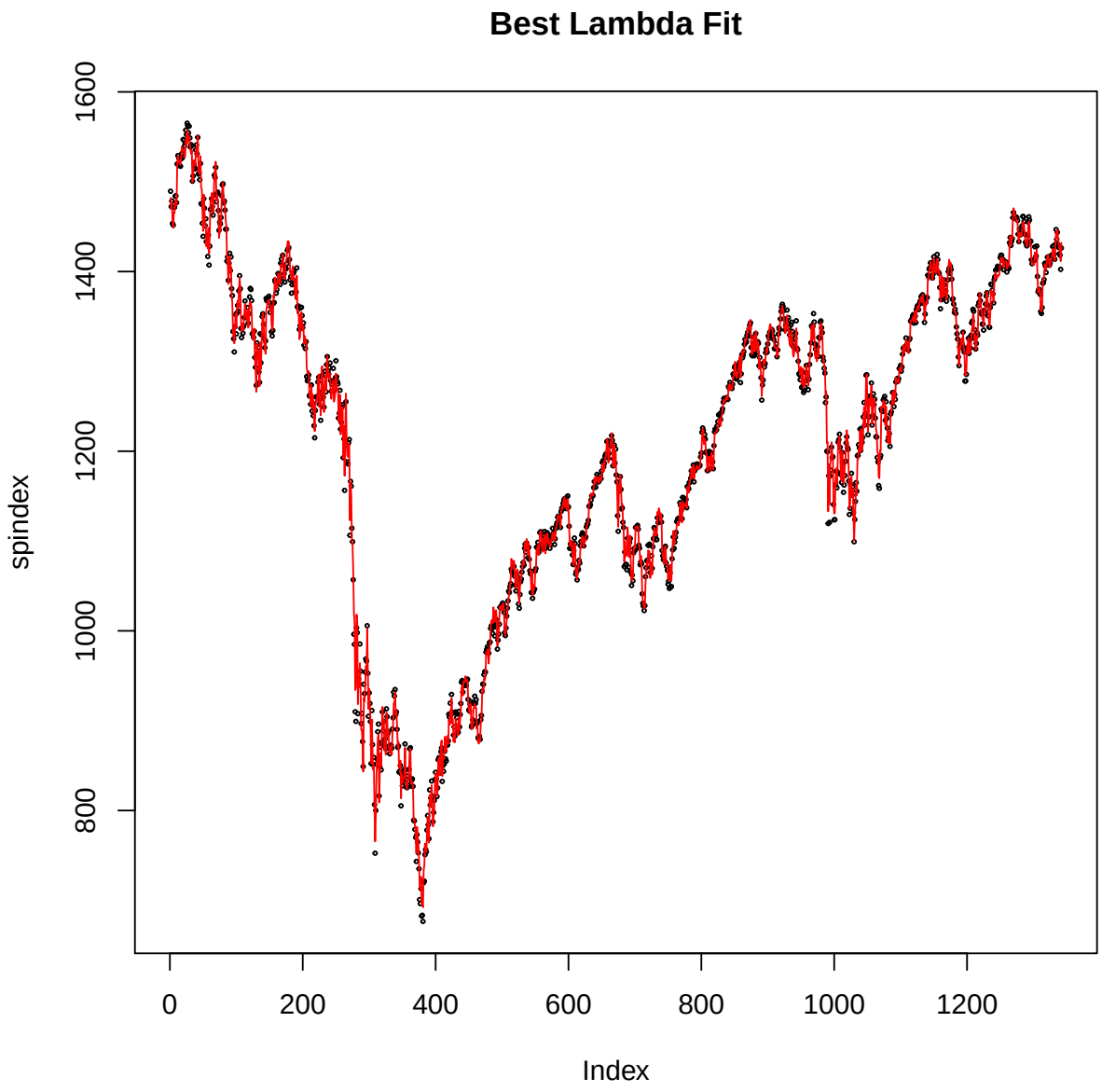

```
# So I chose alpha to be 0.7 in order to minimize the error.
glmnet_fit <- glmnet(price, spindex, family = "gaussian", standardize = FALSE, nlambda = 100,
    alpha = 0.7)
plot(glmnet_fit, xvar = "lambda")
```




```
cv_glmnet <- cv.glmnet(price, spindex, type.measure = "mse", family = "gaussian", alpha = 0.7,
    nfolds = 10)
plot(cv_glmnet)
```




```
predict_glmnet <- predict(glmnet_fit, newx = price, s = cv_glmnet$lambda.min, type = "link")
# We need to calculate the coefficient of lasso
coef_glmnet <- predict(glmnet_fit, newx = price, s = cv_glmnet$lambda.min, type = "coefficients")
plot(spindex, type = "p", cex = 0.3, main = "Best Lambda Fit")
lines(predict_glmnet, col = "red")
```




```
# The name of the company in our sparse portfolio
names[which(coef_glmnet != 0)[-1] - 1]
```

```
##  [1] "3M CO"                          "ALEXION PHARMACEUTICALS INC"
##  [3] "ALLEGHENY TECHNOLOGIES"         "ALTRIA GROUP INC"
##  [5] "AMAZON COM INC"                 "APPLE INC"
##  [7] "BARD C R INC"                   "BIOGEN IDEC INC"
##  [9] "BLACKROCK INC"                  "BOEING CO"
## [11] "BOSTON PROPERTIES INC"          "C F INDUSTRIES HOLDINGS INC"
## [13] "CAPITAL ONE FINANCIAL CORP"     "CHEVRON CORP NEW"
## [15] "CITIGROUP INC"                  "COGNIZANT TECHNOLOGY SOLS CORP"
## [17] "CUMMINS INC"                    "DEERE & CO"
## [19] "DEVON ENERGY CORP NEW"          "ENTERGY CORP NEW"
## [21] "EXXON MOBIL CORP"               "F 5 NETWORKS INC"
```

```
## [23] "FEDEX CORP"                       "FIRST SOLAR INC"
## [25] "FLOWSERVE CORP"                   "FLUOR CORP NEW"
## [27] "FRANKLIN RESOURCES INC"           "FREEPORT MCMORAN COPPER & GOLD"
## [29] "GOLDMAN SACHS GROUP INC"          "GOOGLE INC"
## [31] "GRAINGER W W INC"                 "HARMAN INTL INDS INC NEW"
## [33] "HARTFORD FINANCIAL SVCS GRP INC" "INTERCONTINENTALEXCHANGE INC"
## [35] "LOCKHEED MARTIN CORP"             "M & T BANK CORP"
## [37] "MONSANTO CO NEW"                  "MOODYS CORP"
## [39] "NATIONAL OILWELL VARCO INC"       "NETFLIX INC"
## [41] "NORTHROP GRUMMAN CORP"            "P P G INDUSTRIES INC"
## [43] "PACCAR INC"                       "PRICELINE COM INC"
## [45] "PRUDENTIAL FINANCIAL INC"         "SCHLUMBERGER LTD"
## [47] "SIMON PROPERTY GROUP INC NEW"     "STATE STREET CORP"
## [49] "SUNTRUST BANKS INC"               "TORCHMARK CORP"
## [51] "UNION PACIFIC CORP"               "UNITED STATES STEEL CORP NEW"
## [53] "VORNADO REALTY TRUST"             "WASHINGTON POST CO"
## [55] "WELLPOINT INC"                    "WHIRLPOOL CORP"
## [57] "WYNN RESORTS LTD"                 "ZIMMER HOLDINGS INC"

# 2)
group <- seq(0, 1342, by = 60)
portfolio <- function(n, folds) {
    X <- price[max(1, group[n]):group[n + 1], ]
    Y <- spindex[max(1, group[n]):group[n + 1]]
    glmnet.fit <- glmnet(X, Y, family = c("gaussian"), standardize = FALSE, nlambda = 100,
        alpha = 0.7)
    glmnet.cv <- cv.glmnet(X, Y, family = "gaussian", alpha = 0.7, nfolds = folds)
    lambda <- glmnet.cv$lambda.min
    glmnet.coef <- coef(glmnet.fit, s = lambda)
    coefficient <- glmnet.coef[which(glmnet.coef != 0)]
    prediction <- predict(glmnet_fit, newx = X, s = lambda, type = "link")
    return(list(lambda = lambda, prediction = as.numeric(prediction), coefficient = coefficient,
        compname = names[which(glmnet.coef != 0)[-1] - 1]))
}
# Sample Output
portfolio(1, 10)$compname  # Company List in the first 60 days.

