Stock Trend Evolution

9 April 2020
Agarwal, Taylor; Quelle, Henk; Ryan, Cooper
Mentor: Lanius, Melinda.

Contents

1 Abstract

Our analysis uncovers the mysterious behavior of the United States stock market, that to the untrained eye can seem completely random, but in fact is dictated by surprisingly correlated stock prices. Using the method of Principal Component Analysis (PCA), we grouped stocks that have similar day-to-day and long term structure. Our results and data heavily support the PCA's implementation when observing the stock market. Our work is critical to the world of investing and more specifically to those who wish to see short and long term financial success. Both professionals in the arena of finance and long term amateur investors will find our analysis enlightening and more importantly applicable to their everyday financial life.

2 Introduction

For hundreds of years people have been trying to find new ways to analyze the US stock market to allow for easy access to riches. The market is filled with thousands of publicly traded stocks. Some of the stocks' movements have hidden connections to other stocks that are not easily identified. Some of the stocks have strong effects over others, some dictate markets as a whole, and some are completely unrelated with one another. We tested and discovered new ways to better understand trends in the US Stock Market using a recent machine learning technique called Principal Component Analysis. During the course of this project we aim to identify groups of stocks with similar variance, by looking at their dependence upon certain "eigenstocks". These eigenstocks are inextricably bound to the stocks themselves, as each eigenstock can be interpreted as a component of the stocks in the data set. Determining how much influence a particular eigenstock has on a stock will allow us to group the stock with others of similar influence, thereby isolating groups that rise and fall at similar times. Therefore, by looking at a single stock a group of stocks, we can determine the motion of the group as a whole, thus giving the user actionable information for maximizing profits on these stocks.

3 Model

3.1 The Data

We begin by investigating individual stocks. Stocks are publicly traded shares of ownership in a particular company, and each day the prices of these shares fluctuate according to supply and demand. Each day, the open price is the value of the stock when the market opens for the day, the high is the maximum price it reaches that day, the low is the minimum price it reaches that day, the close the the value at the close of the day, and the volume is the number of shares that are traded during the day. We located a dataset containing each of these values for 7195 stocks for each day between 1960 and 2017 (or the extent of its

availability on the market) on Kaggle [3]. This gives us millions of data points to work with, which would require a large amount of computing power for us to perform any kind of analysis on it. So a cleaning process had to be undergone before we could begin any analyses.

We explored several methods for cleaning the data. Initially, we extrapolated closing values for all 7195 stocks for the entire duration of each, but we found that the vast majority of stocks were not publicly traded for the entirety of the 1960 to 2017 time period, so we decided to reduce our timeframe to the more contemporary time period between 2000 and 2017. Additionally, we noticed that a large portion of the stocks traded at below $5, and experienced very little fluctuation during the day. We decided to disregard these stocks for the sake of saving on computation time while maintaining the variance in the data, whose importance will be discussed later in this section. We selected 100 stocks that are representative of the market for the majority of our analyses. A list of these stocks can be seen in Appendix A.

Normalizing the data is an important step when dealing with stocks. Some stocks are traded at very high values that fluctuate frequently, while others are more steady but lower valued. These stocks needed to be normalized in a meaningful way that preserved the structure of the fluctuations in the stocks while not allowing larger or smaller valued stocks to over-influence the analyses performed here. We decided to use a percent change normalization

$$Percent\ Change = \frac{Close - Open}{Open} \cdot 100 \tag{1}$$

for each day in the dataset. This normalizes the stocks to be independent of their individual prices and focuses in on their daily fluctuations. It also introduces negative values into our data, allowing for easy identification of days when the stock price dropped. Our first idea was to use $\log(Close)$ as our primary normalization method, but we noticed that while it produced useful results when looking at the market as a whole, it did not properly capture the fluctuations of individual stocks quite as well as we would have liked.

Summary statistics of our new dataset are shown below, all values are expressed as percentages.

Min	Median	Mean	Max
-89.9301	0.0262	0.0338	102.7012

3.2 Principal Component Analysis

Principal Component Analysis, or PCA, is a linear algebra technique that reduces the dimensionality of a matrix through statistical and topological reduction methods. PCA analyzes the relationships between each variable of a matrix and evaluates its importance to the matrix itself. In this way, certain dimensions can be disregarded and the important variables come to light. PCA is very valuable to data science, as it allows data scientists to quickly evaluate multidimensional problems with relative ease.

We construct a matrix $A \in \mathbb{R}^{m \times n}$ from our normalized stock data. Each column is a particular stock $\vec{x_i}$ for $i \in [1, n]$ of length m, where i is the stock index, n is the number of stocks in the sample, and m is the number of days in the timeframe. As previously mentioned, the majority of our analyses utilized a 100 stock subset with a timeframe of 4/17/2000 to 11/10/2017. So for this dataset, $n = 100$ and $m = 4420$. Hence our *data matrix* is

$$A = \begin{pmatrix} \vec{x}_1 & \vec{x}_2 & ... & \vec{x}_n \end{pmatrix} \tag{2}$$

For the duration of this paper, x_{ij} will reference the j-th date of the i-th stock. The stocks are ordered alphabetically, so that $\vec{x}_1$ corresponds to the first stock alphabetically.
We now aim to perform PCA on our data matrix. PCA consists of 5 steps:

1. Shift the data to the origin
2. Compute the Covariance Matrix C
3. Compute the eigenvalues and eigenvectors of C
4. Select a small number of eigenvectors that are representative of the data
5. Compute the weight vectors necessary to approximate the original data using only the subset of eigenvectors

These steps and the information held within them come as a result of PCA methods explained by Hargreaves and Mani 2015 [2], Zhang and Turk 2008 [4], and Strang 2019 [6].

3.3 Shift the Data to the Origin

The first step in performing PCA on the data matrix is to find the mean vector

$$\vec{\mu} = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \tag{3}$$

Subtract $\vec{\mu}$ from each $\vec{x}_i$ in A. Then

$$\hat{A} := \begin{pmatrix} \vec{x}_1 - \vec{\mu} & \vec{x}_2 - \vec{\mu} & ... & \vec{x}_n - \vec{\mu} \end{pmatrix} := \begin{pmatrix} \Phi_1 & \Phi_2 & ... & \Phi_n \end{pmatrix} \tag{4}$$

is the new matrix with shifted columns. This shift centers the data around the origin of $\mathbb{R}^m$. This step reveals the numerical stability (extremeness of outliers) and resolution of the data, exposing variance and clear differences between values.

