What Data Series Matter?

Explaining key trends and factors generated by Artificial Intelligence

By: Irene Aldridge[1] and Ying Guan[2]
Cornell University and AbleMarkets

This version: November 3, 2021

Abstract

We show a simple way to let the data speak for themselves. Specifically, we show how a large mixed bag of data, potentially embedded with missing data points and collinearities, and therefore unsuitable for traditional econometric analysis, can be useful in building fast and meaningful big data and artificial intelligence analyses and predictions. What's more, our technique helps the results of the analyses to be easily interpreted by researchers. We use these techniques to build a surprisingly profitable E-mini crude oil futures trading strategy with monthly reallocations, delivering annualized returns of 100%+ with Sharpe ratio exceeding 2.2.

Keywords: asset pricing, artificial intelligence, pca, svd, factors, econometrics

Introduction

Perhaps the key component of Artificial Intelligence and Data Science is the utilization of eigenvalue-based decomposition in finding the optimal factorization of the data on hand. The decompositions, Principal Component Analysis (PCA) or its cousin Singular Value Decomposition (SVD), indeed create optimal linear factors for the data (for proof, refer to Chapter 5 of Aldridge and Avellaneda (2021)). The great advantage of the technique is that any data set can be easily factored in the linear manner, and optimally by construction. Furthermore, for financial data, such factorization fits perfectly with decades of established research with the likes of Sharpe (1964) and Lintner's (1966) Capital Asset Pricing Model (CAPM) and Ross (1976, 1977) Arbitrage Pricing Theory (APT).

One of the key uncertainties surrounding the adoption of PCA and SVD has been the potential lack of interpretability of the results. Both PCA and SVD generate a set of vectors, ranked in importance. Each vector contains a set of coefficients, one for each of the columns of the original data that was decomposed using PCA or SVD. Each vector, when the coefficients are

[1] Cornell University, Cambridge University Judge School of Business, irene@ablemarkets.com

[2] Cornell University, yg532@cornell.edu

multiplied by their respective data columns and added together, comprises an optimal factor. While these factors are now numerically available and clear, finding the real-life equivalent to the set is often left as a daunting task to a researcher. What's more, the researcher is frequently constrained in his or her interpretation of PCA or SVD findings by the researcher's own background and data at his or her disposal.

To explain PCA or SVD-driven factors, researchers to date have deployed various analyses to find the closest fit to the factors from other readily available data sources. For example, as shown in Aldridge and Avellaneda (2021, p. 151), for the S&P 500 stock returns, the first singular vector in fact comprises the returns of the equally-weighted of the S&P 500 portfolio, fully consistent with CAPM and other models that include market returns as a linear factor. Other PCA and SVD-developed factors, however, are more difficult to match with their real-life proxies. Muravyev, Wasquez and Wang (2018) correlate PCA factors with real-world factors to find the best, most correlated, match. Giglio and Xiu (2019) deploy two-pass cross-sectional regressions to match PCA factors and real-life observable factors by their risk premia. Connor, Hagmann and Linton (2012) and Kelly, Pruitt and Su (2017) use instrumental variables to match PCA and real-world factors. Laloux, Cizeau, Potters and Bouchaud (2000), and Avellaneda and Lee (2010) show that covariance-based PCA decomposition produces variance-biased results altogether, and only correlation-based decomposition should be used in PCA factor determination and subsequent matching with real-world examples.

To link financial returns with the underlying PCA factors, Stock and Watson (2002) propose a hybrid PCA-factor and autoregressive model, while Bai and Ng (2013), Doz, Giannone and Reichlin (2011) and Fan, Liao and Mincheva (2013) use multi-factor regressions where they regress returns on contemporaneous or lagged factors determined by PCA. Fan, Liao and Mincheva (2013)'s celebrated POET method decomposes data covariance, as opposed to raw data series.

In this paper, we propose a simple interpretive framework for AI generated using PCA or SVD. Instead of considering all the features in the linear combination of the eigenvector, we look only at the dominant feature as the proxy for that vector. Via simulation, we show that this method overcomes potential collinearity issues and presents a quick and efficient method for synthesizing the information presented in the data. We build trading strategies by constructing multi-factor regressions of target returns on lagged dominant features of the top factors only, instead of the whole factors.

We test our model on a presumed relationship between oil prices, consumer confidence index and market returns. Several researchers have pointed out the linkages among the three variables. For example, Baker and Wurgler (2006) find that consumer confidence indicators and their proxies impact stock returns. In particular, Baker and Wurgler (2006) found that consumer confidence has the most impact on off-the-mainstream-radar stocks, such as small stocks, young stocks, unprofitable stocks, high-volatility stocks, distressed stocks, etc. Johnson and Lamdin (2013) find that changes in gasoline prices inversely affect future consumer confidence numbers: the higher the gasoline prices, the lower the consumer confidence. However, Zafeiriou, Katrakilidis and Pegiou (2019) do not find any meaningful relationships between heating oil prices and consumer confidence index in the European Union. Killian (2009) notes that supply and demand shocks in oil prices affect U.S. markets differently, with demand shocks having less impact on the U.S. economy. At the same time, Killian and Vega (2011) find no relationship between U.S. macroeconomic news and subsequent oil prices, on either daily or monthly horizons. In our analysis, we seek to find relationships among the three variables: U.S. stock prices, oil prices and consumer confidence levels.

Our AI-driven model produces surprising results that result in a profitable trading strategy for U.S. E-mini crude oil futures.

Model

Singular Value Decomposition (SVD) and its close cousin, Principal Component Analysis (PCA), deliver orthogonal vectors that are bases for optimal factorization. Each vector in itself, by construction as discussed in Aldridge and Avellaneda (2021) is the optimal factor, orthogonal to other vectors in the set. Each vector is a linear combination of the features (columns) of the original data set.

Some techniques propose correlating the singular vectors with a chosen dependent variable to “translate” the factors to humans. Here, we propose a much simpler approach. By examining each singular vector, we can infer the most important factor just by observing the largest components in the vector. The constituent with the largest coefficient in the vector will most likely have the most influence within the factor and have the largest correlation with the chosen dependent variables.

The methodology proposed herein easily absolves researchers from dealing with correlated dependent variables creating the problem known as “collinearity” in traditional econometric analysis. By deploying the eigenvalue decomposition and performing the associated optimal factorization, the key factors are elucidated without concerns about inter-factor dependence.

For example, Christoffersen, Fournier and Jacobs (2017) apply eigenvector factorization to options data and then examine how the leading eigenvector-generated factors explain various option features.