##  [1] "AVALONBAY COMMUNITIES INC"      "C M E GROUP INC"
##  [3] "CAPITAL ONE FINANCIAL CORP"     "COGNIZANT TECHNOLOGY SOLS CORP"
##  [5] "CUMMINS INC"                    "FIRST SOLAR INC"
##  [7] "FLUOR CORP NEW"                 "FRANKLIN RESOURCES INC"
##  [9] "FREEPORT MCMORAN COPPER & GOLD" "GOLDMAN SACHS GROUP INC"
## [11] "HARMAN INTL INDS INC NEW"       "NATIONAL OILWELL VARCO INC"
## [13] "PACCAR INC"                     "SCHLUMBERGER LTD"
## [15] "UNITED STATES STEEL CORP NEW"   "WASHINGTON POST CO"
## [17] "WYNN RESORTS LTD"

portfolio(2, 10)$compname  # Company List in the second 60 days.

## [1] "BORGWARNER INC"             "C M E GROUP INC"            "CAMERON INTERNATIONAL CORP"
## [4] "CUMMINS INC"                "FIRST SOLAR INC"            "FLUOR CORP NEW"
## [7] "GOOGLE INC"                 "INTUITIVE SURGICAL INC"     "WASHINGTON POST CO"

portfolio(3, 10)$compname  # Company List in the third 60 days.
```

```
## [1] "BLACKROCK INC"           "C M E GROUP INC"         "FIRST SOLAR INC"
## [4] "FLUOR CORP NEW"          "GOLDMAN SACHS GROUP INC" "GOOGLE INC"
## [7] "INTUITIVE SURGICAL INC"  "WASHINGTON POST CO"
```

```
portfolio(4, 10)$compname  # Company List in the fourth 60 days.
```

```
## [1] "BLACKROCK INC"                "C M E GROUP INC"
## [3] "FIRST SOLAR INC"              "FLUOR CORP NEW"
## [5] "GOOGLE INC"                   "HESS CORP"
## [7] "INTERCONTINENTALEXCHANGE INC" "UNION PACIFIC CORP"
## [9] "WASHINGTON POST CO"
```

```
portfolio(5, 10)$compname  # Company List in the fifth 60 days.
```

```
## [1] "FIRST SOLAR INC"         "GOLDMAN SACHS GROUP INC" "GOOGLE INC"
## [4] "INTUITIVE SURGICAL INC"  "WASHINGTON POST CO"
```

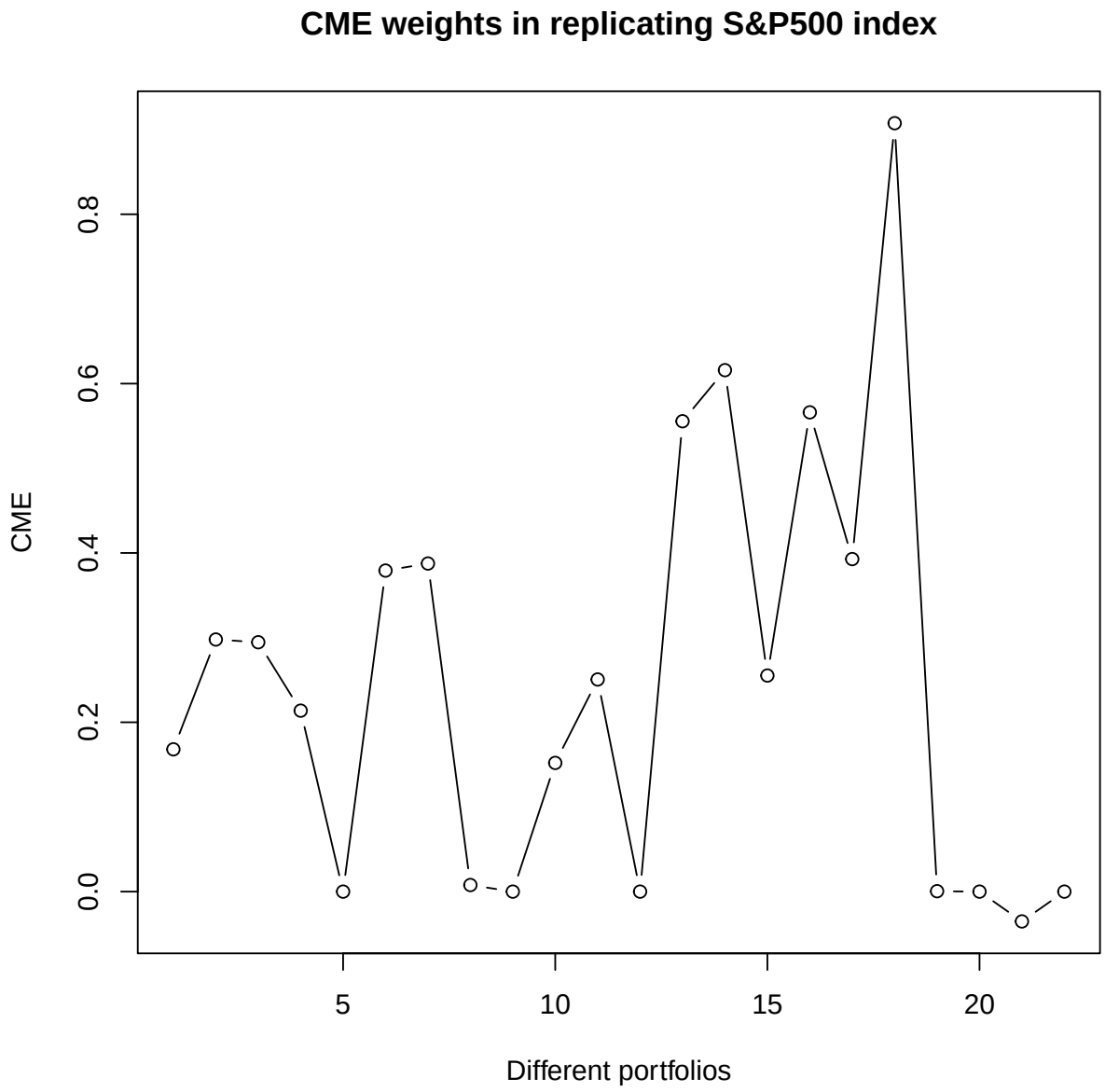
```
portfolio(10, 10)$compname  # Company List in the tenth 60 days.
