



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Master's Thesis of Data Science

# Interactive Visual Exploration System for Multiple Time Series Correlations

다중 시계열 상관관계를 위한  
대화형 시각적 탐색 시스템

August 2023

Graduate School of Data Science  
Seoul National University  
Data Science Major

Ye Ji Chun

# Interactive Visual Exploration System for Multiple Time Series Correlations

Hyunwoo Park

Submitting a master's thesis of  
Data Science

July 2023

Graduate School of Data Science  
Seoul National University  
Data Science Major

Ye Ji Chun

Confirming the master's thesis written by

Ye Ji Chun

August 2023

Chair                      Seunggeun Lee (Seal)

Vice Chair              Hyunwoo Park (Seal)

Examiner                Jay-Yoon Lee (Seal)

# Abstract

This paper proposes an interactive visualization tool to thoroughly investigate correlations within time series data. Although visualizing Pearson correlations among variables within a data is common practice in the process of Exploratory Data Analysis, a simple look into the correlations may not be sufficient in the case of time series data. There are two major problems that a data explorer may overlook, which are the fact that 1) two variables within a time series data may show high positive or negative correlations with some time difference, and 2) the correlations among a given time series may change over time. In order to address these problems, this paper proposes an interactive visualization system that allows users to look into correlations with time differences in variables and change of correlations throughout time segments, through time-shift view and time-segmentation respectively. The time-shift view generates correlations using the Time-Lagged Cross Correlations algorithm, which derives the correlation value with the highest absolute value gained by shifting one of the two columns within a given window range. The time-segmentation view splits the time series data into a number of segments that is set as an input to view the change of correlations as time passes. The correlations of the time series variables are visualized as network graphs for the explorer, along with heatmaps and line charts which are comparatively common methods in visualizing time series data or correlations. Also, a community detection algorithm is implemented to group or color the variables of the data which are denoted as nodes in the network graph visualization, which introduces a relatively novel method to group or cluster time series data. The effectiveness of this proposed tool is demonstrated by applying several time series data as possible examples. Through these examples, we have found that the visualization system is useful in finding patterns in cyclic data through the time-shift view, and identify shifts in correlation through the time-segmentation view.



Possible room for future work is also addressed in the paper.

**Keyword :** correlation, Time-Lagged Cross Correlations, time-series, network, visualization, exploratory data analysis

**Student Number :** 2021-23909

# Table of Contents

Chapter 1. Introduction.....	1
Chapter 2. Related Work .....	5
Chapter 3. Overview.....	11
Chapter 4. Description of Dataset.....	22
Chapter 5. Visualization Results and Case Study.....	28
Chapter 6. Discussion.....	45
Chapter 7. Conclusion and Future Work.....	49
Bibliography .....	51
Abstract in Korean .....	56

# Chapter 1. Introduction

In the process of analyzing time series data, checking the correlation among the columns is typically one of the crucial as well as basic steps of exploring the data. Even before applying machine learning techniques to multivariate time-series data such as those for anomaly detection or prediction, data scientists and analysts often visually look into the Pearson correlations of the variables usually in the form of heatmaps.

There are multiple papers for exploring time series correlations to extract meaning from them prior to data analysis. Some studies aim to extract information from general multivariate time-series data considering the distances of the data and the correlations among them with a matrix. [29, 30] Other research on time series correlations include more domain-specific studies such as in the field of finance and economics. [28]

Although it is common practice to simply looking into the Pearson correlations when exploring time series data as data scientists and analysts do for exploring the relationships among other types of data, this kind of exploration may overlook meaningful patterns and fail to find relationships that takes place between two variables with time differences. For example, two columns of time series may have high correlations when we shift one of the columns by some period, but may fail to find relationships between them when correlations are calculated without giving time shifts. In addition, the correlations may change over time, and therefore the correlations may be different over certain time segments. However, simply calculating the correlations for the entire time stamps given in the data may offset the differences in correlations that change over time.

Therefore, this paper proposes a visual exploration system that aims to allow users to thoroughly look into the correlations among the variables. While conventional heatmaps that visualizes Pearson correlations in a simple way may miss many crucial information, the visualization system supports

interactive exploration with various visualization methods to allow the analyst to capture potential information hidden behind various temporal relationships.

In order to address the stated problem, the proposed visualization system supports two kinds of views: the time-shift view and time-segmentation view. The time-shift view offers a view for the user to check correlations among variables giving time shifts for each column. The algorithm under the hood explores correlations between every two variables within a given range of time-shifts to calculate the correlations that have the highest absolute value as well as the time-lag that was applied for that specific point. This allows the users to find the correlation values taking into account the time shifts between every two variables. On the other hand, the time-segmentation view allows users to explore correlations of the given time series data by specific time segments. The number of segments is one of the inputs for the visualization system, and the time stamps of the data to be explored are divided into the number of segments. Therefore, the user is able to explore change of correlation values over time. This time-segmentation view may later aid data scientists in analyzing time series data, as the users can explore the change of relationships over time.

Another contribution of this paper in investigating time series data is that the proposed tool visualizes correlations not only with heatmaps, which is the most conventional way, but also with networks. Networks are one of the possible methods to visualize the relationships of time series data. With each node indicating the column of the time series and the edges formed according to the absolute correlation values calculated with either time shifts or time segments, the visualization captures the overall view of the relationships within the time series data. The nodes are classified and colored by groups formed by a community detection algorithm and a traditional machine learning method in clustering data. According to a survey paper on time series visualization research, small fraction of papers

has focused on visualizing time series data with network graphs. [7] The traditional line chart is also included as one of the visualizations, which shows the trend of the original data in addition to the correlation values of the data. By going back and forth with the correlation data and the original data.

The methods involved in this visualization is another contribution of this paper on time series visualization research. The visualization system incorporates machine learning methods as well as the Time-Lagged Cross-Correlations algorithm to calculate the correlation values giving time shifts. Although some past studies have included time shifts in deriving the correlation values, the proposed visualization system also derives the shifted days that result in the highest absolute correlation value which differentiates from past related studies.

To summarize, major contributions of this paper are:

- We present a tool to explore multivariate time series data in terms of the relationships between variables using Time-Lagged Cross-Correlations over time, and also derives the amount of shifted time that gives the output value.
- The tool also displays visualizations to view the change of correlation values over time within the data, by dividing the data into some time segments and allowing the user to compare the correlation values for each segment.
- We design an interface using various visualization methods to summarize and exhibit the multifaced aspects of time series correlations including heatmaps and line charts which are more conventional methods to visualize time series data, and network graphs which is comparatively newer method in visualizing time series data. This captures and allows the users to thoroughly view the relationship within the data.
- We cluster the time series data with machine learning algorithms, or

k-medoids and Louvain algorithm to cluster the time series data. The k-medoids is a more conventional data clustering machine learning method. On the other hand, the Louvain algorithm which is one of community detection algorithm clusters the time series data based on the network visualization formed. Using the Louvain algorithm may capture the relationships of the time series data compared to the k-Medoids clustering algorithm.

## Chapter 2. Related Work

### 2.1. Time Series Visualization Serving Various Purposes

There has been a plethora of researches on time series visualization for various purposes. The list of some papers that this study has referred to include those written on visualization tools for time series data pattern search [1], and other studies on preprocessing multivariate time series data. [2] There are also several papers that explore the clusters of time series data, often including an interactive interface. [3, 4, 5, 6]

Prior researches on visualizing correlations for time series also exist. There are domain specific visualization systems that are specialized for financial and economic time series data. [29] Also some papers that attempt to take into account distances among time series data while calculating correlations have been published as references. [28, 30]

In this paper, our visualization method incorporates machine learning techniques to explore the positive and negative correlations of time series data. Some similar attempts have also been implemented for more domain specific areas to recognize patterns in correlations [25] or to cluster time series data. [26] Other visualization systems for exploring time series data include those using parallel coordinates, [17, 18] and Sankey diagram. [19]

### 2.2. Similarity Measures for Time Series Data

According to a survey paper that overviews interactive visualizations for time series data that incorporates machine learning methods, or clustering and classification methods in the case of this survey, [7] the most commonly used methods to calculate similarities or distances are Euclidean Distance and Dynamic Time Warping (DTW). Distance measures such as Euclidean

Distance can be utilized in turn as similarity measures by flipping the values. For example, distance metrics that are naturally between 0 and 1 such as the Euclidean Distance can be turned into a similarity score by calculating  $1 - \text{distance}$ . Cross-correlations are also another method used to calculate distances between two time series variables. Cross-correlation is a measure of distance to check the similarity of the shape of two time series variables at its peak. Equation (1) is the mathematical formula for calculating the Cross-correlation between  $\{X_i\}$  and  $\{X_j\}$ , which is defined by covariance divided by the root-mean variance. To get the formula for sample Cross-correlation statistically, the numerator and denominator of Equation (1) are replaced by sample covariance and sample root-mean variance respectively, as shown in equation (2).

$$\rho_{i,j} = \frac{\gamma_{i,j}}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (1)$$

$$\hat{\rho}_{i,j} = \frac{\sum_{t=1}^N [(X_i^t - \bar{X}_i)(X_j^t - \bar{X}_j)]}{\sqrt{\sum_{t=1}^N (X_i - \bar{X}_i)^2 \sum_{t=1}^N (X_j - \bar{X}_j)^2}} \quad (2)$$

One of the contributions of this paper is that we have used the Time-Lagged Cross-Correlation algorithm, taking into account the lagged time at the peak value as well as the correlation value itself. This allows the user to check the similarity of the shape of two time series variables and also the amount of time difference there are between two similarly shaped time variables. Another point that we have looked into is that we deemed not only positive but negative values as meaningful relationships. Therefore, we calculated the peak of the absolute value of correlations within the time shift range given as an input to capture not only similar shapes but also similar shapes that are flipped upside down.

In like manner, the edges for the network visualization are also formed based on the peak value of the absolute value of the correlation within the input time shifting range. Therefore, both positively and negatively related variables are taken into account in analyzing the correlations and visualizing



them. In other words, having high negative relationships as well as positive relationships are considered as meaning having close relationships with each other.

## 2.3. Time Series Visualization with Clustering Methods

There are several prior researches on data visualization that incorporate time series visualization with clustering methods. According to the survey paper on time series visualizations with interactivity and clustering methods involved, [7] methods of clustering that are frequently used across most studies are hierarchical clustering, model-based clustering methods, which include the self-organizing map (SOM), and partitioning methods that include k-Means, k-Medoids and Fuzzy c-Means algorithms.

We attempted to incorporate both relatively conventional and one that is not so commonly used. We used total two clustering algorithms, which are k-Medoids and Louvain algorithm. The reason for using two clustering algorithms was to simplify the visualization system at the current stage, although many more clustering algorithms may be implemented in the future to compare and contrast the effects of implementing different algorithms. However, we have focused on comparing traditional clustering algorithm to implementing a community detection algorithm in this visualization system. As a relatively new method of clustering proposed in this paper, a community detection algorithm was implemented and considered. Therefore, the k-Medoids algorithm represents a more conventional time series clustering algorithm while the Louvain algorithm tests the effect of implementing a community detection algorithm to capture clusters formed based on the relationships among the variables.

Other classical data clustering methods include k-Means, Fuzzy c-Means etc. as mentioned beforehand. Although the k-Means and k-Medoids algorithm both cluster data based on the distance measure, the k-

Means method forms clusters in a direction that minimizes the total squared error, while the k-Medoids method tries to minimize the sum of dissimilarities within clusters. Another point that we may take into account is that the k-Medoids method derives a data point as the central point of each of the clusters unlike the k-Means. Several community detection algorithms besides the Louvain algorithm include Surprise and Leiden community detection, and the Girvan-Newman algorithm. We used the Louvain algorithm because it is one of the most popular community detection methods that has high computing speed even on large network data. While the Louvain detection algorithm does not take the number of clusters as input, the user must set the number of clusters for k-Means and k-Medoids as input for executing the algorithm.

A contribution of this paper includes the fact that it incorporates a community detection algorithm as one of the clustering algorithms to group time series data. This stems from the observation that clustering time series based on the relationship network may produce meaningful results, since network graphs capture individual relationships between highly correlated variables. [8] Compared to visualization and clustering based on network graphs, it may be difficult for users to identify specific variables with high correlations with more commonly used algorithms such as k-Means and k-Medoids. These more conventional algorithms fit all of the data into one of the clusters, which leads to unclear distinction in identifying which individual variables are highly correlated with other specific variables in terms of absolute value.

Other references that use clustering methods in its time series visualization include those that are more domain specific, such as a study on visualizing click stream data. [21] There is also study that uses distance-based correlations to test network dependence that explores some methods in a deeper way. [30]

## 2.4. Network Visualization

According to a survey paper on interactive visualizations on time series data using machine learning methods, [7] common visualization techniques used in this field of research include line plots, geographic maps, heat maps, histograms and bar graphs. There are some link node visualizations in the proportion, but they are not as common. Time series visualization that uses model-based methods such as the self-organizing map (SOM) specifically adopt link-nodes and glyphs as their common visualization technique.

There are prior studies that analyze time series data with machine learning and deep learning methods using graph or neural-based methods. [8, 9, 10] Recently there has been studies on multivariate time series using Graph Neural Networks for anomaly detection [11, 12, 13, 14] as well as forecasting. [15, 16] Therefore, we look to the possibility of visually aiding or incorporating the process of such time series analysis with the visualization system in this paper as future work.

The contribution of this paper in this aspect is that we utilize network graphs to visualize the relationships derived by Time-Lagged Cross-Correlations by denoting each time series variables with nodes and connecting them with edges if the peak absolute correlation is larger than the threshold that is one of the inputs that is set by the user of the proposed visualization system. This is also related to the other contribution of the paper mentioned in the last subsection, which is the fact that the system provides for the user a community detection algorithm as an option to group the variables, which is visualized through the color of the nodes in the system.

