Stochastic Project 2

Group 20

February 22, 2020

Reading the data using the R file given:

```
data = read.csv("N100StkPrices.csv", header=TRUE)

# cleaning up data
data = na.omit(data)
ticker = data$TICKER

# spun off MDLZ
delete = seq(1, dim(data)[1])[ticker == "MDLZ"]
data = data[-delete, ]

date = apply(as.matrix(data$date), MARGIN=1, FUN="toString")
date = as.Date(date, "%Y%m%d")
ticker = data$TICKER
price = data$PRC
shares = data$SHROUT

# Accounting for changes in ticker names
# KFT changed to KRFT in Oct 2012.
ticker[ticker == "KFT"] = "KRFT"

# SXCI changed to CTRX in Jul 2012.
ticker[ticker == "SXCI"] = "CTRX"

# HANS changed to MNST in Jan 2012.
ticker[ticker == "HANS"] = "MNST"

# convert prices to a matrix, arranged by rows of dates and columns of
tickers
unique_dates = sort(unique((date)))
unique_tickers = sort(unique(ticker))

priceMat = matrix(NA, length(unique_dates), length(unique_tickers))
sharesMat = matrix(0, length(unique_dates), length(unique_tickers))

for (i in 1:length(unique_tickers)) { # Loop to create price and shares
matrices
  tic = unique_tickers[i]
  idx = is.element(unique_dates, date[ticker==tic])
```

```
  priceMat[idx, i] = price[ticker==tic]
  sharesMat[idx, i] = shares[ticker==tic]
}

rownames(priceMat) = as.character(unique_dates)
rownames(sharesMat) = as.character(unique_dates)

rm(list = c("data", "delete", "i", "idx", "price", "shares", "tic", "ticker",
"date"))

# Read Monthly Data -------------------------------------------------------

# read in the data
mdata = read.csv("N100Monthly.csv", header = TRUE, stringsAsFactors = FALSE)

# clean up data
mdate = apply(as.matrix(mdata$date), MARGIN = 1, FUN = "toString")
mdate = as.Date(mdate, "%Y%m%d")
mticker = mdata$TICKER
mprice = mdata$PRC
mshares = mdata$SHROUT
mticker[mticker == "FOXA"] = "NWSA"


unique_mdates = sort(unique((mdate)))
unique_mtickers = sort(unique(mticker))

idx = is.element(unique_mtickers, unique_tickers)

# if (!all(idx)) {
#   print("Warning: Some tickers seem to be missing")
# }
monthlyPriceMat = matrix(NA, length(unique_mdates), length(unique_tickers))
for (i in 1:length(unique_tickers)) {
  tic = unique_tickers[i]
  idx = is.element(unique_mdates, mdate[mticker == tic])
  monthlyPriceMat[idx, i] = mprice[mticker == tic]
}

rm("mdata", "i", "idx", "mprice", "mshares", "mticker", "tic", "mdate")
```

Question 1

```
#Creating a return matrix to get returns of stocks using stock prices
n = ncol(priceMat)
d = nrow(priceMat)
returnMat = matrix(NA, d, n) #d-1 by n matrix for daily returns
rownames(returnMat) = unique_dates
```

```
colnames(returnMat) = unique_tickers

for (i in 1:n){
  returnMat[2:d,i] = (priceMat[2:d,i] - priceMat[1:d-1,i])/priceMat[1:d-1,i]
}
```

Question 2

Here we have used the "Use" function in Cor to handle the NAs

```
corrMatrix = cor(returnMat, use="pairwise.complete.obs") #Calculate
correlation matrix
```

