Economic University of Thessaly – MSc Econophysics

Διατμηματικό Διιδρυματικό Πρόγραμμα Μεταπτυχιακών Σπουδών

Πανεπιστήμιο Θεσσαλίας -
Διεθνές Πανεπιστήμιο της Ελλάδος

# Black Swans 2022-2023

A Complex Network Time Series Analysis Approach


***Chatzikomnitsas Christos***

## Abstract

The occurrence of Black Swan events, characterized by their extreme rarity and significant impact, has posed substantial challenges to the financial industry. In this thesis, we investigate the presence and potential implications of Black Swan events in the financial markets during the years 2022-2023. To analyze and understand these phenomena, we employ a combination of basic time series analysis and complex network approaches.

The study begins with a comprehensive examination of historical financial data, focusing on identifying and characterizing extreme events that deviate from the expected market behavior. We employ basic time series analysis techniques to detect patterns, trends, and anomalies within the financial time series data. By exploring statistical measures, such as volatility, skewness, and kurtosis, we aim to unveil potential indications of Black Swan events during the specified time frame.

Furthermore, we extend our analysis by incorporating complex network methods, recognizing that financial markets exhibit intricate interconnectedness. Through the construction of temporal networks, we capture the dynamic relationships between various financial instruments, institutions, and market segments. This approach enables us to identify critical nodes, measure centrality, and study the propagation of shocks or disruptions within the financial network.

Our findings highlight the importance of combining basic time series analysis with complex network approaches in understanding the occurrence and impact of Black Swan events in finance. By integrating these methodologies, we gain insights into the underlying patterns, systemic risks, and cascading effects associated with extreme events. This research contributes to the broader understanding of risk management, financial stability, and the need for proactive measures to mitigate the adverse consequences of Black Swan events in the financial industry.

By leveraging basic time series analysis techniques and complex network methodologies, this report provides valuable insights into the identification, analysis, and implications of Black Swan events in the financial markets for the years 2022-2023. The findings contribute to enhancing risk assessment frameworks, fostering financial resilience, and improving decision-making processes in an ever-evolving and unpredictable financial landscape.

**Υπεύθυνη δήλωση**

Βεβαιώνω ότι είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην μεταπτυχιακή διπλωματική εργασία.
Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του Διιδρυματικού Διατμηματικού Προγράμματος Μεταπτυχιακών Σπουδών Οικονομική Φυσική – Χρηματοοικονομικές Προβλέψεις, Πανεπιστήμιο Θεσσαλίας, Τμήμα Οικονομικών Επιστημών, Τμήμα Φυσικής, Διεθνές Πανεπιστήμιο της Ελλάδος, Τμήμα Φυσικής
Βόλος, Ιούνιος, 2023

## Contents

## List of Tables



## List of Figures

# 1.Introduction – Black Swans 2022/23

## 1.1 Definition of Black Swans

In finance, a "Black Swan" refers to an unpredictable and rare event that has a severe impact on the financial markets or the economy as a whole. Coined by Nassim Nicholas Taleb in his book "The Black Swan" (1), the term is used to describe an event that is highly unexpected, has a significant deviation from normal expectations, and is often retrospectively rationalized.

Black Swan events are characterized by their extreme rarity, their high impact, and the difficulty in predicting or foreseeing them using traditional statistical models or historical data. These events can lead to significant disruptions, market volatility, and financial losses. Examples of Black Swan events in finance include the 2008 financial crisis, the dot-com bubble burst in the early 2000s, or the global market crash of 1987.

The concept of Black Swan highlights the limitations of traditional risk management approaches that rely on historical data and assume that future events will resemble the past. It emphasizes the need for robust risk assessment methods, the recognition of uncertainty, and the development of strategies to handle extreme and unforeseen events. Below we discuss the black Swans of 2023.

## 1.2 Black Swans of 2022-2023

### 1.2.1 UK Gilt crisis

In September 2022, the U.K. experienced a significant crisis in its Gilt market. Over a span of less than three days, 30-year U.K. Gilt yields surged by more than 1.60 percent, causing unprecedented volatility. This sudden increase posed a severe challenge for U.K. Defined Benefit (DB) pension schemes, which typically manage their liquidity based on gradual bond yield changes over a week or more.

The rapid and substantial market movement exceeded the contingency plans of most institutions, leading to the need for generating collateral to meet substantial margin calls on interest rate swap and FX forward positions. Pension funds and other buy-side institutions were compelled to adapt their contingency plans quickly in response to unexpected market shifts. To prepare for future uncertainties, these institutions recognized the importance of maximizing collateral flexibility by utilizing a wide range of assets and optimizing them swiftly and effectively.

The U.K. Gilt crisis was triggered by concerns over the country's fiscal position, resulting in a fall in Gilt prices and a subsequent rise in yields. The magnitude of the crisis was evident in the intraday range of 127 basis points on the 30-year Gilt, which exceeded the annual range for most of the past 27 years. Fund managers, who hold significant amounts of Gilts to fulfill non-cleared trades obligations, faced a challenging situation. They were also required to post cash as variation margin (VM) on cleared trades, further exacerbating the need for collateral.

The size of U.K. DB liabilities compared to the overall Gilt market was substantial, with £1.4 trillion in liabilities and £2.1 trillion in Gilts. As managers started selling their liability-driven investments (LDI) and Gilts to raise cash, it triggered a cycle of falling prices. The Bank of England intervened by purchasing Gilts to stabilize prices and halt the negative feedback loop.

LDI plays a crucial role in the investment strategy of U.K. pension schemes, aiming to align assets with liabilities and mitigate risks associated with interest rate and inflation fluctuations. The crisis exposed the vulnerability of leveraged plans that used derivatives, introducing liquidity risk to the system. Collateral sufficiency and capital buffers needed to be recalibrated, with a greater emphasis on reducing leverage and increasing collateral coverage.

While the U.K. experienced this Gilt crisis, it is unlikely to have a similar impact on the U.S. LDI market. Structural differences, such as the valuation of pension liabilities on different curves and the use of derivatives and leverage, contribute to varying risk profiles. U.S. pension funds rely more on physical long-duration bonds for hedging, resulting in a more responsive approach to interest rate changes.

In conclusion, the U.K. Gilt crisis highlighted the need for institutions to revise their contingency plans and collateral management strategies. The unique circumstances of the crisis, coupled with structural differences, make it unlikely for a comparable event to occur in the U.S. LDI market. However, the lessons learned from the U.K. crisis can inform risk management practices globally, prompting institutions to adopt more holistic approaches to collateral and funding.

### 1.2.2 SVB collapse

Between March 8 and March 17, a series of significant events unfolded in the banking industry, particularly surrounding the collapse of Silicon Valley Bank (SVB) and its implications. The following is a timeline of the key developments during this period:

March 8:
- SVB announced raising $500 million from General Atlantic and revealed plans for a $1.25 billion common stock sale, along with $500 million of depository shares.
- Earlier in the day, Silvergate, a bank popular among the crypto industry, announced its decision to shut down operations, foreshadowing what was to come.

March 9:
- SVB's stock plummeted 30% as the markets opened, eventually dropping by 60% throughout the day.
- Concerns grew among venture capitalists (VCs) and startups, leading to a significant withdrawal of funds from the bank.
- SVB CEO Greg Becker attempted to calm VCs and startups in a conference video call, urging them to "stay calm." However, the bank's update on deposit outflows effectively halted the share offering.

March 10:
- U.S. regulators took control of SVB, resulting in the bank's closure.

- In the evening, the Federal Deposit Insurance Corporation (FDIC) informed undisclosed SVB employees that they would retain their jobs as part of the newly formed bridge bank for the next 45 days.

March 12:
- Bids for acquiring SVB were due, but no sale occurred. The FDIC reportedly rejected the lone bid from an unnamed company.
- The U.S. government announced its decision to backstop all SVB deposits.
- Signature Bank in New York was shut down by regulators, citing systemic risk.

March 13:
- SVB's U.K. arm was sold to HSBC for £1.
- First Republic Bank's stock experienced a 60% plunge as concerns over a broader banking crisis grew. Other regional banks also witnessed declines in their stock prices.
- SVB reopened as the newly established Silicon Valley Bridge Bank.
- The Federal Reserve announced a review of SVB's failure.

March 15:
- Credit Suisse, which had been facing its own challenges, disclosed its plan to borrow up to 50 billion Swiss francs ($53.68 billion) from the Swiss National Bank to enhance liquidity.

March 16:
- Treasury Secretary Janet Yellen reassured Congress about the soundness of the U.S. banking system.
- Eleven banks injected $30 billion in deposits into First Republic Bank to demonstrate confidence and prevent a similar fate as SVB.

March 17:
- SVB Financial Group filed for Chapter 11 bankruptcy protection in the Southern District of New York.

These events marked a tumultuous period for SVB and had repercussions across the banking sector, prompting regulatory interventions and heightened concerns about the stability of financial institutions.

### 1.2.3 Credit Suisse collapse

Over its 150-year history, Credit Suisse has played a significant role in supporting Switzerland's industrialization and establishing the country as a global finance hub. However, in the past three years, the bank has faced a series of scandals and poor financial performance that have eroded its reputation and competitiveness, both globally and locally against its rival UBS. Below we review the chain of events that led to Credit Suisse's downfall.

In February 2020, a spying scandal emerged, leading to the sudden departure of CEO Tidjane Thiam. The bank had hired private detectives to spy on its former head of wealth management, who had joined UBS. This incident raised concerns about the integrity and transparency.

In March 2021, another blow hit Credit Suisse when Greensill Capital, a firm it heavily invested in, collapsed. This led to the closure of four connected funds and significant financial losses for the bank. Swiss financial regulator FINMA criticized Credit Suisse for breaching its supervisory obligations.

