Singular-Spectrum Analysis for Financial Time Series:
Identifying Trend, Periodicity, and Extracting Noise

Franklin Williams
https://github.com/streater512/AMATH-582
March 19, 2020

Abstract
Singular-Spectrum Analysis (SSA) decomposes a time series into components, that when summed together, recreate the original time series. Components that share similar features can be grouped together and interpreted as the trend, periodicity, and noise of the time series. Although not widely known among statisticians and econometrists, SSA has become a standard tool in meteorology and climatology; it is also a well-known technique in nonlinear physics and signal processing. (Golyandina) In this report, SSA is used for exploratory analysis of three selected financial time series.

Section I: Introduction
This report begins with an introduction to the practicality of SSA as well as an overview of data collected and experiments run. In Section II, a theoretical background of the techniques used will be discussed. Section III covers the algorithm implementation and development in Python. Section IV provides the computational results of our tests. To conclude, Section V summarizes the results and ends with final remarks.

Unlike traditional tools popular within the quantitative finance community such as, the auto regressive integrated moving average (ARIMA) model, SSA is essentially a model-free technique. Another useful property is that SSA does not make any assumptions on the signal provided to the algorithm. Like other popular models, the ARIMA model requires a stationary signal for analysis. It is exceptionally rare to find a raw stationary signal within the field of finance. Because of this, data is often manipulated to force a signal to meet certain stationarity assumptions. One method of manipulation is through differencing. For example, rather than modeling the price of a publicly traded security directly, analysts will model the return patterns of the security. While the return patterns exhibit the characteristics necessary for the model to function, information is lost in the manipulated series. Calculating the return series of the price is akin to imposing a bandpass filter on a raw signal which only accepts frequencies within a designated interval.

The main use of SSA is for exploratory analysis by decomposing a signal into slow-moving trends, oscillatory components, and noise. In this report, three time series were downloaded via the Federal Reserve Economic Data (FRED) API provided by the St. Louis Fed. Data included daily observations of the 10-Year U.S. Treasury Constant Maturity Rate (DGS10), 1-Year U.S. Treasury Constant Maturity Rate (DGS1), and U.S. VIX Index (VIXCLS). For each time series, a trajectory matrix was constructed and singular-value decomposition (SVD) analysis was performed on the trajectory matrix. With the outputs from the SVD analysis, elementary matrices were created. Principal-component analysis (PCA) was performed to illustrate the relative contribution of each elementary matrix to the trajectory matrix. Elementary matrices with similar characteristics are then grouped together into three categories: trend, periodic, and noise. Through diagonal averaging, each elementary matrix can be reconstructed into a time-series. The summation the components within the trend and periodic components create the true signal of the time series, according to SSA. The signal and the noise for each time series are generated and reviewed for exploratory purposes.

