Principal Component Analysis – S&P500 Equity Exploration

Maciej Staniszewski and Adam Foster

Project saved in:

https://github.com/Staneesh/efi-pca

https://github.com/afoster28/efi-pca

Executive Summary

PCA is a dimensionality reduction method whereby multiple variables in a large dataset are transformed into a smaller set of variables, maintaining most of the information in the large dataset. Resulting variables are easier to explore and analyse and are less computationally expensive to process.

This exercise covers a practical demonstration of this approach, as well as its application in determining key single name equity drivers of S&P500 index returns.

Table of Contents

Description of the Problem

In finance, similarly to other domains of science which are heavily reliant on concepts of data analysis and exploration, we often face large sets of data. These can often be difficult to analyse manually, even when plotted graphically. Noticing recurring patterns is highly impractical when the number of variables in the data exceeds reasonable bounds.

Interpreting data on aggregate level is especially difficult. Estimating joint distributions is an elementary step when determining the data generating process behind the dataset. This in turn is a necessary step in inference-based thinking and causal discovery in data. The dimensionality curse restricts us to three easily observable dimensions. When analysing complex datasets consisting of hundreds of variables, it is crucial that a dimensionality-reducing operation is performed, enabling scientists to consider these joint distributions of every variable in a simplified form.

Theoretical Background

A well-known method of representing variables for such analysis is Principal Component Analysis, commonly known as PCA[1]. The main idea behind it is to create new variables from existing ones, that maximize the ratio of original observed variance to the variance captured by these principal components. To do that, an n-dimensional dataset needs to be reduced to a k-dimensional one, where k is smaller than n.

The coordinates of that new system are the namesake principal components. They represent directions in n-dimensional space which retain most of the original variance when casting the data points onto them. The first vector created this way explains the biggest portion of original variance and is thus called the first principal component. Data points projected on this newly created coordinate system can be explored manually and jointly due to the reduced dimensionality.

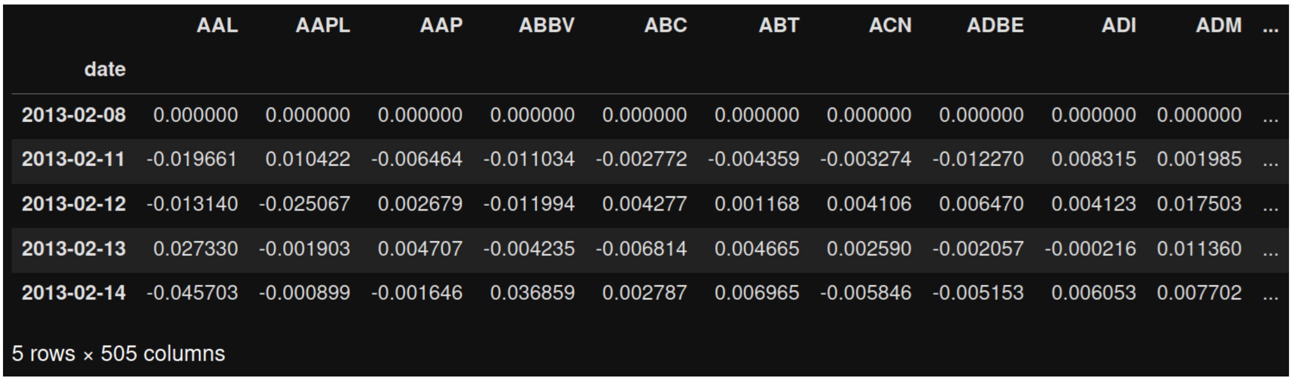
Data

We used a dataset which includes all *S&P500* single-name stock daily close prices in a time period between 2013 and 2018[2]. To obtain the index daily close prices in that period we used the Python yfinance library. We combined these two datasets into one pandas dataframe and cleaned the data: transformed prices into returns, pivoted the data for compatibility with other libraries, ensured

[1] Principal Component Analysis - Wikipedia

[2] S&P500 Single Name Prices - Kaggle Dataset

coherency between all starting dates and ending dates, set not applicable values to zeroes where applicable given standard normality assumptions regarding returns (e.g. when time series ended due to exiting the index). We obtained the following dataset:



	AAL	AAPL	AAP	ABBV	ABC	ABT	ACN	ADBE	ADI	ADM	...
date											
2013-02-08	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
2013-02-11	-0.019661	0.010422	-0.006464	-0.011034	-0.002772	-0.004359	-0.003274	-0.012270	0.008315	0.001985	...
2013-02-12	-0.013140	-0.025067	0.002679	-0.011994	0.004277	0.001168	0.004106	0.006470	0.004123	0.017503	...
2013-02-13	0.027330	-0.001903	0.004707	-0.004235	-0.006814	0.004665	0.002590	-0.002057	-0.000216	0.011360	...
2013-02-14	-0.045703	-0.000899	-0.001646	0.036859	0.002787	0.006965	-0.005846	-0.005153	0.006053	0.007702	...