Just three weeks later, in another major setback, Credit Suisse lost $5.5 billion when Archegos Capital Management defaulted. The bank's failure to manage and control risks in its Prime Services business was highlighted, resulting in substantial losses.

In October 2021, Credit Suisse was fined $475 million by US and British authorities for its involvement in a bribery scandal in Mozambique. The bank's loans to state-owned companies in Mozambique were diverted for bribes, causing an economic crisis in the country.

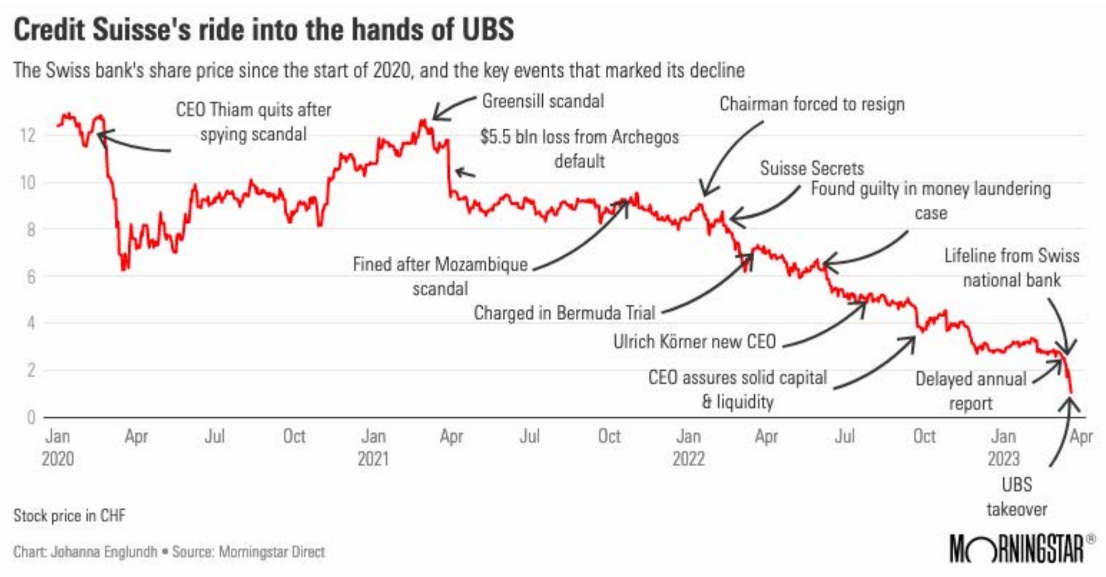


Figure 1 - Credit Suisse's key events - Source: Morningstar.hk

In January 2022, Antonio Horta-Osorio, brought in to lead a turnaround, resigned as chairman after facing accusations of breaking Covid restrictions. He described Credit Suisse's crisis as worse than anything he had experienced in his extensive banking career.

In February 2022, a global media investigation revealed that Credit Suisse had facilitated the stashing of funds by individuals involved in serious crimes, including human rights abuses and corruption. This further tarnished the bank's reputation.

In March 2022, Credit Suisse faced legal troubles, with a Bermuda judge ruling that the bank owed damages to a former Georgian prime minister. Additionally, the bank was found guilty of failing to prevent money laundering by a Bulgarian drug trafficking ring.

Despite efforts to turn the situation around, including job cuts and fresh capital raised, Credit Suisse continued to experience massive customer outflows and financial control issues. In March 2022, the bank's shares plummeted, and the Swiss National Bank provided a liquidity lifeline. However, it was not enough to restore confidence, and Credit Suisse was ultimately taken over by UBS, marking the end of its 167-year history.

The acquisition by UBS was seen as a way to salvage the situation and execute a radical restructuring of Credit Suisse's business. Shareholders experienced significant losses, and UBS emerged as the surviving entity in the deal.

As we see in fig. 1 the downfall of Credit Suisse was the result of a series of scandals, financial losses, and control failures. It highlighted the importance of trust, transparency, and effective risk management in the banking industry, as well as the potential consequences of not upholding these principles.

# 2. Introduction to Complex Network Time Series Analysis

## 2.1 Time Series

### 2.1.1 Definitions

Time series refers to a sequence of data points collected at regular intervals over time. It represents the evolution or variation of a particular variable or set of variables over a continuous time period. Time series data can be univariate, focusing on a single variable, or multivariate, involving multiple variables measured simultaneously. Time series analysis aims to extract meaningful patterns, trends, and dependencies from the temporal data. It involves techniques such as statistical analysis, forecasting, trend detection, spectral analysis, and autocorrelation analysis. Time series analysis helps in understanding the underlying dynamics, making predictions, and uncovering hidden information in the data.

### 2.1.2 Time Series Descriptions

Key concepts in Time Series Analysis as also referred in well-regarded sources (3) are:

1. Time Series Data: Time series data consists of a sequence of observations collected over successive time intervals. The data points are typically recorded at equally spaced time intervals, such as hourly, daily, monthly, or yearly.
2. Trend: Trend refers to the long-term pattern or direction of the data. It represents the overall tendency of the data to increase, decrease, or remain relatively constant over time.
3. Seasonality: Seasonality refers to the regular and predictable patterns that occur at fixed intervals within the time series. These patterns may repeat annually, quarterly, monthly, weekly, or at other intervals.
4. Stationarity: Stationarity is an important assumption in time series analysis. A time series is said to be stationary if its statistical properties, such as mean, variance, and autocovariance, remain constant over time. Stationary time series are easier to model and analyze.
5. Autocorrelation: Autocorrelation measures the relationship between a time series and a lagged version of itself. It helps identify patterns and dependencies in the data. Positive autocorrelation indicates a tendency for consecutive observations to be similar, while negative autocorrelation suggests an inverse relationship.
6. Forecasting: Time series analysis enables forecasting future values based on historical data. Forecasting methods, such as ARIMA (Autoregressive Integrated Moving Average) models, exponential smoothing, or machine learning algorithms, are used to make predictions and estimate future trends.

By understanding these concepts and applying appropriate techniques, analysts can gain insights into the behavior of time series data, make informed predictions, and support decision-making processes in various fields, including finance, economics, weather forecasting, and resource planning.

### 2.1.3 Time Series Models

Brief description of basic time series models:

1. Autoregressive (AR) Model:
The autoregressive (AR) model assumes that the current value of a time series variable is a linear combination of its past values, weighted by coefficients. The order of the model, denoted as AR(p), represents the number of lagged terms used in the model. AR models capture the persistence and memory of the time series.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

Where, y_t represents the current value of the time series, c is a constant term, φ_i (for i = 1 to p) are the autoregressive coefficients, y_{t-i} are the lagged values of the time series, and ε_t is the error term at time t.

2. Moving Average (MA) Model:
The moving average (MA) model represents the current value of a time series variable as a linear combination of its past prediction errors or residuals. Similar to the AR model, the order of the model, denoted as MA(q), represents the number of lagged residuals considered. MA models capture short-term dependencies and can help in smoothing out noise in the data.

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where, y_t represents the current value of the time series, c is a constant term, θ_i (for i = 1 to q) are the moving average coefficients, ε_{t-i} are the past forecast errors, and ε_t is the error term at time t.

3. Autoregressive Moving Average (ARMA) Model:
The autoregressive moving average (ARMA) model combines both the AR and MA models. It incorporates the autoregressive component to capture the time series' linear dependence on past values and the moving average component to capture short-term residual dependencies. The order of the model, denoted as ARMA(p, q), determines the number of lagged terms used from both the AR and MA components.

4. Autoregressive Integrated Moving Average (ARIMA) Model:
The autoregressive integrated moving average (ARIMA) model extends the ARMA model by incorporating differencing to make the time series stationary. Differencing involves computing the difference between consecutive observations to remove trends or seasonality. The order of the model, denoted as ARIMA(p, d, q), represents the number of autoregressive, differencing, and moving average terms, respectively. ARIMA models are useful for non-stationary time series.

5. Exponential Smoothing (ES) Models:

Exponential smoothing models forecast future values of a time series by assigning exponentially decreasing weights to past observations. These models are widely used for forecasting and come in various forms, such as simple exponential smoothing (SES), Holt's linear exponential smoothing (Holt's method), and Holt-Winters' exponential smoothing (seasonal smoothing).

These basic time series models provide a foundation for understanding and analyzing time series data. They are useful for capturing different characteristics, such as trends, seasonality, persistence, and short-term dependencies. More advanced models, such as seasonal ARIMA (SARIMA), vector autoregression (VAR), and state space models, can be built upon these basic models to handle more complex time series patterns and dynamics.

And few non-linear time series models:

1. Nonlinear Autoregressive Exogenous (NARX) Model:
The nonlinear autoregressive exogenous (NARX) model extends the autoregressive (AR) model to incorporate non-linear relationships between the time series variable and its lagged values as well as exogenous variables. It captures non-linear dependencies and can be useful when there are complex interactions or nonlinear patterns in the data.

2. Threshold Autoregressive (TAR) Model:
The threshold autoregressive (TAR) model divides the time series data into different regimes or states based on a threshold. Within each regime, the autoregressive relationship is linear, but the parameters may differ across regimes. The TAR model is suitable when the behavior of the time series changes abruptly at certain thresholds or levels.

3. Markov-Switching Autoregressive (MSAR) Model:
The Markov-switching autoregressive (MSAR) model assumes that the time series data are governed by different regimes or states that follow a Markov process. In each regime, the autoregressive relationship may be different. The model captures changes in the time series dynamics and is useful when the underlying process switches between different states.

4. Neural Network Models:
Neural network models, such as feedforward neural networks, recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, can be used for non-linear time series modeling. These models can capture complex and non-linear relationships in the data by learning from historical patterns and dependencies. Neural networks are particularly useful when the time series exhibit non-linear dynamics or have high-dimensional inputs.