Some visualization tools that visualize non-time series data using network visualization techniques include those for more general purposes such as exploring the operations for GNN models [22] or to visualize data embedding methods. [23] There are also papers on graph visualizations for

data that are more domain specific data such as traffic jam data that target to solve similar tasks. [24, 25, 27] Other studies that attempt to capture temporal changes in relationships with networks also exist. [26]

## Chapter 3. Overview

### 3.1. Input Data

The input data for the system are multivariate time series data with numerical values for each variable and along with a time stamp. For example, let's say we use data as an input with changes of the average temperature for each day in 2017 for 50 countries with the name of each country as the column name. Therefore, there will be a total of 365 rows (365 days in the year 2017) for each day and 50 columns for each country.

Through utilizing the data visualization system, we expect to find the correlations among the temperatures of each country, taking into account various time shifts and time segmentations. We will use this example to explain below the two views provided in the system in this section to have an idea of the goal of the implementation of this system. The system provides mainly two types of views: the time-shift view and the time-segmentation view.

### 3.2. Preparation for the Time-Shift View

The input data for the system are multivariate time series data with numerical values for each variable and along with a time stamp. For example, let's say we use data as an input with changes of the average temperature for each day in 2017 for 50 countries with the name of each country as the column name. Therefore, there will be a total of 365 rows (365 days in the year 2017) for each day and 50 columns for each country.

First type of view offered by the visualization system which is the time-shift view visualizes the results of the Time-Lagged Cross-Correlation values with heatmaps, network and line chart with various time ranges and threshold options that the user chooses as an input. The choices of time-

shift range and threshold for creating the edges for the nodes are provided with a drop-down menu. As shown in figures 1 and 2, The sidebar on the lefthand provides the menus that the user can choose to view the visualization. The user can choose the time shift range, clustering method that will color the nodes of the network, and the threshold level by which the edges of the network are formed if the absolute correlation value exceeds the threshold level.

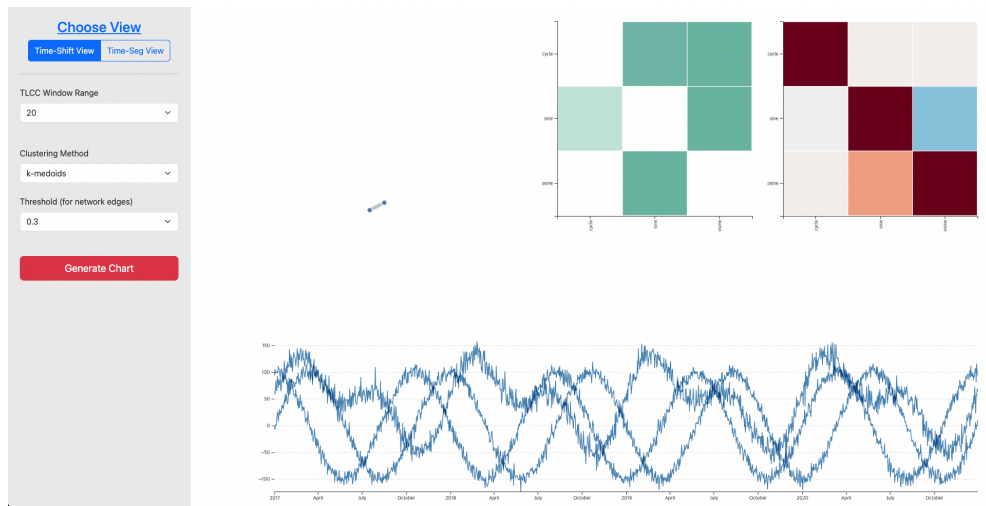


Figure 1 – A screenshot of a sample view of time-shift view

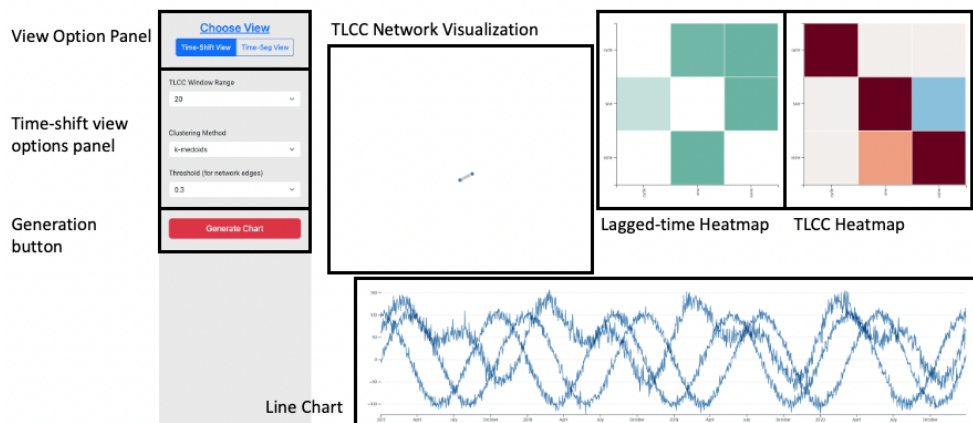


Figure 2 – Names of each section of time-shift view

Although for k-Medoids clustering method needs the number of clusters set by the user as an input, the current visualization system chose the number of clusters as the same number of clusters formed when the Louvain algorithm is run on a network that is formed when there is no threshold level set. Adding the number of cluster options for such clustering may be one of the future works for this project.

As shown in figure 1 and 2, there are a total of four sections for visualizing the results of the Time-Lagged Cross-Correlations algorithm. First there is a line chart that visualizes the original forms of the time series data. This is a common line chart graph to visualize time series data, which was included since it may be one of the crucial information that the user may want to view even when the focus of this tool is to analyze correlations among time series data. A detailed view is given in figure 3.

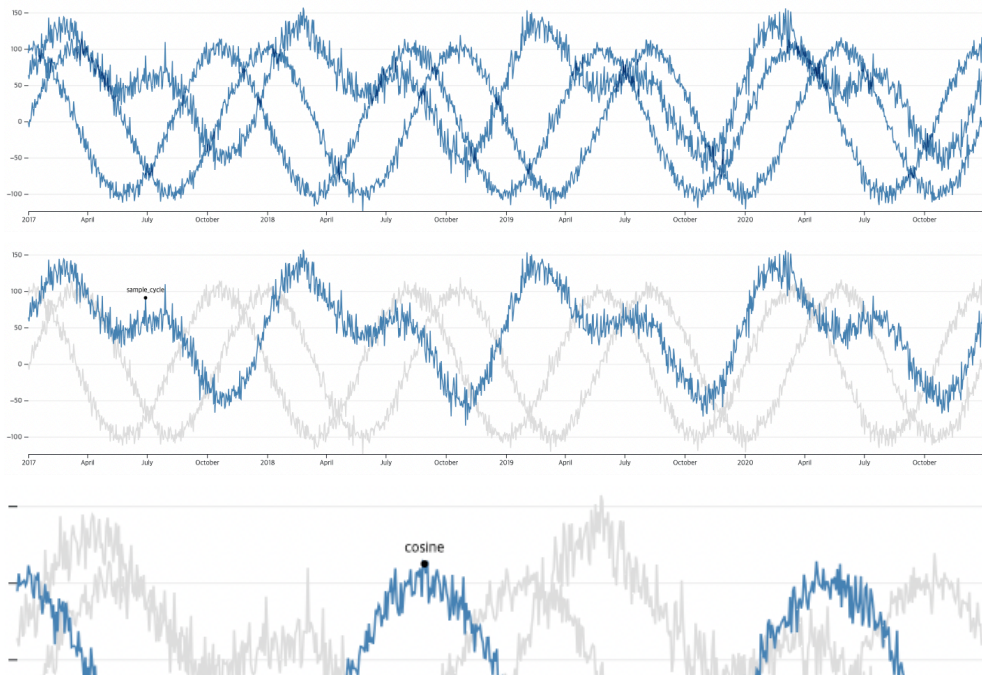


Figure 3 – Line chart visualizations and the color and tooltip effects when the mouse is hovered over each line

The second type of visualization given are the two heatmaps that each visualize the amount of time shifted at the peak level of absolute correlation value and the correlation value at that point. As shown in figure 1 and denoted in figure 2, the heatmap at the top right-hand corner of the visualization system shows the correlation values derived from the Time-Lagged Cross-Correlations algorithm. Correlation values that are close to dark red denotes values that are close to 1.0, while color closer to dark blue denotes a value close to -1.0. The diagonal values from the top left-hand side to the bottom right-hand side has dark red colors which denotes correlation values of 1.0, since the correlation value of each column to itself equals to 1.0. One thing to note here is that the heatmap of the correlation values is not symmetric like normally heatmaps with Pearson correlation values are, because the values differ for a pair of columns depending on which column was shifted in respect to the other.

Another heatmap located right left to the correlation heatmap is another heatmap that visualizes the length of lagged period for each corresponding Time-Lagged Cross-Correlation value. In this heatmap the maximum value is decided by the amount of time range that the user has set as one of the input options, and the minimum value is 0. The closer the value is to 0, the color is closer to white, and as it gets closer to the maximum value the color becomes darker and closer to green. The heatmap for the lagged time values have a sequential color scale while the heatmap for the correlation values have a sequential color scale to fit the visualization purposes. The diagonal values corresponding to the former heatmap in the latter heatmap has values of 0, because the highest correlation values of each column with itself is 1.0 when the time shift is 0. A closer and more detailed view of the heatmaps is shown in figure 4 and 5. Also, slight changes of color of the heatmaps when the window range is provided in figure 6 to check the effects of the input values that has on the output values and its visualization outcome. Even as the time shift range increases by just 10 timestamps, starting from range 20,



then going on to 30, and then 40, we can check that the absolute value of the correlation values slightly increases.

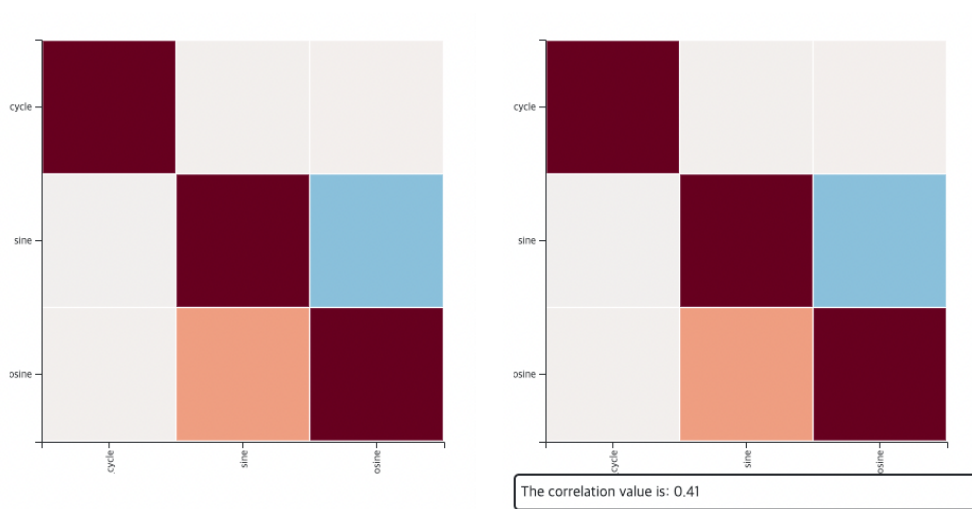


Figure 4 – Heatmap on the top right-hand corner of the visualization system, that visualizes the correlation values derived from the Time-Lagged Cross-Correlation algorithm. The values are shown when the mouse is hovered on each tile.

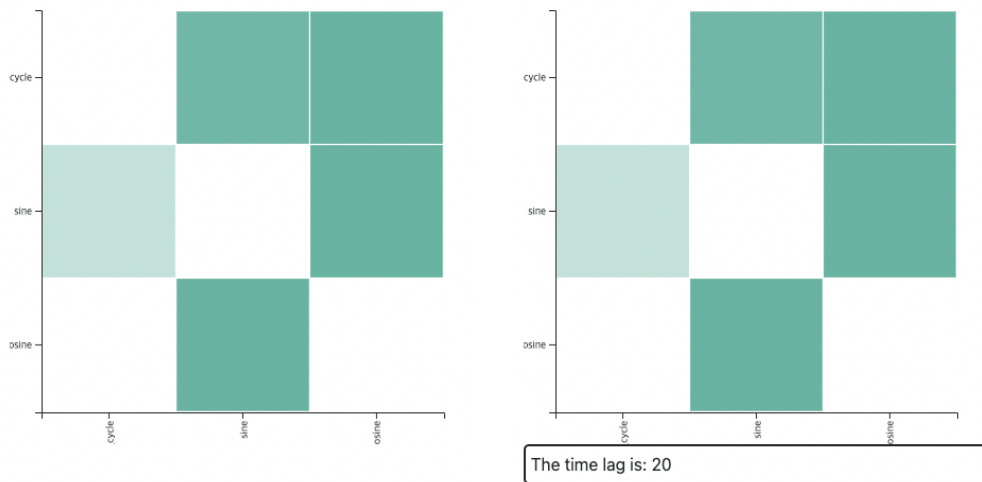


Figure 5 – Heatmap that is on the left side of the heatmap in figure 3 in the visualization system, that visualizes the lagged time derived from the Time-Lagged Cross-Correlation algorithm corresponding to the values in the heatmap in figure 3.

The values are shown when the mouse is hovered on each tile.

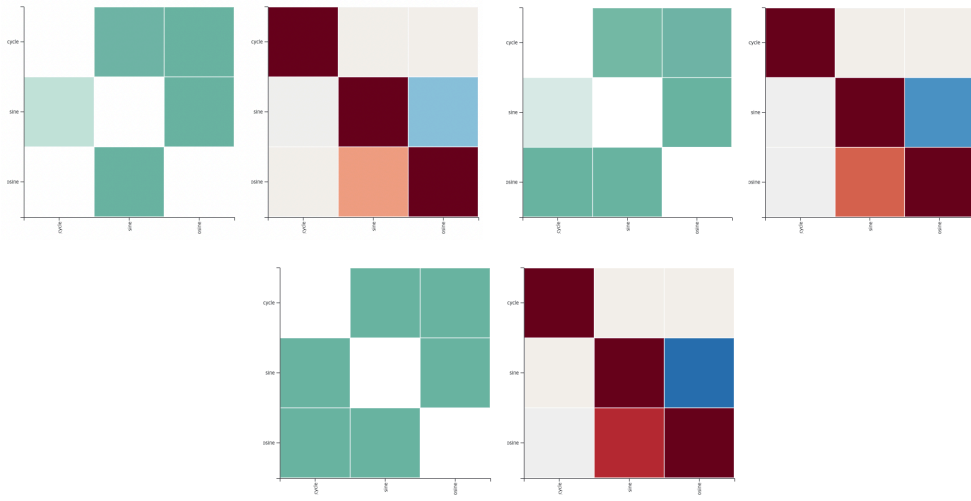


Figure 6 – Screenshots of heatmaps when the time-shift range increases from 20, 30 to 40 respectively. Although relatively slight changes are given in the time-shift range, some of the absolute correlation values become larger.