5 rows × 505 columns

Figure 1: Five rows of preprocessed dataset including daily returns of all S&P500 single-name stocks and the index price between 2013 and 2018.

Methodology

Mathematical Foundations

In order to achieve desired transformation of data, the approach can be broken down into five key steps. At its core, PCA incorporates eigendecomposition and maximisation on pre-processed data to achieve an alternative simpler function form of an equation expressing the relationship between several variables.

Step 1 - Standardisation

Standardisation involves the normalisation using mean and standard deviation and makes varying domains comparable. This prevents variables with larger domains dwarfing the impact of smaller domains, reducing bias.

$$Z = \frac{value - mean}{standard\ deviation}$$

Step 2 - Covariance Matrix

In searching for all possible relationships between the variables and their joint variance, a covariance matrix is created. For n variables, an $n \times n$ covariance matrix is formed.

We can observe a couple of interesting facts about the covariance matrix:

- Diagonal elements of the covariance matrix describe the variance of individual variables. This is because they contain information about the variability of the pair of values $e1, e1$ which is essentially the same as the variability of $e1$ for some variable $e1$.
- Another observation is that the matrix should be symmetrical about the diagonal. This means that elements $(e1, e2)$ vary in the same exact way as elements $(e2, e1)$. Note: that python libraries will raise a warning if the covariance matrix does not have that property.

Step 3 - Eigendecomposition

Principal components are new variables that are constructed as linear combinations of the initial variables. These combinations are done in such a way that the principal components are uncorrelated and most of the information within the initial variables (and therefore the variance) is compressed into the first components.

Across all principal components, the same information is maintained. The advantage is that the minimum amount of information is lost when removing the least important components, hence a large proportion of the data can be represented by few components – they pick up the maximal amount of variance.

The first principal component is calculated to maximise the average of the squared distances from projected points to the origin. The second one is calculated the same way, provided it is uncorrelated with and therefore orthogonal to the first component. This can be continued for the number of dimensions of the data and each component has an associated eigenvector-eigenvalue pair. Eigenvectors describe the direction of the axes where there is the most variance and eigenvalues show the amount of variance carried. Ranking eigenvectors by eigenvalues in descending order shows the principal components in order of significance. The eigenvalue relative to the sum of eigenvalues denotes the percentage of variance covered by the associated principal component.

Methodology

A point x gets projected onto a unit vector u, a new point x' is created whose magnitude is calculated by the inner product between x and u: $x' = (x^T u)u$.

$(x^T u)^2$ can be thought of as the amount of information preserved and is maximal when x is parallel to u and minimal when x is orthogonal to u. PCA is then an optimisation problem whereby it seeks to maximise information preserved subject to the unit vector usage:

$max \sum_i (x_i^T u)^2$ subject to $u^T u = 1$

which simplifies to

$\max u^T Cu$ subject to $u^T u = 1$ where $C = \frac{1}{n}\sum_i x_i \, x_i^T$ (covariance matrix)

This can be expressed as a Lagrangean and solved:

$L = u_T Cu - \lambda(u_T u - 1)$

$2Cu - 2\lambda u = 0$

$Cu = \lambda u$

Therefore, u and λ represent eigenvectors and eigenvalues, respectively. The best direction u_1 to pick is coupled with the eigenvector with the largest eigenvalue λ_1. u_2 can then be selected such that $u_2 \perp u_1$ with λ_2 and so on.

Step 4 - Feature Vector

At this stage, a process of elimination occurs based on principal component significance and the resulting components form the feature vector.

Step 5 - Recast the Data along the PC Axes

Eigenvectors of the feature vector principal components are used to reorient the data from the original axes to the ones represented by the principal components.

$FinalDataset = FeatureVector_T \times StandardisedOriginalDataset_T$

This data can be visualised in a more intuitive manner across fewer dimensions.