3.4 Compute the Covariance Matrix C

Compute the *covariance matrix* C.

$$C = \begin{bmatrix} Cov(\vec{\Upsilon}_1, \vec{\Upsilon}_1) & Cov(\vec{\Upsilon}_1, \vec{\Upsilon}_2) & ... & Cov(\vec{\Upsilon}_1, \vec{\Upsilon}_m) \\ Cov(\vec{\Upsilon}_2, \vec{\Upsilon}_1) & Cov(\vec{\Upsilon}_2, \vec{\Upsilon}_2) & ... & Cov(\vec{\Upsilon}_2, \vec{\Upsilon}_m) \\ ... & ... & ... & ... \\ Cov(\vec{\Upsilon}_m, \vec{\Upsilon}_1) & Cov(\vec{\Upsilon}_m, \vec{\Upsilon}_2) & ... & Cov(\vec{\Upsilon}_m, \vec{\Upsilon}_m) \end{bmatrix} \tag{5}$$

Where $\vec{\Upsilon} \in \mathbb{R}^{1\times n}$ is the shifted data for all stocks on a particular date,

$$\vec{\Upsilon}_j = \vec{y}_j - \mu_j$$

and

$$Cov(\vec{\Upsilon}_a, \vec{\Upsilon}_b) = \frac{1}{n-1}(\vec{\Upsilon}_a \vec{\Upsilon}_b^T) = \frac{1}{n-1}\sum_{i=1}^{n}(y_{ai} - \mu_a)(y_{bi} - \mu_b)$$

This matrix is called the covariance matrix since each entry C_{ab} is the covariance of two dates $\vec{y}_a$ and $\vec{y}_b$. Each diagonal entry C_{aa} is the variance of the date $\vec{y}_a$, while each off diagonal entry C_{ab}, $a \neq b$ is the covariance of date $\vec{y}_a$ and date $\vec{y}_b$. The covariance matrix shows the relationship across all stocks between any two dates; amongst other things, the covariance can tell you what happens to one variable when you adjust another. A positive covariance between dates tells you that the dates experience similar increases and decreases across most stocks. Oppositely, a negative covariance implies that stocks on those dates perform oppositely, so if a stock rises on one date it can be expected to fall on the other. This is an important step in finding the dimensions that can be reduced, since you can determine how related stocks grow and shrink based on a select few variables.

3.5 Compute the Eigenvalues and Eigenvectors of C

To reduce the dimensionality of the covariance matrix, first we find its eigenvalues and eigenvectors. We will call these values λ_j and these column eigenvectors $\vec{v_j}$ for $j = 1, ..., n$. Normalize these eigenvectors such that $||\vec{v_j}|| = 1$. Note that each eigenvector $\vec{v_j} \in \mathbb{R}^{m\times 1}$. This means that each eigenvector has the same size as our original stocks, and can therefore be perceived as an *eigenstock*. Additionally note that the first r eigenvalues are positive. Order these eigenvalues such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$ where r is the rank of C. We expect that $r \leq n$ since there will likely be some linearly dependent columns in C for an interacting system of many variables.

The eigenstock associated with λ_1, the largest eigenvalue, accounts for the most variance in the data, so it would hold the most information about the motion of the data than any other eigenstock. Each successive eigenstock points in a direction orthogonal to each previous eigenstock (i.e. in a unique direction) and accounts for less variance in the data than each previous eigenstock. Therefore the more eigenstocks that are held on to will allow for better interpretation

of the variance in the data, but the point of PCA is to use as few eigenvectors as possible to interpret as much data as possible so that the size of the data is reduced. Figure 1 shows the variance accounted for by the first 68 eigenstocks, which clearly shows that the first eigenstock accounts for the most variance in the data, and the accountance decreases with each successive eigenstock. Once

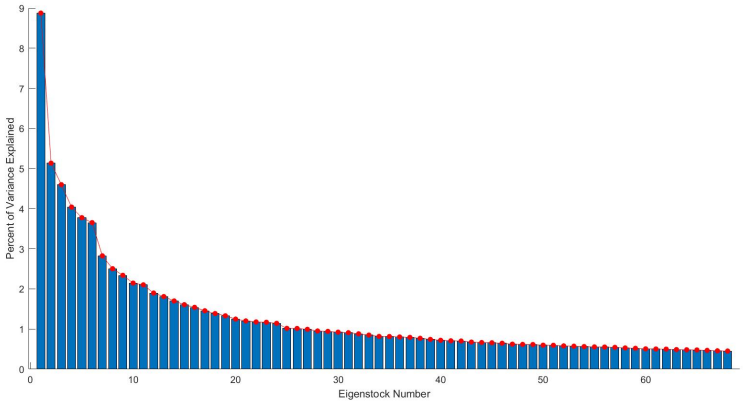

Figure 1: Sample Scree Plot of Percent of Variance Accounted for by the First 68 Eigenstocks

the first k eigenvectors are selected ($k < n$), we placed them as columns in a matrix V, ordered based on the size of their eigenvalues from greatest to least.

$$V = \begin{pmatrix} \vec{v_1} & \vec{v_2} & ... & \vec{v_k} \end{pmatrix} \tag{6}$$

The next section will discuss how we selected k.

3.6 Select a Small Number of Eigenvectors that are Representative of the Data

A common method of determining how to reduce the set of eigenstocks is to select the first k eigenstocks that account for a specified percentage of the variance. The percentage of variance accounted for by eigenstock $\vec{v}_j$ can be determined by λ_j, where

$$Percent\ Explained = \frac{\lambda_j}{\sum_{i=1}^{n} \lambda_i} \times 100\% \tag{7}$$

It can be seen that if all eigenstocks are held onto, the variance is completely accounted for and 100% of the variance is explained by the n eigenstocks. However, this would not reduce the size of the data at all, and one of the key reasons to use PCA would go unutilized. So we instead select $k < n$ eigenstocks based on the variance each accounts for. Figure 2 shows the cumulative percent of

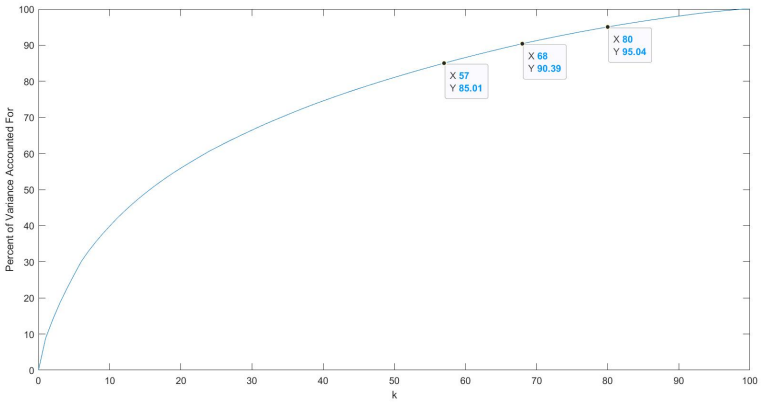

Figure 2: The percent of variance accounted for by the first k eigenstocks

variance accounted for by including each successive eigenstock in the analysis. Notice that the accounted variance exceeds 85% once 57 eigenstocks are held onto, and 90% once 68 eigenstocks are held onto.