The last type of visualization offered in the time-shift view is the network graph visualization. The network graph visualization is drawn based on the heatmap, or the Time-Lagged Cross-Correlation values for all the columns. Each node denotes a column of the time series data, and with the threshold put in as another input value, edges between two nodes are formed if the absolute value of the correlation exceeds the threshold value. The thickness of the edges increases as the absolute value of the edges increase. If a node does not have any connection with other nodes, it is not shown in the visualization. The color of the nodes is decided by the cluster that is formed with the option that is also chosen at the left-hand sidebar by the user. Currently, since there are only 3 maximum nodes in the example, a simple version of how the network graph is visualized is shown in figure 7.

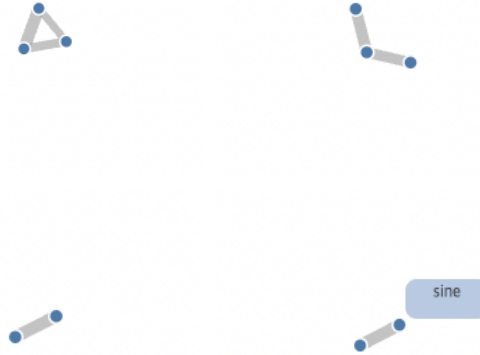


Figure 7 – Screenshots of possible network graphs according to the correlation values and the threshold values set as an input value by the user. Also, the name of the variable pops up when mouse is hovered over a node.

To explain with the given example, let us say the input range was 60 days and we are trying to gain the cross-correlation value for Korea and New Zealand in the example data. The algorithm calculates 60 correlation values between the two columns, shifting New Zealand's temperature data by 0~60 days. If the largest absolute value was  $-0.86$  when the shifted day was 12, the Time-Lagged Cross-Correlation value for Korea and New Zealand becomes  $-0.86$  when the lagged time was 12. Through the line chart visualization, the user is able to check the original temperature data of Korea and New Zealand. With the heatmaps we can check the relationship between the trends of the temperature in Korea and New Zealand, such as how much the weather is most similar with some time differences. We will also be able to view if they have significant correlations with the network graph visualization along with the relationship with other countries in the data. We will also be able to check if those can be clustered into a same group according to the k-Medoids algorithm of the Louvain algorithm.

### 3.3. Preparation for the Time-Segmentation View

The second type of view is the time-segmentation view that allows the users to explore the change of correlations between two variables over time. The user inputs the number of segments, in which the system divides the time series data into by its time stamps. In other words, the time stamp is divided into the number of segments to look into. Then the user can explore the change of correlation between time variables over time. Similar to the time-shift view, the user can change from time-shift view to time-segmentation view by choosing the radio button on the top left-hand corner of the visualization system as shown in figure 8 and 9. The user should choose the number of segments he or she would like to generate, and the segment number to look into though the visualization, the clustering method and the threshold level for the visualization on the left-hand sidebar. After selecting the options, then clicking on the red generation button will create the corresponding visualization.

The visualization provides a heatmap, line chart and network visualization which is similar to the time-shift view, but does not contain the heatmap to indicate the amount of time-lag. This is because the Pearson correlation is calculated once for each segment, and thus does not have different time lag outputs. Moreover, the heatmap in this view is symmetric unlike the one in the time-shift view, since there is only one correlation value for a pair of variables. Another difference with the time-shift view is that the line chart shows not the whole dataset but the part of the segment that the user wants to look into. Therefore, the view in the line chart section changes as the number of total segments and the specific segment chosen changes in the side bar.

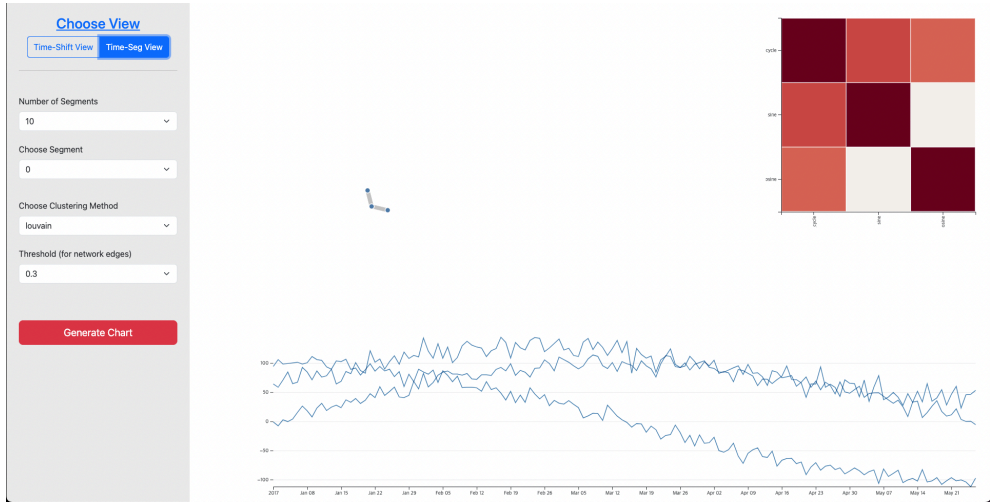


Figure 8 – A screenshot of a sample view of time-segmentation view

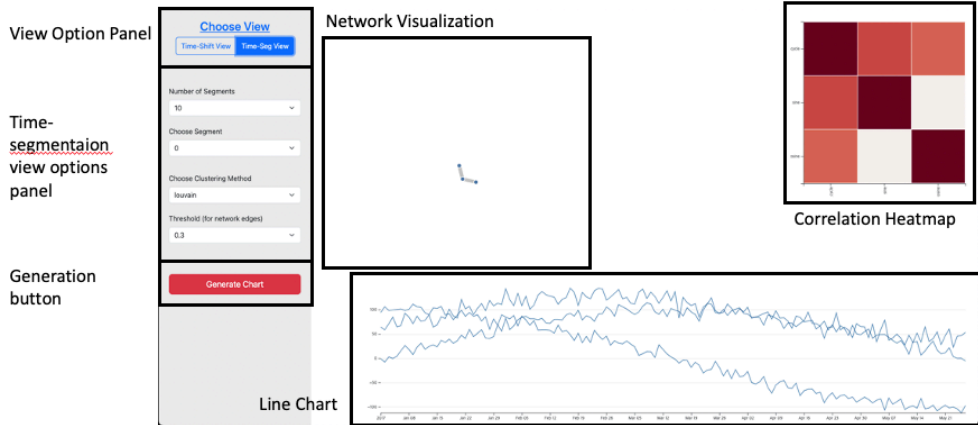


Figure 9 – Names of each section of time-segmentation view

The changes in the visualization according to the change of options is demonstrated with a simple toy data example in figures 10 and 11. Figure 10 shows the change of network visualization with the change of thresholds, and figure 11 shows a change of correlations across time segments, which could derive useful meaning in real world data.

As in the time-shift view, we could imagine the use of the time-segmentation view with a more realistic example. For instance, let's say that the user chose to have 4 segments for the weather data example. Then each segment will approximately contain data for each quarter of the year 2017

for every country. By looking into the visualization system, we can take a look into the change of correlation of New Zealand and Korea over the 4 quarters of the year 2017. Some quarters may show higher positive or negative correlations, but other quarters may show lower positive or negative correlation. Whether each country has much difference between and/or within seasons may have an effect of the results of their change in correlation values.

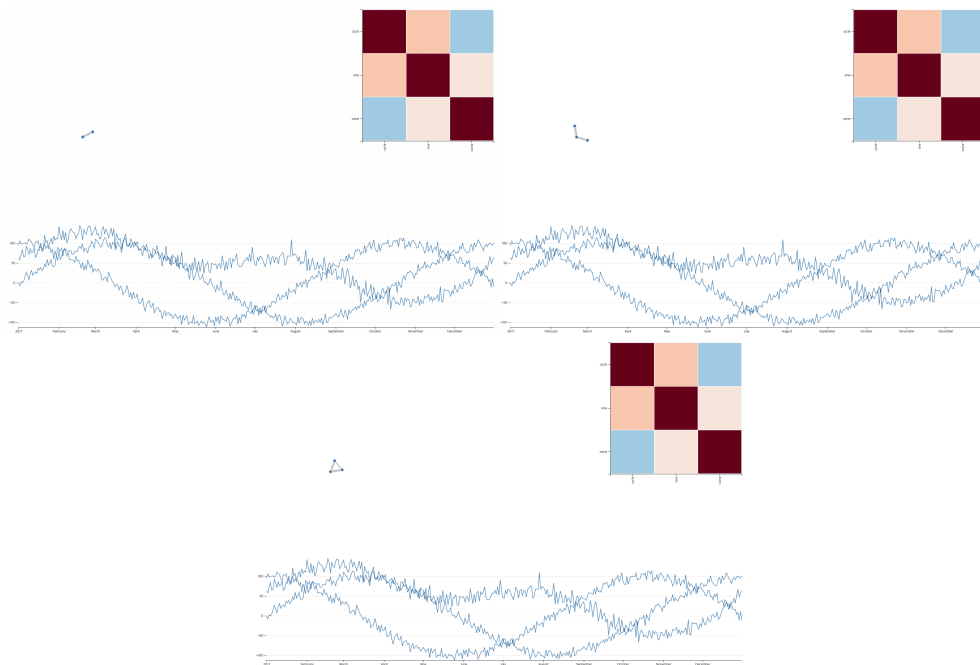


Figure 10 – Change of network graph as the threshold decreases from 0.3 to 0.2 to 0.1 for the first segment out of 4 segments of the data

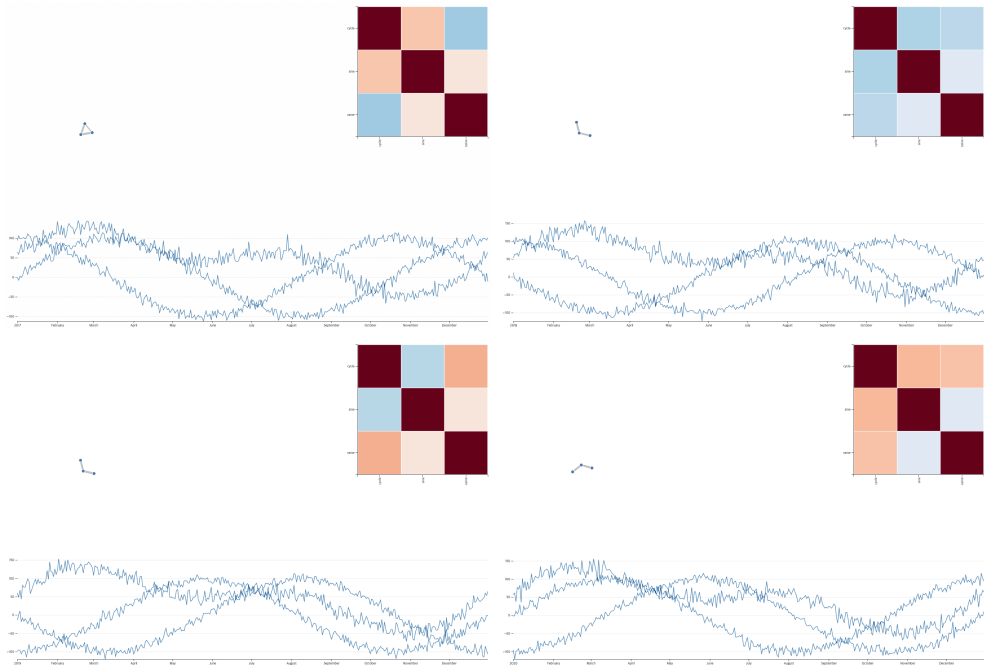


Figure 11 – Change of network graph as the threshold decreases from 0.3 to 0.2 to 0.1 for the first segment out of 4 segments of the data

## Chapter 4. Description of Dataset

In order to test the effectiveness of this visualization tool, we have used 1 toy data to demonstrate the function of this tool as well as 2 time series datasets from the Kaggle website to actually explore the correlations of time series data perhaps in its EDA process. One of the datasets is a weather data in India, and the other is a S&P 500 stock market dataset.

### 4.1. Synthetic Data

Synthetic data was created to test the functioning of the visualization tool. Using sine and cosine functions, 3 columns were generated with daily timestamps starting from January 1<sup>st</sup> of 2017 to December 30<sup>th</sup> 2020. 3 of the column names were 'sine,' 'cosine,' and 'sample cycle' with a total of 1460 rows. The sine and cosine column was generated to have a complete cycle in 292 days, and the value was multiplied by 100. The sample cycle was made by adding two sine functions that were transformed and the cycle was 365 days. Then gaussian white noise was added to all the three columns. The line chart for this data is shown in figure 12 with the data table in figure 13.

We expect to experiment if the Time-Lagged Cross-Correlation algorithm works as we expect it to, since sine and cosine functions should have the same form with some time differences, not considering the white noise that was added. For the 'sine' and 'cosine' columns, it should show almost perfect positive correlation when the 'cosine' column is shifted by 73 days or almost perfect negative correlation the same column is shifted 219 days.



