Project 2

Amanda Coker, Akshata Mohan, Caroline Nelson, Hillary Regan

2/22/2017

Attached are the following R files:
a. similarityMat.R
b. constructFund.R

Question 1

Calculate the daily returns for each stock using the 2012 price data.

```
#Q1
source('readData.R')
returns = matrix(0,nrow=250,ncol=100)

for (j in 1:100){
  stock = as.vector(priceMat[,j])
  for (i in 2:250)
  {
    returns[i,j] = (stock[i] - stock[i-1])/stock[i-1]
  }
}
```

Question 2

As our initial candidate for the similarity matrix, find the correlation matrix for the returns of the 100 stocks. Note that there will be missing data in the price matrix (NA which stands for Not Available). You need to specify 'use' argument in the 'cor' function in order to handle NAs.

```
#Q2
cor_matrix=cor(returns, use="pairwise.complete.obs")
```

Question 3

Code the integer program above as another function that returns the weights for each of the stock that needs to be in your portfolio

weights = constructFund(rho, q, priceMat, sharesMat, unique_tickers, unique_dates)

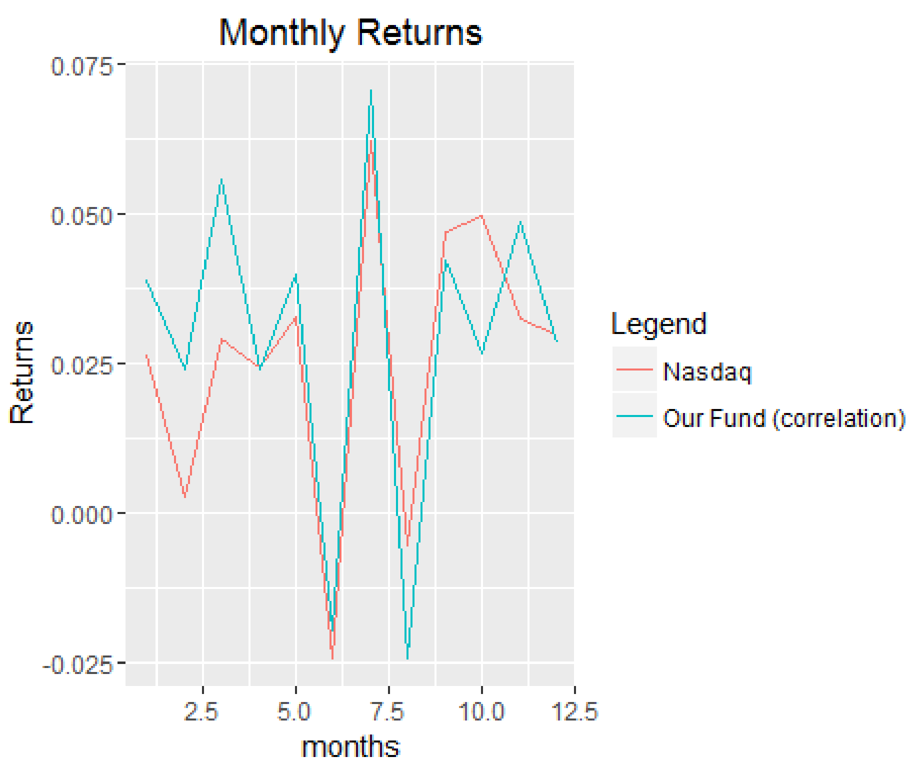
This will amount to simply formulating the integer program, solving it and then using the market capitalization of each company on the last date to compute weights. The output weights will be a vector of size n with only q non-zero elements denoting the weights.

```
#Q3
q = 25
rho <- cor_matrix
source('constructFund.R')
```

Question 4

Use your weights to construct an index portfolio at the end of 2012.

Compare how this index portfolio performs monthly in 2013 as compared to the NASDAQ 100 index using the 2013 stock data provided. Here you may assume that you can directly invest in the Index as if it is a stock. Present your findings using any visualizations or tabulations. You can assume that you will be investing 1 million in your fund. In this case, your shares of each stock you choose to construct the portfolio should be large. As a result, you can leave the shares as non-integers, because the effect that the non-integer parts of shares have should be marginal.