3.7 Compute the Weight Vectors Necessary to Approximate the Original Data

We use the equation

$$W = V^T \hat{A} \tag{8}$$

to find the projection of T onto the space spanned by V. Each column of W, $\vec{\Omega}_i = (w_1, ..., w_k)^T$ for $i = 1, ..., n$, is the vector of weights satisfying $V\vec{\Omega}_i \approx \Phi_i$. In terms of the original data,

$$\vec{x}_i \approx V\vec{\Omega}_i + \vec{\mu} = w_1 v_1 + ... + w_k v_k + \vec{\mu} \tag{9}$$

Then, an approximate recreation of the data matrix can be accomplished by calculating

$$A \approx VW \oplus \vec{\mu} \tag{10}$$

where $\oplus$ indicates the addition of $\vec{\mu}$ across the columns of VW. Figure 3 shows a visual representation of the approximated matrix. The maximum absolute error for any one element of the data matrix is 10.33%, and the median absolute error of all elements is 0.1677%.

3.8 Forming Portfolios

The *weight vector* Ω_i can tell us a great deal about a particular stock. If the magnitude of w_j is very large, then you can conclude that the variance in $\vec{x}_i$ is strongly accounted for by eigenstock $\vec{v}_j$. In fact, the maximum weight (in

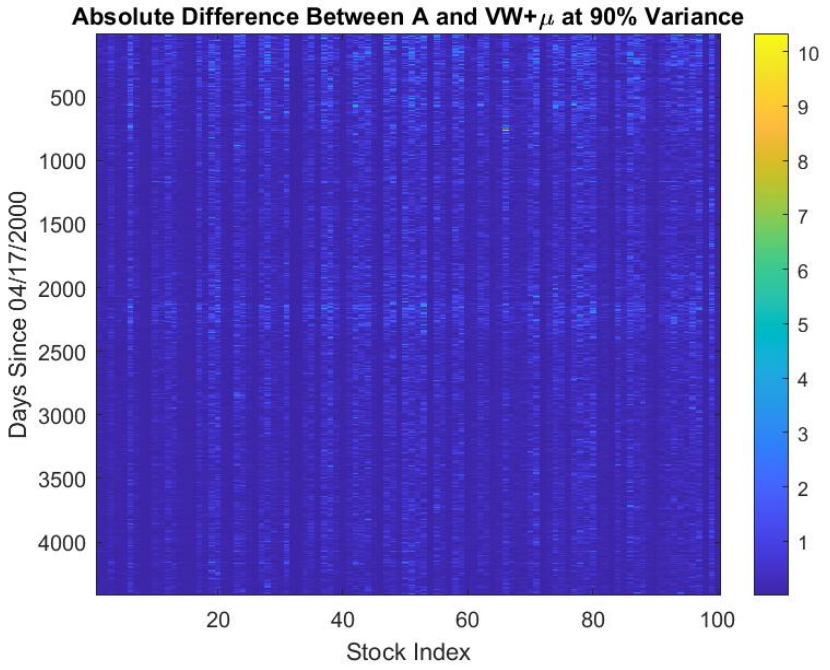

Figure 3: Error from Eigenstock Approximation

absolute value) in Ω_i reveals which eigenstock holds the most information about the stock. Using this knowledge, the eigenstock that stock $\vec{x}_i$ has the strongest association is

$$\vec{v}_j, \text{ where } \{j \mid |w_j| \geq |w_i|, \forall w_i \in \Omega_i\} \tag{11}$$

From this, portfolios can be formed. A stock is said to be a part of portfolio j if it is strongest association is with eigenstock $\vec{v}_j$. Stocks within the same portfolio can therefore be said to have similar structure to the eigenstock, and therefore similar structure to one another.

3.9 Machine Learning

The portfolio of a stock $\vec{X}$ not already in the data matrix can be identified by finding $\vec{\Omega} = V^T(\vec{X} - \vec{\mu})$ for V and μ as previously defined. Then consider it a part of the same portfolio as $\vec{\Omega}_i$ where

$$\vec{\Omega}_i = \arg\min_{\vec{\Omega}_i \in W} ||\vec{\Omega} - \vec{\Omega}_i||_2 \tag{12}$$

4 Results

We expected that the structure of the eigenstocks upon which the portfolios are formed would hold true in both the long term and short term, and we found this to be true with some certainty. Following the procedure outlined in Section 3, we used the 100 stock dataset to form portfolios of stocks that are expected to have similar long term and short term behavior. We found that the percent of days in the timeframe any pair of non-identical stocks rose and fell together is between 49.62-74.71%. On average, each pair of stocks in the portfolios rose and fell together on the same days 58.77% of the time. This is an improvement over picking stocks at random. Narrowing the timeframe to 7 years gave an even better average of 61.29% for the same measurement. This demonstrates that our analysis can "beat the market" on a day to day basis. The linear correlation between any pair of non-identical stocks ranged between -0.0497 and 0.7499. Each pair of stocks in the portfolios had, on average, a linear correlation of 0.2347, which is a moderate improvement over selecting at random. The average correlation can be improved by narrowing the time frame; reducing the timeframe to 7 years produced portfolios with an average linear correlation of 0.3451. This shows that the analysis is also applicable to long term investment, since correlation is a good indicator of long term behavior. Next, we proceed to implementing it in MATLAB. The script can be seen in Appendices B and C.

Our final result is the development of a tool that can be used to analyze the stock market. This tool successfully forms groupings of stocks based on any subsection of the market, while following user-defined time frames and variance thresholds. A few tests were done to verify the robustness of the product including; using the principle eigenstock to recreate a model of the Dow Jones, determining if the grouping held through seemingly isolated, singular company drops in stock value, and viewing the accuracy of the groupings through economic downfalls such as the market crash caused by the 2001 9/11 attacks and the 2008 housing market crash. After we conducted these tests and we determined the resilience of the tool, the outputs, as seen below, produced groupings of stocks of companies that are both related and ones that were seemingly unrelated having rises and falls on the same day, and therefore trending together.