Significance of data features

The sum of the eigenvalues forms the trace of the given matrix:

$$tr(A) = \sum_i s_i \qquad (1)$$

The sum of squared eigenvalues forms the trace of the matrix squared, a metric for variation within the data set:

$$tr(A^2) = \sum_i s_i^2 \qquad (2)$$

Thus, after performing Singular Value Decomposition, we can calculate the proportion of variation in the data table explained by each of the singular vectors as follows:

$$Var\ explained\ by\ s_i = s_i^2 / \sum_i s_i^2 \qquad (3)$$

Since the singular values are always ordered from greatest to the smallest, the first few singular values explain the largest proportion of variation of any dataset.

The k-th eigenfactor or singular factor can be computed as follows: for a given eigenvector *k*, multiply each eigenvector coefficient value *i* with the corresponding dataset column *i*. Then sum up the columns cross-sectionally to obtain the eigenfactor $F^{(k)}$:

$$F^{(k)} = \frac{1}{\sqrt{\lambda^{(k)}}} \sum_i V_i^{(k)} X_i \qquad (4)$$

Next, coefficients of a linear regression of the original data columns on each of the factors are calculated:

$$X_j = \beta_j F^{(k)} + \varepsilon_j \qquad (5)$$

The coefficients in such a factorization are known as factor loadings.

$$X_j = \beta_j \frac{1}{\sqrt{\lambda^{(k)}}} \sum_i V_i^{(k)} X_i + \varepsilon_j$$

The factor loadings $\beta_j$are standard linear regression coefficients and are computed as:

$$\beta_j = \sigma_{FX} / \sigma_F^2 \qquad (6)$$

where σ_X^2 is the variance of factor $F^{(k)}$ and $\sigma_{FX}$is the covariance between X_j and $F^{(k)}$:

$$\sigma_{FX} = E[F^{(k)} X_j] - E[F^{(k)}]\, E[X_j] \qquad (7)$$

Substituting the equation (4) for $F^{(k)}$, we see that

$$\sigma_{FX} = E\left[\left(\frac{1}{\sqrt{\lambda^{(k)}}} \sum_i V_i^{(k)} X_i\right) X_j\right] - E[F^{(k)}]\, E[X_j] \qquad (8)$$

By construction of the eigenfactors, the highest absolute-value factor loading will be obtained for the data feature with the highest absolute-value coefficient in a given singular vector *k.* The largest coefficient will be multiplied by its respective column and will dominate the other columns in the summation. Thus, the resulting eigenfactor will exhibit the highest correlation with the column which dominated the given eigenvector.

Simulation

To test our process, we construct a simulation. In the first example, we build a matrix *A* with 1,000 rows and just four columns. We number the columns here in the computer science notation, beginning with 0, instead of 1.

1. Independent Data Features

To start our experiment, we create a 1000 x 2 dataset with purely independent columns: a sinusoid where the argument is the row number divided by π and the second column is a randomly selected Gaussian noise:

$$A[i, 0] \ = \ sin(\frac{i}{\pi}) \qquad (1)$$

$$A[i, 2] \ \sim \ N(0, 1) \qquad (2)$$

Running a singular value decomposition (SVD) on this data set produces two singular values and two singular vectors. The number of singular values and vectors always corresponds to the number of features or columns in the data. Figure 1 and Table 1 show the singular values and the singular vectors.

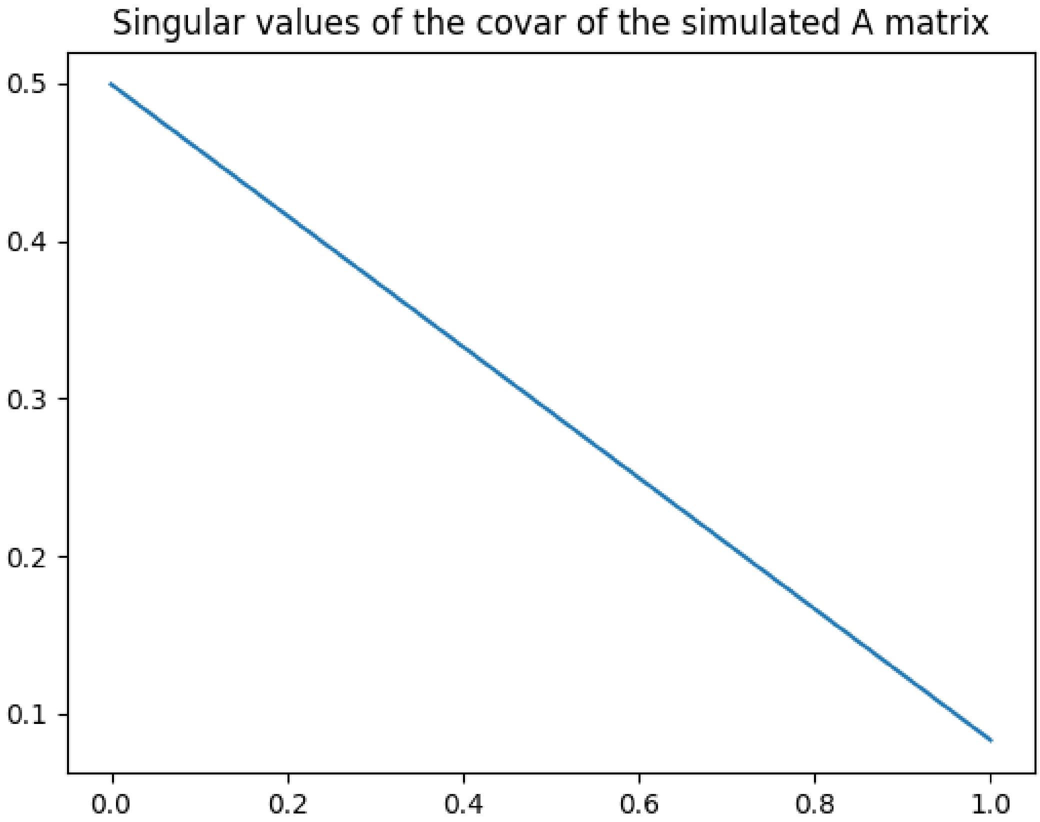

Figure 1. Singular values of a simulated data set *A* with $A[i, 0] \; = \; sin(\frac{i}{\pi})$ and $A[i, 2] \; \sim \; N(0, 1)$.

TABLE 1. Singular vectors of a simulated data set *A* with $A[i, 0] \; = \; sin(\frac{i}{\pi})$ and $A[i, 1] \; \sim \; N(0, 1)$. The numbers in bold show the highest absolute value coefficients.