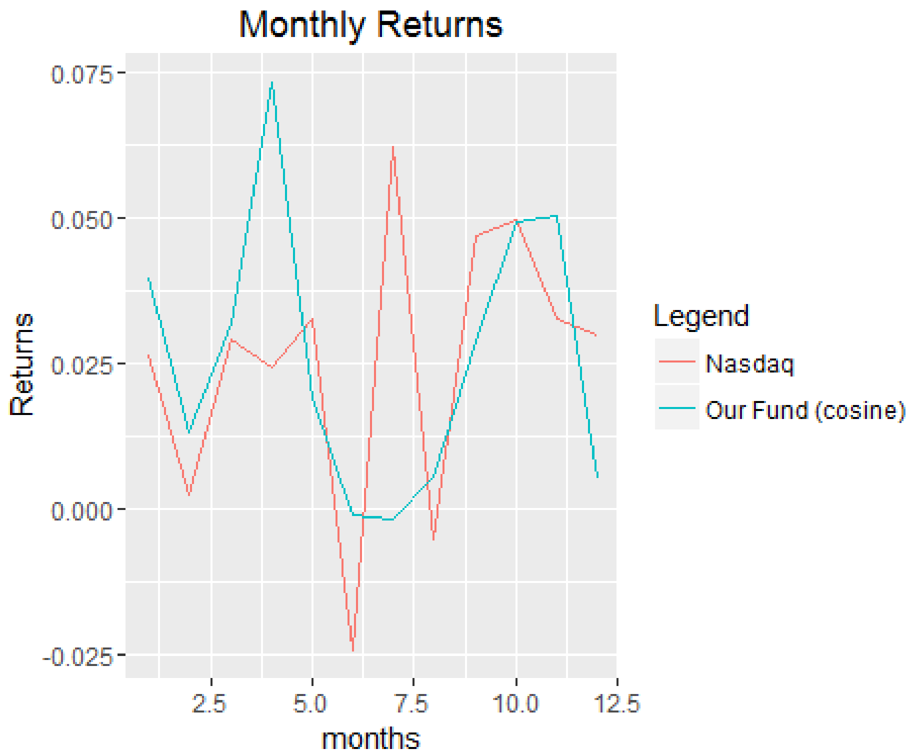
Seen in the graph above, our fund generally mirrors the Nasdaq fairly well. It does however, overestimate the returns for the first seven months or so, then underestimate the returns thereafter. Our fund also exaggerates the shift; where the Nasdaq rises, our fund tends to rise higher.

Question 5

Earlier you used correlation as the similarity measure. Now instead create your own similarity measure and put it in a function similarityMat that has the same inputs and outputs

rho = similarityMat(priceMat, sharesMat, unique_tickers,unique_dates)

Use this rho in your function call to constructFund and as in step 4, evaluate the performance of this fund as well. Please compare the new fund to the previous fund. Try to explain why the performance of the new fund is better (or worse).



We used cosine similarity as the similarity measure to calculate the weights of each stock in this new fund. Cosine similarity measures the cosine of the angle between two vectors, which measures orientation rather than magnitude. Cosine similarity describes how related two vectors are. If two vectors are very similar, then the cosine will be near 1. If the two vectors are unrelated, then the cosine will be near 0. Lastly, if the two vectors are related, but in the opposite direction, then the cosine will be near -1.

This new fund over exaggerates the returns and has a significant delay when the markets turn. Thus, when Nasdaq goes from an up-swing to a downturn, our fund continues to rise then decreases dramatically.

As seen in the comparison of the funds in the two graphs above, the fund created with the correlation matrix performed better than the fund using cosine similarity as the similarity measure. The first fund performed better because using correlation as a similarity measure does not change with scale or shifts in the variables, whereas cosine similarity is invariant to scale, but is subject to changes in the variables.