```

```
##  [1] "APPLE INC"                      "BLACKROCK INC"
##  [3] "C M E GROUP INC"                "EBIX INC"
##  [5] "FLOWSERVE CORP"                 "FREEPORT MCMORAN COPPER & GOLD"
##  [7] "INTUITIVE SURGICAL INC"         "MASTERCARD INC"
##  [9] "PRICELINE COM INC"              "WYNN RESORTS LTD"
```

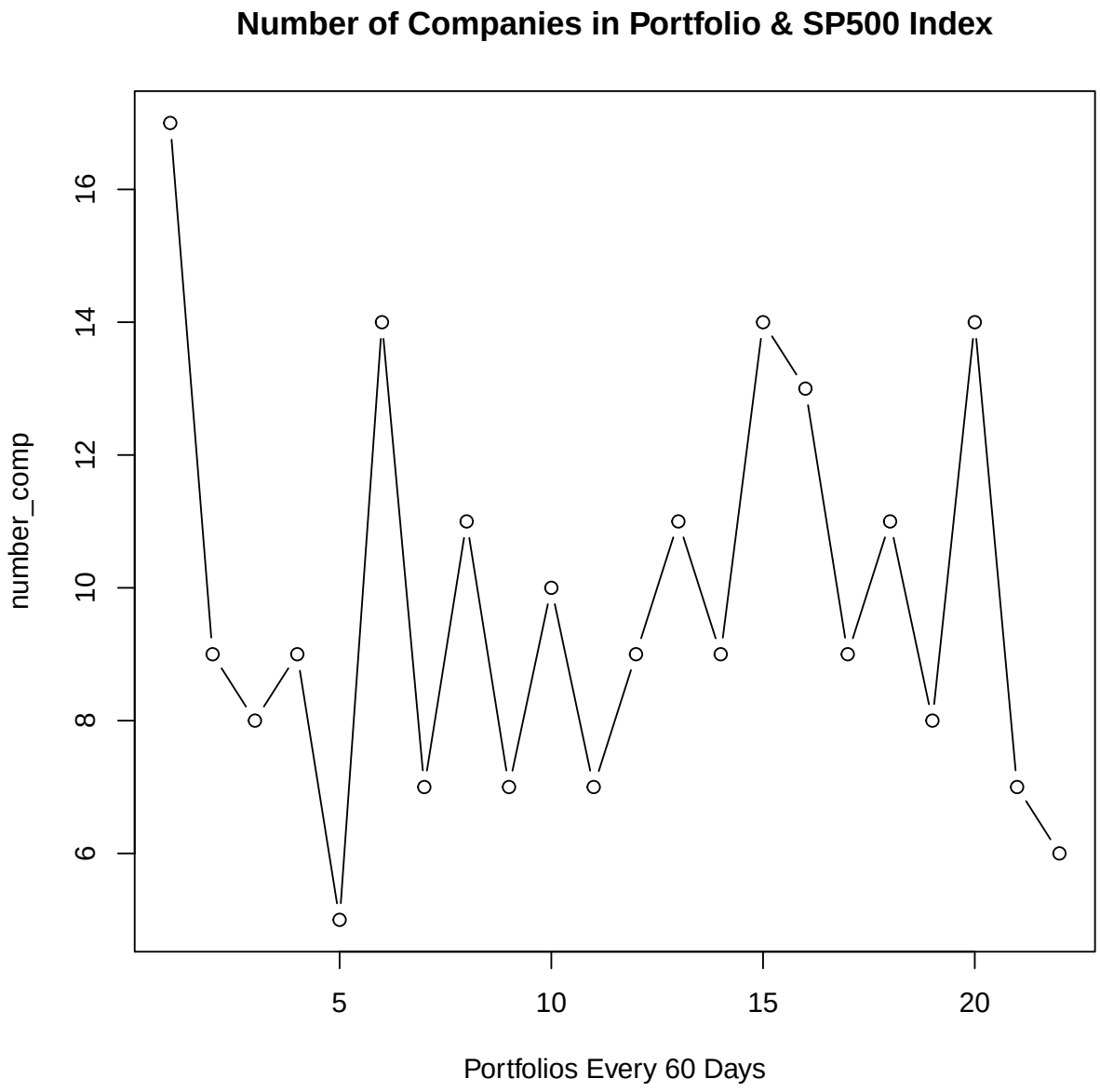
```
portfolio(15, 10)$compname  # Company List in the fiftith 60 days.
```

```
##  [1] "APPLE INC"                      "AUTOZONE INC"
##  [3] "BALL CORP"                      "C M E GROUP INC"
##  [5] "FLOWSERVE CORP"                 "FREEPORT MCMORAN COPPER & GOLD"
##  [7] "GOLDMAN SACHS GROUP INC"        "GOOGLE INC"
##  [9] "INTUITIVE SURGICAL INC"         "LAUDER ESTEE COS INC"
## [11] "NETFLIX INC"                    "PRICELINE COM INC"
## [13] "SCHLUMBERGER LTD"               "WASHINGTON POST CO"
```

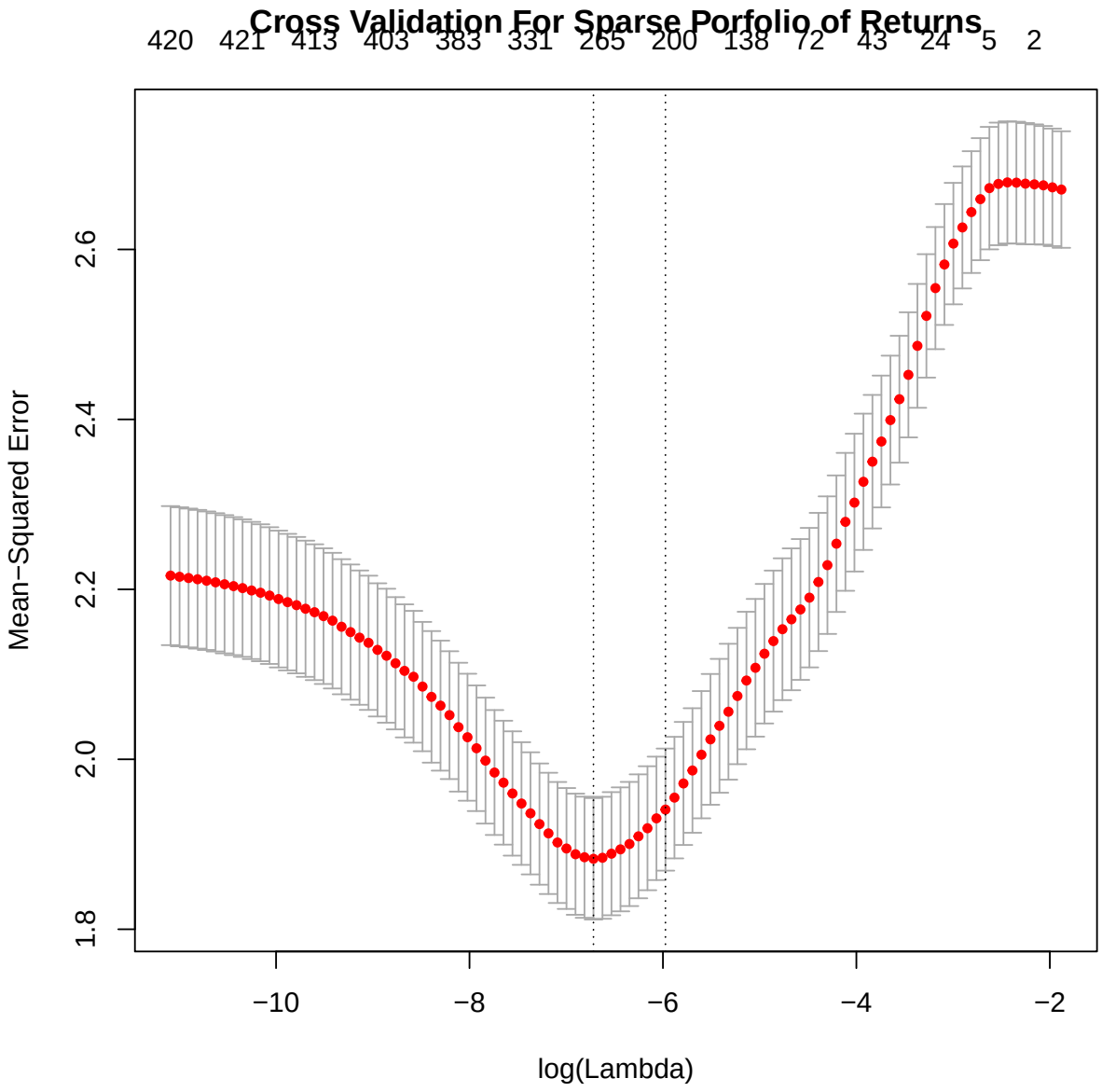
```
CME <- matrix(0, 22)
for (i in 1:22) {
    if (sum(portfolio(i, 10)$compname == "C M E GROUP INC") != 0) {
        j <- which(portfolio(i, 10)$compname == "C M E GROUP INC")
        CME[i] <- portfolio(i, 10)$coefficient[(j + 1)]
    }
}
plot(CME, main = "CME weights in replicating S&P500 index", type = "b", xlab = "Different portfolios")
```