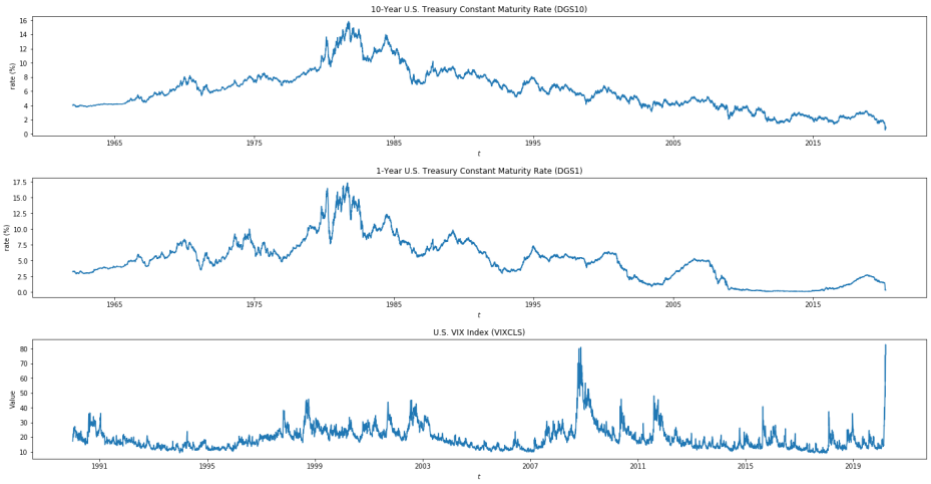

Figure (1): Raw time series: DGS10, DGS1, VIXCLS

Section II: Theoretical Background

SSA is built on the underlying theory described in Singular Value Decomposition (SVD). SVD is a factorization of a matrix into a number of constitutive components all of which have a specific meaning in applications (Kutz). In linear algebra, matrix multiplication can be thought of as a rotation and a stretch (compression) of a given vector. In the case of **y**=**Ax** which projects a hypersphere onto a hyper ellipse, the matrix **A** can be rewritten as:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{U\Sigma V}^{*} \\ \boldsymbol{U} &\in \mathbb{C}^{m\times m} \text{ is unitary} \\ \boldsymbol{V} &\in \mathbb{C}^{n\times n} \text{ is unitary} \\ \boldsymbol{\Sigma} &\in \mathbb{R}^{m\times n} \text{ is diagonal} \end{aligned} \tag{1}$$

The SVD of the matrix **A** shows the matrix first applies a unitary transformation preserving the unit sphere via $\mathbf{V}^{*}$. Following, **Σ** stretches the matrix to create an ellipse with principal semi-axes. Finally, the generated hyper-ellipse is rotated by **U**. Every matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ has an SVD. Two particularly important properties of SVD are as follows:

A is the sum of r rank-one matricies

$$\boldsymbol{A} = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^{*} \tag{2}$$

For any N so that $0 \leq N \leq r$, we can define the partial sum

$$\boldsymbol{A}_N = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^{*} \tag{3}$$

And if N = min{m, n}, define $\sigma_{N+1} = 0$. Then

$$\|A - A_N\|_2 = \sigma_{N+1} \tag{4}$$

Likewise, if using the Frobenius norm, then

$$\|A - A_N\|_2 = \sqrt{\sigma_{N+1}^2 + \sigma_{N+2}^2 + \cdots + \sigma_r^2} \tag{5}$$

Following this logic, we can generate the best approximation of a hyper-ellipsoid with a line segment by taking the longest axis (the one associated with the singular value σ_1). We can then generalize into the best approximation for 2- or *n*-dimensional ellipse by taking the longest 2 or *n* number of axes. After *r* number of steps, the energy within **A** is completely captured. The SVD allows us to project the high dimensional space onto lower dimensional spaces in a formal way similar to the least-squares fit common in statistics.

For SSA, SVD is performed on the trajectory matrix, **X**, created from the raw time series, $\boldsymbol{F}$. To map $\boldsymbol{F}$ into a sequence of multi-dimensional lagged vectors, an integer $\boldsymbol{L}$, is chosen with window length, $2 \leq L \leq \frac{N}{2}$, where *N* denotes the number of observations within $\boldsymbol{F}$. **X** is composed of column vectors containing the first L observations, observations 2 through L+1, etc. The trajectory matrix is shown below:

$$\boldsymbol{X} = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_{N-L} \\ f_2 & f_3 & f_4 & & f_{N-L+1} \\ & \vdots & & \ddots & \vdots \\ f_L & f_{L+1} & f_{L+2} & \cdots & f_N \end{bmatrix} \tag{6}$$

Diagonal averaging is used to reconstruct the time series, $\widetilde{\boldsymbol{F}}_i$, as averages of the corresponding anti-diagonals of $\boldsymbol{X_i}$. This is done through the Hankelisation operator, $\widehat{\boldsymbol{\mathcal{H}}}$, that operates on $\boldsymbol{X_i}$ to generate $\widetilde{\boldsymbol{X}}_i$,

$$\widetilde{\boldsymbol{X}}_i = \widehat{\boldsymbol{\mathcal{H}}}\boldsymbol{X_i} \tag{7}$$

The Hankelisation operator assigns values for $\widetilde{\boldsymbol{X}}_i$ through the following:

The element $\tilde{x}_{m,n}$ in $\widetilde{\boldsymbol{X}}_i$, for $s = m + n$, is given by:

$$\tilde{x}_{m,n} = \begin{cases} \dfrac{1}{\mathrm{s}} \displaystyle\sum_{l=1}^{s} x_{l,s-l}\,;\ 1 \leq s \leq L-1 \\ \dfrac{1}{\mathrm{L}} \displaystyle\sum_{l=1}^{L} x_{l,s-l}\,;\ L \leq s \leq K-1 \\ \dfrac{1}{\mathrm{K+L-s}} \displaystyle\sum_{l=s-K}^{L} x_{l,s-l}\,;\ K \leq s \leq K+L-2 \end{cases} \tag{8}$$