5. Gaussian Process Models:
Gaussian process models are a flexible class of non-linear models that can capture complex dependencies in time series data. They are based on a collection of random variables, where any finite subset follows a multivariate Gaussian distribution. Gaussian process models can handle non-linear patterns, non-stationarity, and uncertainties in the data.

These non-linear time series models go beyond the linear relationships captured by basic models and provide more flexibility in capturing complex patterns, dynamics, and interactions in the data. They are valuable tools for analyzing and forecasting time series data when the relationships are non-linear, exhibit regime shifts, or have complex dependencies.

## 2.2 Complex networks

Complex networks are mathematical and graphical representations used to study interconnected systems composed of various entities. Newman defines complex networks as a collection of nodes representing entities such as individuals, neurons, or computers, with edges representing the connections between them (2). These connections capture the relationships, interactions, or dependencies within the system. Complex networks exhibit intriguing properties such as small-worldness, scale-free degree distribution, and community structure, as described by Newman. By analyzing complex networks, researchers gain insights into the underlying structure, dynamic changes, and emergent behaviors of interconnected systems, contributing to a deeper understanding of complex phenomena.

### Importance and applications of complex network time series analysis

Complex network time series analysis holds great importance and finds applications in various fields due to its ability to capture the interplay between network structures and temporal dynamics. Here are some key reasons for its significance and notable applications:

1. Understanding Complex Systems: Complex network time series analysis allows researchers to gain insights into the behavior and dynamics of complex systems, including social networks, biological networks, transportation networks, and more. It helps in unraveling the underlying mechanisms, identifying influential nodes or edges, and characterizing emergent behaviors and patterns.

2. Predictive Modeling: By integrating network information and temporal dependencies, complex network time series analysis enables the development of predictive models and forecasting techniques. It finds applications in stock market prediction, disease outbreak forecasting, weather forecasting, and other domains where understanding the complex interplay of networks and time is crucial for accurate predictions.

3. Brain Network Analysis: Complex network time series analysis is extensively used in neuroscience to study the human brain. It allows researchers to model and analyze brain connectivity networks over time, enabling the investigation of dynamic brain activity, cognitive processes, and neurological disorders. It aids in understanding how the brain's network structure and temporal dynamics relate to various cognitive and behavioral functions.

4. Social Dynamics and Influence Propagation: Analysis of complex network time series data in social networks helps in studying information diffusion, opinion dynamics, and influence propagation. It assists in understanding how trends, rumors, or innovations spread through a network, and how individuals' interactions and behaviors evolve over time.

5. Financial Markets and Risk Analysis: Complex network time series analysis has applications in financial markets, where it aids in understanding interconnectedness, systemic risk, and market dynamics. It helps in modeling and predicting stock price

movements, identifying key market players, and assessing systemic risks by considering the evolving network structure and temporal dependencies.

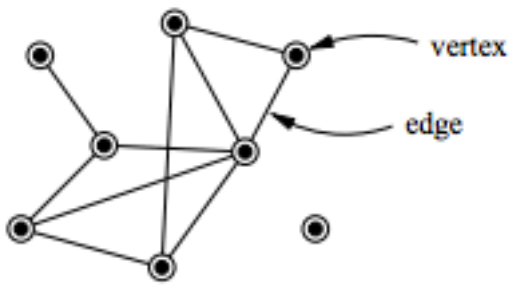
6. Ecological Systems and Environmental Analysis: Complex network time series analysis contributes to the study of ecological systems, such as food webs, species interactions, and ecosystem dynamics. It helps in understanding species coexistence, trophic interactions, and the effects of environmental changes on ecosystems by incorporating temporal dynamics and network structure.

7. Social Media and Online Networks: Analysis of complex network time series data from social media platforms and online networks provides insights into user behaviors, information diffusion, and online community dynamics. It aids in understanding the temporal evolution of online interactions, detecting trends, and predicting user engagement or sentiment.

These are just a few examples of the wide range of applications of complex network time series analysis. Its interdisciplinary nature allows researchers to explore and understand complex phenomena, uncover hidden relationships, and make predictions in various fields of study.

### 2.2.1 Graph terminology and concepts

Graph terminology and concepts that are commonly used in the field of graph theory:




1. Graph: A graph is a collection of vertices (or nodes) connected by edges. It represents a network or a set of relationships between objects. Graphs can be directed (edges have a specific direction) or undirected (edges do not have a direction).
2. Vertices and Edges: Vertices are the individual elements or entities in a graph, often represented as points or circles. Edges are the connections or relationships between vertices, represented as lines or arcs.
3. Degree: The degree of a vertex is the number of edges connected to that vertex. In a directed graph, the degree is divided into the in-degree (number of incoming edges) and the out-degree (number of outgoing edges) of a vertex.
4. Path: A path in a graph is a sequence of vertices connected by edges. It represents a route or a series of steps to go from one vertex to another.

5. Cycle: A cycle is a path in a graph that starts and ends at the same vertex, traversing different vertices and edges. It forms a closed loop within the graph.
6. Connectedness: A graph is connected if there is a path between every pair of vertices. A disconnected graph consists of two or more disconnected components.
7. Weighted Graph: In a weighted graph, each edge is assigned a weight or value. The weights can represent distances, costs, or any other relevant quantity associated with the edges.
8. Subgraph: A subgraph is a graph that is formed by selecting a subset of vertices and edges from an original graph.
9. Directed Acyclic Graph (DAG): A directed acyclic graph is a directed graph that does not contain any cycles. It often represents a partial order or a directed flow of events.
10. Bipartite Graph: A bipartite graph is a graph whose vertices can be divided into two disjoint sets, such that there are no edges between vertices within the same set.
11. Spanning Tree: A spanning tree of a connected graph is a subgraph that is a tree (no cycles) and spans all the vertices of the original graph. It forms a connected and acyclic subset of the graph.

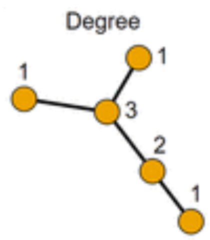
These are just a few of the fundamental graph terminology and concepts. Graph theory provides a rich framework for analyzing and modeling various real-world systems, such as social networks, transportation networks, computer networks, and biological networks.

### 2.2.2 Topological metrics

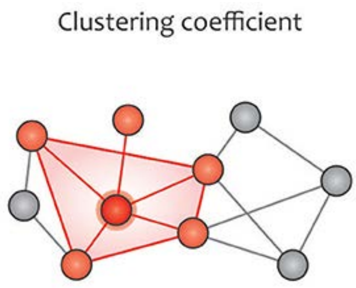
In complex networks, topological metrics refer to quantitative measures or properties that characterize the structure and connectivity of the network. These metrics provide insights into the organization, efficiency, robustness, and other structural properties of the network.

Here are some commonly used topological metrics in complex network analysis:

1. Degree: The degree of a node in a network is the number of edges connected to that node. It measures the connectivity or centrality of a node within the network.



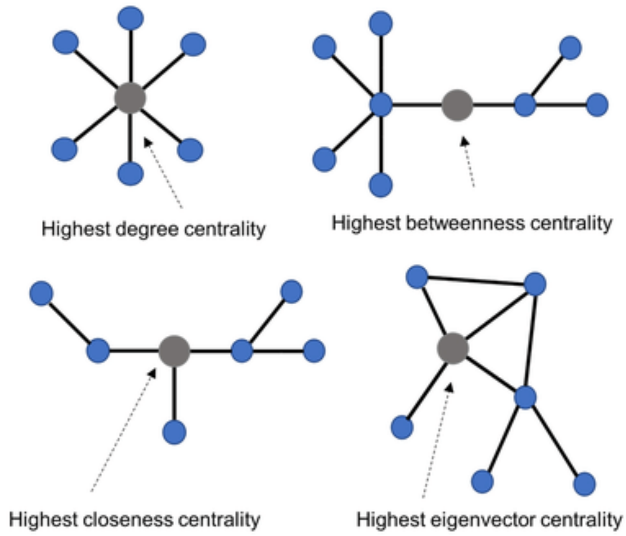

2. Clustering coefficient: The clustering coefficient quantifies the degree to which nodes in a network tend to cluster together. It measures the density of connections between the neighbors of a node.



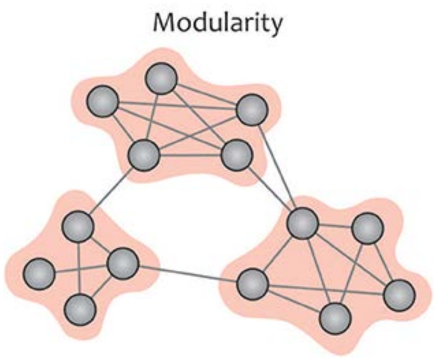

3. Path length: The path length is the shortest distance between two nodes in a network, typically measured as the number of edges or nodes traversed to go from one node to another. It characterizes the efficiency of information or resource flow in the network.

4. Degree distribution: The degree distribution describes the probability distribution of node degrees in the network. It provides insights into the heterogeneity or homogeneity of connectivity patterns in the network.