Columns	V[0]	V[1]
$A[i, 0] \; = \; sin(\frac{i}{\pi})$	**-0.999**	0.020
$A[i, 1] \; \sim \; N(0, 1)$	-0.020	**-0.999**
% variation explained	60.3	39.7

var_explained = [0.601 0.399]

As shown in Table 1, the signal comes out loud and clear in the first eigenvector, dominating the noise. The coefficient for the signal in V[0] is -0.999, large in magnitude, especially in comparison with miniscule noise coefficient, -0.020.

If we were to select the most important data from the original data set, according to SVD, we would multiply the first singular vector by the original data set *A*. In doing that, all the entries of the first column of *A* would be multiplied by -0.999, and all the entries of the second column would be multiplied by 0.020, retaining the dominance of the signal.

The dominant features of the covariance of matrix *A* remain the same for the top eigen factors of the covariance decomposition following the POET method of Fan, Liao and Mincheva (2013).

To measure the correlation of our eigenvector-based factors with the original data set columns, we next create eigenfactors, per equation (4) above, repeated here for convenience:

$$F^{(k)} = \frac{1}{\sqrt{\lambda^{(k)}}}\sum_{i} V_{i}^{(k)} X_{i} \qquad (4)$$

In this case, our eigenfactor is a 1000 x 2 matrix. The correlations of the eigenfactors with the original two columns are as shown in Table 1:

TABLE 2. Correlation matrix of eigenfactors and the original data set A comprising a sinusoid signal and noise.

	A[:,0]: first data column	A[:,1]: second data column
F0: First eigenfactor	-0.9999	-0.0041
F1: Second eigenfactor	-0.0006	0.9999

As the correlation matrix in Table 2 shows, as expected, the sinusoid signal in the original data is most correlated with the first eigenfactor F0, which itself is dominated by the sinusoid signal as shown in Table 1. The noise is, in turn, highly correlated with the noise column in the original dataset.

2. Collinear Data Features

Next, we intentionally construct a data set with collinear values. The first column is our first "signal": a sinusoid with the argument representing a function of the row number:

$$A[i, 0] \; = \; sin(\frac{i}{\pi}) \qquad (1)$$

The second column is just the first column amplified five times with Gaussian noise added:

$$A[i, 1] \; = 5 \, * \, A[i, 0] \; + \; \epsilon, \; \epsilon \sim N(0, 1) \qquad (2)$$

The third column contains a randomly selected Gaussian number:

$$A[i, 2] \; \sim N(0, 1) \qquad (3)$$

And the fourth column is the third column multiplied by 3:

$$A[i, 3] \; = 3 \, * \, A[i, 0] \qquad (4)$$

Running a singular value decomposition (SVD) on matrix A produces singular values plotted in Figure 2 and the singular vectors shown in Table 3. There are four singular values, each corresponding to each column in the matrix. Likewise, there are also four singular vectors.

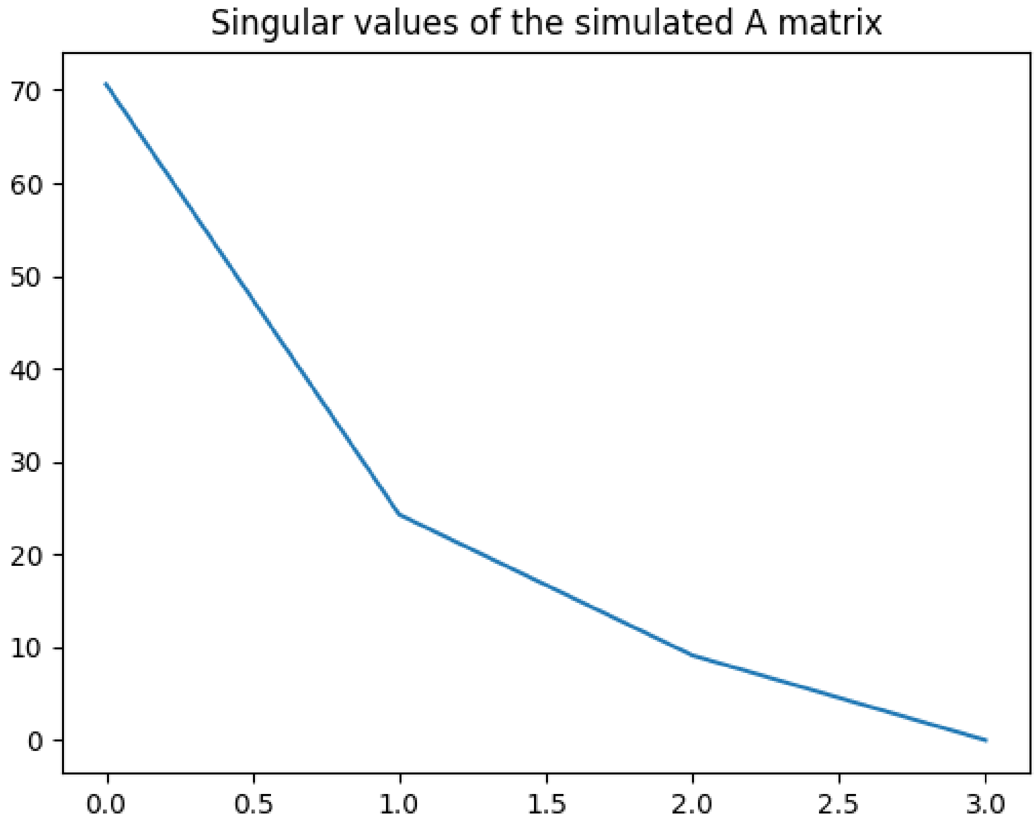

Figure 2. Four singular values of the simulated matrix *A* where the columns comprise a sinusoid, an amplified sinusoid with noise, Gaussian noise and the amplified sinusoid.

TABLE 3. Singular vectors of the simulated matrix *A* where the columns comprise a sinusoid, an amplified sinusoid, Gaussian noise and same noise amplified. Numbers in bold correspond to the largest absolute value coefficient for each singular vector V[j].