Because the trajectory matrix is already a Hankel matrix, it can be expressed in terms of its Hankelised elementary matrices, where *d* is the rank of the trajectory matrix:

$$\boldsymbol{X} = \sum_{i=1}^{d} \widetilde{\boldsymbol{X}}_i \tag{9}$$

To quantify which time series should be grouped together, a weighted correlation matrix is created from the reconstructed time series, $\widetilde{\boldsymbol{F}}_i, \widetilde{\boldsymbol{F}}_j$:

$$W_{i,j} = \frac{(\widetilde{\boldsymbol{F}}_i, \widetilde{\boldsymbol{F}}_j)_w}{\|\widetilde{\boldsymbol{F}}_i\|_w \|\widetilde{\boldsymbol{F}}_j\|_w} \tag{10}$$

Where $(\widetilde{\boldsymbol{F}}_i, \widetilde{\boldsymbol{F}}_j)_w$ is defined as the weighted inner product:

$$(\widetilde{\boldsymbol{F}}_i, \widetilde{\boldsymbol{F}}_j)_w = \sum_{k=1}^{N} w_k \tilde{f}_{i,k} \tilde{f}_{j,k} \tag{11}$$

Where $\tilde{f}_{i,k}$and $\tilde{f}_{j,k}$ are the *k*th values of $\widetilde{\boldsymbol{F}}_i$ and $\widetilde{\boldsymbol{F}}_j$, and w_k is given by:

$$w_k = \begin{cases} k, & 0 \le k \le L-1 \\ L, & L \le k \le K-1 \\ N-k, & K \le k \le N-1 \end{cases} \tag{12}$$

Finally, $\left\|\widetilde{\boldsymbol{F}}_k\right\|_w = \sqrt{(\widetilde{\boldsymbol{F}}_k, \widetilde{\boldsymbol{F}}_k)_w}$ for $k = i, j$.

Section III: Algorithm Implementation and Development

Data were taken from the FRED database through their built-in API. To gain access to the API, a request can be submitted through the help page and a free key is generated for the end user. The data are slightly modified to exclude any dates where the securities did not trade. Roughly 95% of tradable days remained for each of the time series. DGS10 and DGS1 contain 14536 observations spanning from 1962 through 2020. The VIXCLS time series is shorter, ranging from 1990 through 2020.

A convenient Python class was developed by D'Arcy in his Kaggle notebook "Introducing SSA for Time Series Decomposition." The class takes in two parameters, the time-series like object to be analyzed (in the form of a pandas or numpy series) and the window length, L, to be used for decomposition. The trajectory matrix is formed using the algorithm described in Section II. Using numpy's built in SVD tool, the matrices U, S, and V are found. These inputs are used to generate the various elementary matrices. The Hankelisation operator is built into the reconstruct function within the class. This takes the elementary matrices and generates the reconstructed time series to be analyzed. Using these Hankelised time series, the weighted correlation table is constructed following the above definitions.

Section IV: Computational Results

Figure (2) illustrates the trajectory matrix generated from each of the three time series to be decomposed. The highlighted streaks running diagonally through each matrix correspond to the highest observations within the respective time series. While necessary to conduct SSA, the trajectory matrix does not on its own reveal a large amount of information on the components of the time series. In order to denote trends, periodicity, and noise, we must inspect the elementary matrices that when combined formed the below trajectory matrices.