```
# I need to figure out the stability of company list.
number_comp <- matrix(NA, 22)
name_comp <- matrix(NA, 22, 17)
for (i in 1:22) {
    number_comp[i, ] <- length(portfolio(i, 10)$compname)
}
plot(number_comp, main = "Number of Companies in Portfolio & SP500 Index", type = "b", xlab = "Portfolio
```




```
# 4).
Return.lasso <- glmnet(price, SPReturn, family = "gaussian", standardize = FALSE, alpha = 1)
Return.cv <- cv.glmnet(price, as.matrix(SPReturn), family = "gaussian", alpha = 1)
plot(Return.cv, main = "Cross Validation For Sparse Porfolio of Returns")
```




```
Return.coef <- coef(Return.lasso, s = Return.cv$lambda.min, alpha = 1)
length(which(Return.coef != 0)) - 1  # The number of coefficients minus one intercept

## [1] 408

portfolio_return <- function(n, folds) {
    X <- price[max(1, group[n]):group[n + 1], ]
    Y <- SPReturn[max(1, group[n]):group[n + 1]]
    glmnet.fit <- glmnet(X, Y, family = c("gaussian"), standardize = FALSE, nlambda = 100,
        alpha = 0.7)
    glmnet.cv <- cv.glmnet(X, Y, family = "gaussian", alpha = 0.7, nfolds = folds)
    lambda <- glmnet.cv$lambda.min
    glmnet.coef <- coef(glmnet.fit, s = lambda)
    coefficient <- glmnet.coef[which(glmnet.coef != 0)]
```

```
    prediction <- predict(glmnet_fit, newx = X, s = lambda, type = "link")
    return(list(lambda = lambda, prediction = as.numeric(prediction), coefficient = coefficient,
        compname = names[which(glmnet.coef != 0)[-1] - 1]))
}
portfolio_return(1, 10)$compname
```

```
##  [1] "A F L A C INC"                    "AMAZON COM INC"
##  [3] "APOLLO GROUP INC"                 "APPLE INC"
##  [5] "ASSURANT INC"                     "AVALONBAY COMMUNITIES INC"
##  [7] "BIOGEN IDEC INC"                  "CELGENE CORP"
##  [9] "CONSOL ENERGY INC"                "DOMINION RESOURCES INC VA NEW"
## [11] "F M C CORP"                       "FEDEX CORP"
## [13] "FIRST SOLAR INC"                  "FLIR SYSTEMS INC"
## [15] "FLOWSERVE CORP"                   "FRANKLIN RESOURCES INC"
## [17] "FREEPORT MCMORAN COPPER & GOLD"   "GARMIN LTD"
## [19] "GOOGLE INC"                       "HARTFORD FINANCIAL SVCS GRP INC"
## [21] "HUMANA INC"                       "INTERCONTINENTALEXCHANGE INC"
## [23] "INTERNATIONAL BUSINESS MACHS COR" "INTUITIVE SURGICAL INC"
## [25] "JACOBS ENGINEERING GROUP INC"     "KIMBERLY CLARK CORP"
## [27] "L 3 COMMUNICATIONS HLDGS INC"     "LEGG MASON INC"
## [29] "M & T BANK CORP"                  "P N C FINANCIAL SERVICES GRP INC"
## [31] "PACCAR INC"                       "PARKER HANNIFIN CORP"
## [33] "STATE STREET CORP"                "T ROWE PRICE GROUP INC"
## [35] "UNION PACIFIC CORP"
```

```
portfolio_return(2, 10)$compname
```

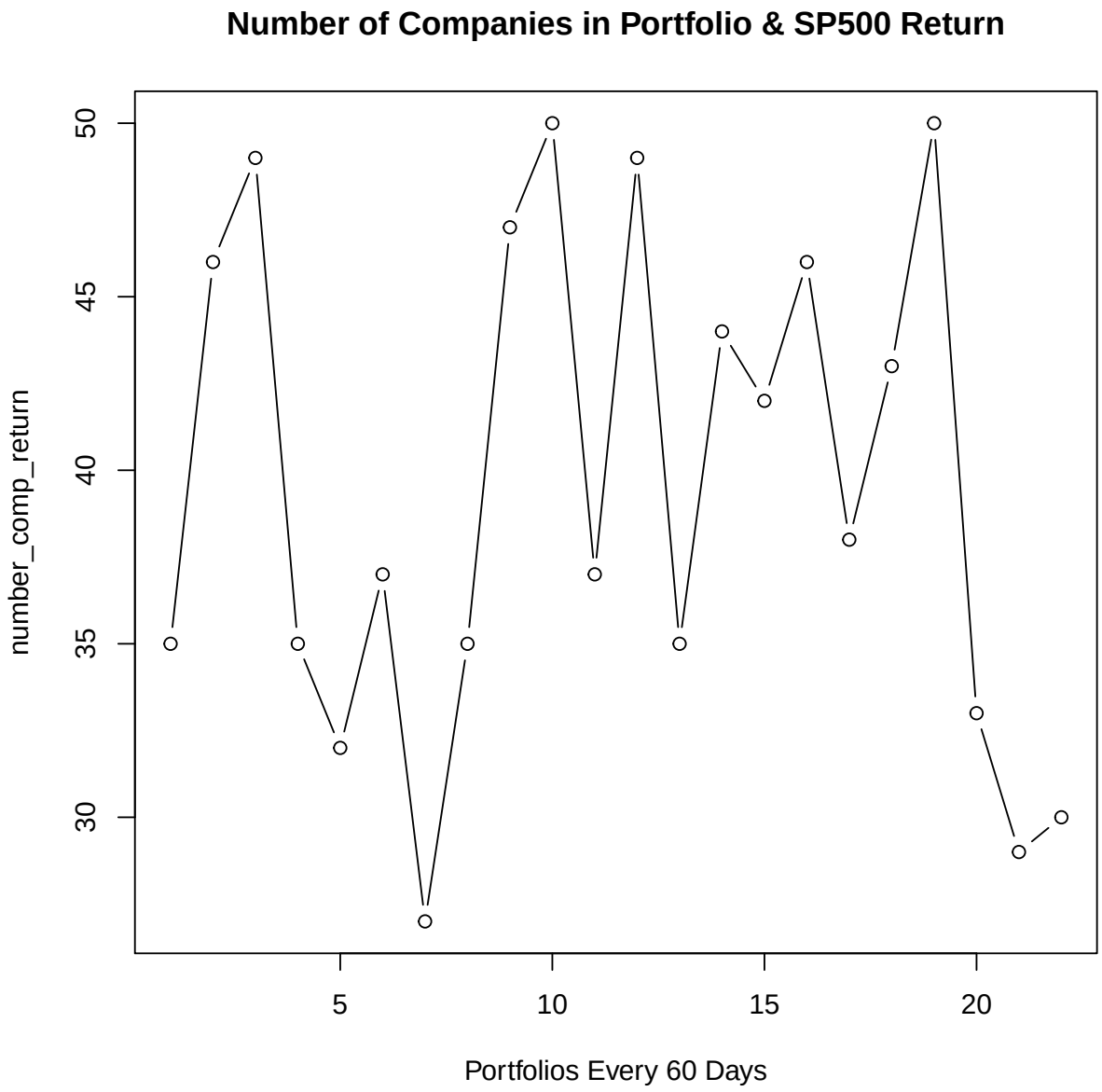
```