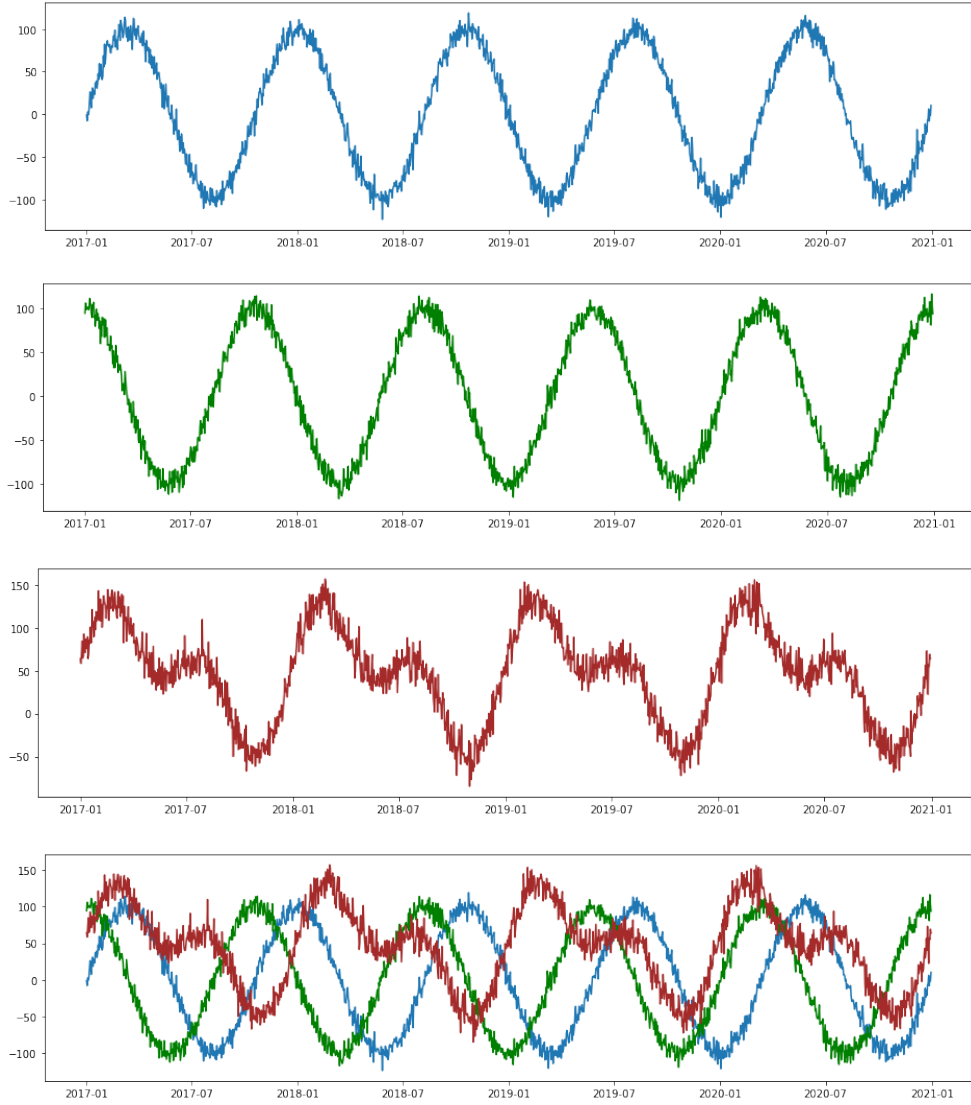


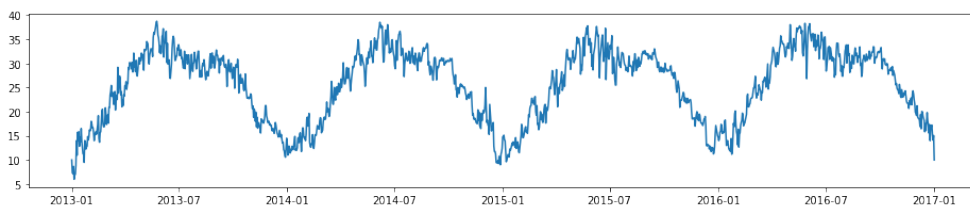
Figure 12 – 3 columns of the synthetic data. From top to bottom, ‘sine,’ ‘cosine,’ and ‘sample cycle’ columns and the three columns plotted together respectively.

	sample_cycle	sine	cosine
date			
2017-01-01	64.098554	-1.107648	94.249341
2017-01-02	58.513156	-7.641302	105.936861
2017-01-03	71.027445	2.632984	98.349706
2017-01-04	84.662213	-0.348903	99.277925
2017-01-05	64.711053	3.958170	100.363330
...	...	...	...
2020-12-26	22.577625	6.186513	104.797360
2020-12-27	57.197780	-7.859772	80.818156
2020-12-28	55.011026	1.044068	116.243842
2020-12-29	69.145328	-1.368544	102.339058
2020-12-30	63.822804	10.366974	93.788007

Figure 13 – Synthetic data table made of sine and cosine functions

## 4.2. Weather Data

A second dataset that we explored with the visualization tool is a weather data of Delhi, India from the Kaggle website. There are four columns each denoting the mean temperature, humidity, wind speed and mean pressure for each day as the timestamp. The date of the timestamp ranges from January 1<sup>st</sup> of 2013 to January 1<sup>st</sup> of 2017. The data contains 1462 rows of daily data. The line chart of each column and the figure of the data table are shown in figures 14 and 15 respectively. This dataset may also have a cycle with a length of a year, as one can conjecture through the pictures in the line graph.



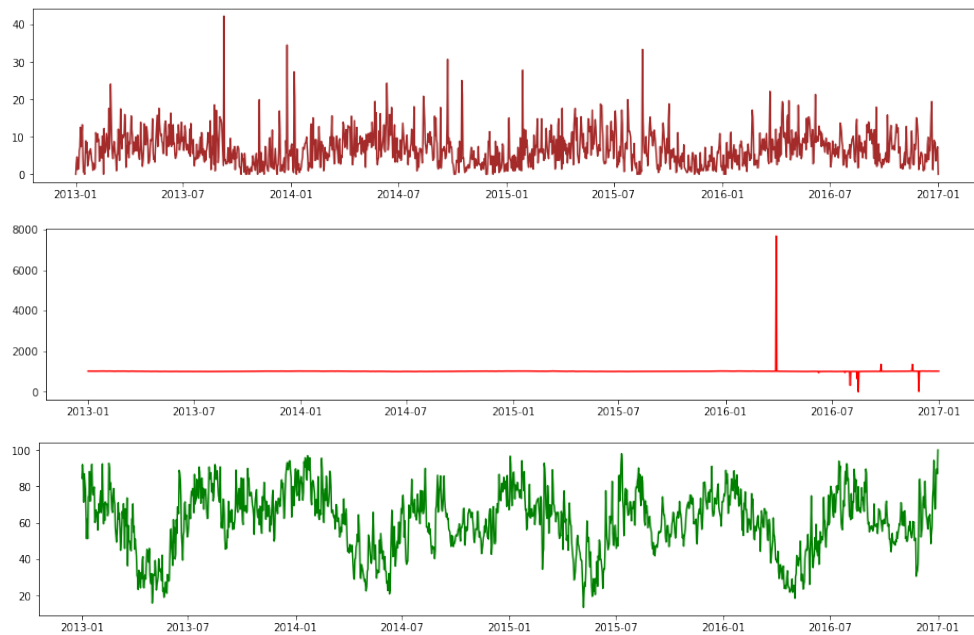


Figure 14 – Line chart of the four columns from the weather data of Delhi, India. Columns denoting mean temperature, humidity, wind speed, mean pressure respectively.

	meantemp	humidity	wind_speed	meanpressure
date				
2013-01-01	10.000000	84.500000	0.000000	1015.666667
2013-01-02	7.400000	92.000000	2.980000	1017.800000
2013-01-03	7.166667	87.000000	4.633333	1018.666667
2013-01-04	8.666667	71.333333	1.233333	1017.166667
2013-01-05	6.000000	86.833333	3.700000	1016.500000
...	...	...	...	...
2016-12-28	17.217391	68.043478	3.547826	1015.565217
2016-12-29	15.238095	87.857143	6.000000	1016.904762
2016-12-30	14.095238	89.666667	6.266667	1017.904762
2016-12-31	15.052632	87.000000	7.325000	1016.100000
2017-01-01	10.000000	100.000000	0.000000	1016.000000

Figure 15 – Table of the weather data of Delhi, India

### 4.3. Stock Market Data

The third dataset also from the Kaggle website is a daily stock market data set from NASDAQ, NYSE and S&P 500. A vast amount of data exists

from year in this dataset, from January 4<sup>th</sup> of 1970 to the current date for 412 companies. Also, because the columns include ‘Date,’ ‘Volume,’ ‘High,’ ‘Low,’ ‘Open.’ and ‘Close,’ and ‘Adjusted Close,’ we used the ‘Adjusted Close’ column to get rid of redundant information. We also cut the data and cleansed it to use a period of about 6 and 2/3 years, from January of 2016 to August of 2022. When the dates were decided the dataset contained 401 companies with valid data. However, because there were too many columns for us to explore in the current visualization system, we picked 49 companies to look into, a fraction of companies out of the 401 companies. These data may have some cyclic factors every year, but overall, the cyclic factors may not also be very strong since the other factors affecting the market trend may be more influential in the data. Figure 16 shows the table of the stock market data that we used and figure 17 shows the line chart of the 49 stocks.

	CSCO	UAL	TROW	ISRG	NVR	MRO	BA	GILD	NLSN	EQIX	...	DOV
0	21.543217	55.610001	57.520290	60.821110	1555.660034	11.698455	126.005112	78.083855	37.813046	260.976501	...	43.847973
1	21.445328	55.060001	57.757931	61.328888	1594.089966	11.643703	126.516312	79.079742	38.417454	267.437408	...	43.015686
2	21.216927	55.200001	56.463131	61.471111	1587.060059	10.293181	124.507408	79.908295	38.193905	272.479248	...	42.105148
3	20.727495	52.630001	54.955261	59.655556	1552.319946	9.736546	119.287811	76.681679	37.374210	267.331604	...	41.500500
4	20.213589	51.889999	53.562134	59.615555	1541.420044	9.444541	116.579384	76.841011	36.703564	273.061066	...	41.144825
...	...	...	...	...	...	...	...	...	...	...	...	...
1671	47.410000	36.910000	124.449997	216.809998	4200.910156	25.709999	160.070007	63.680000	27.780001	667.780029	...	133.139999
1672	47.070000	37.389999	124.820000	220.470001	4289.009766	26.290001	163.600006	63.619999	27.799999	671.469971	...	132.979996
1673	47.270000	38.270000	126.389999	225.020004	4364.990234	26.370001	169.380005	63.590000	27.840000	689.530029	...	134.970001
1674	45.889999	36.570000	119.680000	211.089996	4224.990234	26.160000	164.529999	62.369999	27.840000	670.739990	...	127.989998
1675	45.595001	36.113300	118.175003	210.080002	4204.000000	27.219999	163.419998	61.525002	27.875000	663.059998	...	126.959999

Figure 16 – Stock market data table

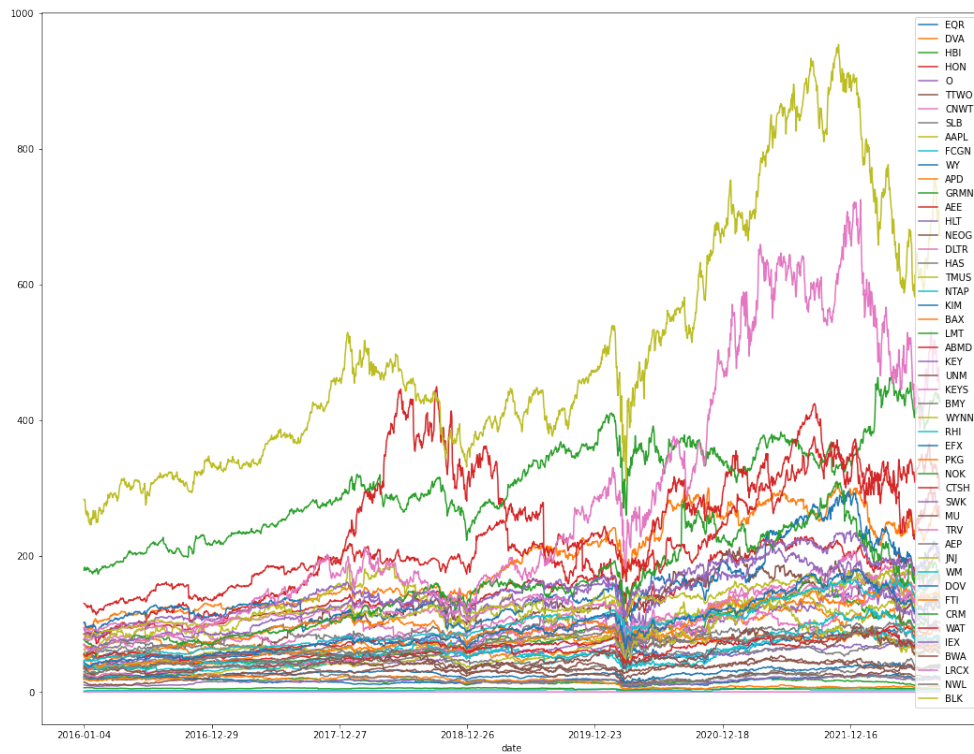


Figure 17 – Line chart of the forty-nine columns from the stock market data.  
Columns denoting stock market prices over the time stamps for each stock.

# Chapter 5. Visualization Results and Case Study

## 5.1. Synthetic Data Exploration

First experiment with the visualization system was with the synthetic data created with sine and cosine functions. Two out of three columns of this data, 'sine' and 'cosine', should show high correlations either in a positive or negative way with some time difference. Both columns have a complete cycle in 292 timestamps. We expect that when the cosine column is shifted for 73 timestamps ( $= 292 / 4$ ), or  $73 + 292 * n$  where  $n$  is an integer equal to or larger than 0, then it will show almost a perfect positive correlation with the sine column. We also expect that when the cosine column is shifted for 219 timestamps ( $= 292 * 3/4$ ), or  $219 + 292 * n$  where  $n$  is equal to or larger than 0, then it will show almost a perfect negative correlation with the sine column. In the case of shifting the sine column, we expect that it will have almost perfect negative correlation when it is shifted by  $73 + 292 * n$  columns, and positive correlation when it is shifted by  $219 + 292 * n$  columns. On the other hand, the 'sample cycle' data runs on a different cycle, or 365 days. Therefore, it is unclear how the correlations will be derived.