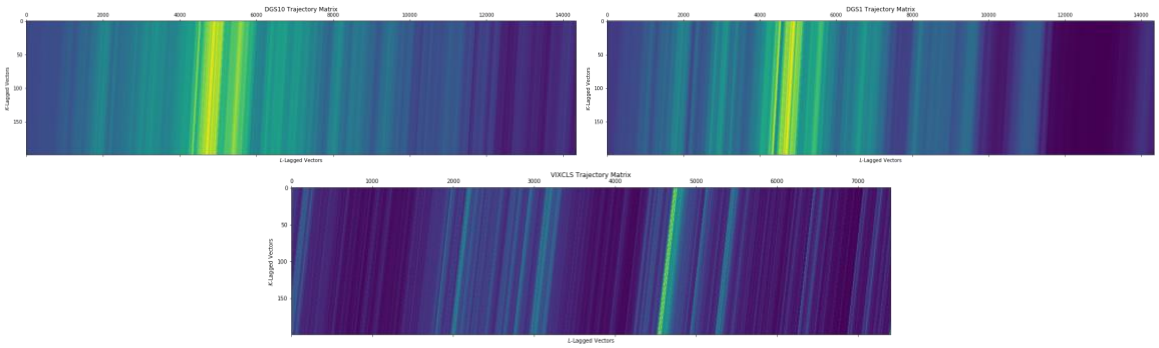

Figure (2) Top row left to right: Trajectory Matrix for DGS10 and DGS1.
FBottom row: Trajectory matrix for VIXCLS

In all three examples, the first elementary matrix plays the largest role in reconstructing the trajectory matrix. This can be seen in both the illustrations of the first ten elementary matricies and is quantified through PCA analysis. In each set of elementary matrices, there is a clear distinction between the first elementary matrix and the remaining. In each of the first graphics the pulse that appears across the illustration can be interpreted as the overal trend the time series follows. Inspection of the second matrix for each of the time series appears to show rapid oscillations in a steady manner. These are interpreted as the short term periodicity of the timeseries. No discernable pattern can be found in the remaining elementary matrices which is interpreted as the nosie components of the time series.

To further quantify these interpretations, PCA analysis is conducted on the trajectory matrix. The inferences of the elementary matrices are confirmed. The first componenet of the DGS10 and DGS1 trajectory matrix contributes to nearly all of the variance, roughly 99.6% in both cases. While in the case of VIXCLS, the first componenet contributes a slightly smaller amount, roughly 95.0%.