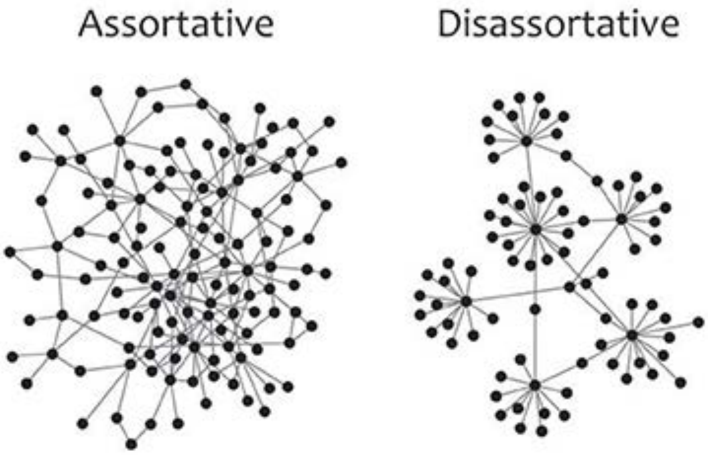
5. Centrality measures: Centrality metrics, such as betweenness centrality, closeness centrality, and eigenvector centrality, quantify the importance or influence of a node within the network. They identify nodes that play critical roles in information flow, communication, or control.




6. Modularity(related to Clustering coefficient): Modularity measures the extent to which a network can be divided into communities or modules. It captures the presence of densely connected groups of nodes with relatively sparse connections between them.




7. Assortativity: Assortativity measures the tendency of nodes with similar attributes or properties to be connected. It quantifies the degree of homophily or heterophily in the network.




These topological metrics help researchers and analysts understand the structural properties, resilience, efficiency, and functional characteristics of complex networks across various domains, including social networks, biological networks, transportation networks, and technological networks.

### Basic Complex Network types

## 2.3 Motivation for combining Complex Networks and Time Series Analysis

The motivation for combining complex networks and time series analysis stems from the understanding that many real-world systems can be represented as both dynamic networks and time-evolving processes. By integrating these two fields, researchers aim to gain a deeper understanding of the underlying structure and dynamics of complex systems.

In the past, couple of interesting papers have been published on the topic and below we refer to some of the ones that pioneered the research of the area.

Mantegna in his paper (12) offers a comprehensive overview of complex networks in financial markets. It discusses the application of complex network analysis to financial time series, focusing on correlation-based networks. The paper highlights the importance of network topology in understanding the structure and dynamics of financial markets, providing insights into the interconnectedness of different market entities.

Onnela et al in their paper (13) investigate the interconnections between equities in financial markets using complex network analysis. It applies the correlation-based approach to construct networks based on a large dataset of stocks traded on the Finnish Stock Exchange. The study explores the topological properties of these networks, shedding light on the underlying structure of interconnected equities in financial markets.

Finally Kenett et al in their paper (14) explore the dynamic correlation networks of stock returns using complex network analysis. It introduces the concept of time-dependent correlation matrices to capture the temporal evolution of interconnections in financial markets. The study investigates the changing network properties during different market conditions, providing insights into the dynamic nature of correlations and network structures in stock markets.

These papers contribute to the understanding of complex network analysis in the context of financial time series. They provide valuable insights into the interdependencies, structure, and dynamics of financial markets, offering a framework to study the relationships and patterns within these complex systems.

Here are some key motivations for combining complex networks and time series analysis, not only on the financial context:

1. Capturing Interactions and Dependencies: Complex networks provide a framework to capture and analyze the interactions, dependencies, and relationships among components in a system. Time series analysis, on the other hand, focuses on the temporal behavior and patterns within the data. By combining these approaches, we can study how the evolving network structure influences the dynamics of the system and vice versa, uncovering hidden relationships and dependencies.

2. Characterizing Complex Dynamics: Many real-world systems exhibit complex and nonlinear dynamics. By representing the system as a network and analyzing the associated time series data, we can uncover the emergent behaviors, identify critical nodes or edges, and study how changes in the network structure impact the system's dynamics. This integrated analysis provides insights into the underlying mechanisms driving the complex behavior of the system.

3. Unveiling Temporal Evolution: Time series analysis allows us to study the temporal evolution of a system's variables. By incorporating network analysis, we can investigate how the network structure evolves over time, identifying important events, phases, or transitions. This combined approach helps in understanding the temporal patterns and transitions in complex systems, shedding light on critical time points and dynamic processes.

4. Predictive Modeling and Forecasting: Integrating complex network and time series analysis enables the development of predictive models and forecasting techniques. By considering the network structure and temporal dependencies, we can leverage the

collective behavior and interactions of the system's components to make more accurate predictions or forecasts. This has applications in various fields, such as predicting stock market behavior, epidemic outbreaks, or social dynamics.

5. Interdisciplinary Applications: The combination of complex networks and time series analysis has diverse applications across disciplines. It is employed in fields such as neuroscience, social sciences, finance, ecology, and engineering, to understand and model complex phenomena, including brain dynamics, social networks, financial markets, ecological systems, and more.

By integrating the methodologies and concepts from complex networks and time series analysis, researchers can gain a more comprehensive understanding of complex systems' structure, dynamics, and behaviors, leading to insights and advancements in various domains.

# 3. From Time Series to Complex Networks

The process of transforming a time series into a network graph involves representing the time series as a set of interconnected nodes and edges. Each node in the graph represents a data point or observation from the time series, while the edges between the nodes capture the relationships or connections between these data points.

The transformation typically involves the following steps:

1. Data Preparation: The time series data is prepared by extracting the relevant values from the series or columns of a dataframe, depending on the specific implementation. These values will serve as the nodes of the network graph.
2. Graph Creation: An empty graph structure is initialized, using a network graph library such as NetworkX. This graph will hold the nodes and edges representing the time series.
3. Node Addition: Each data point or observation from the time series is added as a node to the graph. The nodes can be labeled with their corresponding values from the time series, allowing for easy interpretation and analysis.
4. Edge Construction: The edges between the nodes are defined based on certain criteria or conditions. This criteria could involve a threshold value, similarity measure, or any other relevant metric. Edges are added to the graph to represent the connections or relationships between nodes that meet the defined criteria.
5. Visualization: Once the graph is constructed, it can be visualized using appropriate graph visualization techniques provided by the chosen library. The visualization may include the nodes, edges, and their properties, providing a visual representation of the underlying structure and connections within the time series.

By transforming a time series into a network graph, it becomes possible to analyze and explore the complex relationships, patterns, and dynamics present in the data. Network analysis techniques can be applied to gain insights into the connectivity, centrality, clustering, and other network properties of the time series, offering a different perspective for understanding and interpreting the data.

## 3.1 Natural Visibility Graph

In the context of complex networks, a Natural Visibility Graph (NVG) is a representation of a time series or sequence as a network structure. It is a graph-based approach that captures the underlying patterns and relationships in the time series data.

Lacasa et al. (8) introduced the concept of the Natural Visibility Graph (NVG), which transforms time series data into a complex network representation. The NVG method is based on the visibility algorithm that captures the pairwise visibility between data points. The paper demonstrates the application of the NVG to various time series datasets and discusses the topological properties and information content of the resulting visibility graphs.

Charakopoulos et al. (9) further expanded on the concept of the Natural Visibility Graph (NVG) and explored its basic properties and applications. The paper discusses the algorithmic construction of NVG, its relationship with other graph representations, and its robustness to noise and sampling rate. Additionally, it presents applications of NVG in various fields such as climate science, finance, and neuroscience, highlighting its potential in analyzing complex time series data.

The construction of a Natural Visibility Graph involves the following steps:

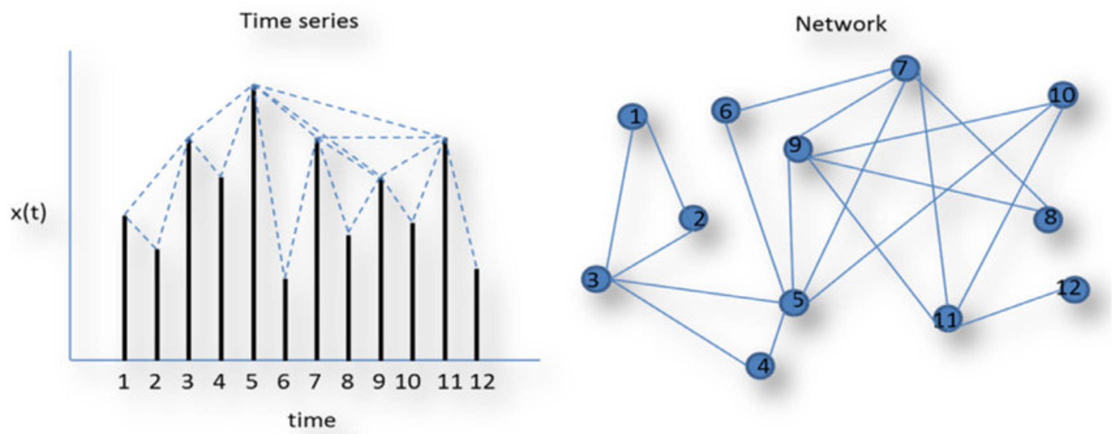
1. Time Series: Start with a given time series, which is a sequence of data points collected at successive time intervals.

2. Extrema Identification: Identify the local extrema in the time series. Local extrema are the points where the values of the time series reach a local maximum or minimum.

3. Line Segments: Connect consecutive local extrema with line segments. Each line segment represents a direct visibility between two extrema points in the time series.

4. Visibility Criterion: Determine the visibility criterion to decide if two extrema points can "see" each other. Typically, this criterion is based on the relative heights or values of the extrema points. For example, one common criterion is that an extrema point A can "see" another extrema point B if there are no other points with higher values between A and B.

5. Graph Construction: Create a graph representation where the local extrema points are represented as nodes, and the line segments between them represent the edges. The resulting graph is the Natural Visibility Graph.




The Natural Visibility Graph provides a network-based representation of the time series data, capturing the inherent patterns and visibility relationships within the sequence. It enables the application of various network analysis techniques and measures to gain insights into the underlying dynamics and properties of the time series.

Researchers and practitioners often use Natural Visibility Graphs in areas such as time series analysis, complex systems, and network science to study the characteristics, dynamics, and information flow within temporal data.