Column	V[0]	V[1]	V[2]	V[3]
$A[i,0] \ = \ sin(\frac{i}{\pi})$	-0.316	-0.371	-0.001	**-0.949**
$A[i,1] \ = 5 * A[i,0] \ + \ \epsilon,$ $\epsilon \sim N(0,1)$	-0.006	**0.702**	**-0.712**	0.000
$A[i,2] \ \sim N(0,1)$	-0.009	**0.712**	**0.702**	-0.000
$A[i,3] \ = 3 * A[i,0]$	**-0.949**	-0.011	-0.002	0.316
% variation explained	88.1%	10.4%	1.5%	0.0%

As Table 3 shows, the technique deems the pure amplified sinusoid to be the dominant signal in the data, as it has the highest coefficient in the first eigenvector V[0]. Pure Gaussian noise comes out on top in the first eigenvector. The noisy amplified sinusoid comes third, dominating V[2]. Finally, the original sinusoid signal appears dominant in the last eigenvector.

Considering the results in Table 3, we note that SVD ignores or diminishes collinear data features in the analysis results. Thus, the first singular vector primarily uses the amplified sinusoid, while the collinear sinusoid is the major feature of the last singular vector, typically discarded in the global factor analysis. (It may be kept in an idiosyncratic cross-sectional analysis of data.) The second singular vector is focused on the amplified noise.

The correlation of eigenfactors computed according to equation (4) and the respective columns of the data matrix A are shown in Table 4:

TABLE 4. Correlations of eigenvector-driven eigenfactors and the columns of the original data matrix A.

	A[:,0]	A[:,1]	A[:,2]	A[:,3]
F0	**-0.9999**	-0.0308	-0.0580	**-0.9999**
F1	0.0005	**-0.9995**	-0.0002	0.0005
F2	0.0009	0.0002	**-0.9983**	0.0009
F3	**0.9829**	0.0079	0.0468	**0.9829**

The correlation matrix shows that, as predicted, the eigenfactors are nearly perfectly correlated with the data columns of A that featured prominently in the eigenvectors corresponding to the eigenfactors. For instance, the most significant factor in the first eigenvector F0 was the amplified sinusoid. It shows up with nearly perfect, yet negative, correlation in the first eigenfactor along the original sinusoid (since both are perfectly correlated with each other!). The pure and the amplified sinusoids also appear dominant in the last eigenfactor, F3, that often represent the idiosyncratic features of the dataset. The second eigenvector, V[1], had two comparable dominant features: the amplified sinusoid corrupted by noise and pure Gaussian noise. As the factor table 3 shows, it is the corrupted sinusoid that turns out to have the highest correlation with the second eigenfactor F1 based on the second eigenvector V[1]. The pure noise comes out ahead in the third eigenfactor, F2. The third eigenfactor is roughly in the middle of the dataset and is likely to be discarded by analyses along with the second eigenfactor, F1.

To make sure that our results in Figure 2 and Table 3 are not accidental, we rearrange the order of columns and rerun the analysis. Figure 3 and Tables 4 and 5 show the results.

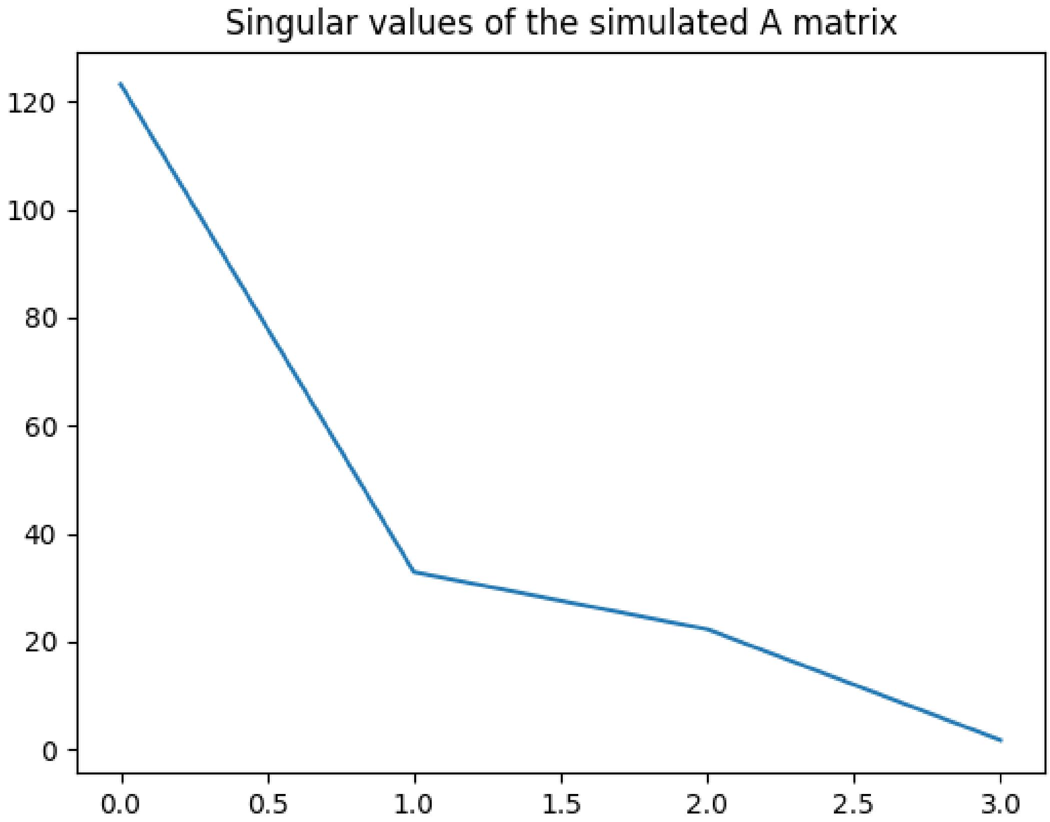

Figure 3. Four singular values of the simulated matrix *A* where the columns comprise an amplified and randomized sinusoid, the original sinusoid and pure Gaussian noise.

TABLE 5. Singular vectors of the simulated matrix *A* where the columns comprise an amplified and randomized sinusoid, pure Gaussian noise, the original sinusoid, and Gaussian noise amplified. Numbers in bold correspond to the largest absolute value coefficient for each singular vector V[j].

Column	V[0]	V[1]	V[2]	V[3]
$A[i,0] = 5 * A[i,2] + \epsilon$, $\epsilon \sim N(0,1)$	-0.231	0.006	**-0.973**	-0.000
$A[i,1] \sim N(0,1)$	-0.308	-0.001	0.073	**-0.949**
$A[i,2] = sin(\frac{i}{\pi})$	0.003	**-0.999**	-0.007	-0.000
$A[i,3] = 3 * A[i,1]$	**-0.923**	-0.005	0.219	-0.316

% variation explained	84.7%	11.8%	3.4%	0.0%

In Table 5, the amplified noise trumped the signal in the first eigenvector, just as it drowns the signal in many spike models (see, for example, Johnstone (2001), Bail, Ben-Arous and Peche (2005), Karoui (2005), Baik and Silverstein (2005), Paul (2007), Benaych-Georges and Nadakuditi (2011), and Benaych-Georges and Nadakuditi (2011), and summarized in Aldridge and Avellaneda (2021).