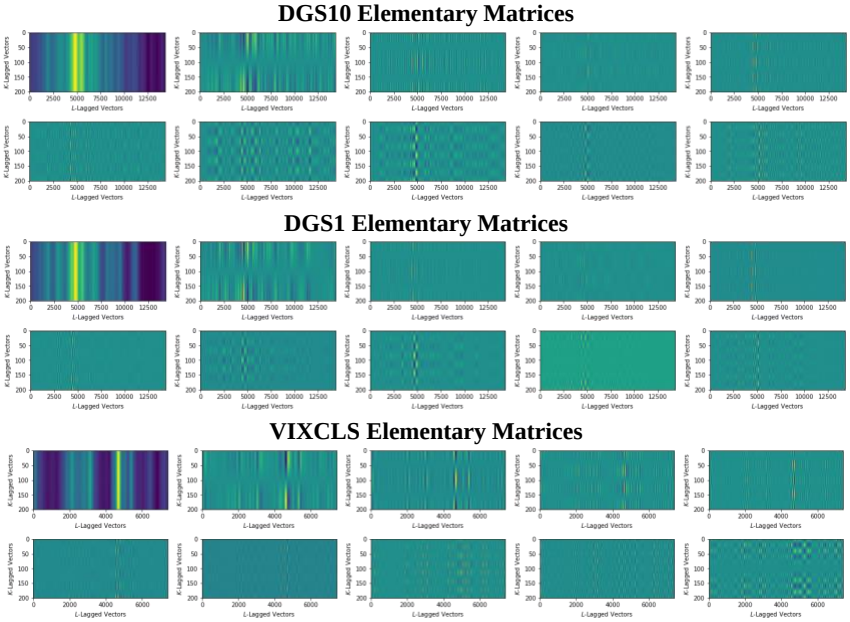

Figure (3) Elementary Matricies. Top row: 1-5. Bottom row: 6-10

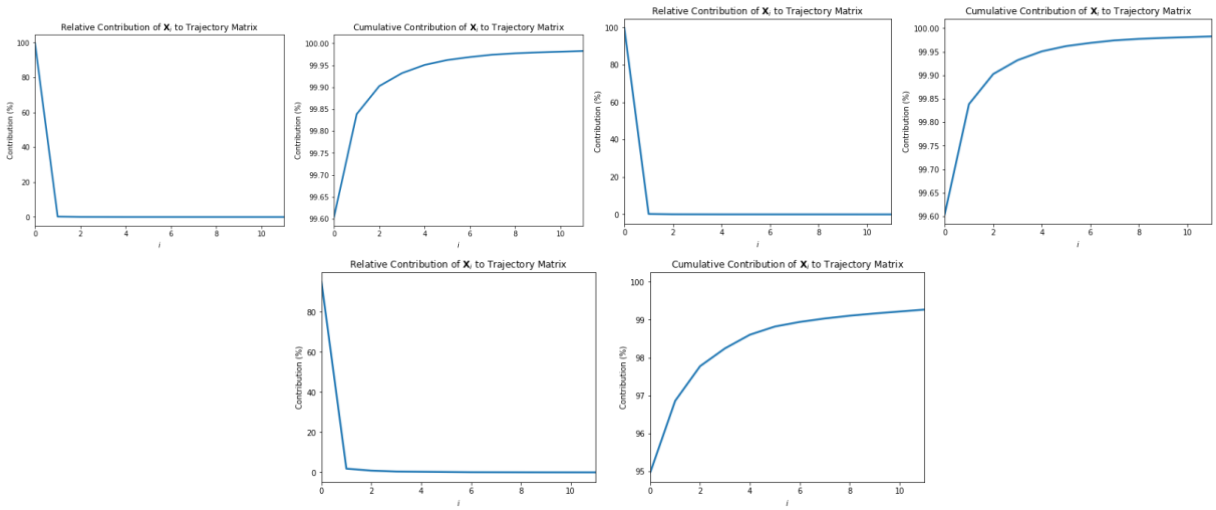

Figure (4) PCA Analysis. Top row left to right: DGS10 and DGS1. Bottom row: VIXCLS

A final check to confirm groupings comes in the form of the weighted correlation matrix. Figure (5) indicates that in all three of the time series, the first two reconstructed time series, $\widetilde{\boldsymbol{F}}_1$ and $\widetilde{\boldsymbol{F}}_2$ are moderatley correlated with each other and have roughly zero correlation to the remaining reconstructed series. Additionally, while there appear to be clusters of correalted reconstructed series after the first two, they are all grouped together and treated as noise, due to their small explanatory contributions to the original trajectory matrix

Figures (6)-(8) visualize the various components of the reconstructed time series. In the tope panel of each of the figures, the original time series is ploted in light gray. The reconstructed trend, $\widetilde{\boldsymbol{F}}_1$ is overlayed in blue. On the bottom of the top panel, the remaining 199 decomposed series are ploted. The main periodicity cruve can be seen in organge throughout the top panel as it indicates smaller movements about the trend line. In the bottom panel, the time series, $\widetilde{\boldsymbol{F}}_1$ and $\widetilde{\boldsymbol{F}}_2$, are summed to indicate the signal componenet of the original series, as shown in blue. Below in orange, the noise components are more clearly visable.

In the bottom panel of Figure (6) a clear demarcation occurs near the 5000th observation, corresponding roughly 1980. A sudden increase in the noise portion of the overall signal can be seen. This increase in noise corresponds directly to the 'tight money, high inflation, and heightened nuclear fear all contributed to real rates becoming unusually high in the early 1980s.' (Hendershott, Peek). Interestingly, as these fears for investors have passed, there appears to be an increased amount of noise in the signal compared to the early 1960s and 1970s.