4.1 Tests

Figure 4 shows the first test run to determine if the method of using PCA to create eigenstocks was still accurately modeling the market. To do this, the eigenstock that covered the most variance in the data, $\vec{v}_1$ was graphed and compared to the movements of the Dow Jones. The Dow Jones was chosen to compare the model to because it is used to model the movements of the entire market on a day to day basis. Since the movements of our eigenstock that covers the most variance and the movement of the Dow Jones are so similar it showed that the method of using eigenstocks was still accurately modelling the market. Another example of two of the tests run can be seen in Figure

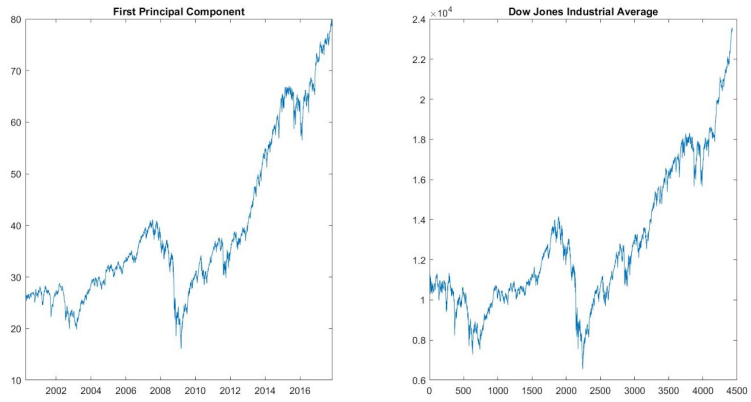

Figure 4: First Principal Component and Dow Jones

5. This is a graph of portfolio 1 taken from the analysis of 100 Fortune 500 depicting the time frame from mid 2007 to late 2012. The large drop in the Aflac stock price was an isolated incident that did not directly affect any of the other stocks involved in this graph. It can be seen here that even during the 2008 recession and through an individual stock's isolated price drop, the group still trends together with rises and falls matching one another.

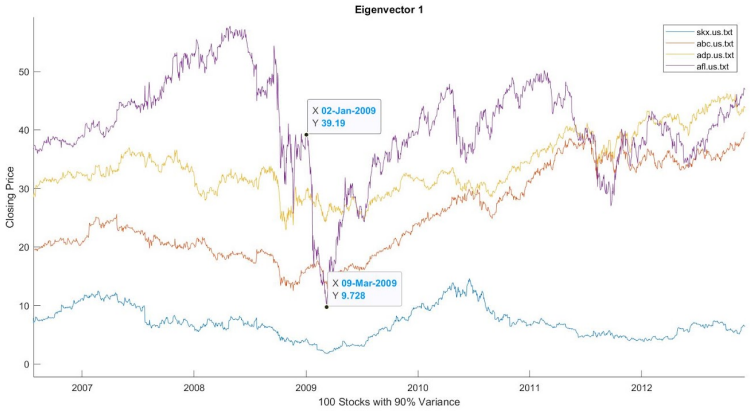

Figure 5: Graph During the 2008 Recession and Aflac Price drop

4.2 Related Stock Groupings

As stated above, the groupings that we produced were sometimes all stocks that one might assume to trend together. For example, Figure 6 depicts 3 of the

largest health care related stocks in the country; Anthem (ANTM), McKesson Corporation (MCK), and UnitedHealth Group (UNH). Figure 7 represents the

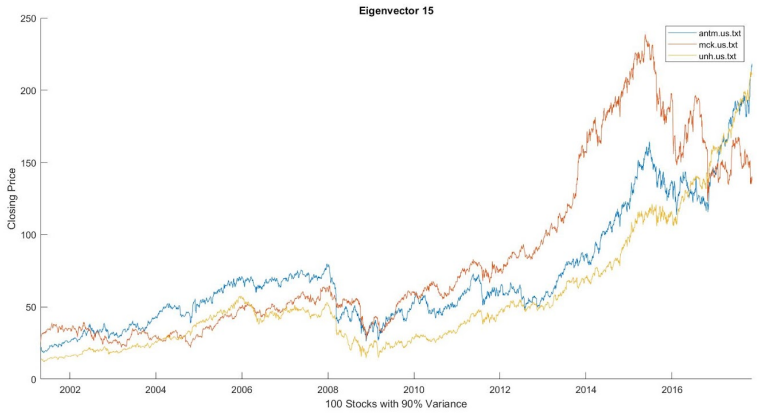

Figure 6: Grouping of Anthem (ANTM), McKesson Corporation (MCK), and UnitedHealth Group (UNH)

stocks that trended with British Petroleum, before, after, and during its price drop after the oil spill in the Gulf of Mexico. These companies would also all

Figure 7: Grouping of British Petroleum (BP) are General Dynamics (GD), Lockheed (LMT), Northrop (NOC), and Xerox (XRX)

be expected to trend together due to their need for oil based products and their production of oil based products. However, even though the data and groupings found of related companies trending together is important, the more desired output of this tool is seen in the subsection below that discusses companies that one would not naturally expect to trend together.

4.3 Unexpected Stock Groupings

There were tens of examples of seemingly unrelated stocks trending together from our analysis of a mere 100 stocks. Figures 8, 9, and 10 are just a few examples of unexpected stock groupings. Figure 8 shows a grouping of stocks trending together during the time frame of the announcement of the Iraq invasion in 2003. Raytheon was included in the group by means of machine learning,

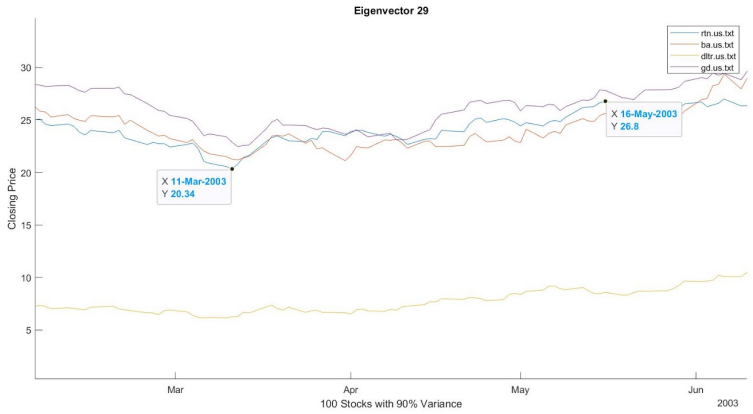

Figure 8: Grouping of Raytheon (RTN), Boeing (BA), General Dynamics (GD), and Dollar Tree (DLTR)

as discussed in Section 3.9. It was expected to see major defense contractors trending together, however it was unexpected to see Dollar Tree trending along side them. Figure 9 depicts a grouping of three stocks. It can be argued here that all three stocks are quite unrelated due to the fact that Best Buy sells electronic products, Dillards sells clothing, and Goodyear is a tire manufacturer and seller. Figure 10 shows three stocks, two of which are technology companies and one a home improvement store. Now, it is not understood what relationships there actually are inside these corporations that cause them to be grouped together, because that is not the purpose of this research. But, it is important that the companies inside these groupings are all have similar upward and downward stock movements both daily and long term.