First, in the time-shift view, experiment was run on 5 different time-shift ranges, or 50, 100, 150, 200, and 250 time-shift ranges. 5 different thresholds for creating edges were given, or 0.1, 0.3, 0.5, 0.7 and 0.9. The results somewhat matched our hypothesis for the relationship between the columns to some extent. An example screenshot of this view is shown in figure 16.

One point to look into was the relationship between the sine and cosine columns. The results for when the time-shift range input changed from 50, 100, 150, 200 to 250 is shown in figure 17, where the clustering method and

threshold are fixed to k-Medoids and 0.1 respectively. When the time-shift range was 50, the output all of the green columns in the time-lag heatmap's value was 50. However, when the time-shift range was set equal to or over 100 and equal to or lower than 200, the time-shift values in the heatmap remained the same: 1) when the sine column is shifted in respect to the cosine column, the point with the highest absolute correlation value is  $-0.99$  when 73 periods are shifted, and 2) when the cosine column is shifted in respect to the sine column, the point with the highest absolute correlation value is  $0.99$  when 72 periods are shifted. When the time shift range reached 250 the values in case 1) became  $0.99$  at 219, and  $-0.99$  at 220 in case 2). These results match our assumptions.

In the case of sample cycle column which does not have the same cycle length with the other two columns, the time lag values did show some patterns in the given time shift ranges. However, we were able to find that the absolute correlation values in relation to the other two columns were not high in all of the time shift range options.

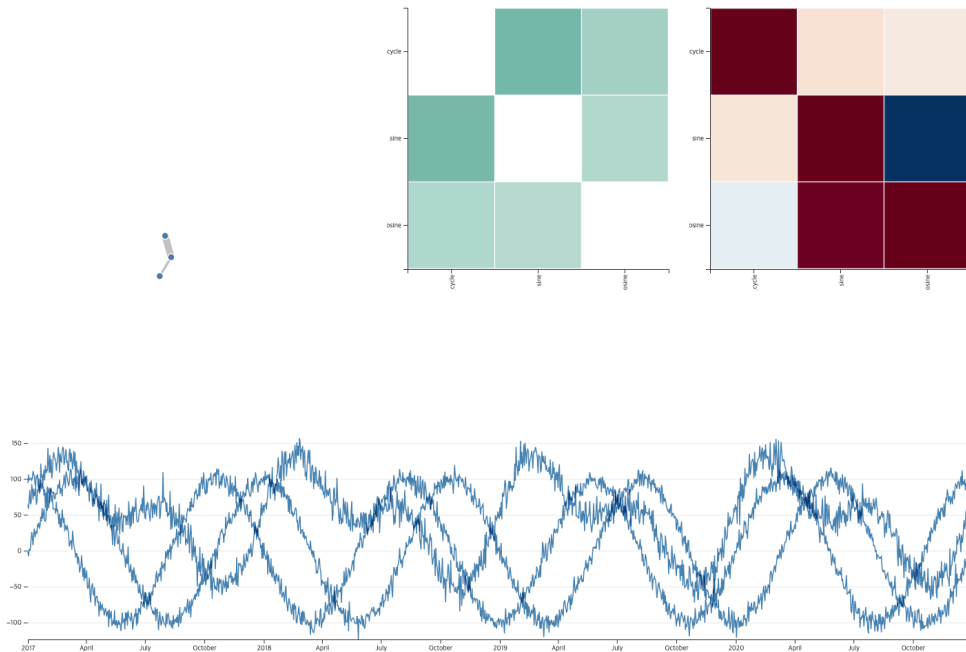


Figure 18 – Synthetic data time-shift view example screenshot

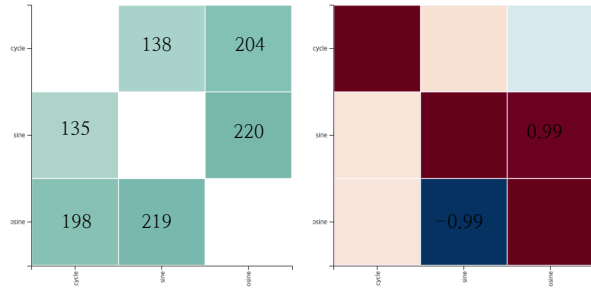
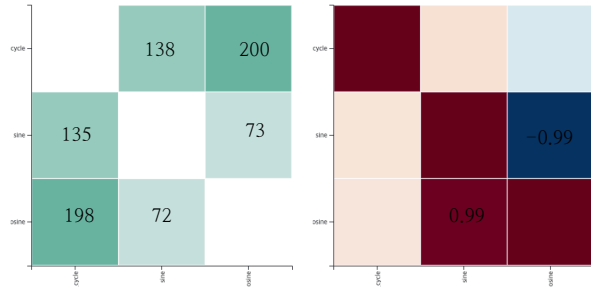
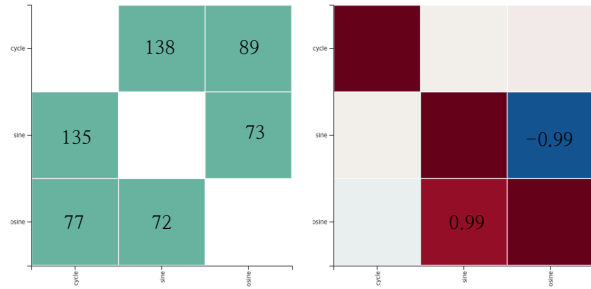
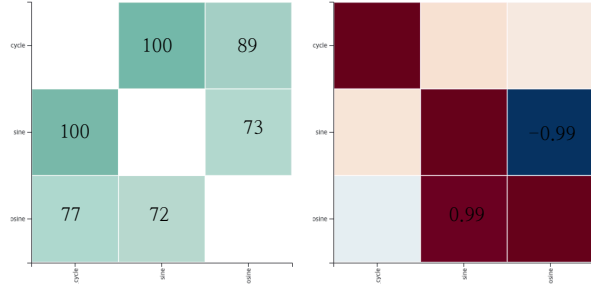
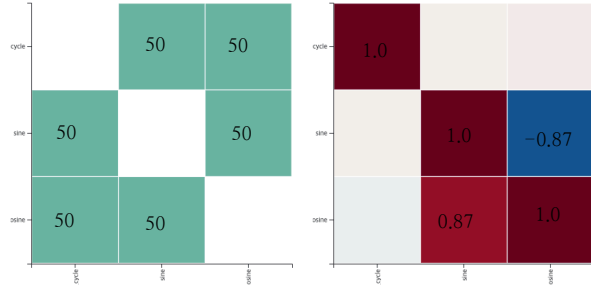


Figure 19 – Experiment results with synthetic data, changing time shift range from 3 0

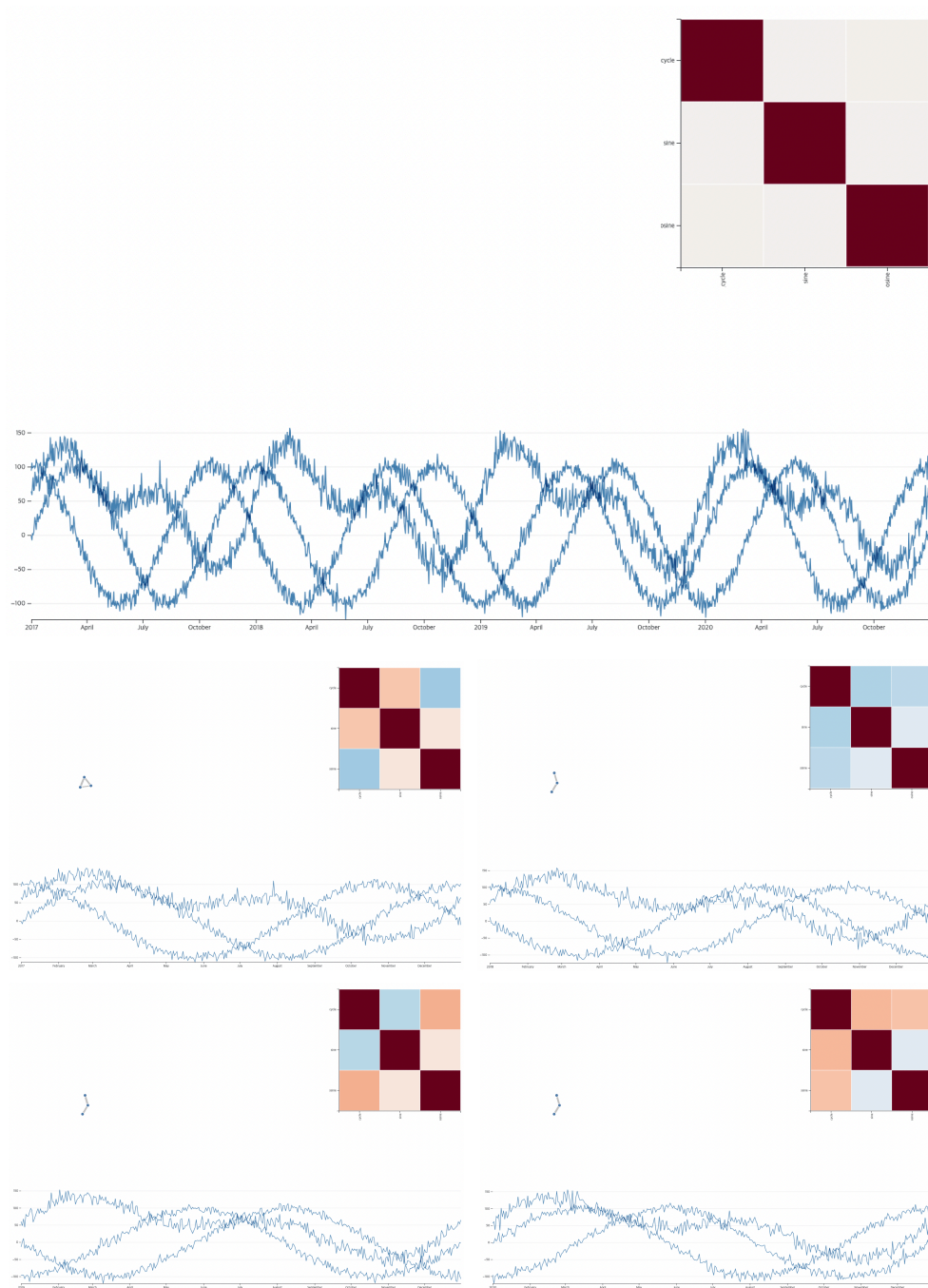


50, 100, 150, 200 to 250 under threshold level 0.1 and k-Medoids as the clustering method

For the time-segmentation view, the number of segments that the user could choose was 1, 4, 8, 12, and 16. If the user chooses to see the correlations under 16 segments, then the user can choose next a number between 0~15 as the segment number to look into. Two of the clustering options were k-Medoids and Louvain, while the threshold options were 0.1, 0.3, 0.5, 0.7 and 0.9, The results of this finding show that when the time series data is segmented, it seems to show some level of correlation although the correlations are almost close to 0 when the Pearson correlation values are calculated for the whole dataset. The detailed results are shown in figure 18. The trend seemed to show that as the time series data is segmented into smaller units, the absolute correlation values may tend to go higher then when more longer periods are observed.

The results at least with this data showed that perhaps exploring into time-segmentation may have some drawbacks, as one may conclude that some variables are related to one another in a segmented view, but is actually not when a longer period of the data is observed. Perhaps a deeper and more of a multifaceted view may need to be derived to look into the time-segmentation view in a meaningful way.

Overall, through this sample dataset we found that the time-shift view functions as we have planned, but that the time-segmentation view may need further exploration and complements to give useful and meaningful visualization results for the users. However, since this case of synthetic data, it did not have enough columns to thoroughly explore the network graph visualization and the effects of each clustering method. The value of network graph visualization method and each clustering method may be further explored in the next two examples.