##  [1] "ABERCROMBIE & FITCH CO"           "ALLEGHENY TECHNOLOGIES"
##  [3] "AMERICAN EXPRESS CO"              "APACHE CORP"
##  [5] "APOLLO GROUP INC"                 "BAKER HUGHES INC"
##  [7] "BLACKROCK INC"                    "C M E GROUP INC"
##  [9] "CAMERON INTERNATIONAL CORP"       "CELGENE CORP"
## [11] "CERNER CORP"                      "CUMMINS INC"
## [13] "DEERE & CO"                       "DENBURY RESOURCES INC"
## [15] "ENTERGY CORP NEW"                 "EOG RESOURCES INC"
## [17] "FIRST SOLAR INC"                  "FREEPORT MCMORAN COPPER & GOLD"
## [19] "GAMESTOP CORP NEW"                "GOOGLE INC"
## [21] "HARMAN INTL INDS INC NEW"         "INTERCONTINENTALEXCHANGE INC"
## [23] "INTERNATIONAL BUSINESS MACHS COR" "INTUITIVE SURGICAL INC"
## [25] "JACOBS ENGINEERING GROUP INC"     "LABORATORY CORP AMERICA HLDGS"
## [27] "LAM RESH CORP"                    "LEGG MASON INC"
## [29] "M & T BANK CORP"                  "MASTERCARD INC"
## [31] "MONSANTO CO NEW"                  "NORDSTROM INC"
## [33] "NORTHROP GRUMMAN CORP"            "P P G INDUSTRIES INC"
## [35] "PACCAR INC"                       "PEPSICO INC"
## [37] "PRICELINE COM INC"                "PRINCIPAL FINANCIAL GROUP INC"
## [39] "PUBLIC SERVICE ENTERPRISE GP INC" "SHERWIN WILLIAMS CO"
## [41] "TARGET CORP"                      "TEXTRON INC"
## [43] "UNITED STATES STEEL CORP NEW"     "V F CORP"
## [45] "WASHINGTON POST CO"               "WYNN RESORTS LTD"
```

```
portfolio_return(5, 10)$compname
```

```
##  [1] "ALEXION PHARMACEUTICALS INC"     "APACHE CORP"
##  [3] "APPLE INC"                       "AVALONBAY COMMUNITIES INC"
```

```
##  [5] "BLACKROCK INC"                    "BOSTON PROPERTIES INC"
##  [7] "C F INDUSTRIES HOLDINGS INC"      "C M E GROUP INC"
##  [9] "CHUBB CORP"                       "CONSOL ENERGY INC"
## [11] "DIAMOND OFFSHORE DRILLING INC"    "DUN & BRADSTREET CORP DEL NEW"
## [13] "EXXON MOBIL CORP"                 "FIRST SOLAR INC"
## [15] "FRANKLIN RESOURCES INC"           "GOLDMAN SACHS GROUP INC"
## [17] "GOOGLE INC"                       "HARTFORD FINANCIAL SVCS GRP INC"
## [19] "INTERCONTINENTALEXCHANGE INC"     "INTUITIVE SURGICAL INC"
## [21] "JOHNSON & JOHNSON"                "L 3 COMMUNICATIONS HLDGS INC"
## [23] "M & T BANK CORP"                  "METLIFE INC"
## [25] "NIKE INC"                         "NORTHERN TRUST CORP"
## [27] "SIMON PROPERTY GROUP INC NEW"     "TORCHMARK CORP"
## [29] "V F CORP"                         "VORNADO REALTY TRUST"
## [31] "WASHINGTON POST CO"               "WYNN RESORTS LTD"

portfolio_return(10, 10)$compname

##  [1] "ANADARKO PETROLEUM CORP"          "APOLLO GROUP INC"
##  [3] "AUTOZONE INC"                     "BARD C R INC"
##  [5] "BAXTER INTERNATIONAL INC"         "BEST BUY COMPANY INC"
##  [7] "BLACKROCK INC"                    "C S X CORP"
##  [9] "CABOT OIL & GAS CORP"             "CONSOL ENERGY INC"
## [11] "CUMMINS INC"                      "DEERE & CO"
## [13] "DIAMOND OFFSHORE DRILLING INC"    "EASTMAN CHEMICAL CO"
## [15] "EBIX INC"                         "ENTERGY CORP NEW"
## [17] "EXXON MOBIL CORP"                 "GARMIN LTD"
## [19] "GOLDMAN SACHS GROUP INC"          "GOOGLE INC"
## [21] "HEALTH CARE REIT INC"             "INTERCONTINENTALEXCHANGE INC"
## [23] "INTUITIVE SURGICAL INC"           "JOY GLOBAL INC"
## [25] "KIMBERLY CLARK CORP"              "LEGG MASON INC"
## [27] "M & T BANK CORP"                  "MASTERCARD INC"
## [29] "MONSANTO CO NEW"                  "NORTHERN TRUST CORP"
## [31] "PRICELINE COM INC"                "REGENERON PHARMACEUTICALS INC"
## [33] "SEMPRA ENERGY"                    "SIGMA ALDRICH CORP"
## [35] "SMUCKER J M CO"                   "STARWOOD HOTELS & REST WLDWD INC"
## [37] "STERICYCLE INC"                   "V F CORP"
## [39] "VARIAN MEDICAL SYSTEMS INC"       "VORNADO REALTY TRUST"
## [41] "WYNN RESORTS LTD"

portfolio_return(15, 10)$compname