Table 6 shows correlations of corresponding eigenfactors with the original data columns. The first eigenfactor F0 is nearly perfectly correlated with both Gaussian noise and its amplified version. The second eigenfactor F1 is in nearly perfect synchronization with the original sinusoid. The third eigenfactor F2 is dominated by the amplified and randomized sinusoid, and, finally, the last eigenfactor F3 representing idiosyncratic features is once again attuned to the pure sinusoid.

TABLE 6. Correlations of eigenvector-driven eigenfactors and the columns of the data matrix A.

	A[:,0]	A[:,1]	A[:,2]	A[:,3]
F0	-0.0674	**0.9994**	-0.0586	**0.9994**
F1	-0.0480	0.0329	**0.9983**	0.0329
F2	**-0.9966**	-0.0064	-0.0074	-0.0064
F3	0.0374	0.0239	**-0.9984**	0.0239

Similar to the analysis of uncorrelated data, the dominant features of the covariance of collinear matrix *A* remain the same for the top eigen factors of the covariance decomposition following the POET method of Fan, Liao and Mincheva (2013).

Empirical Results

To test our factorization approach on real data, we construct two toy models to illustrate the principle. In the first example, we examine the relationship between U.S. Consumer Confidence Index (CCI) and the S&P 500 ETF returns (NYSE: SPY) on a monthly basis. In the second example, we add E-mini crude oil futures data to link the monthly oil prices into the model. We obtain a profitable prediction for one-month-ahead E-mini crude oil futures trading. However, the CCI data is proving to be largely irrelevant in price prediction, at least on the month-to-month horizon.

Model I

We construct our model first by including just the Consumer Confidence Index (CCI) and the S&P 500 ETF (NYSE:SPY) observations in the same table. The SPY data are included to approximate for the broad market. We include the U.S. CCI data from The Organisation for Economic Co-operation and Development (OECD). OECD collects and reports CCI figures monthly. The U.S. CCI data are available from January 1960. The SPY data come from Yahoo! Finance and are available beginning February 1993.

We create a “mixed bag” of contemporaneous, not lagged, data comprising both CCI and SPY data. In addition to base values of reported CCI levels, we compute and include monthly percent change in CCI. Similarly, in addition to the monthly level data (Open, High, Low, Close, Adjusted Close and Volume) for SPY we include the monthly return on the SPY based on the Adjusted Close figures.

Even though the lengths of the data samples for CCI and SPY do not match, we do not delete any observations from the CCI time series. Instead, we fill in the SPY series with zeros from 1960 through January 1993, where the SPY observations are not available. We do not include any dividend information for the SPY.

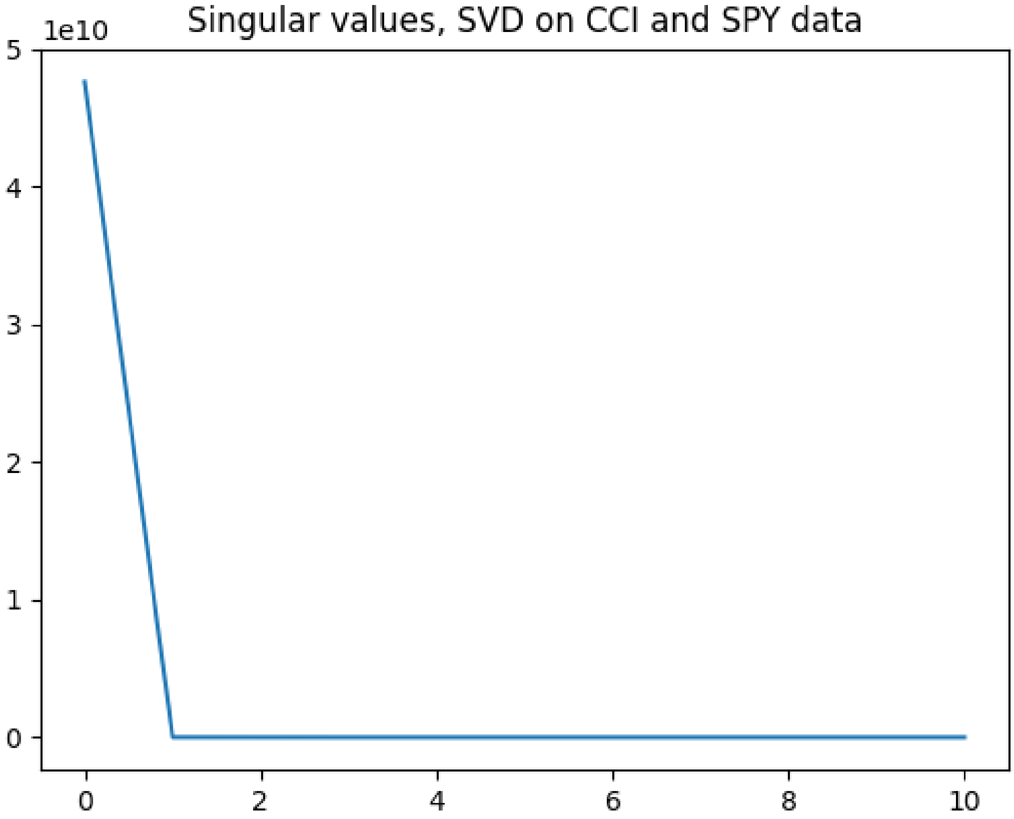

Figure 1. Singular values of SVD comprising CCI and SPY data.

Table 1 shows the eigenvectors of the SVD decomposition of CCI and SPY data. Each vector contains coefficients that, when multiplied by data in their respective columns and added together in a linear combination, form orthogonal axes or bases for the data, also known as Principal Components (PCs) of the data.

These orthogonal bases or PCs usually stump researchers as they can be difficult to interpret. What, for example, does a linear combination of the Date column and CCI Return mean? Most researchers believe that at this point they need to come up with factors outside of the given data set to find as close a match for the entire eigenvector basis. However, the identified correlations in many studies may be reasonably small in magnitude. For example, Muravyev, Vasquez and Wang (2018) find that the first principal component (PC) in their options data has a 66% correlation with the call-put volatility spread and the second PC has a -39% correlation with the option skew.