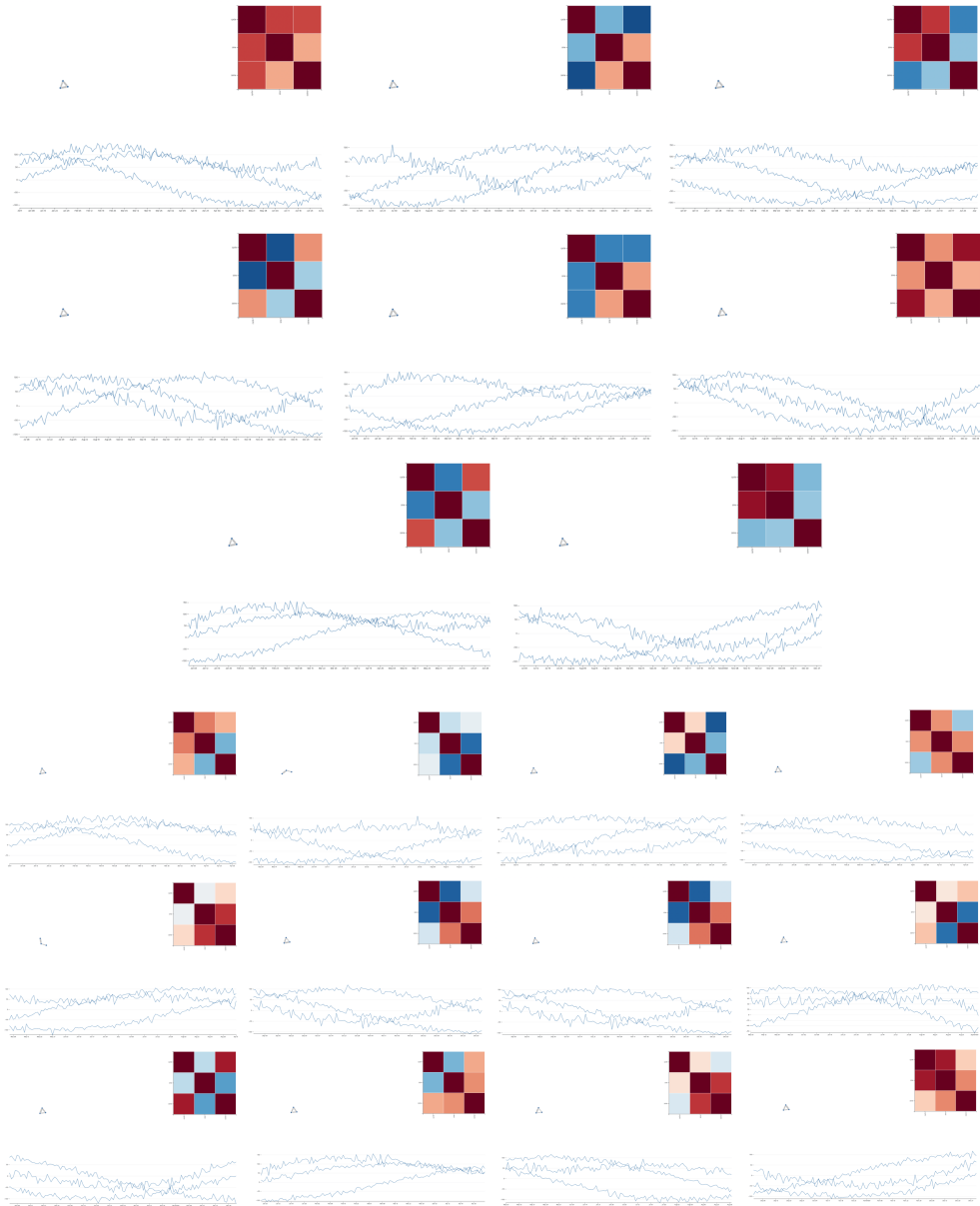


Figure 20 – Time segmentation view with the synthetic data, with k-Medoids as the clustering method, 0.1 as the threshold. From top to bottom, left to right, are screenshots of segments when the data is segmented into 1, 4, 8, 12, 16 segments respectively.

## 5.2. Weather Data Exploration

By looking into the weather data with four columns related to the weather condition in Delhi, India, we were able to find which columns had positive and negative correlations. As shown in figure 19, the temperature and humidity had a negative correlation without time shifts when the time shift range was given to 10. We were able to conjecture which data was closely related even though we were not experts in this field. However, through exploring with the visualization system, we were able to find that the dataset could have a cycle around a length of a year. A figure of a time-shift view of the weather data is shown in figure 21.

We tested to give various time ranges to check the Time-Lagged Cross-Correlations. The given time ranges were 10, 30, 60, 90, 200, and 400. We found that the mean temperature, humidity and wind speed columns overall have high absolute correlations. On the other hand, the wind pressure column did not seem to show high absolute correlations with the other three columns. Especially the temperature and humidity, wind speed and humidity constantly show negative correlations when the time lag is 0 over the time ranges. Another characteristic we can find is that when the wind speed is lagged in respect to the temperature, it constantly shows the highest positive correlation when the time lagged is 43 days when the time range is high enough. The screen shots are shown in figure 22.

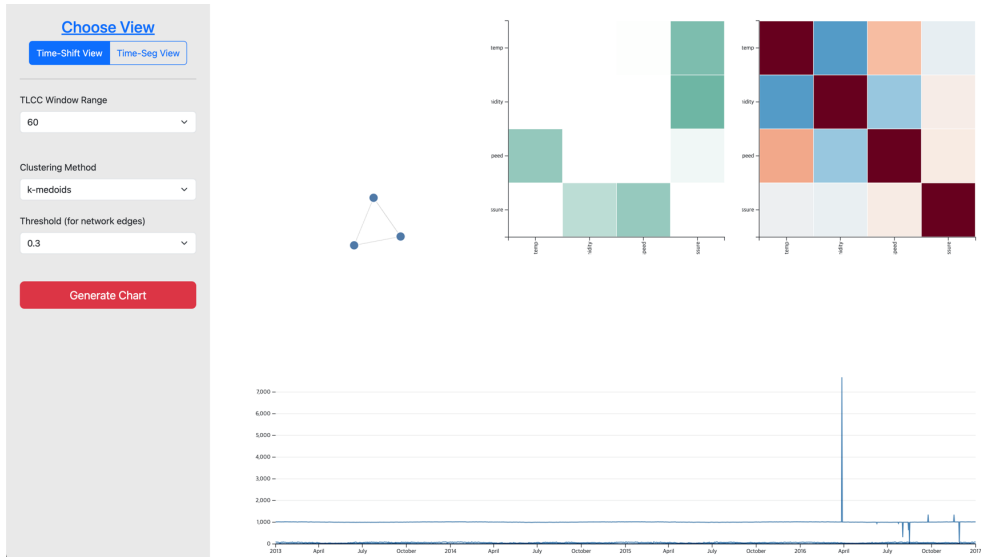


Figure 21 – Time shift view of the weather data

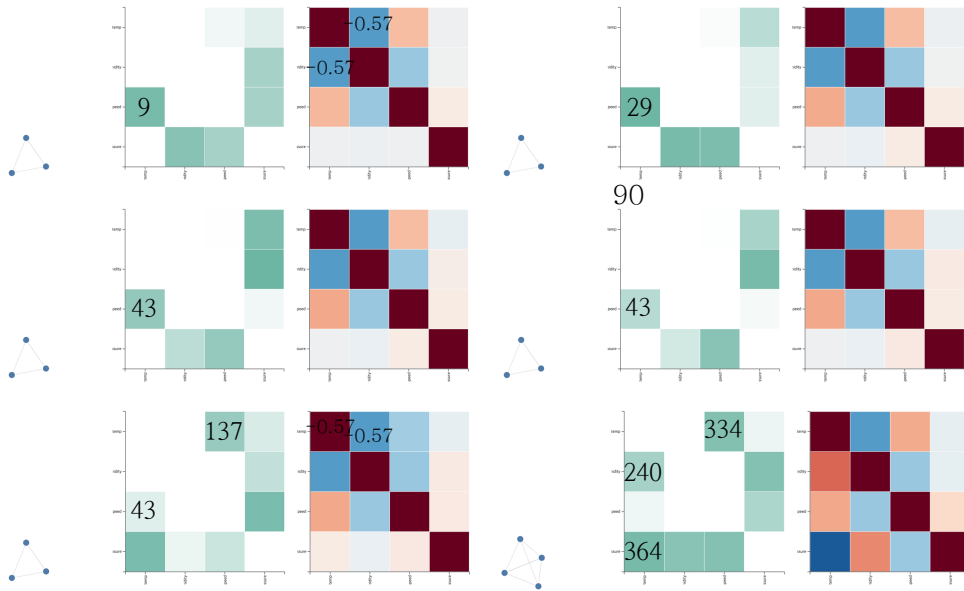


Figure 22 – Screenshots of time-shift view of the weather data, when the time ranges are 10, 30, 60, 90, 200 and 400 respectively from left to right, top to bottom with the threshold of the edges is set at 0.3.

One thing to note was that in case of the wind pressure column, it showed low level of correlation with other columns, and the time lag heatmap tended to show high level of time lags. This may point to the fact that possibly columns that have varying level of time lags for each time range and columns that show dark colors in the time lag correlation heatmap may tend to have low level of actual relationship with other columns. This may mean that maybe looking into correlations with which its corresponding time lag does not vary often may produce meaningful results. In this case, the correlations between the three columns, mean temperature, humidity and wind speed may have meaningful correlations because the time lag heatmap shows some consistency over the various time ranges.

In addition, when the time range is set to 400, the colors of the correlation heatmap becomes darker overall, with the network graph of all the columns completely connected at the threshold level of 0.3. Also, considering the exploration with the synthetic data example, we can conjecture that the wind speed and temperature columns may have some similarity in its shape with a cycle length of around 377 ( $= 334 + 43$ ) days. The wind pressure column also seems to have higher correlations with other columns when this column is shifted in regard to the other three columns, with time lags under 364 days. For this characteristic, we will have a look into the time segmentation view of the data.

Next, we looked into the time-segmentation view with the weather data. The number of segments given to look at were 1, 4, 8 and 16. These number of segments were given to look at the whole dataset, yearly view, semi-annul view, and quarters of every year respectively, given the fact that the timestamp of the dataset ranges over a period of 4 years. The screenshot of the time-segmentation view is shown in figure 23.

Some interesting patterns were observed when the time series data was split into several segments. When the correlation for the whole data was

calculated, the wind pressure column seemed to have low correlations with the other columns. However, when the data was split into 4 segments, in other words yearly data, the wind pressure column actually showed high level of correlation as well as the other three columns for the first three years, and only showed low correlation values in the last one year of the data. The total correlation of the wind pressure column when there was only one segment, which showed low absolute correlation seems to have been affected by the fourth year of the data. By looking into the raw data through the line chart visualization, it seems that there are some outliers in the wind pressure dataset in the fourth year of the dataset. Therefore, some process for taking care of outliers may be needed in the data cleaning process, or we can check if there were actual exceptional phenomenon in the year 2016 of Delhi, India. In addition, when we look into the 8-segment view and 16 segment view, we can notice that there are some semi-annual and quarterly trends each year in the data.





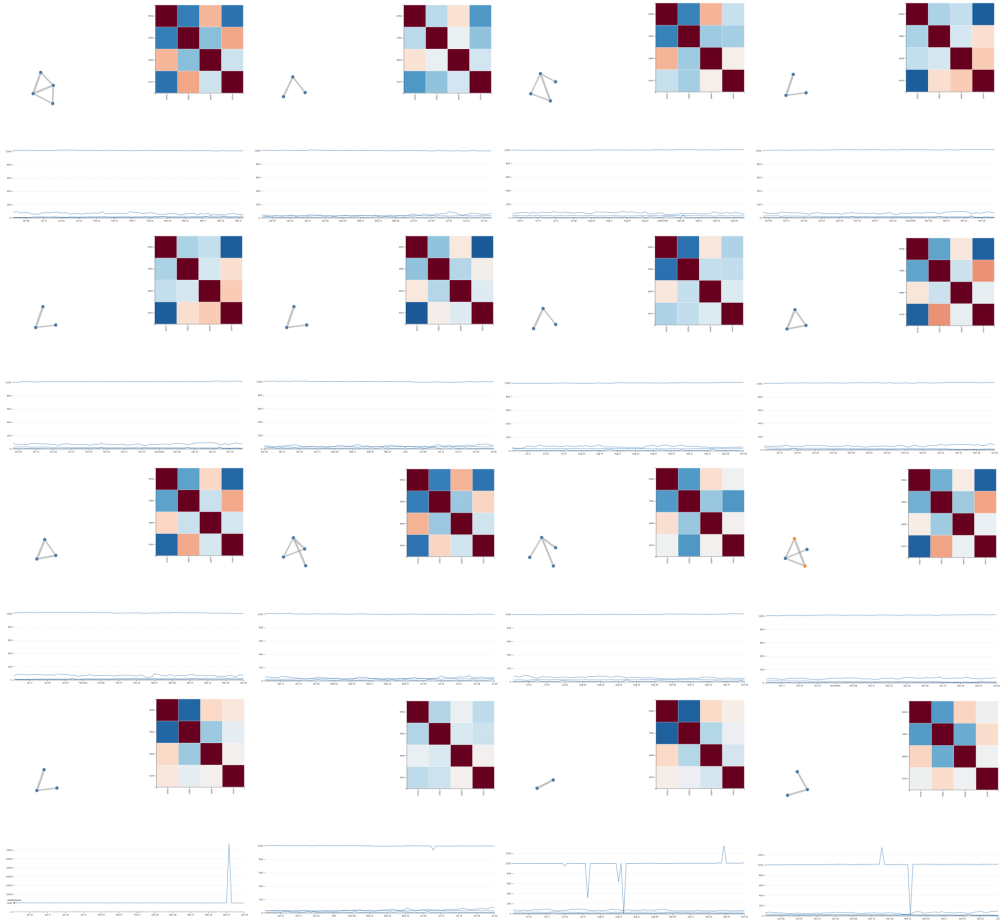


Figure 23 – Screenshots of time-segmentation view of the weather data, when the numbers of segment(s) are 1, 4, 8, 16 respectively from left to right, top to bottom with the threshold of the edges is set at 0.3, with clustering method set to k-Medoids.

## 5.2. Stock Market Data Exploration

By looking into the weather data with four columns related to the weather condition in Delhi, India, we were able to find which columns had positive and negative correlations. As shown in figure 19, the temperature and humidity had a negative correlation without time shifts when the time shift range was given to 10. We were able to conjecture which data was closely related even though we were not experts in this field. However,

through exploring with the visualization system, we were able to find that the dataset could have a cycle around a length of a year. A figure of a time-shift view of the weather data is shown in figure 21.

With the stock market data, we were able to capture interesting findings by looking into the time-shift view of the visualization system. One of the meanings of looking into the stock market data was that we were able to look into the meaning of clusters in this system. Also, we found that interestingly, the lines with negative correlations tend to have higher time lags, which were some facts to look into. With more columns in the data, we were able to find higher number of clusters in the network graph visualization. A sample time shift view is shown in figure 24.