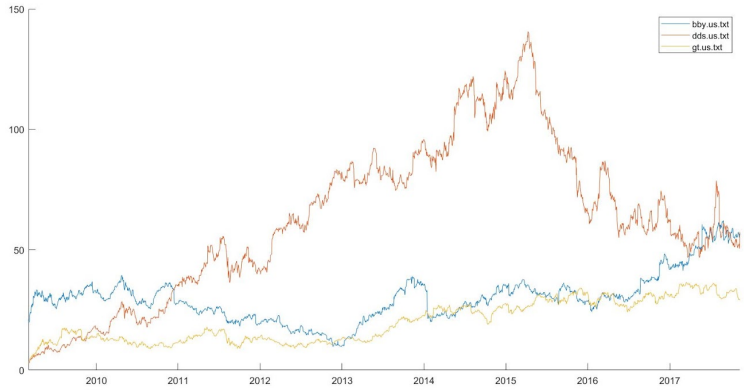

Figure 9: Grouping of Best Buy (BBY), Dillards (DDS), and Goodyear Tires (GT)

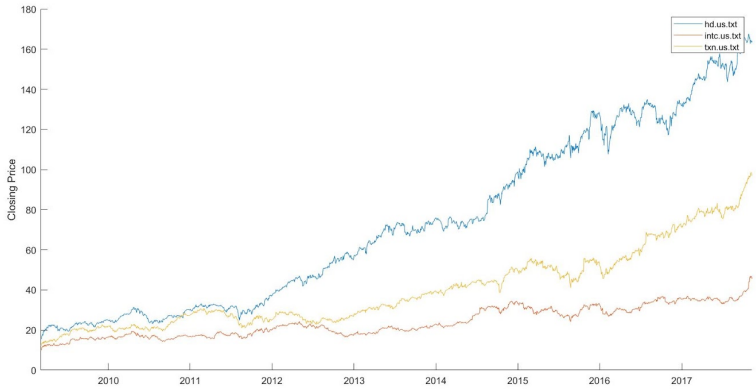

Figure 10: Grouping of Intel (INTC), Texas Instruments (TXN), and Home Depot (HD)

4.4 Developed Application

The final product of the research and modeling done in this paper was a user friendly application that is run through MATLAB. The first step to analyze a data set using the program is to choose a file that needs to be analyzed. This is the bar at the top center in both Figures 11 and 12. Once this is done, there are many different parameters that can be adjusted to meet the user's needs for analyzing stocks. On the left are the options of adjusting the time frame from start date to end date. A user would use these options to choose how far back into the data they would like to look to see relationships between the stocks

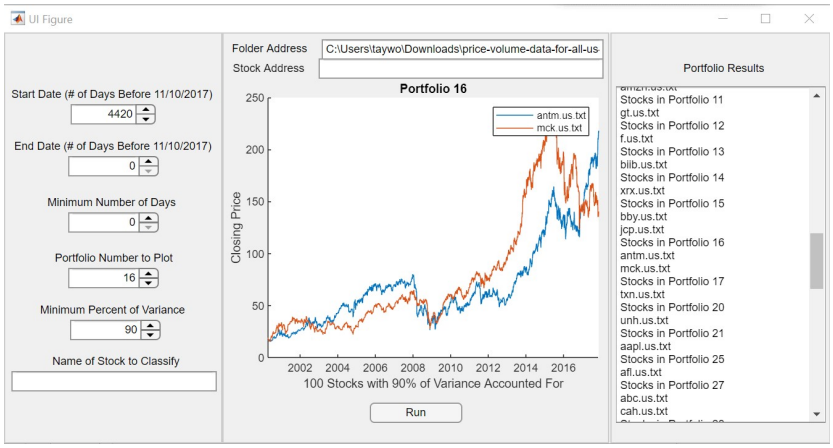

Figure 11: Application Analysis of 100 Fortune 500 Stocks

being analyzed. Below these options is a place to select the minimum number of days that they would like each stock to have been public for. When the program is run this action is completed first to sort the data set into stocks that have been public long enough to meet the previously specified time frame. The user can then specify the percent variance they would like to use to analyze the data. After this is over the user will then click "Run". After running the application the column of differently numbered portfolios will appear on the right. This shows all the different groupings of stocks that have been produced by the data. Then, as seen in Figure 11 the user can pick which portfolio they would like to

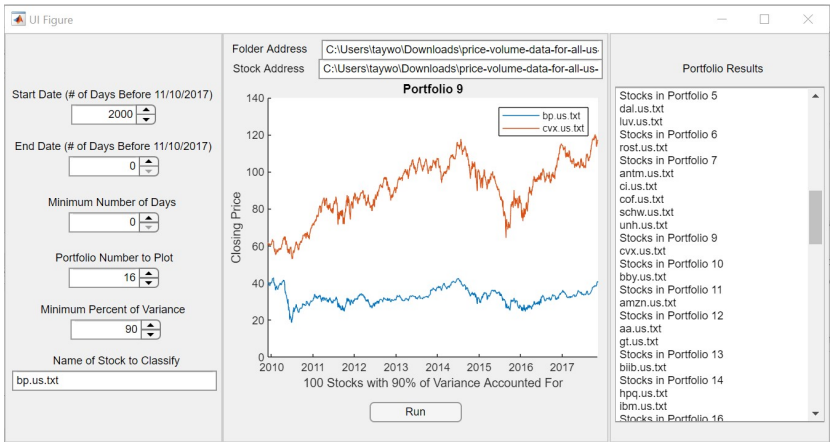

Figure 12: Application Analysis of Outside Stock With initial 100

graph and they will see it appear in the center after clicking run again. If the user would like to analyze a stock that is outside the previously analyzed data

set and see where it is grouped with the already analyzed set (as in Section 3.9) they can do that as well. This is shown in Figure 12. In this example, the user wants to analyze where British Petroleum is grouped in the already input set of stocks so they add the BP file in the second bar at the top center of the screen. The user can also search for a specific stock to see which portfolio it is grouped in and what stocks it trends with, this can be done, as seen in Figure 12 with BP, by putting in the name of the file containing the desired stock to be viewed. The program only ever graphs a maximum of 5 stocks no matter how many are in any given portfolio to keep the graphs more organized. When there are more than 5 stocks in any given portfolio, the program selects which stocks to graph based on their alphabetical order.

