MATH 628 FINAL PROJECT

PCA Analysis

Chunlin Shi

Noah Collins

1. Abstract

In this project, we found 1-year daily trading data for the following stocks: AAL, AMD, AVGO, BAC, BRK-A, CVX, DAL, EOG, INTC, JPM, LUV, NVDA, SLB, UAL, WFC, XOM. We repeated the analysis of Sections 2.1-2.2 of the paper by Avellaneda and Lee. We then discussed the signs of the second and the third eigenvectors partition stocks into stocks and discovered that stocks in the same group have the same sign on the second and third eigenvectors, which corresponds to the grouping of the stocks by industries.

2. Introduction

Principal Component Analysis is a powerful statistical tool that allows us to explore multidimensional relationships within datasets. In our paper's case, our dataset was 15 stocks and their trading information over a year. PCA analysis helps capture the most significant sources of variation, which in the case of stocks, unveils patterns and correlations that may not be apparent to the naked eye. We seek to delineate groups of stocks that share common trends, behaviors, or underlying factors and to compare these results with official industry groupings to demonstrate the efficacy of PCA as a tool for financial professionals, researchers and investors.

3. Data Collection Methodology

To conduct this research, we gathered data from the website of Center for Research in Security Prices (CRSP) of Wharton Research Data Services. The time range of the daily trading data is from December 31, 2021, to December 30, 2022. We collected following variables for our study:

Names Date: Date of trading.

Ticker Symbol: Ticker for stocks accordingly.

North American Industry Classification System: Code used to proxy belonging industry of the stock.

Price or Bid/Ask Average, Shares Outstanding: Variables used to construct market cap weighted portfolios.

Returns without Dividends: Daily return data used for PCA analysis and eigen portfolio construction.

4. Implementation of the Analysis in the Paper by Avellaneda and Lee.

a) Section 2.1

According to the Paper, the daily return is calculated by:

$$R_{ik} = \frac{S_{i(t_0-(k-1)\Delta t)} - S_{i(t_0-k\Delta t)}}{S_{i(t_0-k\Delta t)}}, \; k = 1, \cdots, M, \; i = 1, \cdots, N$$

Where $M = 252$ and $N = 16$, where $S_{it}$is the price of stock i at time t adjusted for dividends and $\Delta t = \frac{1}{252}$

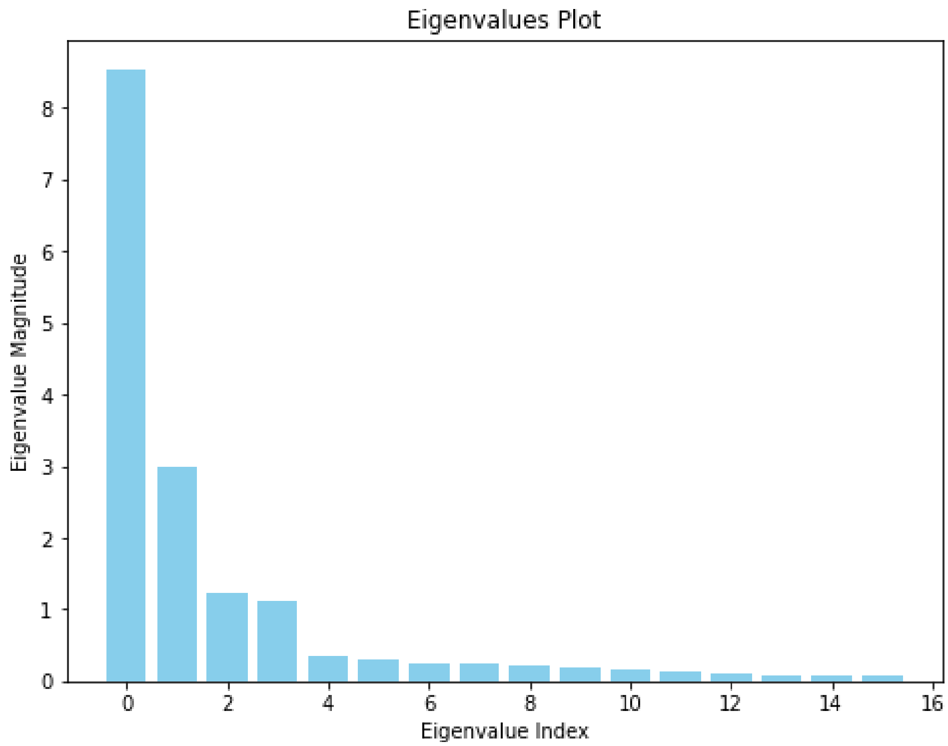
The return was standardized as follows:

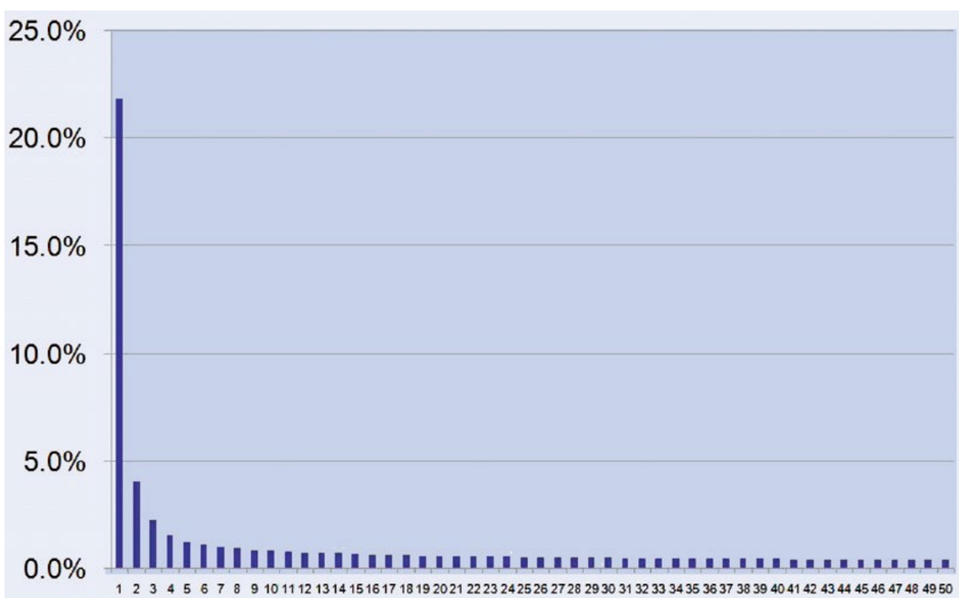
$$Y_{ik} = \frac{R_{ik} - \overline{R_i}}{\overline{\sigma_i}}$$

Where,

$$\overline{R_i} = \frac{1}{M}\sum_{k=1}^{M} R_{ik}$$

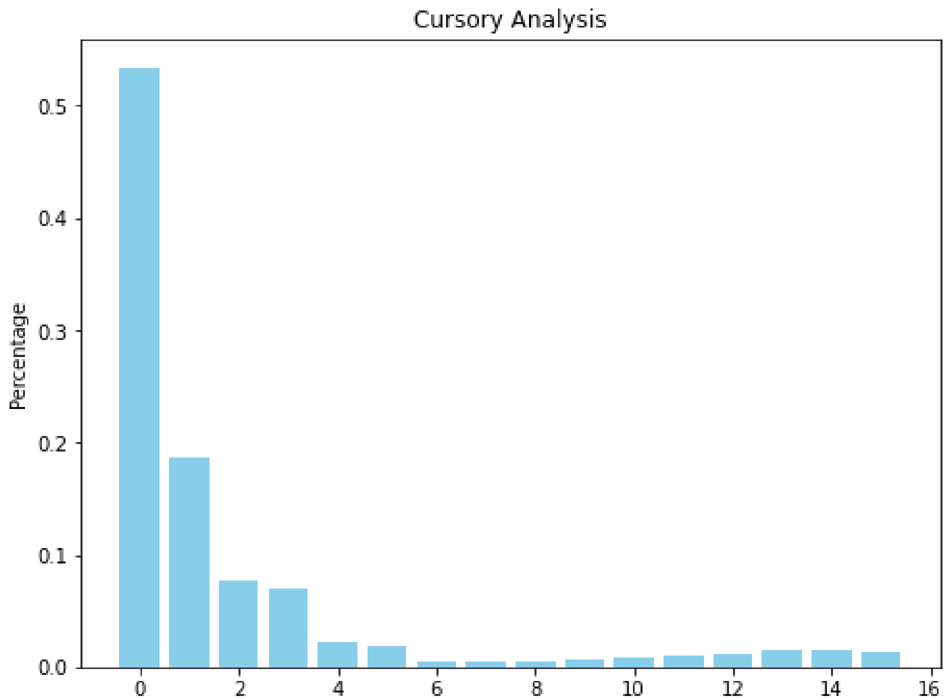
In our procedures, we used daily return adjusted for dividends from CRSP and used *StandardScalar in sklearn* in Python to calculate standardized return matrix. Then, we applied PCA analysis with 3 principal components on the standardized return matrix. We also calculated the correlation matrix based on the standardized return matrix, and generated eigenvalues and plotted them from the largest value to the lowest value. Below is the graph we plot and the graph in the paper:

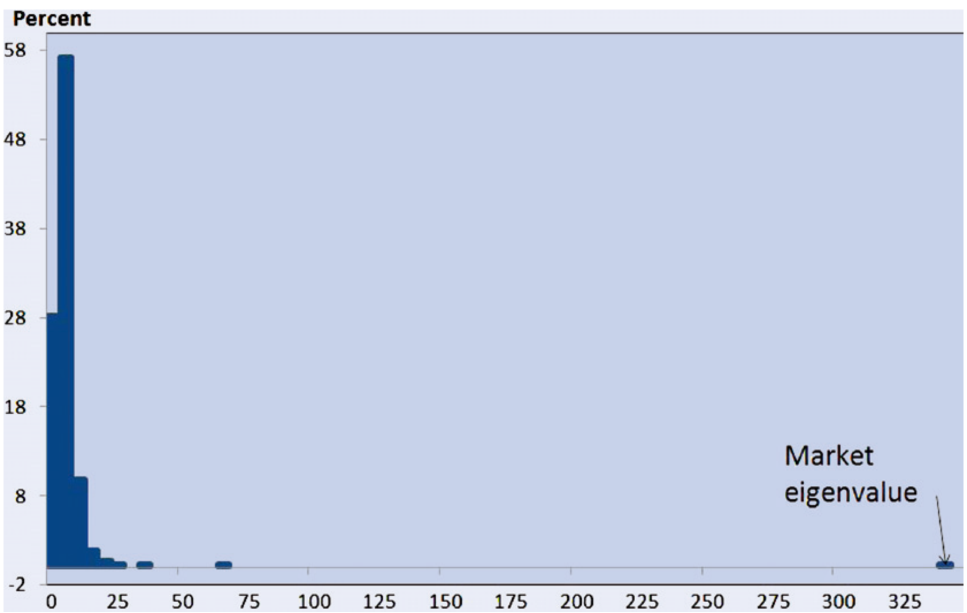




From graphs above, we can see that both graphs show a decreasing trend of the magnitude of eigenvalues.

We later conducted a cursory analysis on eigenvalues. Below is the graph we plot and the graph in the paper:

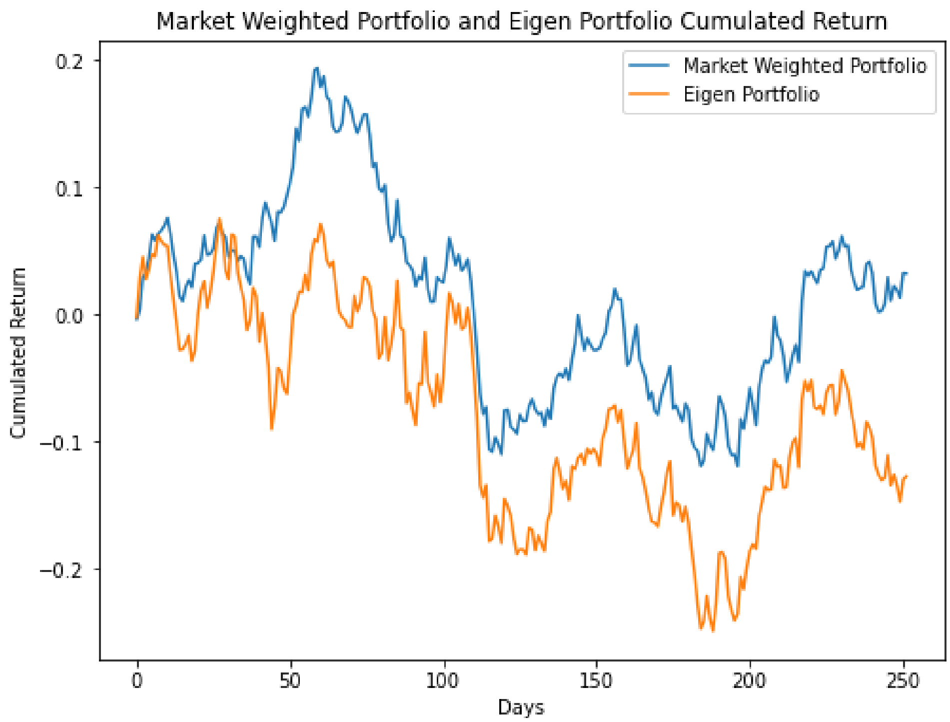


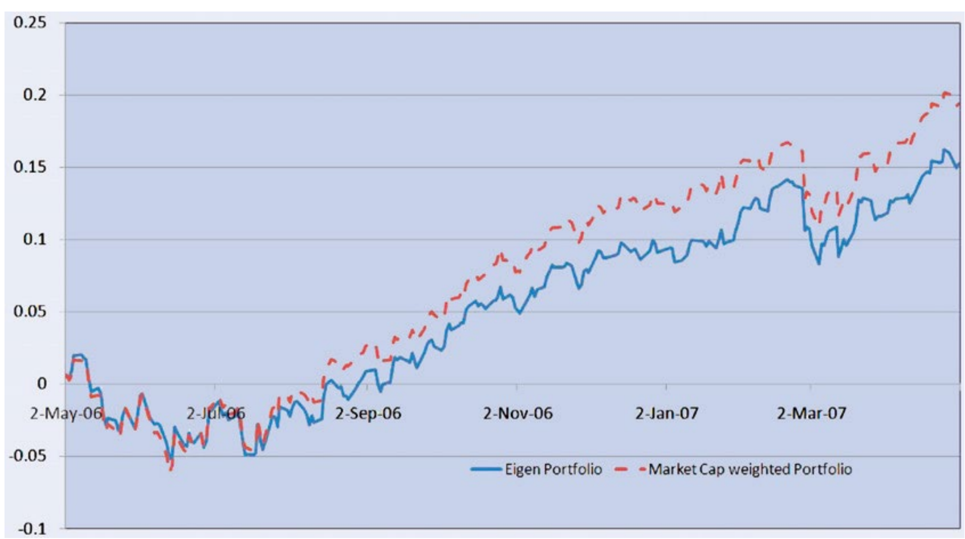


From graphs above, we can see that both graphs show similar trends that high density is shown on the left side of the graph while a minor increase on the right side.

b) Section 2.2

In this section, we constructed the eigen portfolio and the market cap weighted portfolio to calculate their cumulative returns. Below is the cumulative return graph we plot and the graph in the paper:





We discovered that our graph showed high similarity with graph provided in the paper that both portfolios showed a comparative evolution, and the cumulative return of the eigen portfolio is slightly lower than market cap weighted portfolio. Hence, this result matches the view in the paper that two portfolios are not identical but are good proxies for each other.

c) **The relationship between Signs of the Second and Third Eigenvectors and the Belonging Industry of the Stock**

In this section, we used the result of PCA analysis with 3 principal components on the standardized return matrix. We retrieved the second and third eigenvectors of each stock and merged them with the industry code. Below is the table:

Stock	**Second Eigenvector**	**Third Eigenvector**	**North American Industry Classification System**
EOG	-0.450143	-0.049374	211120
SLB	-0.423821	-0.103848	213112
XOM	-0.468001	-0.032569	324110
AMD	0.073633	0.412608	334413
AVGO	0.069199	0.376026	334413
INTC	0.034908	0.417784	334413
NVDA	0.098801	0.420549	334413
CVX	- 0.451722	- 0.026942	447190
AAL	0.218933	- 0.210610	481111
DAL	0.203492	- 0.259594	481111
LUV	0.174883	- 0.220278	481111
UAL	0.218353	- 0.266802	481111
BAC	0.000662	- 0.165386	522110
JPM	0.025836	- 0.158595	522110
WFC	0.032731	- 0.194900	522110
BRK	- 0.064327	0.001756	524126

From the table above, we can see that stocks in the same industry have the same signs of the second and third eigenvectors.

5. Conclusion

Through replication of the original results using a smaller dataset, our project aimed to validate the robustness and generalizability of the findings to a more constrained context. Despite the scale reduction, our results closely mirror those reported in the original paper, demonstrating the consistency of PCA and its ability to group stocks in a matter that corresponds to industries.

6. References

Marco Avellaneda & Jeong-Hyun Lee (2010) Statistical arbitrage in the US equities market, Quantitative Finance, 10:7, 761-782, DOI: 10.1080/14697680903124632

Wharton Research Data Services. (2022). Wharton Financial Data Services. Retrieved from https://wrds-www.wharton.upenn.edu/

Pandas Development Team. (2023). pandas (1.3.3). Retrieved from https://pandas.pydata.org/

NumPy Community. (2023). NumPy (1.21.2). Retrieved from https://numpy.org/

Matplotlib Development Team. (2023). Matplotlib (3.4.3). Retrieved from https://matplotlib.org/

Scikit-learn Developers. (2023). scikit-learn (0.24.2). Retrieved from https://scikit-learn.org/