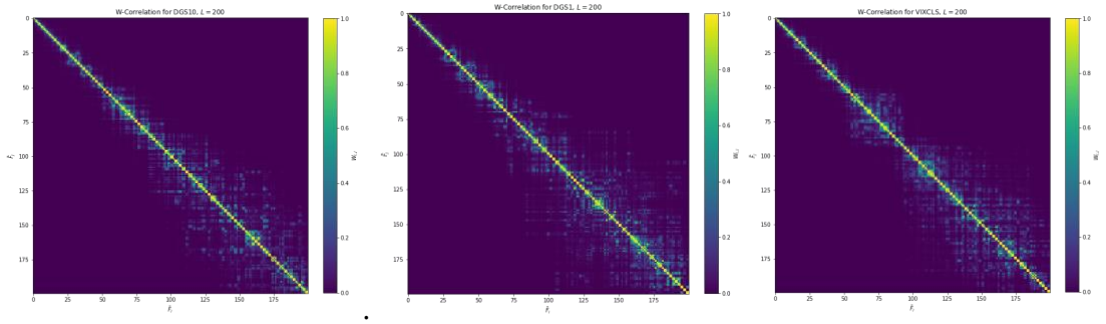

Figure (5) Weighted Correlation Matrix. Left to right: DGS10, DGS1, VIXCLS

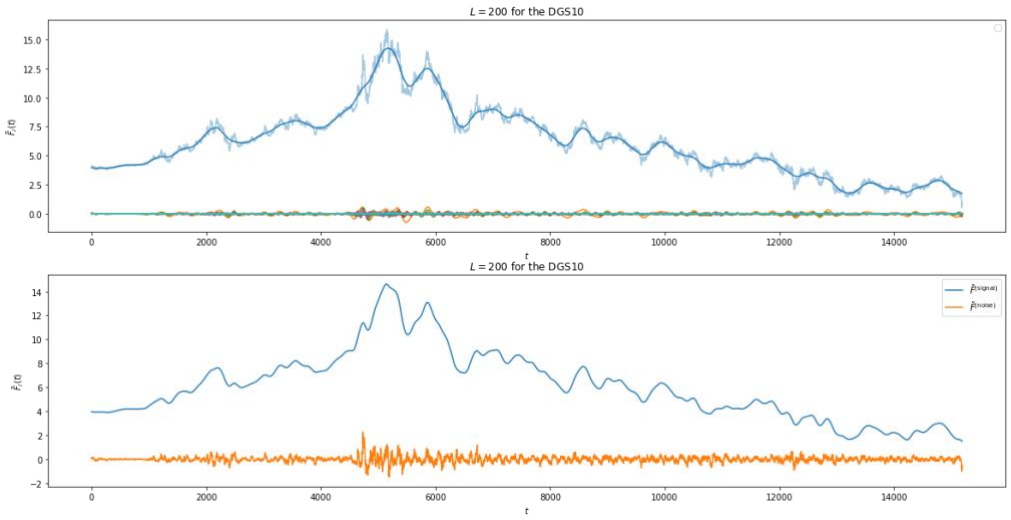

Figure (6) Top: DGS10 reconstructed series and original.
Bottom: Signal (blue), Noise (orange)

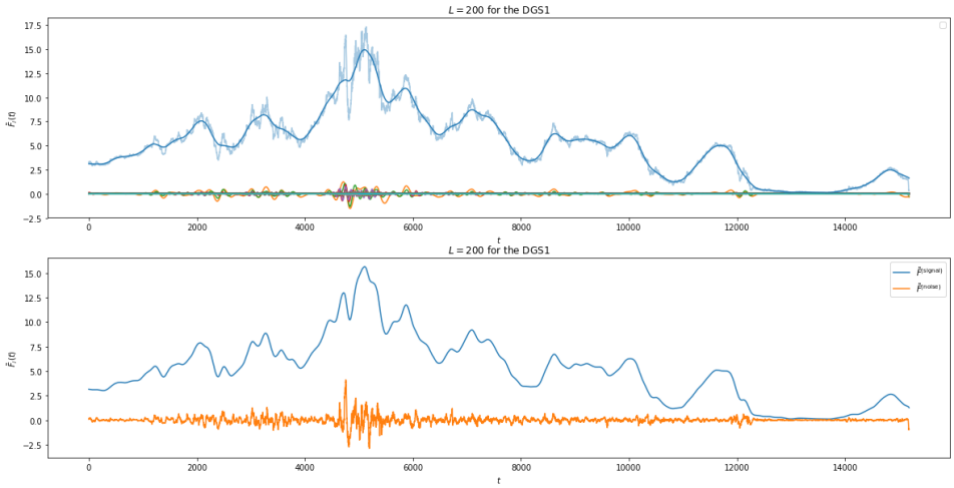

Figure (7) Top: DGS1 reconstructed series and original.
Bottom: Signal (blue), Noise (orange)

Similarities to DGS10 are expected to be seen when exploring the DGS1 data, as they are similar securities with differeing maturities. However, key differences can be noted. Wtihin the signal portion of the reconstructed series, there are many more oscilations. This is due to the monetary policy of the Federal Reserve. Increasing (decreasing) short-term rates are one of the tools the Federal Reserve uses to decrease (increase) economic activity within the United States. These oscilations of trend are clearly seen over the past fifteen years as the Fedearl Reserve cut short-term rates to near zero after the Global Financial Crisis in 2008. Interestingly, although this was a volatile time in United States economy, it did not create large instances of noise within the reconstructed signal. The organge time series appears realatively dormant compared to the large pulse seen in the 1980s.