5 Applications of the Model

5.1 Application Concept

The application behind Principal Component Analysis (PCA) and the linear algebraic modeling of the stock market is in short, a different perspective of how to view overall trends and variance in the market. The idea of an "eigenstock," post PCA, is not that it is necessarily representative of a real stock that someone would go to the market to acquire. Rather, an eigenstock can be observed as a close approximation that should reveal similar behavior(Shahnawaz and Ghazanfar 2017). The groupings of these stocks based off of their highest weights of the eigenstock in linear combination with other eigenstocks, will in theory give us stocks that vary in nearly the exact same manner. So, this concept of grouping eigenstocks based on their highest eigenstock weight, is far more useful than say, using this to identify one or a few stocks that dominate the market. Taking specific groups in association with one-another will allow for an individual to look at the stocks that are related based on its grouping in a more magnified way. Looking at stocks in groups is useful if for example, there is a barrier to enter the grouping based on the highest weight of its eigenstocks in linear combination. This should help in some cases, to determine a dominant stock in a grouping, meaning that the common eigenstock weight far exceeds the barrier of entry. This will allow us to key in on this stock to determine the overall direction of the portfolio. This is applicable because usually the market investor would like to play it safe and diversify one's portfolio of stock investments, and an individual having the knowledge of which stocks are in association with each-other is a good method to observe when selecting a series of stocks that won't necessarily be heavily dependent on one's entire investment sum.

5.2 Observed Utilities

Our research allows for two primary methods of investment on two different ends of the investing spectrum. Investing spectrum is a different way to describe the duration and type of method applied when investing. The two methods that seem to be relevant in relation to this model are very small windows of time such as a short and long term investing. The principal components allow the different stocks in the model to be grouped by the variance a particular grouping of stocks may experience, as mentioned before. This is clearly advantageous in the world of investing for the long term, because diversifying one's portfolio with stocks that rely on dissimilar principal components will help hedge any sort of significant financial loss. This in essence means that investors will need to distribute their financial capital into stocks that are associated with different eigenstocks. The short term investment strategy is slightly more volatile because the investments of a day trader can change at a moments notice, so a keen focus on the desired stocks of interest is required. Nonetheless the process of PCA is still applicable and more advantageous than other forms of reasoning, especially to an aspiring stock trader who desires to short the market. Usually the shorting of a stock comes from a tip or the identification of a feature in a company that may cause it's short term valuation to decrease. Before PCA, this practice was only applicable to the specific stock that was in question. With PCA and observing the results of past declines, an average of when the other stocks of the same eigenstock will begin to fall can be established. So rather than being limited to the short term gains of one specific stock, there is potential for multiple stocks to be shorted in similar windows of time.

5.3 Subtracting Natural Bias

To summarize the explanation based off of the method selected in the research, the idea is to safely invest a consumer's income into a diverse portfolio of stock investments so that the outcome is profit. This method allows us to group stocks based on their average trends, so in some cases the observer would expect to see stocks of similar industry grouped together but it should be noted that industries in seemingly unrelated fields of business can and will be grouped together. It is important that the user understands that this is a mathematical model that is heavily dependent on the transformation of raw data. The numbers are what the entirety of this research is based on and all bias to specified stocks is disregarded. There are far too many factors in business that can affect the valuation of a companies share price, so grouping them by how they change is far more efficient to get a relative sense of how certain stocks may trend in the future as well as in the present, so that the investor may make a far more educated decision.

6 Conclusions

The PCA approach is a new methodology that can be applied when viewing and interacting in the stock market. Each eigenstock that has been obtained can be used to group stocks based off their similarity to its structure. There is a clear application of this methodology for short term and long term investing strategies. The results are promising and properly justified by error calculations and the Dow Jones model. It is very clear that the trends from the actual data and the transformed data are nearly identical. The Dow Jones representation helped confirm the initial assertion that PCA could be used in mapping behavioral aspects of the stock market. There were then other significant examples that were analyzed such as the British Petroleum (BP) oil spill which is considered to be an isolated incident but when referencing the data, the particular stocks grouped along with BP have similar trends. Although the decline is not as significant as British Petroleum there is still a clear drop off in stock value from the other stocks within it's grouping. These are just a few observations to look back on but they suggest that this is a feasible approach in furthering the understanding of movement in the stock market.

While the mathematics is sound and the technical aspects involved in the computational process of PCA have proven to be legitimate, this research is not without scientific challenges. The main challenge that has become an evident factor is the significant decrease in size of the stock market that was used to formulate the these results. The use of one hundred stocks encompassed within the Fortune 500 that spans seventeen years, while significant in it's own right does not necessarily provide us with the ability to claim this to be nothing but the truth. The other primary issue was the ability to compute accurately stocks that may have opened their doors to be openly traded at a later date than others in the data set. The problem was partially from the lack of data, and also the issue of computing over blank spaces of percent change, that do not exist in the data set. The process while having its fair share of issues is still palpable in shorter durations of time. In the future the idea would be to completely analyze the stock market in it's entirety by gaining access to a more sufficient computational device. This would allow for a more illuminated idea of the issues described and help formulate a plot to solve them. In the meantime this is a sufficient start to obtaining a behavioral solution to the stock market from an approach that has not been explored with this much depth in other mathematical models.