Project Specification

In our project we decided to explore a well-known financial equity index – *Standard & Poor's 500* or *S&P500*[3]. Its price[4] is determined by the joint market performance of five hundred largest US-based companies traded publicly. We wanted to find the companies which drive the price variability of the index and determine if the performance of *S&P500* can be expressed as a linear combination of the underlying assets. We relied on PCA to decompose daily returns of the index into principal components and obtained main drivers from the coordinates of the first component. We arrived at a quantitative solution which helps analysts to understand price moves of complex financial products, such as an equity index.

[3] Standard and Poor's 500 - Wikipedia

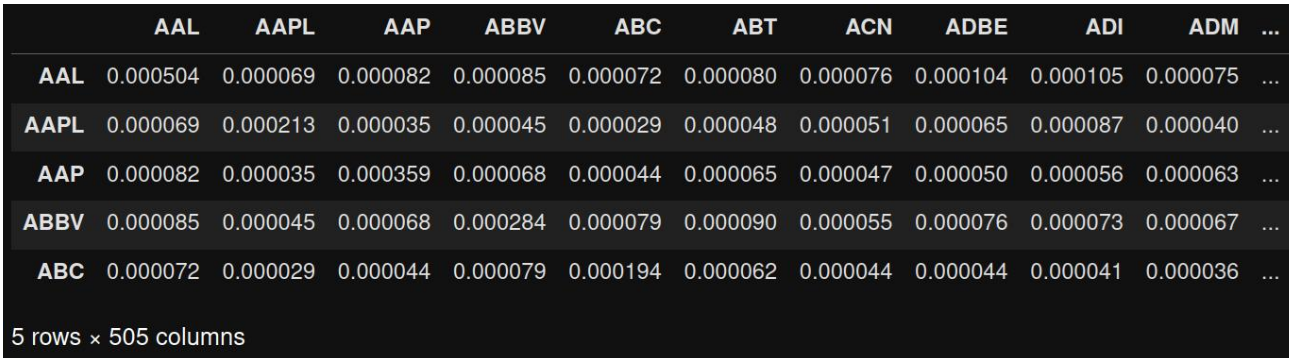
[4] S&P500 Index Price - S&P Global

Execution

We used Python programming language inside a Jupyter Notebook[5] to interactively display both code and its results. The following packages were included:

- numpy[6] for basic mathematical operations
- pandas[7] for dataset manipulations
- ipykernel Jupyter kernel for Python code execution
- matplotlib[8] for data visualisation with plots
- yfinance for financial data from Yahoo Finance[9]

We were operating on the dataset described in detail in section Data. We used cov() Python method to compute the covariance matrix which sample is presented below:



	AAL	AAPL	AAP	ABBV	ABC	ABT	ACN	ADBE	ADI	ADM	...
AAL	0.000504	0.000069	0.000082	0.000085	0.000072	0.000080	0.000076	0.000104	0.000105	0.000075	...
AAPL	0.000069	0.000213	0.000035	0.000045	0.000029	0.000048	0.000051	0.000065	0.000087	0.000040	...
AAP	0.000082	0.000035	0.000359	0.000068	0.000044	0.000065	0.000047	0.000050	0.000056	0.000063	...
ABBV	0.000085	0.000045	0.000068	0.000284	0.000079	0.000090	0.000055	0.000076	0.000073	0.000067	...
ABC	0.000072	0.000029	0.000044	0.000079	0.000194	0.000062	0.000044	0.000044	0.000041	0.000036	...

5 rows × 505 columns

Figure 2: Five rows of covariance matrix of daily returns of S&P500 single-name stocks and the index price between 2013 and 2018.

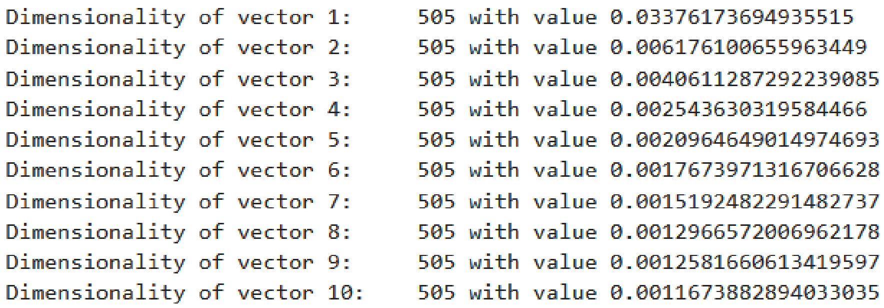
After such pre-processing we were ready to start considering eigenvectors and eigenvalues. The results of our analysis are present in the next section: Results.