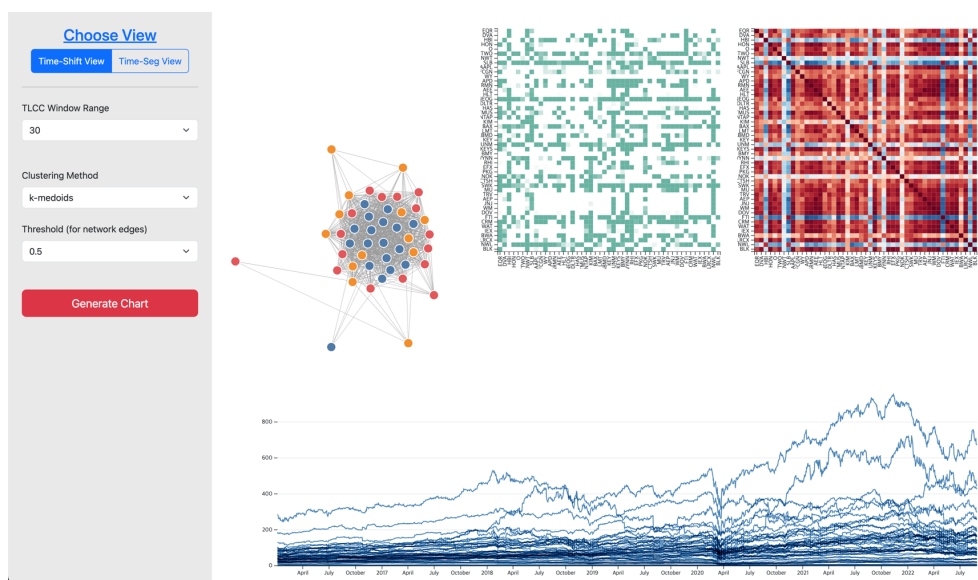


Figure 24 – A time-shift view of the stock market data

The stock market dataset was explored with time range of 30, 60 and 90 days. The screen shot for each time range is given below in figure 25. Although background knowledge in this field may be needed for deeper understanding of the data, the dataset seems to show high level of positive correlation among the columns, possibly because most stocks follow the trend of the stock market. We can conjecture from the previous exploration

of datasets that correlations that have high corresponding time lag values even with varying time ranges may have insignificant relationships. While most columns showed positive correlations, some columns that showed negative correlations seemed to show a high number of time stamps lagged over various time ranges, which the explorer may look into and may come to a conclusion that it does not have significant correlation compared to relationships within other columns.

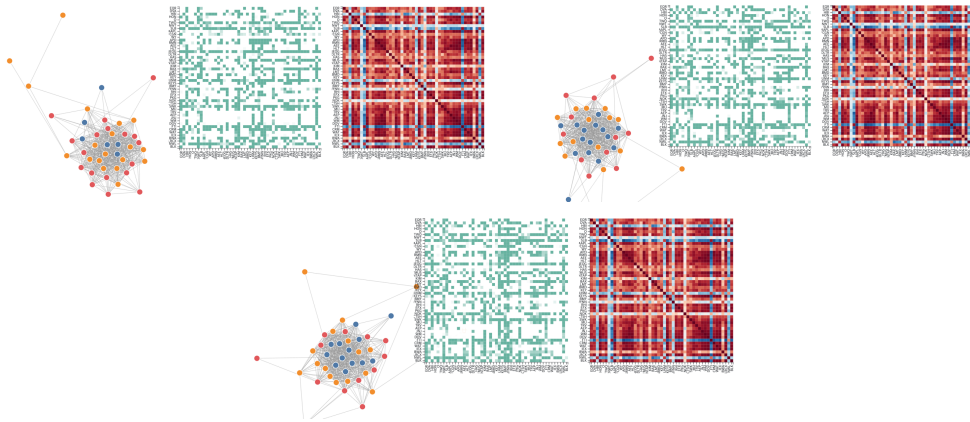


Figure 25 – Screenshots of time-shift view of the stock market data, when the time ranges are set to 30, 60 and 90 days respectively from left to right, top to bottom with the threshold of the edges is set at 0.7, with clustering method set to Louvain community detection algorithm.

To have a closer look into the threshold of the network, we changed the threshold level from 0.5, 0.7 to 0.9 when the time range was 90 and the clustering method was set to Louvain community detection algorithm. As the threshold level was raised, the explorer was able to find columns that are very closely correlated to one another. When the threshold level was low, the user was able to explore the communities or clusters formed with a larger portion of the columns of the dataset included. The images are shown in figure 26.

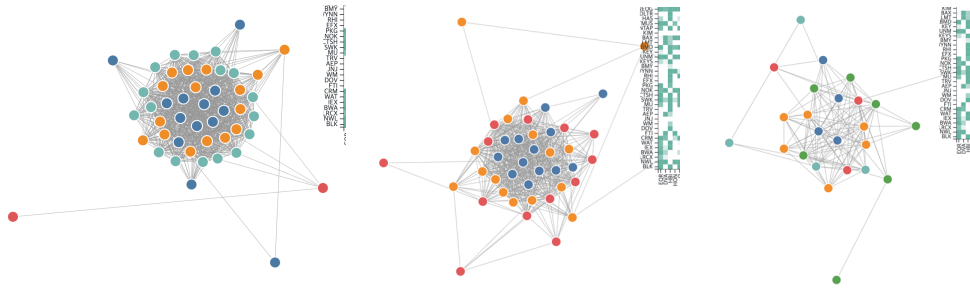


Figure 26 – Screenshots of network graphs formed, when the threshold levels are 0.5, 0.7, and 0.9 respectively from left to right, top to bottom with clustering method set to Louvain community detection algorithm.

Going on to the time-segmentation view, we were able to check that in the case of stock market data, the user of the visualization system can look into time periods with specific events that the user wishes to look at. An example view of time-segmentation view with the stock market data is given in figure 27. Also, screenshots of view of the data when it is segmented into 18 segments are shown in figure 28. As an example, segment number 8 and 11 may show time periods with notable events in the stock market trend. Segment number 12 shows one of the periods when the columns of the data are relatively not as correlated as other periods.

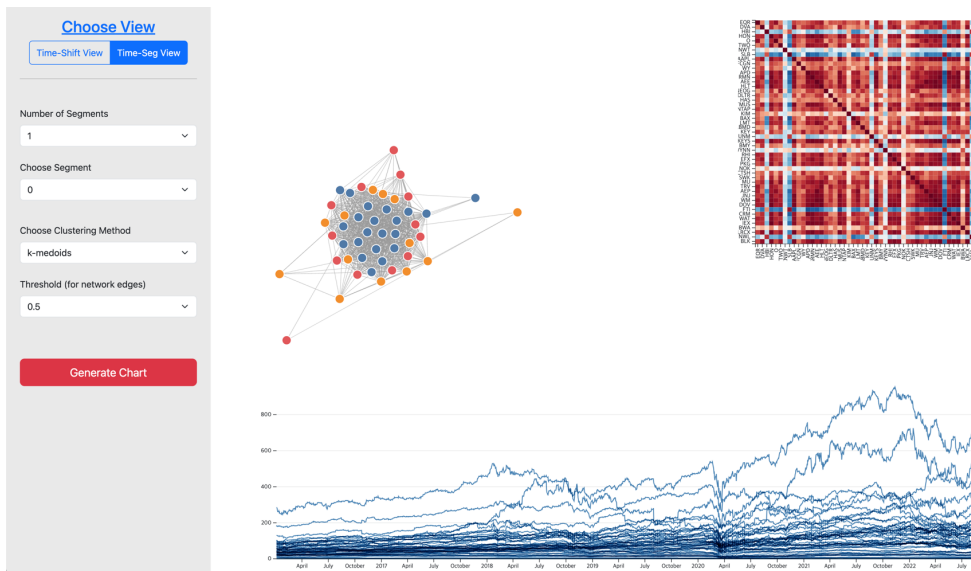


Figure 27 – A time-segmentation view of the stock market data

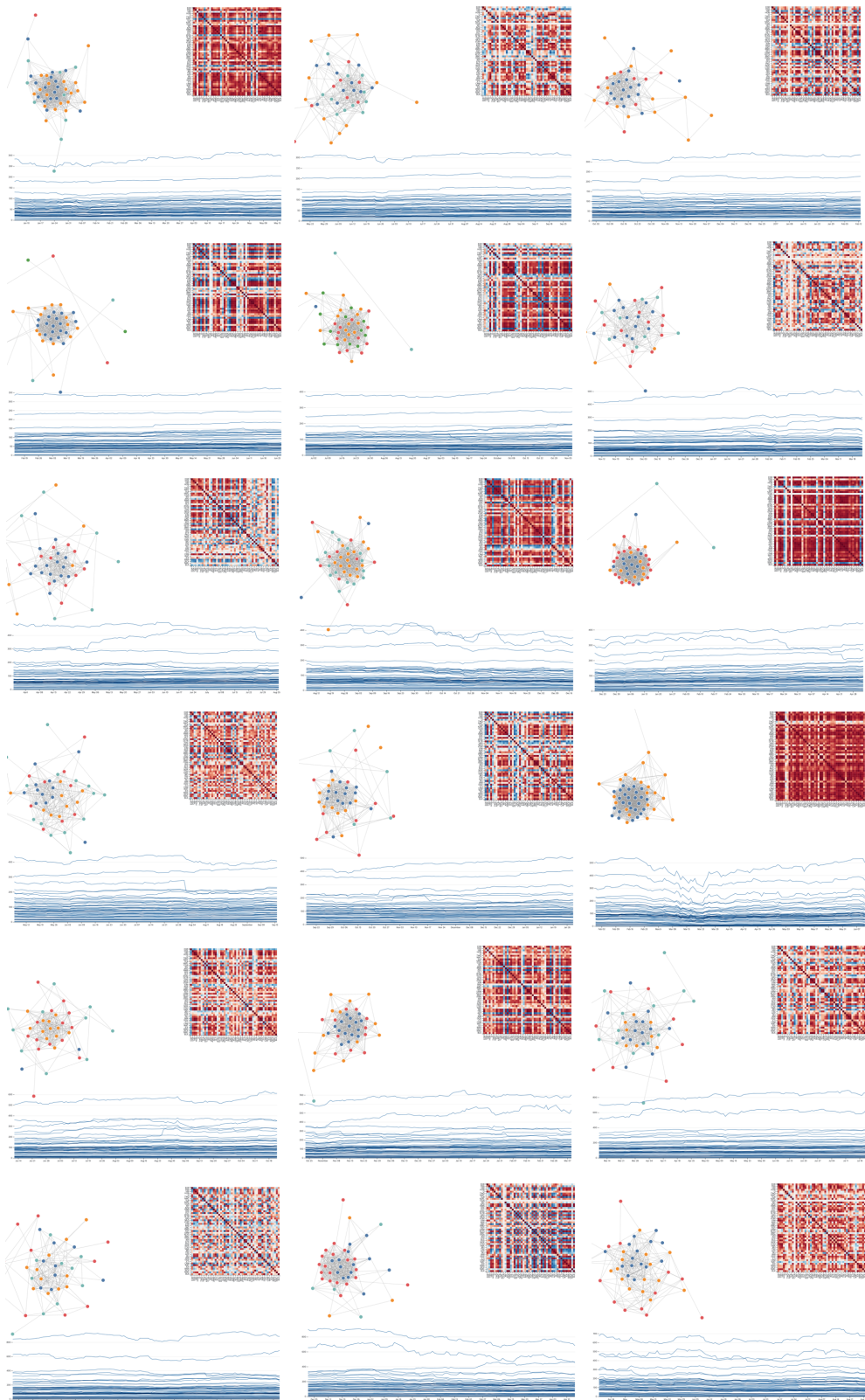


Figure 28 – Screenshots of time-segmentation view of the stock market data, when the numbers of segment(s) are 18, starting from left to right, top to bottom with the threshold of the edges is set at 0.7, with clustering method set to Louvain community detection algorithm.

## Chapter 6. Discussion

This section introduces some options that were considered in the visualization system but were not implemented. The main reasons for this fact that as the system was in the process of development, we considered which options would be the most crucial options, and that options that were thought to have secondary importance should be added in the future after further exploration of the time series data with the given tool. Since practical usage of this visualization tool is important, decisions for which options should be added could be possibly added with further exploration with real data.

### 6.1. Option to Choose a Specific Lagged Time for the Time-Shift

One of the options that were considered in this way was an option that enables the user to choose the specific shifted time instead of the entire result of the Time-Lagged Cross-Correlations algorithm. If the Time-Lagged Cross-Correlation algorithm was found to be an effective method to explore time series data, the user of this tool may want to have a detailed view of the process that this algorithm that was run on the data.

Specifically, if the user chooses the range of the time-shift window, then the user can view the correlation results for each step of the shifts as well as the total result of the Time-Lagged Cross-Correlation algorithm. For example, if the window range was set to 20, the viewer can also look into the heatmaps, network, and line chart visualization when the shifts were given through 0~20 for every pair of columns. In this case, the line chart will stay the same while the heatmaps and networks differ for each time step.