Appendices

A Stocks Used in Our Analyses

Stock	Start Date	Min	Max
"aa.us.txt"	"1970-01-02"	-17.272	13.445
"aapl.us.txt"	"1984-09-07"	-11.907	13.12
"abc.us.txt"	"1995-04-04"	-12.155	11.712
"adbe.us.txt"	"1986-08-14"	-19.952	24.773
"adm.us.txt"	"1983-04-06"	-16.843	13.812
"adp.us.txt"	"1983-04-06"	-9.7137	11.724
"afl.us.txt"	"1984-07-19"	-28.896	30.685
"aig.us.txt"	"1984-09-07"	-38.025	102.7
"amzn.us.txt"	"1997-05-16"	-16.203	28.928
"antm.us.txt"	"1993-01-28"	-10.34	15.954
"axp.us.txt"	"1972-01-07"	-14.703	15.511
"ba.us.txt"	"1970-01-02"	-7.4356	9.8481
"bac.us.txt"	"1986-05-29"	-21.294	21.494
"bbby.us.txt"	"1992-06-05"	-9.8829	19.943
"bby.us.txt"	"1985-04-19"	-36.543	24.003
"biib.us.txt"	"1991-09-17"	-14.599	17.913
"brk-b.us.txt"	"1996-05-09"	-8.6043	26
"c.us.txt"	"1970-01-02"	-30.7	22.96
"cah.us.txt"	"1988-01-04"	-13.813	10.003
"cat.us.txt"	"1970-01-02"	-9.4592	10.697
"ccl.us.txt"	"1989-01-05"	-10.486	14.279
"ci.us.txt"	"1982-03-31"	-24.219	19.261
"cl.us.txt"	"1977-01-03"	-7.568	7.535
"clx.us.txt"	"1983-03-21"	-9.3521	9.652
"cmcsa.us.txt"	"1988-07-07"	-13.39	23.074
"cof.us.txt"	"1994-11-16"	-19.437	30.006
"cop.us.txt"	"1982-01-04"	-14.88	10.753
"cost.us.txt"	"1986-07-09"	-7.217	11.827
"csco.us.txt"	"1990-03-26"	-14.436	26.873
"cvs.us.txt"	"1984-12-17"	-11.065	11.46
"cvx.us.txt"	"1970-01-02"	-13.219	16.154
"dal.us.txt"	"1980-01-02"	-24.999	29.623
"dds.us.txt"	"1989-06-30"	-89.93	42.129
"de.us.txt"	"1982-01-04"	-13.508	18.841
"dis.us.txt"	"1970-01-02"	-8.1505	12.703
"dltr.us.txt"	"1995-03-09"	-10.04	12.691
"dte.us.txt"	"1970-01-02"	-8.6692	10.218
"ecl.us.txt"	"1988-01-05"	-10.366	12.694
"exc.us.txt"	"1980-01-02"	-8.351	15.467
"f.us.txt"	"1977-01-03"	-27.772	38.817
"fdx.us.txt"	"1980-01-02"	-9.6197	14.333
"gd.us.txt"	"1977-01-03"	-9.5778	9.6357
"ge.us.txt"	"1962-01-02"	-11.172	14.964
"gis.us.txt"	"1983-06-10"	-8.0752	5.2069
"gs.us.txt"	"1999-05-04"	-14.942	20.116
"gt.us.txt"	"1970-01-02"	-17.012	24.628
"hd.us.txt"	"1981-09-22"	-6.9582	12.987
"hon.us.txt"	"1970-01-02"	-17.19	12.117
"hpq.us.txt"	"1970-01-02"	-13.471	17.215
"hsy.us.txt"	"1985-07-01"	-7.6875	16.332
"ibm.us.txt"	"1962-01-02"	-9.7572	13.013
"intc.us.txt"	"1972-01-07"	-11.378	13.514
"itw.us.txt"	"1987-11-05"	-9.2893	9.3645
"jcp.us.txt"	"1982-01-04"	-15.147	18.647
"jnj.us.txt"	"1970-01-02"	-8.0796	8.0189
"jpm.us.txt"	"1970-01-02"	-15.726	22.564
"jwn.us.txt"	"1986-07-09"	-12.979	24.968
"k.us.txt"	"1984-12-17"	-9.4905	7.1504
"ko.us.txt"	"1970-01-02"	-8.9733	8.8667
"kr.us.txt"	"1977-01-03"	-9.7976	12.57
"kss.us.txt"	"1992-05-19"	-13.913	13.275
"lmt.us.txt"	"1977-01-03"	-11.091	15.658
"low.us.txt"	"1985-07-01"	-7.3547	18.736
"luv.us.txt"	"1980-01-02"	-15.06	15.573
"mar.us.txt"	"1993-10-13"	-13.847	15.068
"mcd.us.txt"	"1970-01-02"	-10.802	15.364
"mck.us.txt"	"1994-11-15"	-16.799	8.9849
"met.us.txt"	"2000-04-05"	-19.512	19.872
"mmc.us.txt"	"1987-12-30"	-24.255	17.395
"mrk.us.txt"	"1970-01-02"	-10.574	11.586
"msft.us.txt"	"1986-03-13"	-7.5393	11.617
"mu.us.txt"	"1989-05-16"	-17.157	23.67
"nke.us.txt"	"1987-08-19"	-10.108	10.697
"noc.us.txt"	"1981-12-31"	-10.703	9.727
"omc.us.txt"	"1990-03-26"	-12.301	12.476
"orcl.us.txt"	"1988-03-02"	-13.135	26.738
"pep.us.txt"	"1977-01-03"	-10.169	10.64
"pfe.us.txt"	"1982-01-04"	-9.4306	9.4791
"pg.us.txt"	"1970-01-02"	-8.8068	8.8901
"ppg.us.txt"	"1983-04-06"	-10.1	17.567
"rost.us.txt"	"1986-07-09"	-9.3774	16.657
"sbux.us.txt"	"1992-06-26"	-10.213	12.13
"schw.us.txt"	"1989-06-30"	-13.307	26.421
"swk.us.txt"	"1985-07-01"	-11.632	12.113
"syk.us.txt"	"1988-02-01"	-12.302	20.423
"syy.us.txt"	"1987-07-23"	-12.615	11.559
"t.us.txt"	"1984-07-19"	-7.3312	12.046
"tgt.us.txt"	"1983-04-06"	-11.203	14.505
"tjx.us.txt"	"1988-01-05"	-14.295	14.278
"txn.us.txt"	"1981-12-31"	-10.293	14.172
"unh.us.txt"	"1990-03-26"	-16.867	29.472
"unp.us.txt"	"1980-01-02"	-12.146	9.9806
"ups.us.txt"	"1999-11-10"	-8.5607	7.9539
"vz.us.txt"	"1983-11-21"	-7.4604	11.962
"wba.us.txt"	"1985-07-01"	-10.4	11.476
"wfc.us.txt"	"1984-11-01"	-17.569	20.855
"wmt.us.txt"	"1972-03-20"	-9.8785	9.2356
"x.us.txt"	"1991-04-12"	-15.993	20.47
"xom.us.txt"	"1970-01-02"	-12.366	12.602
"xrx.us.txt"	"1977-01-03"	-25.777	19.606