Results & Sensitivity Analysis

The following top eigenvectors were obtained through the decomposition. Considering the large number of variables, poor extent of correlation among many of the underlying equities and the existence of index weights, the top several principal components explain a commendable portion of

[5] Jupyter Notebook - Main Page
[6] NumPy - Python
[7] Pandas - Python Data Analysis Library
[8] Matplotlib - Visualizing with Python
[9] Yahoo Finance - Top ETFs

the returns of the index portfolio. The total number of variables exceeds the 500 index constituents total due to removal and addition of constituents throughout the time period considered.



```
Dimensionality of vector 1:     505 with value 0.03376173694935515
Dimensionality of vector 2:     505 with value 0.006176100655963449
Dimensionality of vector 3:     505 with value 0.0040611287292239085
Dimensionality of vector 4:     505 with value 0.002543630319584466
Dimensionality of vector 5:     505 with value 0.0020964649014974693
Dimensionality of vector 6:     505 with value 0.0017673971316706628
Dimensionality of vector 7:     505 with value 0.0015192482291482737
Dimensionality of vector 8:     505 with value 0.0012966572006962178
Dimensionality of vector 9:     505 with value 0.0012581660613419597
Dimensionality of vector 10:    505 with value 0.0011673882894033035
```

Figure 3: Top 10 eigenvectors

The top five principal components explain the following proportions of total variance:

1. 26.6%
2. 4.9%
3. 3.2%
4. 2.0%
5. 1.7%

The top principal component explains as much as 26.6% of overall variance. It could potentially be used for future index return predictions whereby instead of using all 500 underlying equities only this one principal component would reflect index returns with 26% accuracy based on historical returns. S&P500 data availability makes this rather unnecessary.

Due to the exploratory nature of this paper, we show another application of the results of this PCA: showing which are the key equities driving S&P500 returns, as opposed to limiting ourselves to principal components which are non-interpretable in terms of equities on their level of aggregation.

The algorithm developed for this purpose is as follows:

- Obtain the covariance matrix from underlying data
- Extract the top eigenvector (feature vector)
- Select the several top coordinates by magnitude, i.e. contribution to eigenvector construction
- Map these coordinates back to the underlying equities
- Retrieve these equities and plot them against the S&P500

The following five equities were selected:

- RCL – Royal Caribbean Cruises Ltd. – world's second-largest cruise holding company
- YUM – Yum! Brands, Inc. – fast food company including KFC, Pizza Hut and Taco Bell brands
- NWL – Newell Brands – manufacturer, marketer and distributor of consumer and commercial products
- NTAP – NetApp, Inc. – hybrid cloud data services and data management company
- XRX – Xerox Holdings Corporation – corporation that sells print and digital document products and services in more than 160 countries