While the SSA analysis for DGS10 and DGS1 reveal long oscilating trends with relatively small amounts of noise, the opposite is true of the reconstructed VIXCLS signal. The VIXCLS is an index that attempts to track volatility within the United States stock market. Because of quick reaction to news and other current events, there are often large and unexpected increases within the index. Periods of low volatility can quickly end should investors react to news events in a herd. This can be seen within the SSA reconstruction by a roughly stable trendline that is overtaken by the noise component. Spikes corresponding to the Global Financial Crisis in 2008 and the recent COVID-19 global pandemic illustrate this.

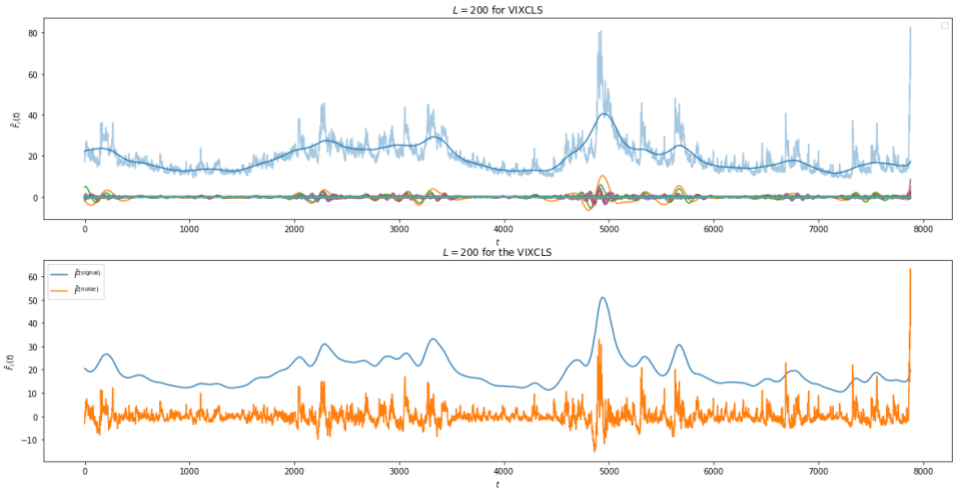

Figure (8) Top: VIXCLS reconstructed series and original.
Bottom: Signal (blue), Noise (orange)

Section V: Summary and Conclusion

SSA can be a useful tool to extract trend information from a noisy signal. Although not common within the quantitative finance community, the illustrations above show one use of SSA analysis on financial time series data. Within all three examples in this report, the first two reconstructed time series explained nearly 100% of the variance within the respective trajectory matrix. Within DGS10 and DGS1, it appears the noise component of the raw signal contributes relatively less to the overall signal, compared to the VIXCLS time series. While only two parameters are needed for the SSA algorithm, the raw signal and the window length, there is one significant drawback. The SVD analysis of the trajectory matrix, can be computationally expensive, especially when analyzing a large time series. In order to better extract trend information from a large time series, the window length must increase. This can create such large trajectory matrices that the SVD analysis becomes too computationally expensive to perform. In the cases above, the window length of 200 was used. Larger windows generated roughly the same output. However, once the window length was roughly 30% of the time series length, the SVD analysis became too computationally expensive.

Section VI: References

D'Arcy, Jordan: *Introducing SSA for Time Series Decomposition*, 2018, Kaggle Notebook

Kutz, Nathan J: *Data-Driven Modeling & Scientific Computation. Methods for Complex Systems and big Data*. 2013 Oxford Press.

Golyandina, Nina. Nekrutkin, Vladimir. Zhigljavsky, Anatoly: *Analysis of Time Series Structure, SSA and related Techniques*. 2001 Chapman & Hall/CRC

Peek, Joe. Hendershott, Patric: *NBER Working Paper No. 3036*. July 1989.

Appendix A:
Functions Implemented

Compute rank of trajectory matrix, X: np.linalg.matrix_rank(X)
Compute SVD analysis of trajectory matrix X: np.linalg.svd(X)
Compute Hankelisation of trajectory matrix X: Hankelise(X):
Python class for SSA analysis: SSA(signal, window size)
Sum decomposed components: SSA.reconstruct()
Compute and plot weighted correlation matrix: plot_wcorr(SSA)

Appendix B: Code

GitHub Link: FRED_SSA.ipynb