##  [1] "AGILENT TECHNOLOGIES INC"         "ALEXION PHARMACEUTICALS INC"
##  [3] "APACHE CORP"                      "APOLLO GROUP INC"
##  [5] "APPLE INC"                        "AUTOZONE INC"
##  [7] "AVALONBAY COMMUNITIES INC"        "BALL CORP"
##  [9] "C M E GROUP INC"                  "CH ROBINSON WORLDWIDE INC"
## [11] "CONSOL ENERGY INC"                "F 5 NETWORKS INC"
## [13] "F M C TECHNOLOGIES INC"           "FIRST SOLAR INC"
## [15] "FLOWSERVE CORP"                   "FREEPORT MCMORAN COPPER & GOLD"
## [17] "GOLDMAN SACHS GROUP INC"          "GOOGLE INC"
## [19] "GRAINGER W W INC"                 "HELMERICH & PAYNE INC"
## [21] "INTERNATIONAL BUSINESS MACHS COR" "INTUITIVE SURGICAL INC"
## [23] "JOY GLOBAL INC"                   "LABORATORY CORP AMERICA HLDGS"
## [25] "LOCKHEED MARTIN CORP"             "M & T BANK CORP"
## [27] "MASTERCARD INC"                   "MONSANTO CO NEW"
```

```
## [29] "NATIONAL OILWELL VARCO INC"       "NETFLIX INC"
## [31] "NEWMONT MINING CORP"              "NORFOLK SOUTHERN CORP"
## [33] "P N C FINANCIAL SERVICES GRP INC" "PALL CORP"
## [35] "PRICELINE COM INC"                "ROCKWELL AUTOMATION INC"
## [37] "ROSS STORES INC"                  "STARWOOD HOTELS & REST WLDWD INC"
## [39] "UNITED STATES STEEL CORP NEW"     "WASHINGTON POST CO"
## [41] "WESTERN DIGITAL CORP"             "WISCONSIN ENERGY CORP"
## [43] "WYNN RESORTS LTD"

portfolio_return(20, 10)$compname

##  [1] "ABERCROMBIE & FITCH CO"          "AKAMAI TECHNOLOGIES INC"
##  [3] "ALEXION PHARMACEUTICALS INC"     "AMAZON COM INC"
##  [5] "APPLE INC"                       "AVALONBAY COMMUNITIES INC"
##  [7] "BAXTER INTERNATIONAL INC"        "BIOGEN IDEC INC"
##  [9] "C M E GROUP INC"                 "CELGENE CORP"
## [11] "CROWN CASTLE INTERNATIONAL CORP" "EDWARDS LIFESCIENCES CORP"
## [13] "F 5 NETWORKS INC"                "FLOWSERVE CORP"
## [15] "FOSSIL INC"                      "GOLDMAN SACHS GROUP INC"
## [17] "GOOGLE INC"                      "INTERCONTINENTALEXCHANGE INC"
## [19] "INTUITIVE SURGICAL INC"          "JOY GLOBAL INC"
## [21] "KOHLS CORP"                      "M & T BANK CORP"
## [23] "MASTERCARD INC"                  "NOBLE ENERGY INC"
## [25] "ONEOK INC NEW"                   "P P G INDUSTRIES INC"
## [27] "PRECISION CASTPARTS CORP"        "PRICELINE COM INC"
## [29] "SALESFORCE COM INC"              "SANDISK CORP"
## [31] "SCHLUMBERGER LTD"                "UNITED STATES STEEL CORP NEW"
## [33] "WASHINGTON POST CO"

number_comp_return <- matrix(NA, 22)
for (i in 1:22) {
    number_comp_return[i, ] <- length(portfolio_return(i, 10)$compname)
}
plot(number_comp_return, main = "Number of Companies in Portfolio & SP500 Return", type = "b",
    xlab = "Portfolios Every 60 Days")
```



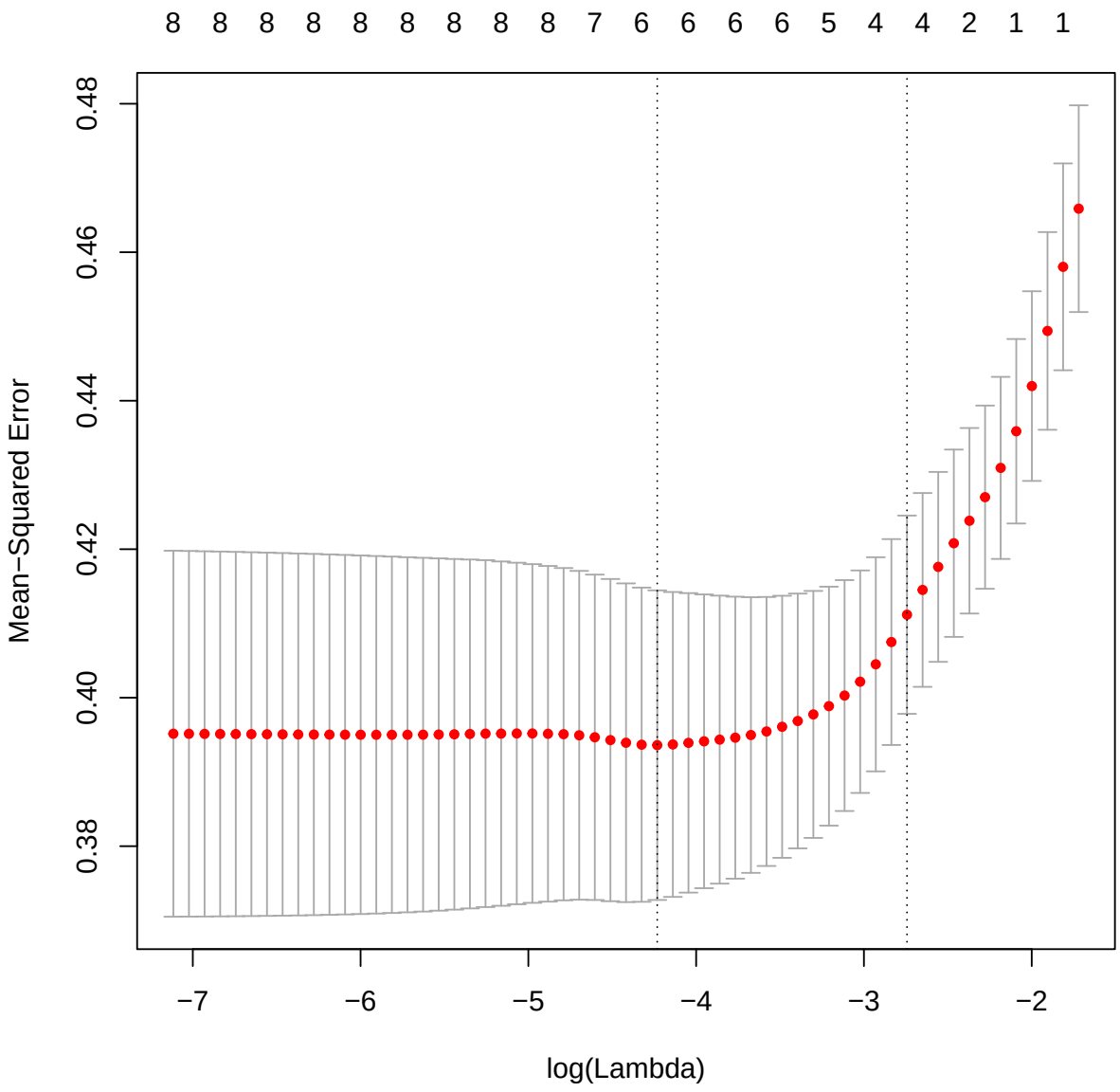