## 3.2 Average Degree

Average Degree: The average degree of a network is the average number of connections or edges that each node has. It provides an indication of the overall connectivity of the network. To calculate the average degree, we sum up the degrees of all nodes in the network and divide by the total number of nodes.

## 3.3 Average Path Length

Average Path Length: The average path length of a network quantifies the typical distance between pairs of nodes in the network. It measures how easily information can flow between nodes. A shorter average path length indicates a higher level of global connectivity. It is calculated by finding the shortest path between all pairs of nodes and taking the average of these path lengths.

## 3.4 Number of communities

Number of Communities: Communities in a network refer to groups of nodes that are more densely connected within the group than with nodes outside the group. The number of communities is a measure of the network’s division into distinct groups or clusters. Community detection algorithms are applied to identify these groups.

## 3.5 Clustering coefficient

Clustering Coefficient: The clustering coefficient of a node in a network quantifies the degree to which its neighboring nodes are connected to each other. It measures the tendency for nodes in the network to form clusters or groups. The local clustering coefficient of a node is calculated as the ratio of the number of connections between its neighbors to the maximum possible number of connections.

## 3.6 Modularity

Modularity: Modularity is a measure that quantifies the extent to which a network is divided into distinct communities or modules. It evaluates the quality of the division by comparing the number of edges within communities to the expected number of edges if connections were randomly distributed. A higher modularity value indicates a stronger community structure.

These measures provide valuable insights into the structure, connectivity, and organization of complex networks. They help characterize and analyze various types of networks, including social networks, biological networks, and technological networks, among others. By studying these measures, we can gain a better understanding of the network's behavior, resilience, information flow, and other properties.

# 4. Basic Time Series Analysis

## 4.1 Description of time-series

The time series data used in this analysis consists of various financial indicators spanning a specific time window from just before the Covid-19 pandemic break to the middle of March 2023. The dataset, retrieved from investing.com, includes stock prices of Credit Suisse and SVB Bank (see fig.2), representing two prominent financial institutions. Additionally, it incorporates the prices of UK Gilt and US2Y bond (fig. 3), providing insights into the bond markets. The data also includes (fig.4) the FED rate, which reflects the monetary policy decisions of the Federal Reserve. Furthermore, the prices of GBP/USD pair and Gold are considered, offering perspectives on currency exchange rates and the precious metals market. For the purpose of our calculations, we have focused on the Date and Price columns, extracting the necessary information from the initial data source. By analyzing these diverse financial indicators over the specified time period, valuable insights can be gained regarding market trends, volatility, and potential impacts of significant events like the Covid-19 pandemic.

## 4.2 Preprocessing and data preparation

For preprocessing and data preparation, the first step involved loading the CSV files containing the financial data into Python dataframes. Then, to ensure proper analysis, the data was transformed to specific datatypes, ensuring that each variable had the appropriate format. To streamline the analysis, any unnecessary columns that were not required for the study were dropped from the dataframes. The next step involved merging the different dataframes into a single consolidated dataframe, combining the relevant information from each source. To handle missing or empty records, a technique known as backfilling and forward filling was applied. This process involved filling the missing values by propagating the last known value forward or the subsequent value backward. By employing this technique, all records in the dataframe were aligned, ensuring uniformity in the dataset, which is crucial for accurate analysis and modeling. These preprocessing and data preparation steps laid the foundation for further analysis and exploration of the financial time series data.

## 4.3 Plots

Plots are essential in time series analysis as they provide visual representations of the data, allowing us to observe patterns, trends, and relationships that may not be immediately apparent from the raw numbers. Here are a few reasons why plots are important in time series analysis:

1. Visualizing Data: Plots help us visualize the data, enabling a better understanding of its overall structure and characteristics. By visualizing the time series, we can identify patterns, cycles, and irregularities that might not be evident in numerical form alone. It allows us to gain insights into the underlying dynamics of the data and detect any unusual observations or outliers.

2. Trend Identification: Plots help us identify trends in the time series. A trend is a long-term movement in the data that can be upward (increasing trend), downward (decreasing trend), or flat (no trend). By plotting the data over time, we can visually identify the presence and direction of trends, which can be crucial for forecasting, decision-making, and understanding the behavior of the variable.
3. Seasonal and Cyclical Patterns: Plots can reveal seasonal or cyclical patterns in the data. Seasonality refers to regular patterns that repeat at fixed intervals, such as daily, weekly, or yearly patterns. Cyclical patterns, on the other hand, are non-regular fluctuations that occur over longer time periods. By examining the plots, we can detect these patterns and incorporate them into our analysis and forecasting.

Overall, plots serve as a powerful tool in time series analysis by providing visual insights into the data, enabling trend identification, pattern detection, model assessment, and aiding in decision-making. They complement quantitative analysis and enhance our understanding of the behavior and characteristics of the time series data.