Question 3

Constraint 1 gives us 1 row, constraint 2 gives us n rows and constraint 3 gives us (n^2) rows. So we have matrix A initialized with 0s for (n^2 +n +1) rows and (n^2 +n) columns representing (n^2) x's and (n) y's.We have foudn the weights, normalized them and used them in the next question to solve for the portfolio shares.

```
library(lpSolve)
constructFund <- function(rho, q, priceMat, sharesMat, unique_tickers,
unique_dates){
  n = length(unique_tickers)
  d = length(unique_dates)

  #Formulating the A, b, c of the integer program to be solved
  c = c(as.vector(rho),rep(0,n))
  A = matrix(0, n^2+n+1, n^2+n)
  A[1,(n^2+1):(n^2+n)] = rep(1,n)
  for (i in 1:n){
    A[(i+1), (n*(i-1)+1):(n*i)] = rep(1,n)
  }
  A[(n+2):(n^2+n+1), 1:n^2] = diag(1,n^2)
  A[(n+2):(n^2+n+1), (n^2+1):(n^2+n)] = matrix(rep(diag(-1,n),n), nrow=n^2,
byrow=T)
  b = c(q, rep(1,n), rep(0,n^2))
  dir = c(rep('=',(n+1)),rep('<=',n^2))
  s <- lp('max', c, A, dir, b, all.bin=TRUE)

  share_last = sharesMat[d,]
  price_last = priceMat[d,]
  market_cap = share_last*price_last

  similar = matrix(0,n,n)
  for (i in 1:n){
    similar[i,] = market_cap[i]*s$solution[(n*(i-1)+1):(n*i)]
```

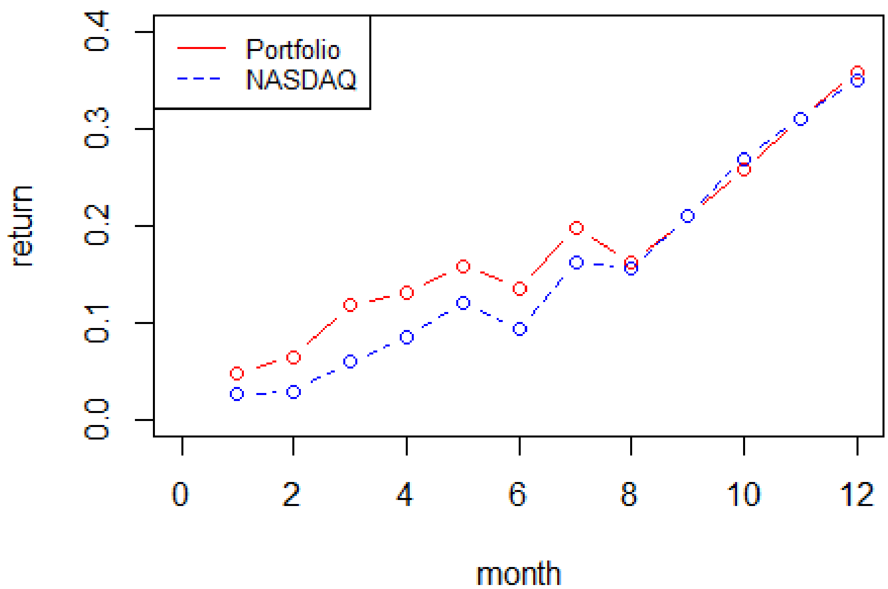
```
  }
  weights = colSums(similar)
  weights_adj = weights/sum(weights)
  return(weights_adj)
}
```

Question 4

Here we have used the weights to choose what goes into our portfolio. We calculated both values and returns for NASDAQ as well as our own portfolio for comparison. The comparison plot has been plotted too.

```
library(ggplot2)
q = 25
investment = 1000000
weights = constructFund(corrMatrix,q,priceMat, sharesMat, unique_tickers,
unique_dates)
weights_dollar = weights*investment

portfolioShare = weights_dollar/priceMat[d,]
nasdaqShares = investment/2660.93

#multiply the number of shares with prices of each month
nasdaqValues = nasdaqShares*c(2731.53, 2738.58, 2818.69, 2887.44, 2981.76,
2909.60, 3090.19, 3073.81, 3218.20, 3377.73, 3487.82, 3592)
#portfolioValues = colSums(apply(monthlyPriceMat, 1, function(x)
x*portfolioShare))

portfolioValues = c(rep(0,12))

for(i in 1:12){
value_ij = 0
  for(j in 1:100){
    value_ij = value_ij + monthlyPriceMat[i,j]*portfolioShare[j]
  }
  portfolioValues[i] = value_ij
  }

#Calculating returns for NASDAQ and our portfolio
nasdaqReturns = (nasdaqValues - investment)/investment
portfolioReturns = (portfolioValues - investment)/investment

Comparison = data.frame(c(1:12), nasdaqReturns, portfolioReturns)
colnames(Comparison) = c('MONTH', 'NASDAQ', 'PORTFOLIO')
View(Comparison)

plot(Comparison$MONTH, Comparison$PORTFOLIO, col = 'red', type = 'b',
     xlab = 'month', ylab = 'return', xlim = c(0,12), ylim = c(0, 0.4))
```

```
lines(Comparison$MONTH, Comparison$NASDAQ, col = 'blue', type = 'b', lty=2)
legend("topleft",legend = c('Portfolio','NASDAQ'), col = c('red','blue'), lty
= 1:2, cex = 0.8)
```



From October onwards, both NASDAQ and our portfolio show negligible differences. Before October, Portfolio returns are higher than the NASDAQ values.