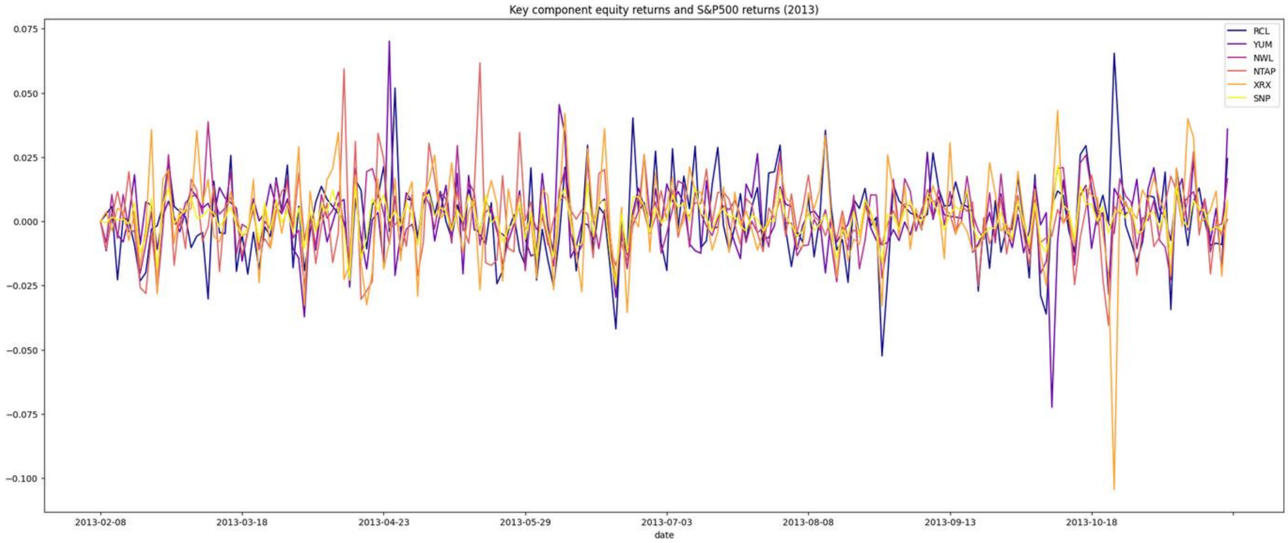


Figure 4: Key component equity returns and S&P500 returns (2013)

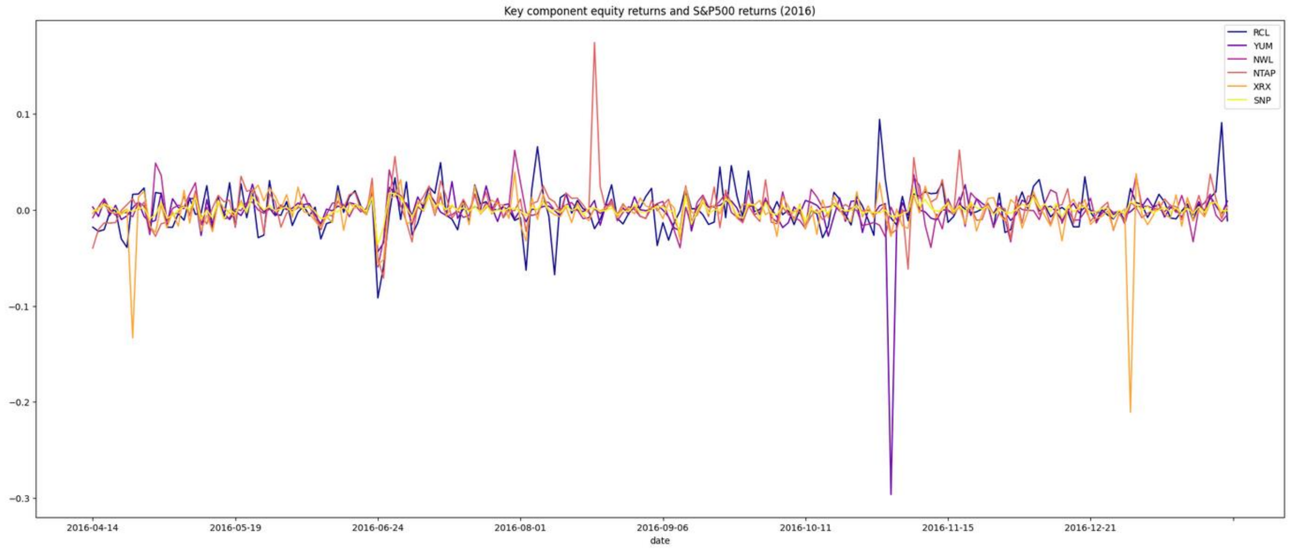


Figure 5: Key component equity returns and S&P500 returns (2016)

Summary and Recommendations

PCA was used to represent individual returns of equities within the S&P500 index using principal components. The most significant principal component explained over 26% of the index's variance. Additional methodology allowed for the extraction of the top 5 equities driving this main component. For the 2013-2018 time period, these spanned large corporations across multiple industries, like tourism, retail & tech – representative of the broad index.

This analysis provides a simpler and more practical representation of the returns of all S&P500 constituents. It also provides a framework for extracting key single name equities within principal components for equity-specific tracking, which saves both analytical and computational resources.

Bibliography

https://towardsdatascience.com/principal-component-analysis-pca-from-scratch-in-python-7f3e2a540c51

https://stats.stackexchange.com/questions/143905/loadings-vs-eigenvectors-in-pca-when-to-use-one-or-another

https://www.kaggle.com/datasets/camnugent/sandp500

https://builtin.com/data-science/step-step-explanation-principal-component-analysis