Figure 13: The 100 Stock Dataset

B MATLAB Code for PCA

```
function [V,E,mu,W] = PCA(A, p)
    % This function performs PCA on the data matrix, where the columns
    % of data are samples and the rows are features
    % Input Arguments:
    % data - The data to be analyzed, m >> n
    % p - the minimum percent of variance to be accounted for
    % Output Arguments:
    % V - The principal components of data
    % E - The eigenvalues associated with each principal component
    % mu - The mean sample
    % W - The weight matrix,
    %% Important Info
    nsamples = size(data,2);
    nfeatures = size(data,1);
    % Compute mean stock
    mu = mean(A,2);
    % Compute Shifted matrix
    T = A - mu;
    % Compute Covariance matrix
    C = cov(T');
    % Compute Eigenvectors and eigenvalues
    [V,D] = eigs(C,nsamples);
    E = diag(D);
    % Compute percent explained
    explained = E./sum(E);
    % Save eigenvectors up until requested explained
    tot_explained = 0;
    j = 1;
    V_red = [];
    while tot_explained < p
        v_temp = V(:,j);
        v = v_temp./norm(v_temp);
        V_red = [V_red, v];
        tot_explained = tot_explained + explained(j);
        j = j + 1;
    end
    V = V_red;
    %% Find weight vectors
    W = V'*T;
end
```

C MATLAB Code for Forming Portfolios

```
clear
%% User Inputs
save_file = 'big_name_stock_data.mat';
% Place Path to Folder Here
testfiledir = 'C:\Users\taywo\Downloads\price-volume-data
    -for-all-us-stocks-etfs\Big Name Stocks 2001';
% Set minimum number of days in stock file
min_days = 260*7;
% Set the start date for analysis
start_date = 4420;
% Set the end date for analysis
end_date = 260*0;
% Portfolio number to plot
portfolio_no = 1;
% File to Classify
% class_stock = 'luv.us.txt';
% class_file = 'C:\Users\taywo\Downloads\price-volume-
    data-for-all-us-stocks-etfs\Stocks\luv.us.txt';
class_file = '';
% Desired percent of variance to be explained
percent_explained = 0.9;
%% Read in Files that Meet Criteria
matfiles = dir(fullfile(testfiledir, '*.txt'));
nfiles = length(matfiles);
disp('Total Number of files:')
disp(nfiles)
j = 1;
for i = 1:nfiles
    fid = fullfile(testfiledir, matfiles(i).name);
    M = importdata(fid);
    if isempty(M)
        continue
    end
    if string(M.textdata(end,1)) == '2017-11-10' && size(
        M.textdata,1) >= min_days
        data{j} = M;
        file_names{j} = matfiles(i).name;
        j = j + 1;
    end
end
nfiles = j - 1;
disp(string(nfiles)+' Files Loaded')
%% Calculate percent change and place into a matrix based
```

```
    on date
% Matrix format is rows are samples, and columns are
    features
% Columns are Date, Open, High, Low, Close, Volume,
    OpenInt
% Percentage change of stock over the day as values
% Files always end at November 10, 2017
% Some files are empty
min_sz = 10000000;
% Identify the smallest of the bunch
for i = 1:nfiles
    A = data{i};
    if size(A.data, 1) < min_sz
        min_sz = size(A.data, 1);
    end
end
ul_pc_matrix = [];
if start_date < min_sz && start_date ~= 0
    min_sz = start_date;
end
% Calculate Percent Change and Form the Data Matrix
for i = 1:nfiles
    A = data{i};
    pc = ((A.data(:,4)-A.data(:,1))./A.data(:,1)).*100;
    sz = size(pc, 1)+1;
    pc =  pc(sz-min_sz:end,:);
    ul_pc_matrix = [ul_pc_matrix, pc];
end
date_diff = start_date - end_date;
ul_pc_matrix = ul_pc_matrix(1:date_diff,:);
disp('Data Cleaned and Ready for Analysis')
%% Perform PCA
A = ul_pc_matrix;
[V,E,mu,W,explained] = PCA(A, percent_explained);
%% Important Info
nsamples = size(A,2);
nfeatures = size(A,1);
%% Form Porfolios Based on Highest Weighted Eigenstock
disp('')
hold on;
[~,best_eigenstock] = max(abs(W));
j = 1;
stock_names = {};
% Classify new stock
if size(class_file,2) > 1
    new_data = importdata(class_file);
```

```
84      new_data_pc = ((new_data.data(:,4)-new_data.data(:,1)
            )./new_data.data(:,1)).*100;
85      sz = size(new_data_pc, 1)+1;
86      new_data_pc =  new_data_pc(sz-min_sz:end,:);
87      new_data_pc = new_data_pc(1:date_diff,:);
88      phi = new_data_pc - mu;
89      omega = V'*phi;
90      test_matrix = abs(W - omega);
91      [~, index] = min(sum(test_matrix.^2,1).^(1/2));
92      portfolio_no = best_eigenstock(index);
93      xlabs = new_data.textdata(sz+1-min_sz:end,1);
94      plot(datetime(xlabs), new_data.data(sz-min_sz:end,4))
95      stock_names{j} = class_stock;
96      j = j + 1;
97  end
98  % Create report and plot
99  saved_indexes = [];
100 for i = 1:size(V,2)
101     indexes = find(best_eigenstock == i);
102     if ~isempty(indexes)
103         disp('Stocks in Portfolio '+string(i))
104         for index = indexes
105             disp(file_names{index})
106             if i == portfolio_no
107                 saved_indexes = indexes;
108                 sz = size(data{index}.data, 1)+1;
109                 xlabs = data{index}.textdata(sz+1-min_sz:
                        end,1);
110                 if j < 5
111                     plot_data = data{index}.data(sz-
                            min_sz:end,4);
112                     plot(datetime(xlabs(1:date_diff,:)),
                            plot_data(1:date_diff,:))
113                     stock_names{j} = file_names{index};
114                     j = j + 1;
115                 end
116             end
117         end
118     end
119     indexes = [];
120 end
121 title('Portfolio '+string(portfolio_no))
122 legend(stock_names)
123 xlabel(string(nsamples)+' Stocks with '+ ...
124         string(percent_explained*100)+'% Variance')
125 ylabel('Closing Price')
```

References

[1] Fortune Media. "FORTUNE." Fortune, Fortune, 3 Apr. 2020 `fortune.com`.

[2] Carol Anne Hargreaves, and Chandrika Kadirvel Mani, *The Selection of Winning Stocks Using Principal Component Analysis*, American Journal of Marketing Research, Vol. 1, No. 3, pp. 183-188, (2015)

[3] Marjanovic, Boris. *Huge Stock Market Dataset*. Kaggle, 2017. `https://www.kaggle.com/borismarjanovic/`

[4] Muhammad Waqar, Hassan Dawood,Muhammad Bilal Shahnawaz, Mustansar Ali Ghazanfar, and Ping Guo, *Prediction of Stock Market by Principal Component Analysis*, Proceedings of the 2017 13th International Conference on Computational Intelligence and Security

[5] Sheng Zhang and Matthew Turk. *Eigenfaces*. Scholarpedia, 3(9):4244 (2008).

[6] Strang, Gilbert. *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.