However, just eyeballing the data, we notice that each vector has very strong coefficients for specific columns of the original data. These coefficients make certain columns dominant in each eigenvector. If we were to take the resulting eigenvectors one by one and and correlate them with the underlying data columns, the magnitude of the correlations will be highest with the columns that have the highest coefficient in the eigenvector.

As shown in Table 1, the largest-magnitude coefficients of the the eigenvectors are: V[0]: SPY Volume, V[1]: Date (YYYYMM), V[2]: SPY Date (YYYYMM), V[3] and V[4]: SPY Adj Close**, V[5]: SPY Low, V[6]: SPY Close*. CCI Level finally appears as the dominant feature in V[7]. Not shown in Table 1 are the coefficients for the remaining eigenvectors V[8]: SPY High, V[9]: Monthly SPY Return, and finally V[10]: Monthly CCI Return.

Why is Date (YYYYMM) such a significant factor and why does it have an even higher significance than SPY Date (YYYYMM)?

Table 1. Eigenvectors from SVD of CCI and SPY Data. (The largest coefficient of each vector is in bold).

Feature	V[0]	V[1]	V[2]	V[3]	V[4]	V[5]	V[6]	V[7]
Date YYYYMM	-5.4e-05	**9.3e-01**	3.7e-01	2.2e-07	-3.5e-05	-1.2-05	2.4e-05	4.9e-04
CCI Level	-2.6e-08	4.7e-04	1.9e-04	-4.4e-04	6.9e-02	2.3e-02	-4.8e-02	**-9.8e-01**
Monthly CCI Return	8.9e-14	3.7e-10	-1.4e-09	-1.8e-07	-4.3e-05	-1.3e-04	8.6e-05	-1.1e-04
SPY Date YYYYMM, zero-filled till 199302	-5.4e-05	3.7e-01	**-9.3e-01**	-1.5e-03	-1.5e-04	4.6e-05	3.6e-05	-9.4e-06
SPY Open	-4.3e-08	2.5e-04	-6.4e-04	4.4e-01	4.7e-01	3.3e-01	5.3e-01	-6.9e-02
SPY High	-4.4e-08	2.6e-04	-6.6e-04	4.5e-01	3.0e-01	4.3e-01	-4.0e-01	1.6e-01
SPY Low	-4.1e-08	2.5e-04	-6.3e-04	4.3e-01	1.2e-01	**-7.4e-01**	3.3e-01	4.6e-02
SPY Close*	-4.3e-08	2.6e-04	-6.5e-04	4.4e-01	-4.3e-03	-3.1e-01	**-6.4e-01**	-7.8e-02
SPY Adj Close**	-3.7e-08	2.1e-04	-5.2e-04	**4.7e-01**	**-8.2e-01**	2.4e-01	2.0e-01	-5.7e-02
SPY Volume	**-9.9e-01**	-7.0e-05	3.0e-05	-1.3e-08	3.0e-10	-1.2e-09	-1.7e-10	-7.2e-10
Monthly SPY Return	1.4e-13	3.4e-08	-8.6e-08	2.0e-05	-2.4e-03	-3.1e-03	-6.0e-03	-1.1e-04

Indeed, correlation of the principal components with the respective data columns (features) was nearly identical to the columns' coefficients in the input data table, as shown in Table 2.

Eigenvector	Highest eigenvector coefficient in Table 1	Original Data Feature (column) with the highest eigenvector coefficient in Table 1	Correlation of the original data feature and the PC generated as the dot product of the original data table and the Eigenvector
0	-9.9e-01	SPY Volume	-1
1	9.3e-01	Date (YYYYMM)	-0.36797078
2	-9.3e-01	SPY Date YYYYMM, zero-filled till 199302	-0.86254635
3	4.7e-01	SPY Adj Close**	0.67475104
4	-8.2e-01	SPY Adj Close**	-0.05729275
5	-7.4e-01	SPY Low	-0.03049728
6	-6.4e-01	SPY Close*	-0.01829721
7	-9.8e-01	CCI Level	-0.77454226
8	-5.7e-01	SPY High	-0.00673263
9	9.9e-01	Monthly SPY Return	0.49381423
10	9.9e-01	Monthly CCI Return	0.95371259

As Table 2 shows, a data feature with the largest coefficient in a given eigenvector i does not always produce PC[i] and feature correlation equal to the coefficient. For example, in VT[0], the largest coefficient is -1 for SPY Volume. The correlation of the PC[0] with the SPY Volume column also happens to be -1! In the case of VT[3], on the other hand, the largest coefficient is 0.47 for SPY Adj Close, while the correlation of the PC[3] and SPY Adj Close is 67%. However, in all the cases in the example of Table 2, the signs of the coefficients and the respective correlations match up perfectly.

What if features do have high correlation with some PCs that our methodology above fails to pick up? Table 3 shows the correlation matrix between all the PCs and all the features.

TABLE 3. Correlation of Data Features and Data PCs

PCs: Features	0	1	2	3	4	5	6	7	8	9	10
Date YYYYMM	-62%	-37%	-67%	25%	-2%	-3%	2%	20%	4%	2%	3%
CCI Level	36%	46%	-36%	11%	24%	3%	-7%	-77%	-11%	1%	2%
Monthly CCI Return	7%	8%	-4%	-1%	-11%	-20%	9%	-6%	-2%	14%	95%
SPY Date YYYYMM, zero-filled till 199302	-57%	-26%	-86%	0%	-0%	-0%	0%	0%	0%	0%	0%
SPY Open	-53%	-29%	-66%	58%	3%	1%	2%	0%	1%	0%	0%
SPY High	-53%	-29%	-66%	58%	2%	2%	-1%	0%	-0%	0%	0%
SPY Low	-52%	-27%	-66%	58%	1%	-3%	1%	0%	-1%	0%	0%
SPY Close*	-52%	-28%	-66%	58%	-0%	-1%	-2%	0%	0%	0%	0%
SPY Adj Close**	-50%	-28%	-58%	67%	6%	1%	1%	0%	0%	0%	0%
SPY Volume	-100%	-94%	-8%	0%	-0%	-0%	0%	0%	0%	0%	0%
Monthly SPY Return	9%	17%	-25%	8%	-48%	-38%	-54%	-0%	2%	49%	-0%

While some factors, like CCI return, may appear significant in the later eigenvectors, they are generally not considered significant for the data set. As discussed in Aldridge and Avellaneda (2021), the eigenfactor significance can be separated into that important for the entire data set ("macro-significance") and that specific to individual observations (idiosyncratic or "micro-significance"). The macro-significant factors are found in the first one or several eigenvectors, while the micro-significant factors comprise the very last few eigenvectors. The cut-off number of which vectors fall into the macro- or micro sets usually falls around "the elbow" on the singular value plot. From Figure 1, the first eigenvector appears "before the elbow" in the global significance. More precisely, the cut-off can be determined using the Marcenko-Pastur method.