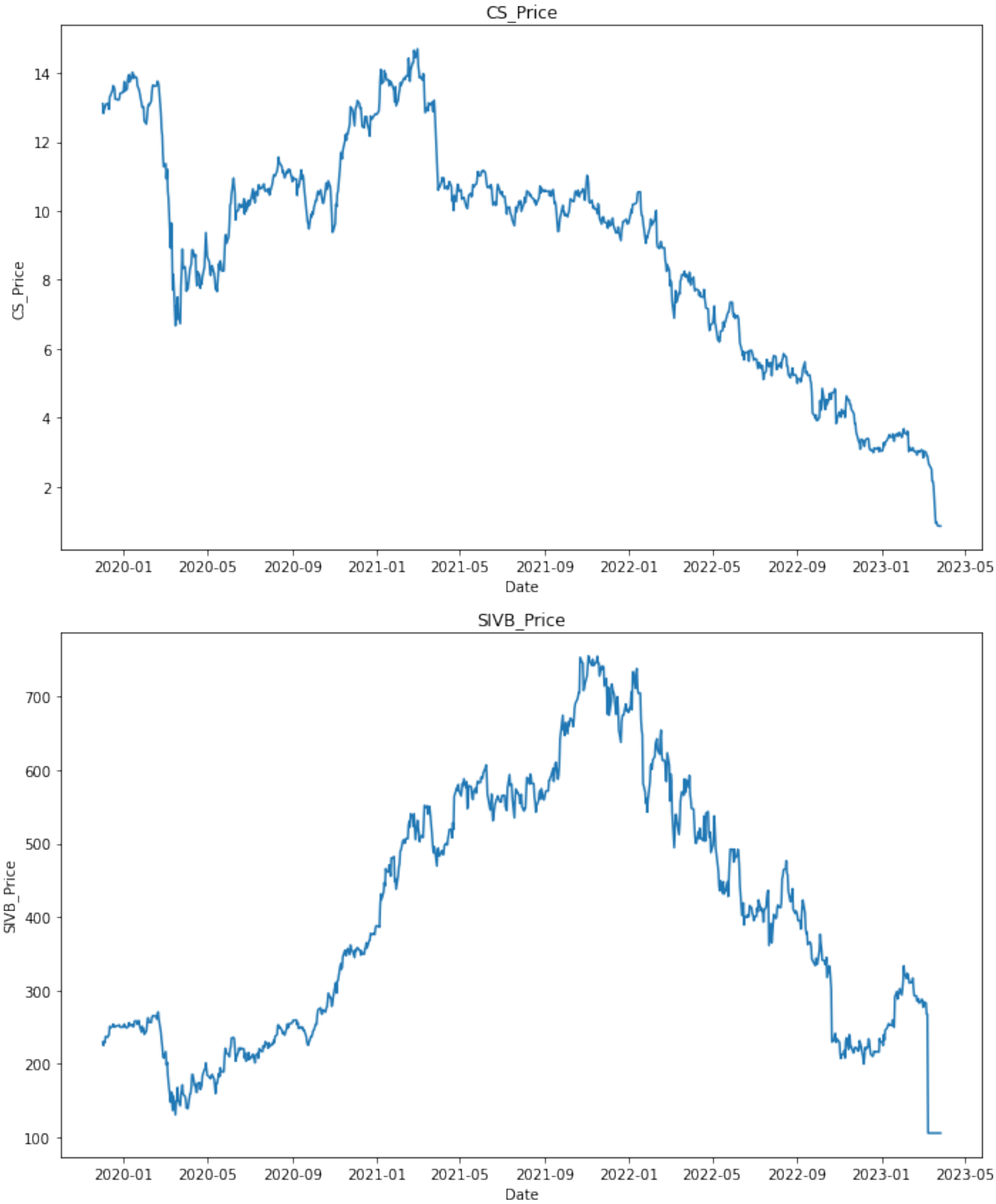



Figure 2 - CS and SVB Prices plots

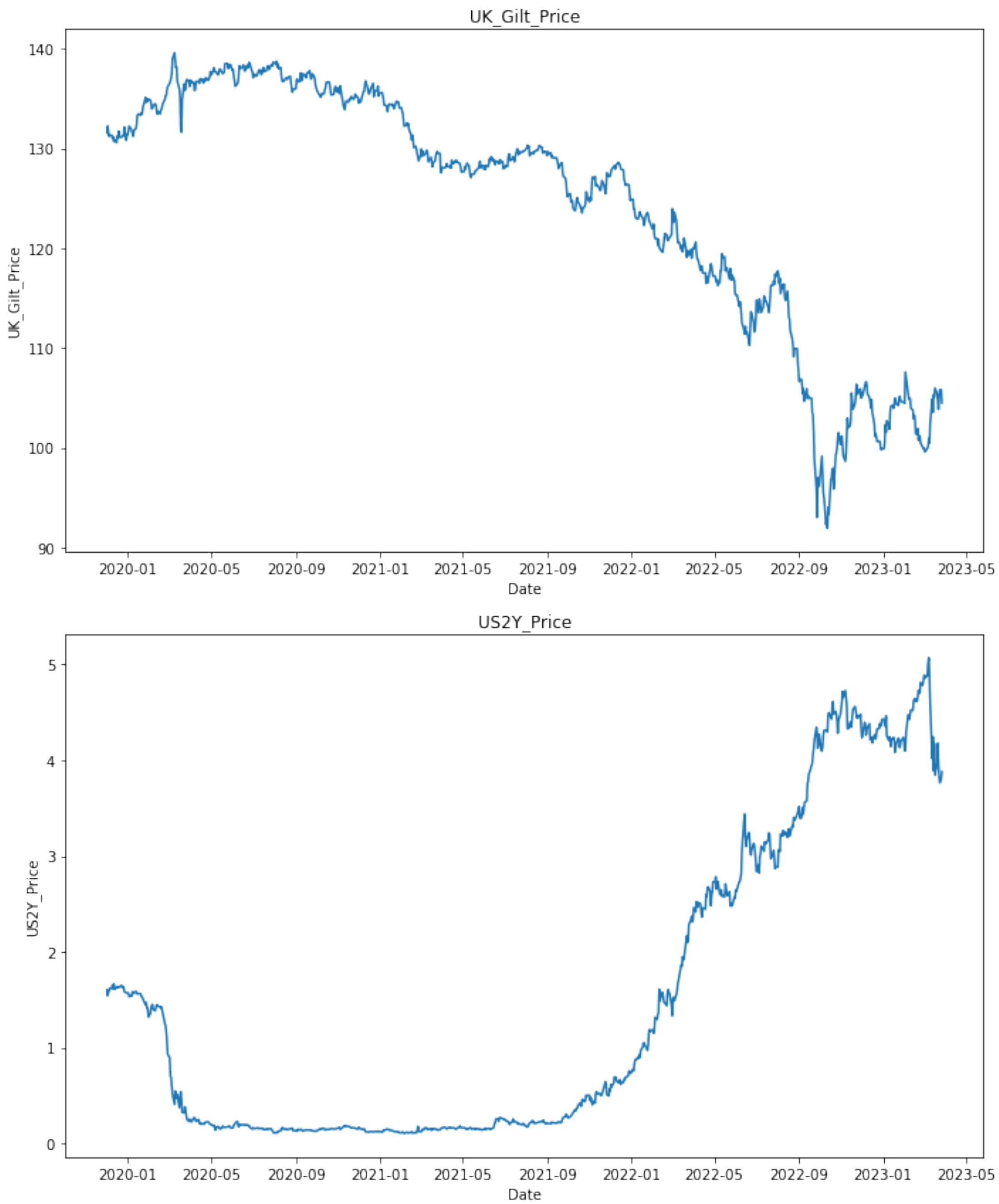



Figure 3 - UK Gilt and US2Y prices

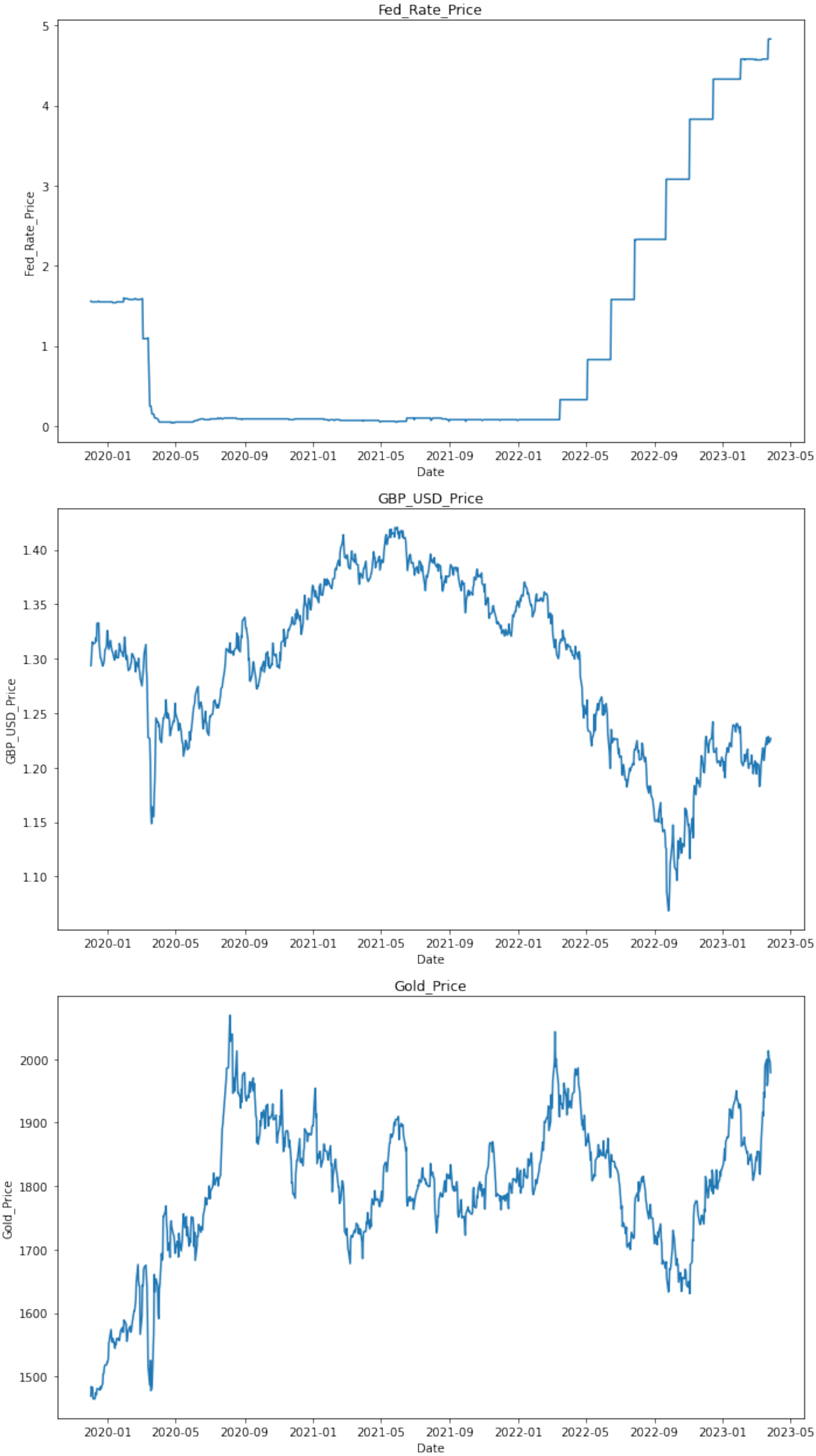



Figure 4 - FED Rate, GPB/USD and Gold prices

## 4.4 Descriptive Statistics

When checking descriptive statistics for a time series, there are several key items we can refer to. Here are the common ones:

1. Mean: The average value of the time series, which provides an indication of its central tendency.

2. Median: The middle value in the time series when it is sorted in ascending or descending order. It represents the value below and above which 50% of the data lie, providing insights into the data's central tendency.

3. Standard Deviation: A measure of the dispersion or variability in the time series values around the mean. It quantifies the average amount by which each data point deviates from the mean.

4. Minimum and Maximum: The lowest and highest values observed in the time series, providing information about the range of values.

5. Quartiles: The values that divide the time series into four equal parts. The first quartile (Q1) represents the 25th percentile, the second quartile (Q2) represents the median, and the third quartile (Q3) represents the 75th percentile. Quartiles help assess the data's distribution and identify potential outliers.

6. Skewness: A measure of the asymmetry of the distribution of the time series values. Positive skewness indicates a longer right tail, while negative skewness indicates a longer left tail.

7. Kurtosis: A measure of the peakedness or flatness of the distribution of the time series values. It compares the tails of the distribution to that of a normal distribution. Positive kurtosis indicates a relatively peaked distribution with heavy tails, while negative kurtosis indicates a relatively flat distribution.

8. Correlation: The degree of linear relationship between the time series and other variables. Correlation coefficients range from -1 to 1, with positive values indicating positive correlation, negative values indicating negative correlation, and values close to 0 indicating weak or no correlation.

9. Autocorrelation: The correlation of a time series with its lagged values. It measures the relationship between each observation and its past observations at different time lags, providing insights into the presence of patterns or trends in the data.

10. Percentiles: Values that divide the time series into equal parts. Common percentiles include the 10th, 25th, 50th (median), 75th, and 90th percentiles, which help understand the distribution and identify extreme values.

By referring to these descriptive statistics, we can gain a better understanding of the characteristics, distribution, and patterns present in the time series data.

Skewness seems to be one of the most important so in more depth we can see the following:
In the context of time series analysis, skewness is a statistical measure that quantifies the asymmetry of the distribution of data points in a time series. It provides insight into the shape of the distribution and the extent to which it deviates from a symmetric distribution.

Skewness measures the degree and direction of skew, or the lack of symmetry, in a distribution. It indicates whether the distribution is skewed to the left (negative skewness) or to the right (positive skewness) relative to the mean of the data.

- Positive Skewness: If the distribution is positively skewed, it means that the tail of the distribution is skewed to the right, and there are more extreme values on the right side of the distribution. In other words, the distribution is elongated towards higher values. The mean is typically greater than the median in a positively skewed distribution.

- Negative Skewness: If the distribution is negatively skewed, it means that the tail of the distribution is skewed to the left, and there are more extreme values on the left side of the distribution. In this case, the distribution is elongated towards lower values. The mean is typically less than the median in a negatively skewed distribution.