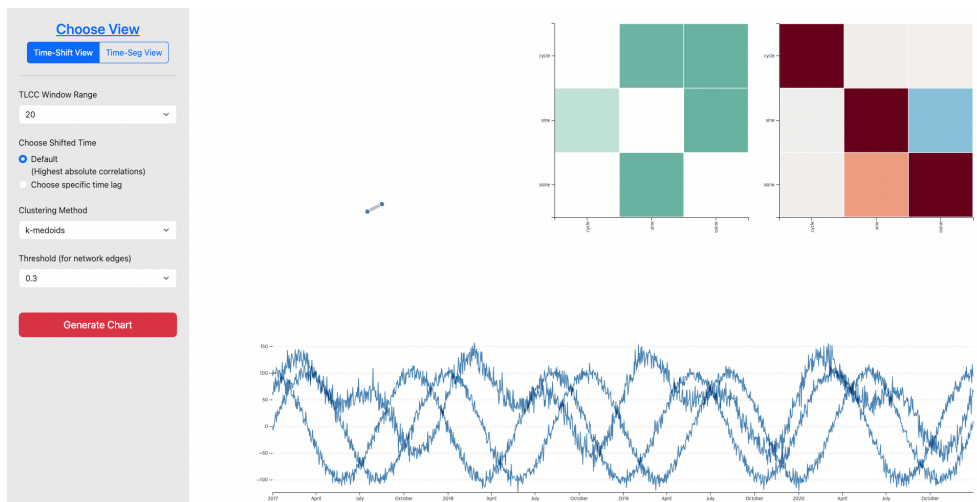


Figure 29 – Interface with the option to choose a specific lagged time under the Time-Shift View

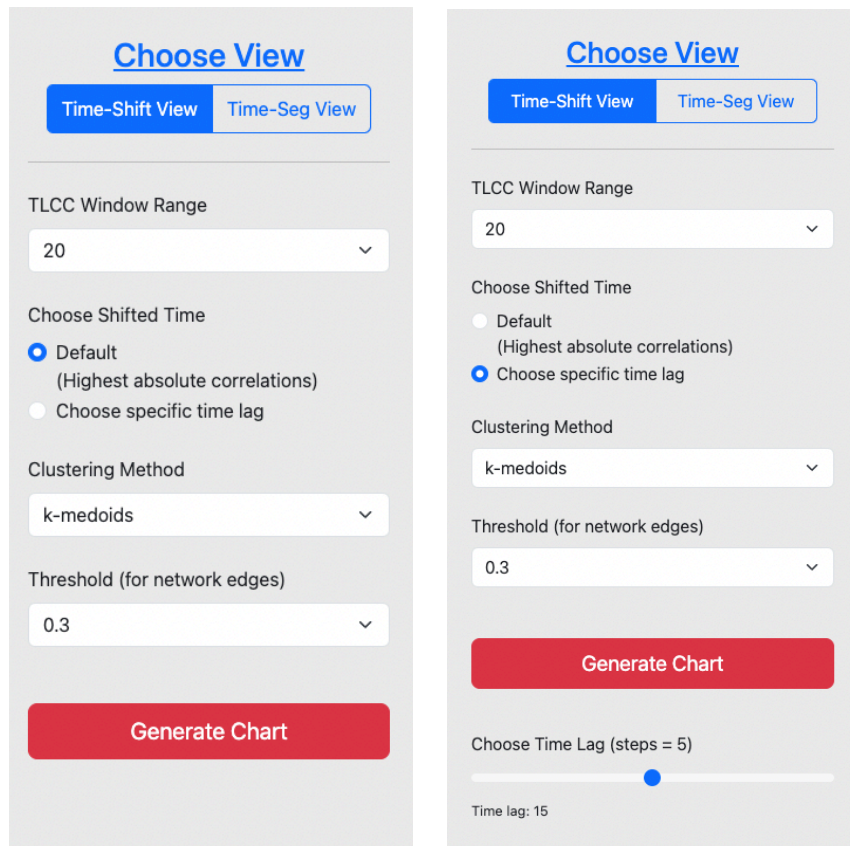


Figure 30 – The side bar when the default option for the time-shift view is chosen, versus when the user chooses to view correlations given for specific time lags using slide bars when the option other than 'Default' is chosen.



Although this option may be useful when the information for each time shift is important, we were not able to find out if this would be one of useful information in time series data analysis in correlations. For example, drawing networks for edges formed for each step of the time shift and viewing the heatmaps may not be as effective, and could be even superfluous information. Therefore, the considered option in the next subsection was also considered that could capture the change of correlations through steps in time shifts in a more effective and information-compact visualization.

## 6.2. Line Chart Visualizing the Change of Correlations over the Given Time-Shift Range

Another option that was considered was a line chart that allows the user to view the change of correlations over the amount of shifted time periods. Therefore, the x-axis of the line chart becomes the amount of time lagged, and the y-axis becomes the correlation value of a pair of columns. The range of the x-axis would be integer numbers starting from 0 to the number given as the input for the range of time-shift the user wants to explore. The range of the y-axis would be  $[-1, 1]$ , since the values indicate the correlation values for each time shift given. The number of lines drawn in this line chart would be  $n * (n-1)$  when there are  $n$  number of columns in the dataset. This is because two lines are drawn for a pair of columns, let us say column A and column B, for each case when column A is shifted in respect to column B and vice versa.

This line chart would be useful when the user would like to check the validity of the correlations derived from the Time-Lagged Cross-Correlations in a sense, because the results may be deemed to be trustworthy if the correlation values over the shifted time range changes with some continuity while correlation values are calculated for each time shift. In addition, since the Time-Lagged Cross-Correlations derives the point where the absolute value of the correlation value reaches its peak, the algorithm

may not clearly show cases where a pair of columns show both high positive correlation value and negative value at two different points. By looking into the line chart, the user will easily spot all the local minimum and maximum points other than the global minimum or maximum point derived by the algorithm.

### **6.3. Heatmaps to Visualize the P-values Corresponding to Each Correlation Value**

Another method of validating the correlation values derived from the Time-Lagged Cross-Correlation algorithm is to check the p-values for each of the correlation values derived. It is one of the methods for checking the validity of correlation values. Therefore, we considered visualizing this information also into heatmaps. Such heatmap could be added along with the heatmaps visualizing the correlation values and lagged time at the point where the absolute correlation value reached its peak in the case of time-shift view, and with the heatmap visualizing correlation value for each segment of data in the case of time-segmentation view.

## Chapter 7. Conclusion and Future Work

This paper proposes an interactive visualization tool that allows the user to explore time series data using Time-Lagged Cross-Correlation and time-segmentation in various facets using network graphs, heatmaps and line charts. This allows data analysts and data scientists to thoroughly explore the relationships of time series data either in the EDA process or even in the process of analyzing the data. The tool will aid the users to look into the relationships among the variables in a deeper way compared to when they generate simple heatmaps to visualize Pearson correlations.

Although the visualization explores Time-Lagged Cross-Correlation between variables, there may be other possible relationships to explore, such as logarithmic relationships, exponential relationships, etc. There are also other similarity or distance measures such as the Euclidean Distance or Dynamic Time Warping (DTW). Through further research, we may develop as future work an interactive visualization system that allows the users to explore these different relationships and methods, and perhaps compare the different outcomes of using these methods.

Moreover, the current visualization system is more of a simple version of a visualization system that has the potential to become richer in expanding on its options. For example, in the case of clustering methods, the current visualization system offers only two options that each represent a more conventional method of clustering data and a community detection algorithm that is more novel in clustering time series data. While the visualization allows the user to compare and contrast the differences of using these two clustering methods, we may need to increase options such as k-Means, Girvan-Newman community detection algorithms, etc. This will enlarge the potential of looking into correlations in various aspects and trials with different methods that fit best for each dataset to explore.

These options may become more useful if the visualization system offers a real-time analysis function embedded in it, that allows the user to input the time range window for the Time-Lagged Cross-Correlation algorithm etc., unlike the current way where the options are already set at the analysis step and the user can only choose among the set options with the visualization system. For more development in these option choices and functions, further study should be made with datasets to figure out which options would be useful and economical, to not try to make superfluous options.

In addition, we conjecture that this tool will be mainly useful in the process of exploring data. However, further research may be done to develop a tool to explain the process of analyzing time series data using machine learning and deep learning. Especially in the case of Graph Neural Networks, or GNN, we look into the possibility of connecting with our visualization system to help in its process of analyzing data. By capturing the trends of the changes in the relationships between two variables, the tool may aid improving the performance of deep learning algorithms such as these.

# Bibliography

- [1] Hailin Li et al., Time series visualization based on shape features, ScienceDirect, 2013 Jurgen Bernard et al., Visual-Interactive Preprocessing of Multivariate Time Series Data, Computer Graphics forum, 2019
- [2] Jurgen Bernard et al., Visual-Interactive Preprocessing of Multivariate Time Series Data, Computer Graphics forum, 2019
- [3] Anders Holst et al., Interactive Clustering for Exploring Multiple Data Streams at Different Time Scales and Granularity, ACM, 2019
- [4] Zafar Ahmed et al., An Adaptive Parameter Space-Filling Algorithm for Highly Interactive Cluster Exploration, IEEE Conference on Visual Analytics Science and Technology (VAST), October 2012
- [5] Jungen Bernard et al., Guided Discovery of Interesting Relationships Between Time Series Clusters and Metadata Properties, ACM, 2012
- [6] Dominik Sacha et al., SOMFlow: Guided Exploratory Cluster Analysis with Self-Organizing Maps and Analytic Provenance, IEEE Transactions on Visualization and Computer Graphics, 2018
- [7] Mohammed Ali et al., Clustering and Classification for Time Series Data in Visual Analytics: A Survey, IEEE Access, 2019

- [8] Hailin Li et al., Multivariate time series clustering based on complex network, ScienceDirect, 2021
- [9] Chao-Lung Yang et al., Multivariate Time Series Data Transformation for Convolutional Neural Network, IEEE, 2019
- [10] Shengdong Du et al., Multivariate time series forecasting via attention-based encoder-decoder framework, ScienceDirect, 2019
- [11] Hang Zhao et al., Multivariate Time-Series Anomaly Detection via Graph Attention Network, IEEE International Conference on Data Mining (ICDM), 2020
- [12] Ailin Deng et al., Graph Neural Network-Based Anomaly Detection in Multivariate Time Series, AAAI, 2021
- [13] Julien Audivert et al., USAD: UnSupervised Anomaly Detection on Multivariate Time Series, KDD '20, 2020
- [14] Nan Ding et al., Multivariate-Time-Series-Driven Real-time Anomaly Detection Based on Bayesian Network, Sensors, 2018
- [15] Xiang Yin, Multi-Attention Generative Adversarial Network for Multivariate Time Series Prediction, IEEE Access, 2021

- [16] Zonghan Wu et al., Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks, KDD '20, 2020
- [17] Lena Cibulski et al., COMPO\*SED: Composite Parallel Coordinates for Co-Dependent Multi-Attribute Choices, IEEE Transactions on Visualization and Computer Graphics, 2022
- [18] Teng-Yok Lee, Visualization and Exploration of Temporal Trend Relationship in Multivariate Time-Varying Data, IEEE Transactions on Visualization and Computer Graphics, 2009
- [19] Patrick Riehmann et. al, Interactive Sankey Diagrams, IEEE Conference Publication, November 2005
- [20] Vung Pham et al., MTSAD: Multivariate Time Series Abnormality Detection and Visualization, IEEE International Conference on Big Data (Big Data), 2019
- [21] Jishang Wei et al., Visual cluster exploration of web clickstream data, IEEE Conference on Visual Analytics Science and Technology (VAST), October 2012
- [22] Zipeng Jiu et al., Visualizing Graph Neural Networks With CorGIE: Corresponding a Graph to Its Embedding, IEEE Transactions on Visualization and Computer Graphics, 2022

- [23] Aindrila Ghosh et al., VisExPreS: A Visual Interactive Toolkit for User-Driven Evaluations of Embeddings, IEEE Transactions on Visualization and Computer Graphics, 2022
- [24] Zuchao Wang et al., Visual Traffic Jam Analysis Based on Trajectory Data, IEEE Transactions on Visualization and Computer Graphics, Vol. 19, 2013
- [25] Joscha Elrich et al., IRVINE: A Design Study on Analyzing Correlation Patterns of Electrical Engines, IEEE Transactions on Visualization and Computer Graphic, 2020
- [26] Benjamin Bach et al., Small multipiles: Piling time to explore temporal patterns in dynamic networks, Computer Graphics Forum, vol. 34, 2015
- [27] Florian Ferstl et al., Time-Hierarchical Clustering and Visualization of Weather Forecast Ensembles, IEEE Transactions on Visualization and Computer Graphics, 2016
- [28] Konstantinos Fokianos et al., Testing independence for multivariate time series via the auto-distance correlation matrix, Biometrika, Volume 105, Issue 2, Pages 337–352, June 2018
- [29] David S. Matteson et al., Time-Series Models of Dynamic Volatility



and Correlation, IEEE Signal Processing Magazine, Volume 28, Issue 5, September 2011

- [30] Youjin Lee et al., Network dependence testing via diffusion maps and distance-based correlations, Biometrika, Volume 106, Issue 4, Pages 857–873, December 2019

## Abstract

이 논문은 시계열 데이터 내의 상관관계를 다양한 각도에서 조사하기 위한 대화형 시각화 도구를 제안한다. 데이터 탐색 과정에서 변수들 간의 피어슨 상관관계를 시각화하는 것은 흔한 방법이지만, 시계열 데이터의 경우에는 단순히 상관관계를 살펴보는 것만으로는 충분하지 않을 수 있다. 시계열 데이터의 상관관계를 탐색하는 중에 간과할 수 있는 두 가지 주요 문제가 있다고 보았다. 첫째, 시계열 데이터에서 변수들은 특정한 시간 차이를 두고 높은 양의 또는 음의 상관관계를 보일 수 있다. 둘째, 주어진 시계열 데이터에서의 상관관계는 시간이 지남에 따라 변할 수 있다. 이러한 문제를 해결하기 위해 본 논문은 사용자가 변수 간의 시간 차이에 따른 상관관계를 살펴보고 시간 구간별로 상관관계의 변화를 관찰할 수 있는 대화형 시각화 시스템을 제안한다. 먼저 이 시스템에서 시간 이동 뷰는 시간 교차 상관관계 알고리즘을 활용하여 시간 차이를 반영한 시각화를 제공한다. 이 알고리즘은 주어진 범위 내에서 두 변수 중 하나의 시간대를 이동시키며 얻은 절대값이 가장 높은 상관관계 값을 계산한다. 또한, 시간 분할 뷰는 구간별로 상관관계의 변화를 관찰하기 위해 시계열 데이터를 여러 구간으로 나누어 상관관계에 대한 시각화를 생성한다. 시계열 변수들의 상관관계를 시각화한 방법으로는 시계열 데이터나 상관관계를 일반적으로 사용되는 방법인 히트맵과 라인차트뿐 아니라 네트워크 그래프를 사용하였다. 또한, 커뮤니티 탐지 알고리즘을 사용하여 네트워크의 노드로 표현된 시계열 데이터의 변수들을 그룹화 또는 클러스터링하여 각 노드의 색상을 지정하였다. 본 논문에서 제안한 도구의 효과를 확인하기 위해 여러 시계열 데이터 예시에 적용해 보았다. 이러한 예시를 통해, 시간 이동 뷰를 통해 주기가 있는 데이터에서 패턴을 발견하는 것, 시간 분할 뷰를 통해 상관관계의 변화를 확인하는 데 본 시각화 시스템이 유용하다는 것을 알 수 있었다. 논문에서는 또한 향후 연구에 대한 가능성도 언급하였다.