It is important to point out that prior to SVD factorization outlined below, we tested CCI data, SPY prices and E-mini oil futures prices in the linear and XGBoost frameworks and obtained no significant relationships or predictability. In our linear and XGBoost modeling, we did not try SPY volume as a factor. The factorization process above allowed us to identify truly important factors, including SPY volume, from the pool of all available variables instead of relying just on our own priors and their limitations.

Building Factor-Based Trading Strategies

Building a strategy from the identified factors can be simple or complex. Here, we opt for the simplest approach to illustrate a point. Since in SPY volume is identified to be the top factor and the factor coefficient within the first eigenvector is negative, we sell when the volume rises and buy when the volume falls. Specifically, we construct a simple strategy framework the pseudocode for which looks like this:

1) If SPY Volume today > mean(SPY Volume over the past 5 days) - stdev(SPY Volume over the past 5 days), SELL SPY today at close, buy it back tomorrow at close
2) If SPY Volume today < mean(SPY Volume over the past 5 days) + stdev(SPY Volume over the past 5 days), BUY SPY today at close, sell it back tomorrow at close to bring the position back to 0

The cumulative performance of such a basic strategy is illustrated in Figure 2. As Figure 2 shows, the strategy delivers positive results, but underperforms the simple buy-and-hold strategy of SPY. The two strategies come very close in terms of Sharpe Ratio: simple SPY buy-and-hold produces 2.36, while our SPY volume-based strategy delivers Sharpe of 2.30 over the same period of time.

Performing a similar analysis on the CL E-mini futures data, we obtain a significant improvement in Sharpe ratio vis-a-vis the base case of CL E-mini futures. While the simple buy and hold of E-minis generated the Sharpe ratio of 0.6616, the volume-improved strategy produced Sharpe of 2.2484.

To generate this strategy, in addition to tracking SPY Volume as above, we add the condition for CL E-mini futures Volume. Since the CL E-mini futures volume coefficient is positive, we buy when the change in volume is positive and sell when it is negative, opposite to our treatment of SPY volume. The exact conditions are as follows:

1) If SPY Volume today > mean(SPY Volume over the past 5 days) - stdev(SPY Volume over the past 5 days) AND if CL Volume today < mean(CL Volume over the past 5 days) + stdev(CL Volume over the past 5 days), SELL CL E-mini futures today at close, buy back tomorrow at close.
2) If SPY Volume today < mean(SPY Volume over the past 5 days) + stdev(SPY Volume over the past 5 days) AND if CL Volume today > mean(CL Volume over the past 5 days) - stdev(CL Volume over the past 5 days), BUY CL E-mini futures today at close, sell it back tomorrow at close to bring the position back to 0.

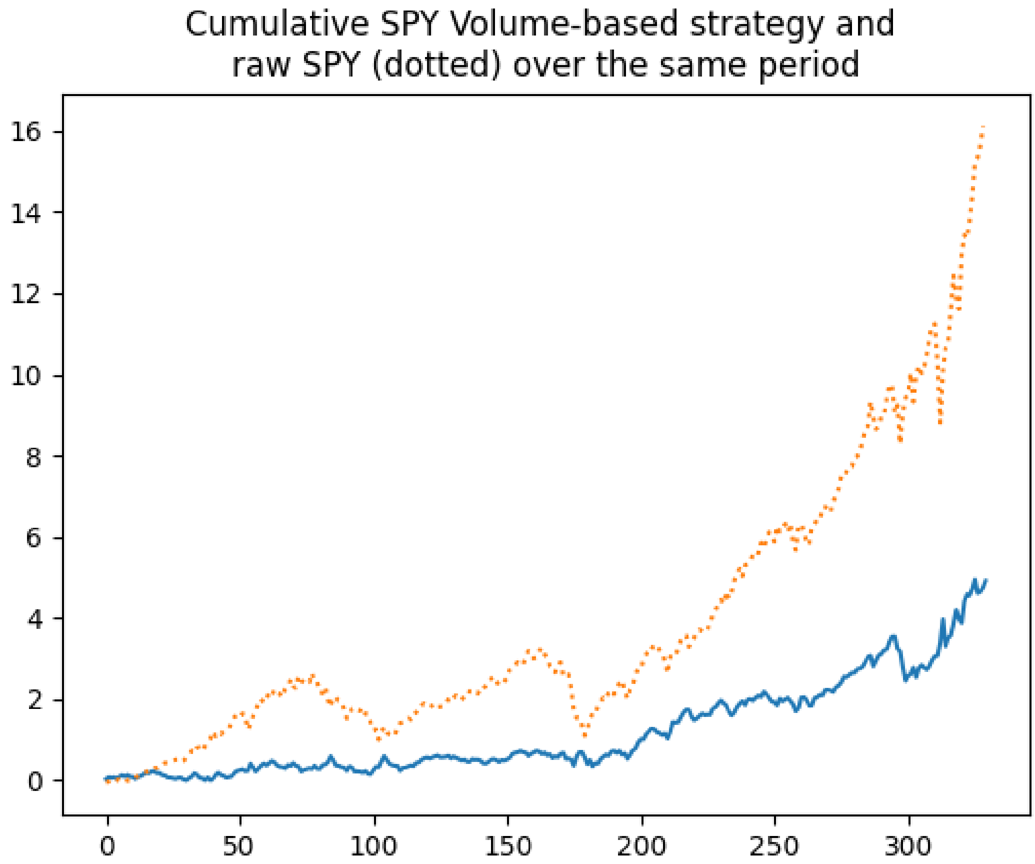

Figure 2. Cumulative SPY Volume-based strategy for SPY and raw SPY over the same period of time.