Skewness is a valuable measure in time series analysis as it provides insights into the shape of the data distribution and the presence of asymmetry. It helps to understand the behavior and potential biases in the data. For example:

- Skewness can indicate the presence of outliers or extreme values that affect the shape of the distribution.
- Skewness can impact forecasting accuracy and model assumptions, as certain statistical models assume a symmetric distribution.
- Skewness can be used in risk analysis to assess the potential for tail events or extreme values.

It's important to note that skewness is just one aspect of the distribution, and it should be considered in conjunction with other statistical measures and visualizations to obtain a comprehensive understanding of the time series data.

| | CS_Price | SIVB_Price | UK_Gilt_Price | US2Y_Price | Fed_Rate_Price |
|---|---|---|---|---|---|
| count | 870.000000 | 870.000000 | 870.000000 | 870.000000 | 870.000000 |
| mean | 8.971690 | 401.873713 | 124.239862 | 1.460116 | 1.014379 |
| std | 3.267852 | 171.752208 | 12.300067 | 1.600109 | 1.459306 |
| min | 0.860000 | 106.040000 | 91.930000 | 0.105200 | 0.040000 |
| 25% | 6.530000 | 245.937500 | 116.660000 | 0.161250 | 0.080000 |
| 50% | 9.940000 | 389.955000 | 128.280000 | 0.502500 | 0.090000 |
| 75% | 10.840000 | 555.877500 | 134.855000 | 2.731050 | 1.580000 |
| max | 14.700000 | 755.030000 | 139.580000 | 5.070100 | 4.830000 |
| skewness | -0.445326 | 0.249293 | -0.801581 | 0.859689 | 1.398968 |
| kurtosis | -0.720537 | -1.150600 | -0.555100 | -0.818006 | 0.558552 |

| | GBP_USD_Price | Gold_Price |
|---|---|---|
| count | 870.000000 | 870.000000 |
| mean | 1.292522 | 1793.743908 |
| std | 0.076922 | 112.281608 |
| min | 1.068400 | 1464.900000 |
| 25% | 1.227600 | 1735.500000 |
| 50% | 1.304650 | 1803.000000 |
| 75% | 1.359500 | 1867.050000 |
| max | 1.420900 | 2069.400000 |
| skewness | -0.396599 | -0.663302 |
| kurtosis | -0.723297 | 0.567979 |

Table 1 - Descriptive statistics

Using Python we have calculated the matrix shown in Table 1 including most of the major measures discussed earlier.

## 4.5 Returns

Returns play a crucial role in time series analysis because they provide valuable insights into the changes and relative movements of a variable over time. Here are a few reasons why returns are important:

1. Relative Performance: Returns allow us to compare the performance of a variable or asset over different time periods. By analyzing returns, we can assess the relative gains or losses and understand the volatility or stability of the variable.
2. Trend Detection: Returns help in identifying trends in the time series data. Positive returns indicate an upward trend, suggesting growth or positive movement in the variable. Negative returns, on the other hand, indicate a downward trend or decline in the variable. Analyzing the patterns of returns can provide insights into the underlying dynamics and help identify potential turning points or shifts in the data.
3. Risk and Volatility: Returns are closely related to risk and volatility. Higher returns generally imply higher risk or volatility in the underlying variable. By analyzing the distribution of returns, measuring their variability, or calculating metrics like standard deviation, we can assess the level of risk associated with the variable and make informed decisions.
4. Statistical Analysis: Returns are widely used in various statistical analyses and modeling techniques. They often exhibit desirable statistical properties such as stationarity (constant mean and variance) and can be easier to work with

compared to the original time series data. Returns can be used in regression analysis, forecasting models, and other quantitative methods to study relationships, estimate parameters, and make predictions.

5. Investment and Portfolio Management: Returns are crucial in investment and portfolio management. Investors and analysts use historical returns to assess the performance of investments, calculate risk-adjusted returns, and make informed decisions on asset allocation and portfolio diversification. Returns also play a role in evaluating investment strategies, measuring performance benchmarks, and analyzing risk-return trade-offs.

Overall, returns provide a concise and standardized representation of the changes in a variable over time. They offer valuable information for understanding trends, risk, performance, and statistical properties of time series data, enabling better analysis, modeling, and decision-making in various fields.