```
# 5).
sp500_lm <- lm(spindex[1:800] ~ price[1:800, ])
prediction <- predict(sp500_lm, interval = "none", type = "response")
sum((prediction - spindex[1:800])^2)/800

## [1] 0.09587

lm.test <- cbind(rep(1, 542), price[801:1342, ]) %*% sp500_lm$coefficients
sum((lm.test - spindex[801:1342])^2)/542

## [1] 275

sp500_ridge <- glmnet(price[1:800, ], spindex[1:800], standardize = FALSE, family = "gaussian",
    alpha = 1)
ridge.cv <- cv.glmnet(price[1:800, ], spindex[1:800], family = "gaussian", type.measure = "mse",
```

```
    alpha = 1)
ridge.fit <- predict(ridge.cv, price[801:1342, ], s = ridge.cv$lambda.min, type = "link")
sum((ridge.fit - spindex[801:1342])^2)/542

## [1] 488.4
```

Appendix: Code for Problem Three

```
setwd("/Volumes/æIJĽèČ¡åĞžæšą/Stat 154/HW_4")
SA <- read.csv("SouthAfrica.csv", header = TRUE)
Response <- SA[, 11]
Response_train <- Response[1:300]
Response_test <- Response[301:length(Response)]
X <- SA[, c(2, 3, 4, 5, 7, 8, 9, 10)]
X_train <- X[1:300, ]
X_test <- X[301:length(Response), ]
Train.error <- matrix(NA, 4)
Test.error <- matrix(NA, 4)
# I wrote a function to detect the false positive and false negative rate.
fpfn <- function(predict, original) {
    fp <- sum(original[which(predict == 1)] != 1)
    fn <- sum(original[which(predict == 0)] != 0)
    return(c(`False Positive` = fp, `False Negative` = fn))
}
# LDA and variants LDA classifier
library(MASS)
LDA <- lda(X_train, grouping = as.factor(Response_train))
LDA_train <- predict(LDA, X_train)$class
# Train error of LDA
Train.error[1, ] <- sum(as.factor(Response_train) != LDA_train)/length(Response_train)
sum(as.factor(Response_train) != LDA_train)/length(Response_train)

## [1] 0.3133

# Test error of LDA:
LDA_test <- predict(LDA, X_test)$class
Test.error[1, ] <- sum(as.factor(Response_test) != LDA_test)/length(Response_test)
sum(as.factor(Response_test) != LDA_test)/length(Response_test)

## [1] 0.2346

fpfn(LDA_test, Response_test)

## False Positive False Negative
##             11             27

# QDA classifier
QDA <- qda(X_train, grouping = as.factor(Response_train))
QDA_train <- predict(QDA, X_train)$class
# Train error of RDA
Train.error[2, ] <- sum(as.factor(Response_train) != QDA_train)/length(Response_train)
sum(as.factor(Response_train) != QDA_train)/length(Response_train)
```

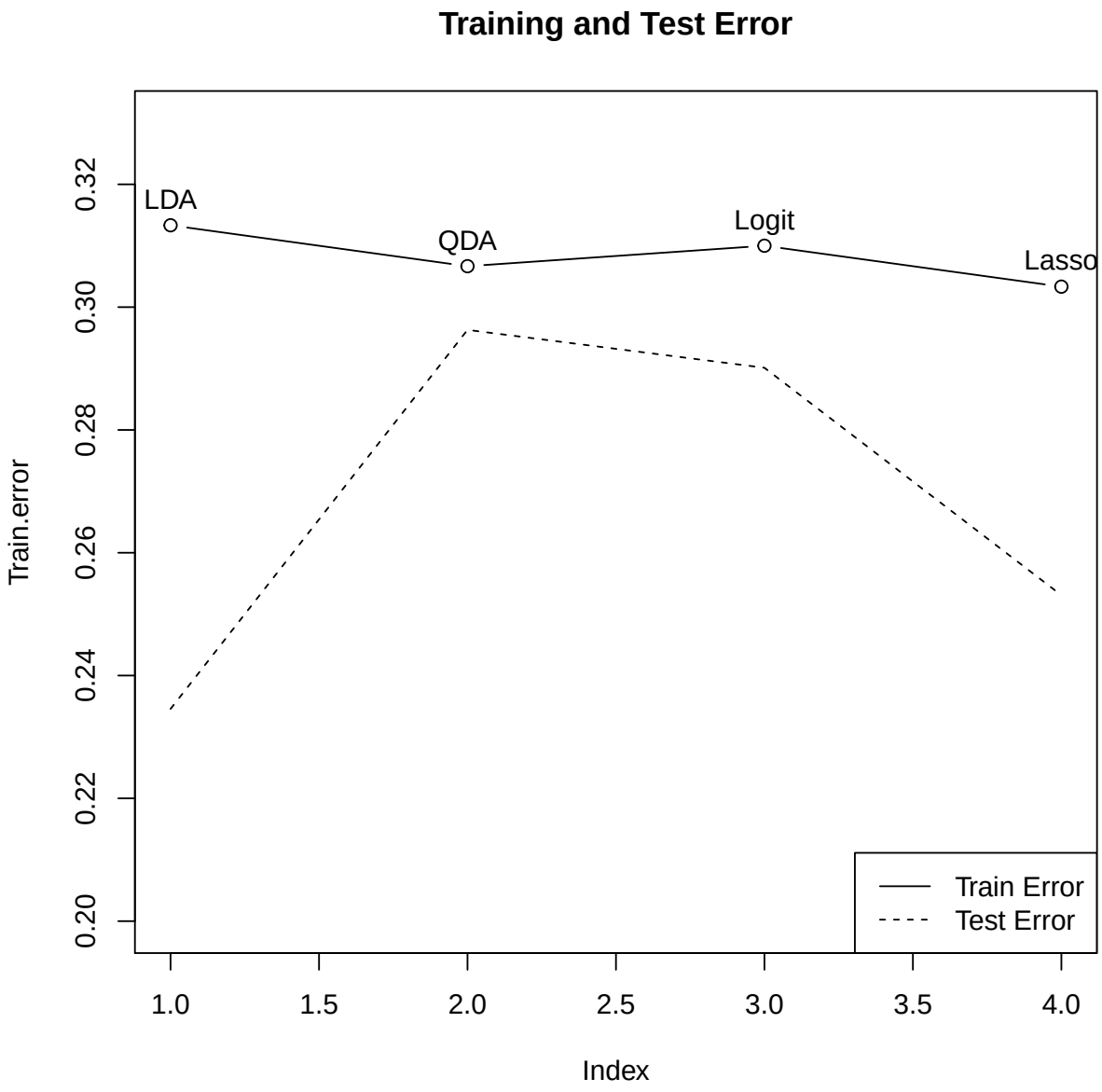
```
## [1] 0.3067

# Test error of LDA:
QDA_test <- predict(QDA, X_test)$class
Test.error[2, ] <- sum(as.factor(Response_test) != QDA_test)/length(Response_test)
sum(as.factor(Response_test) != QDA_test)/length(Response_test)

## [1] 0.2963

fpfn(QDA_test, Response_test)

## False Positive False Negative
##             26             22