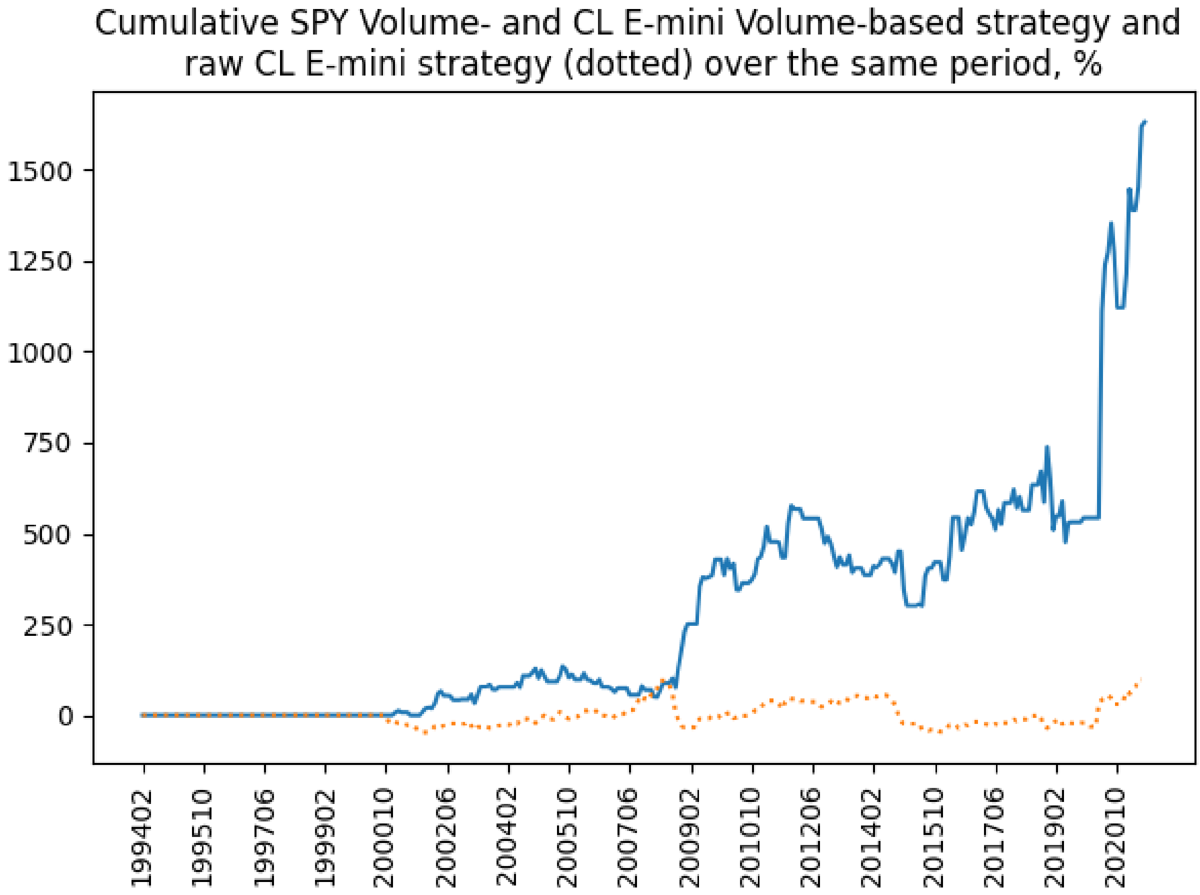

Figure 3. Cumulative performance of the strategy based on SPY volume and CL E-mini volume.

Conclusions

Our proposed factor interpretation provides a quick and efficient framework for sorting out key drivers within datasets. The framework is particularly effective in datasets with many columns (features) since the richer the data, the more likely we are to find the true explanatory variables driving the dataset. By selecting just the most prominent component within each resulting factor, we absolve any collinearity issues and are able to generate profitable predictive models.

References

Aldridge, I. and M. Avellaneda, 2021. “Big Data Science in Finance.” ISBN: 978-1119602989, Wiley & Sons: Hoboken, NJ.

Avellaneda, M. and J.-H. Lee, 2010. “Statistical Arbitrage in the US Equities Market.” *Quantitative Finance* 10(7): 761-782.

Bai, J. and S. Ng, 2013. “Principal component estimation and identification of static factors.” *Journal of Econometrics* 17: 18-29.

Baker, M. and J. Wurgler, 2006. “Investor sentiment and the cross-section of stock returns.” The Journal of Finance, 61(4), 1645-1680.

Christoffersen, Peter, Mathieu Fournier and Kris Jacobs, 2018. “The Factor Structure in Equity Options.” The Review of Financial Studies, Volume 31, Issue 2, Pages 595–637.

Connor, G., M. Hagmann, and O. Linton, 2012. “Efficient Semi-Parametric Estimation of the Fama-French Model and Extensions.” *Econometrica* 80(2): 713-754.

Doz, C., D. Giannone and L. Reichlin, 2011. “A two-step estimator for large approximate dynamic factor models based on Kalman filtering.” *Journal of Econometrics* 164: 188-205.

Fan, J., Y. Liao and M. Mincheva, 2013. “Large Covariance Estimation by Thresholding Principal Orthogonal Components. *Journal of the Royal Statistical Society, Series B, Statistical Methodology* 1:75.

Giglio, S., and D. Xiu, 2019. “Asset Pricing with Omitted Factors.” Chicago Booth Research Paper No. 16-21.

Kelly, B., S. Pruitt and Y. Su, 2017. “Instrumented Principal Component Analysis.” Working Paper.

Kilian, L., 2009. “Not all oil price shocks are alike: disentangling demand and supply shocks in the crude oil market.” American Economic Review, 99 (2009), pp. 1053-1069

Kilian, L. and C. Vega, 2011. “Do energy prices respond to the U. S. macroeconomic news? A test of the hypothesis of predetermined energy prices.” Review of Economic Statistics, 93 (2011), pp. 660-671.

Laloux, L., P. Cizeau, M. Potters and J.-P. Bouchaud, 2000. “Random Matrix Theory and Financial Correlations.” *Mathematical Methods in Applied Sciences* 1(2): 217-222.

Lintner, J., 1965. “The valuation of risk assets on the selection of risky investments in stock portfolios and capital budgets.” Review of Economics and Statistics 47: 13-37.

Muravyev, D., Vasquez, A., and Wang, W., 2018. “Making better use of Options to Predict Stock Returns.” Working paper, Boston University.

Ross, Stephen A. 1976. “The Arbitrage Theory of Capital Asset Pricing.” Journal of Economic Theory. 13:3, pp. 341–60.

Sharpe, William F. 1964. “Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk.” Journal of Finance. 19:3, pp. 425– 42.

Stock, J.H. and M.W. Watson, 2002. “Forecasting using principal components from a large number of predictors, *Journal of the American Statistical Association* 97: 1167-1179.

Zafeiriou, E., C. Katrakilidis and C. Pegiou, 2019. “Consumer Confidence on Heating Oil Prices: An Empirical Study of their Relationship for the European Union in a Nonlinear Framework.” European Research Studies Journal, Volume XXII, Issue 1, pp. 63-90.