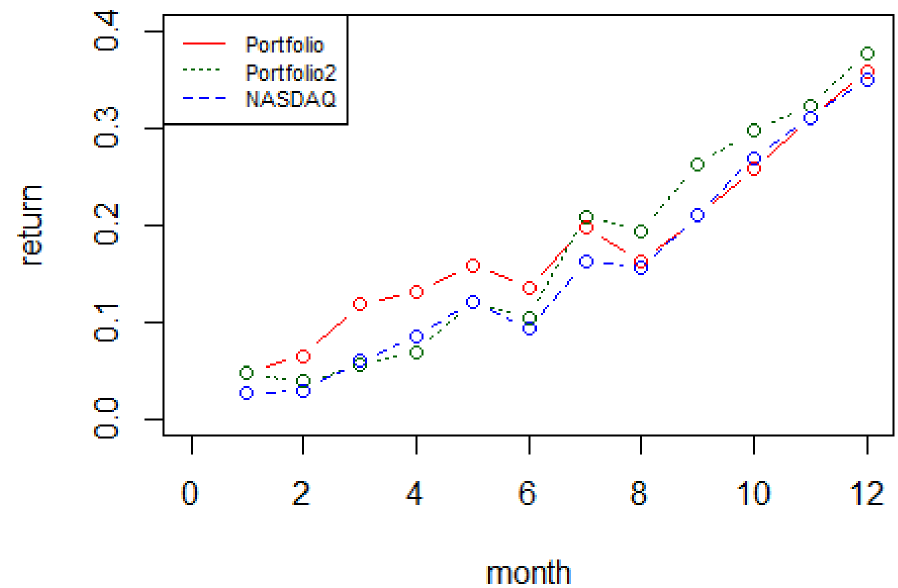
Question 5

We chose the market cap correlation matrix as our similarity measure here. Earlier, with stock returns as the correlation matrix, we would replace one stock with another that have similar returns. With market capitalization correlation, we would be able to replace one company with another based on their size. We believe that size is an important factor in stock returns, as it is a factor discussed in the Fama French 3 factor model.

We now have a new portfolio to compare with NASDAQ. The similarity matrix has also been defined below.

```
similarityMat <- function(priceMat, sharesMat, unique_tickers,unique_dates){
  market_cap = priceMat*sharesMat
  corrMatrix = cor(market_cap, use = "pairwise.complete.obs")
  return(corrMatrix)
}
rho = similarityMat(priceMat,sharesMat,unique_tickers, unique_dates)
```

```
weights2 = constructFund(rho,q,priceMat, sharesMat, unique_tickers,
unique_dates)
weights_dollar2 = weights2*investment
portfolioShare2 = weights_dollar2/priceMat[d,]
#portfolioValues2 = colSums(apply(monthlyPriceMat, 1, function(x)
x*portfolioShare2))
portfolioValues2 = c(rep(0,12))

for(i in 1:12){
value_ij = 0
  for(j in 1:100){
    value_ij = value_ij + monthlyPriceMat[i,j]*portfolioShare2[j]
  }
  portfolioValues2[i] = value_ij
}

portfolioReturns2 = (portfolioValues2 - investment)/investment
Comparison$PORTFOLIO2 = portfolioReturns2
View(Comparison)

plot(Comparison$MONTH, Comparison$PORTFOLIO, col = 'red', type = 'b',
     xlab = 'month', ylab = 'return', xlim = c(0,12), ylim = c(0, 0.4))
lines(Comparison$MONTH, Comparison$PORTFOLIO2, col = 'dark green', type =
'b', lty=3)
lines(Comparison$MONTH, Comparison$NASDAQ, col = 'blue', type = 'b', lty=2)

legend("topleft",legend = c('Portfolio','Portfolio2','NASDAQ'), col =
c('red','dark green','blue'), lty = c(1,3,2), cex = 0.7)
```



We see that the new portfolio performs better than the old after 7th month (July). However, both these portfolios track the NASDAQ quite closely.