# Logistic Regression.
logistic <- glm(Response_train ~ as.matrix(X_train) + 0, family = "binomial")
logistic_train <- predict(logistic, type = "response")
logistic_train[logistic_train >= 0.5] <- 1
logistic_train[logistic_train < 0.5] <- 0
Train.error[3, ] <- sum(Response_train != logistic_train)/300
sum(Response_train != logistic_train)/300

## [1] 0.31

logistic_test <- as.matrix(X_test) %*% logistic$coefficients
logistic_test[which(logistic_test < 0)] <- 0
logistic_test[which(logistic_test > 0)] <- 1
Test.error[3, ] <- sum(Response_test != logistic_test)/length(Response_test)
fpfn(logistic_test, Response_test)

## False Positive False Negative
##             16             31

# Logistic Lasso
library(glmnet)
lasso <- glmnet(as.matrix(X_train), Response_train, family = "binomial", alpha = 1)
cv.lasso <- cv.glmnet(as.matrix(X_train), Response_train, family = "binomial", type.measure = "mse")
plot(cv.lasso)
```




```
lasso_train <- predict(lasso, as.matrix(X_train), s = cv.lasso$lambda.min, type = "class")
Train.error[4, ] <- sum(as.factor(lasso_train) != as.factor(Response_train))/length(Response_train)
sum(as.factor(lasso_train) != as.factor(Response_train))/length(Response_train)

## [1] 0.3033

lasso_test <- predict(lasso, as.matrix(X_test), s = cv.lasso$lambda.min, type = "class")
Test.error[4, ] <- sum(as.factor(lasso_test) != as.factor(Response_test))/length(Response_test)
sum(as.factor(lasso_test) != as.factor(Response_test))/length(Response_test)

## [1] 0.2531

# Detect false positive and false negative
fpfn(lasso_test, Response_test)
```

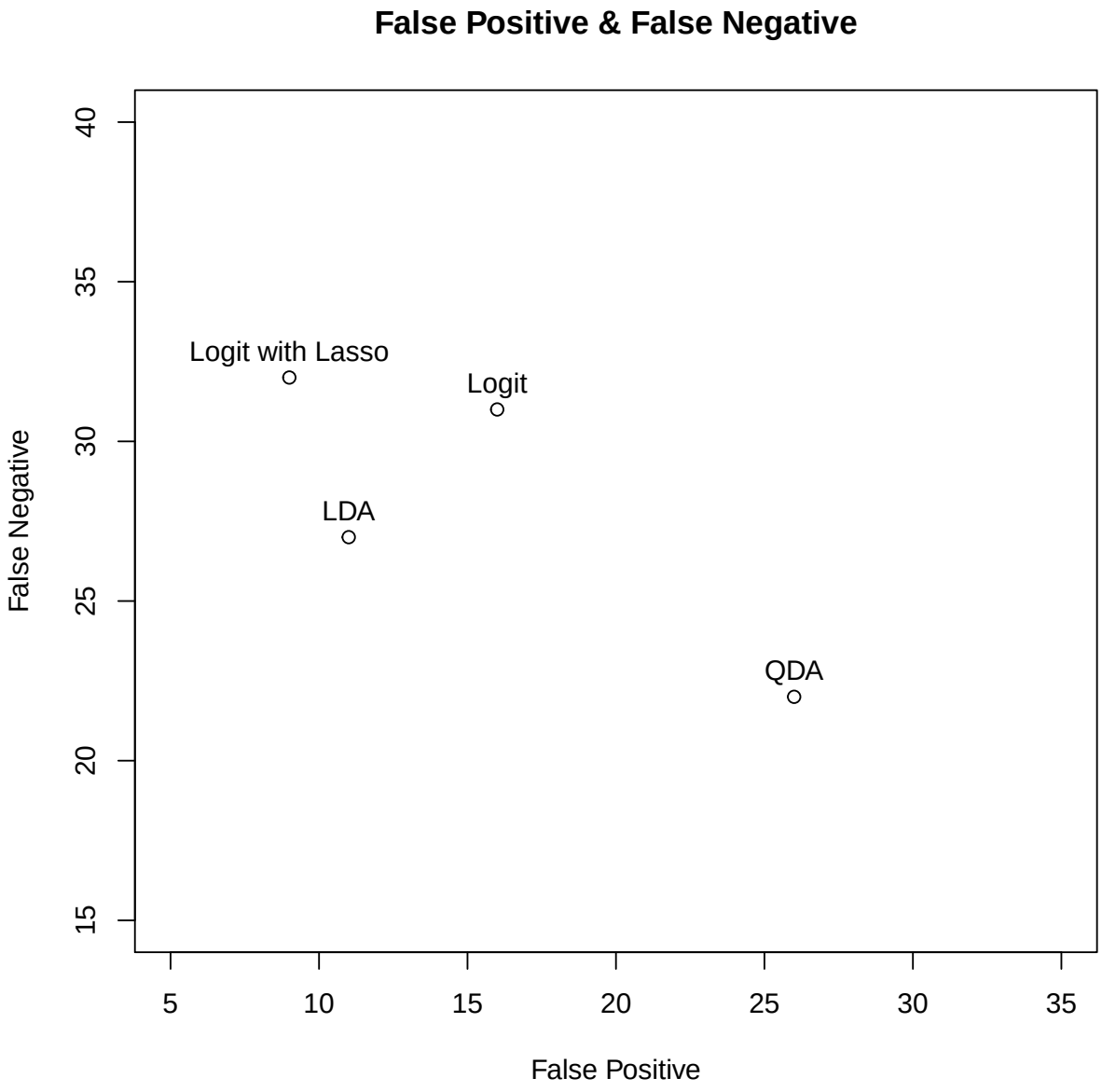
```
## False Positive False Negative
##              9             32
```

```
row.names(Train.error) <- c("LDA", "QDA", "Logit", "Logit with Lasso")
with(plot(Train.error, type = "b", main = "Training and Test Error", ylim = c(0.2, 0.33)),
    text(x = Train.error, labels = c("LDA", "QDA", "Logit", "Lasso"), pos = 3))
lines(Test.error, lty = 2)
legend("bottomright", c("Train Error", "Test Error"), lty = c(1, 2))
```




```
FPFN <- rbind(fpfn(LDA_test, Response_test), fpfn(QDA_test, Response_test), fpfn(logistic_test,
    Response_test), fpfn(lasso_test, Response_test))
row.names(FPFN) <- c("LDA", "QDA", "Logit", "Logit with Lasso")
with(plot(FPFN, main = "False Positive & False Negative", ylim = c(15, 40), xlim = c(5, 35)),
    text(x = FPFN[, 1], y = FPFN[, 2], labels = c("LDA", "QDA", "Logit", "Logit with Lasso"),
```

```
        pos = 3))
```


False Positive & False Negative
False Negative
False Positive
40
35
30
25
20
15
5
10
15
20
25
30
35
Logit with Lasso
Logit
LDA
QDA