4

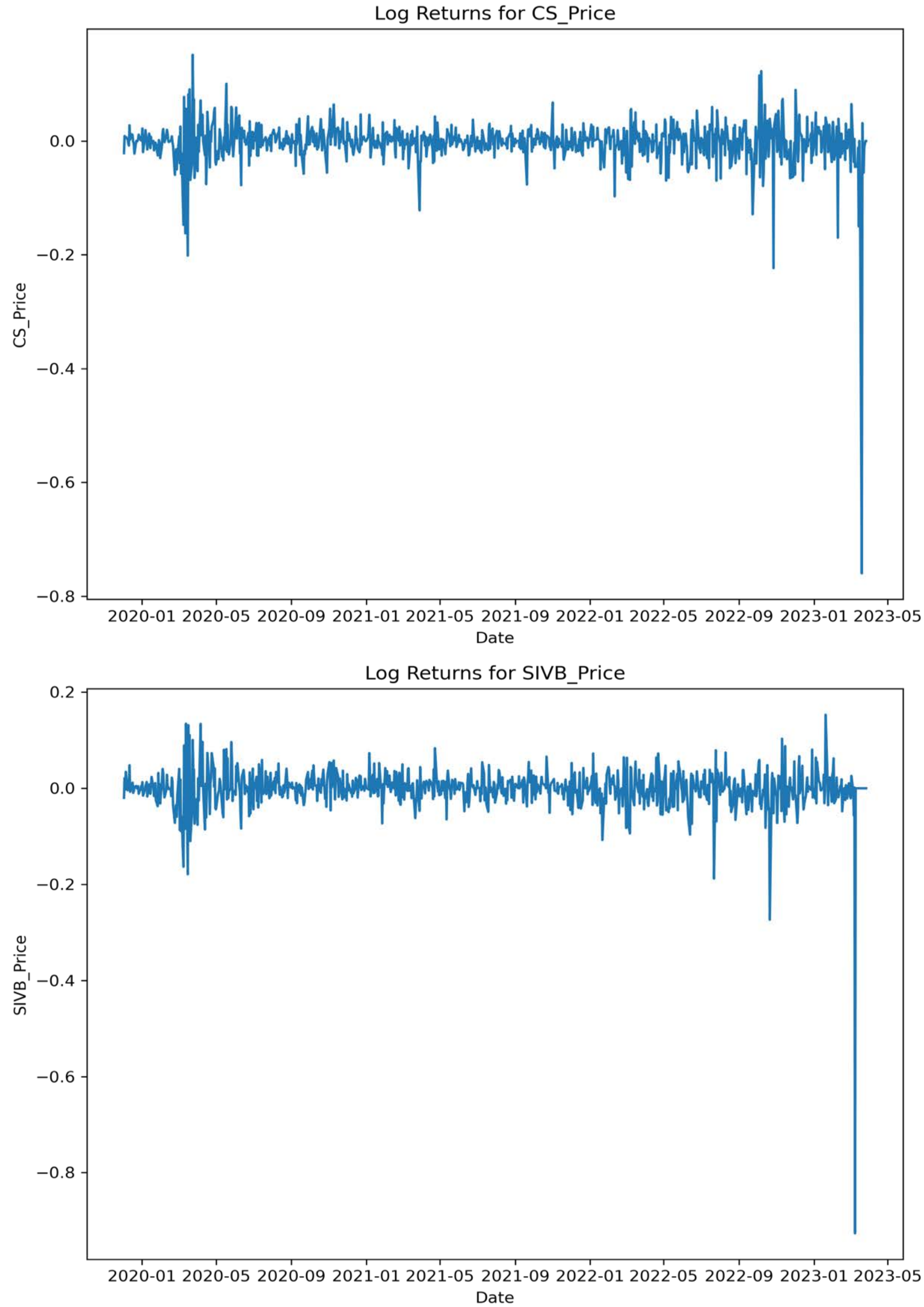



Figure 5 - CS and SIVB Returns

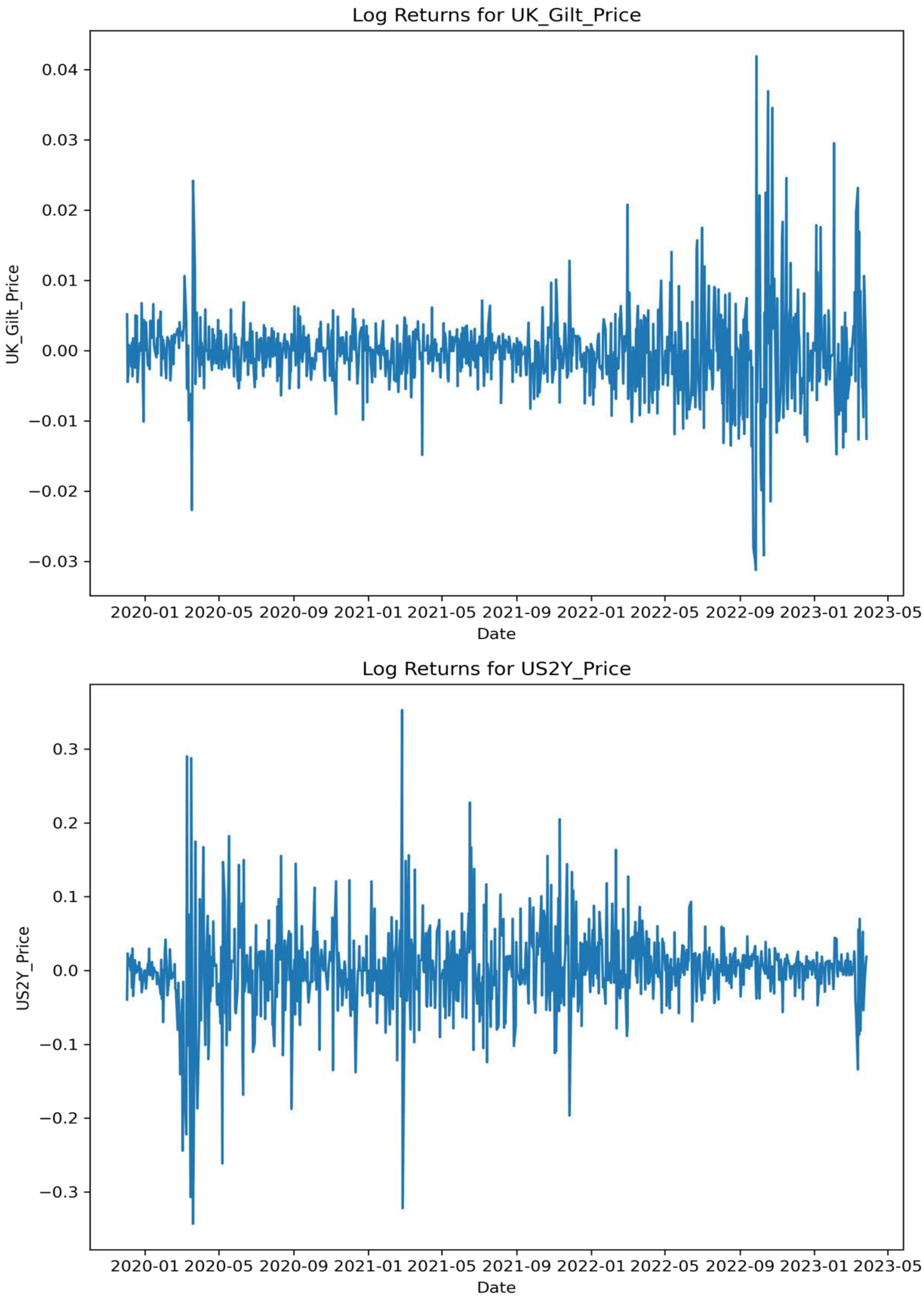



Figure 6 - UK Gilt and US2Y Returns

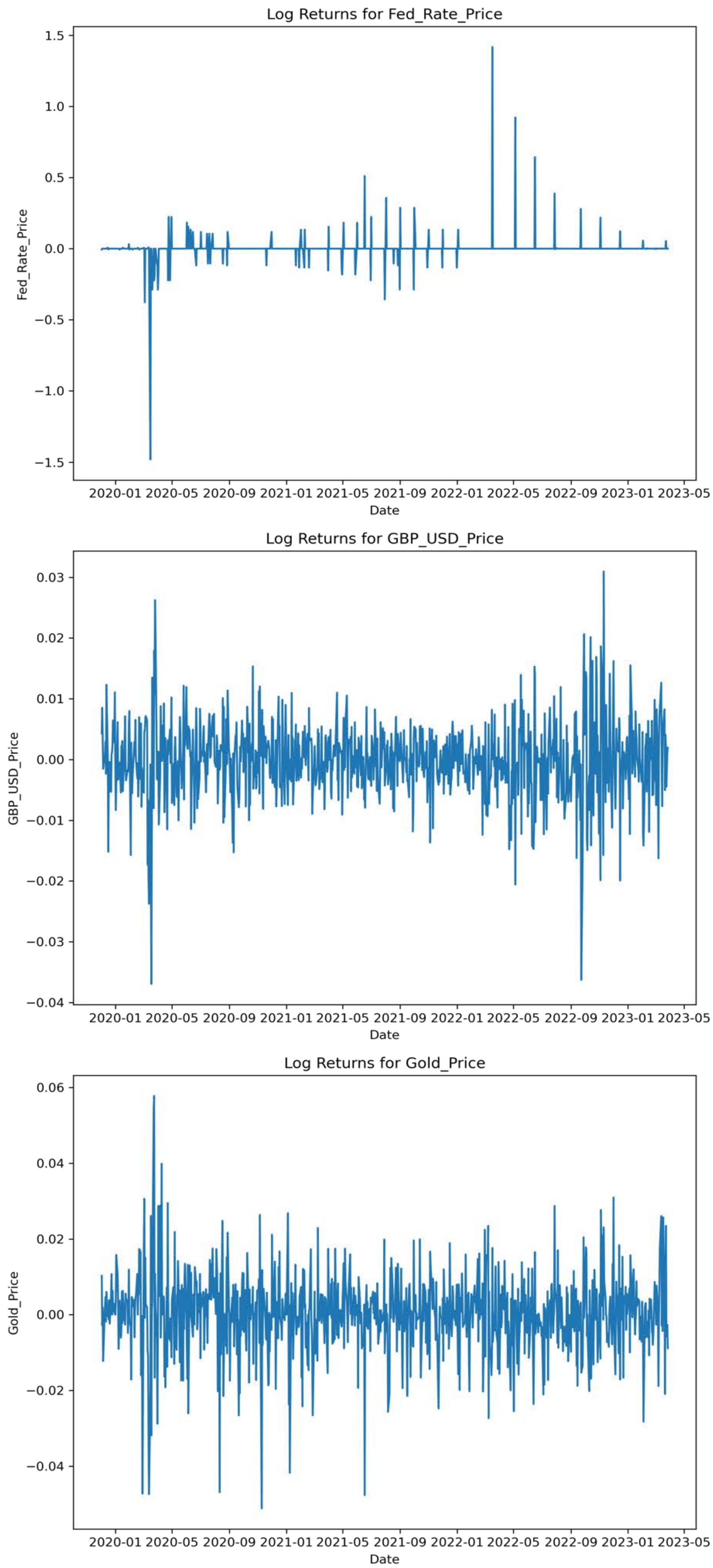



Figure 7 - FED Rate,GBP/USD and Gold Returns

## 4.6 Auto-correlation functions

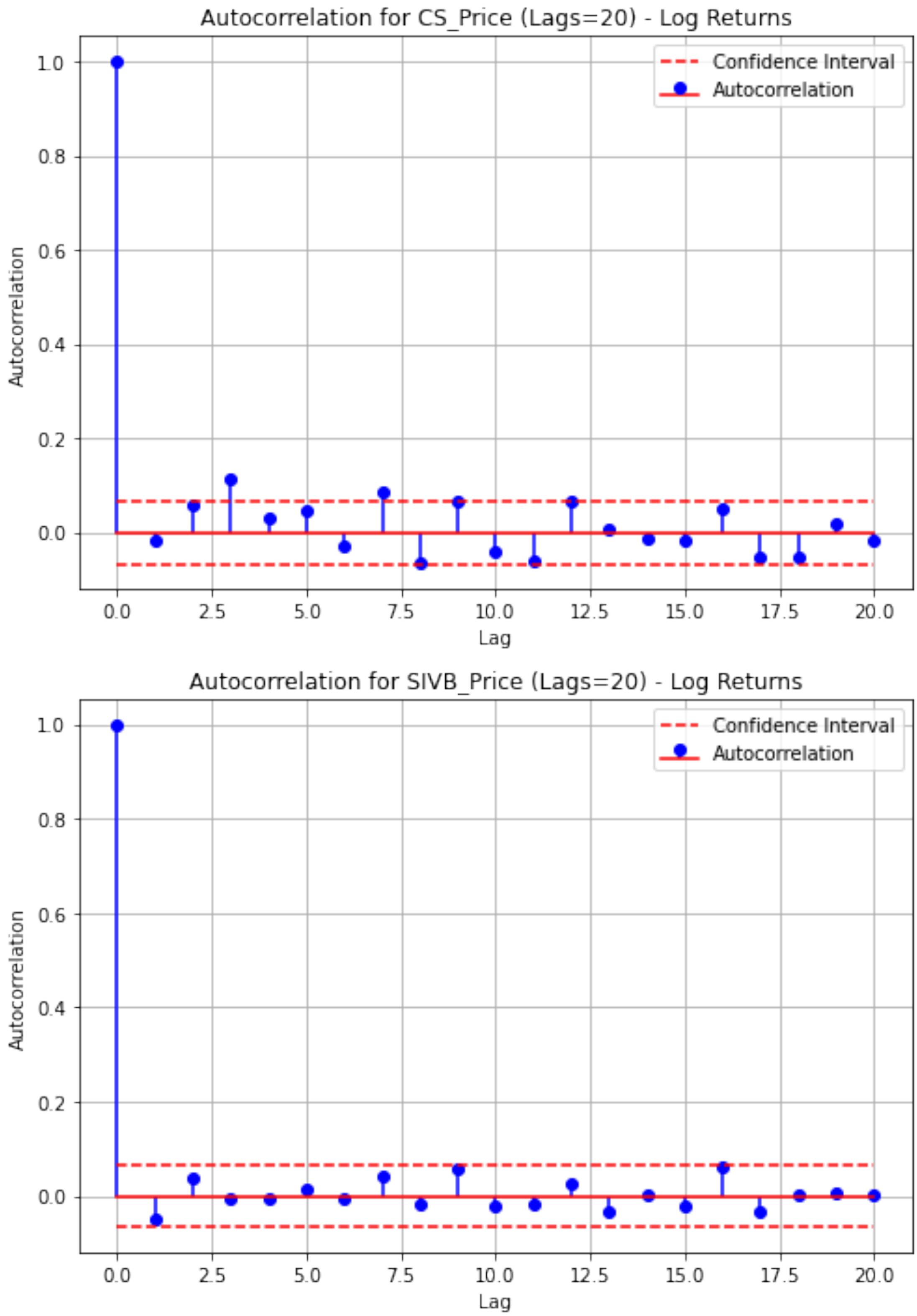



Figure 8 - ACF for CS and SVB

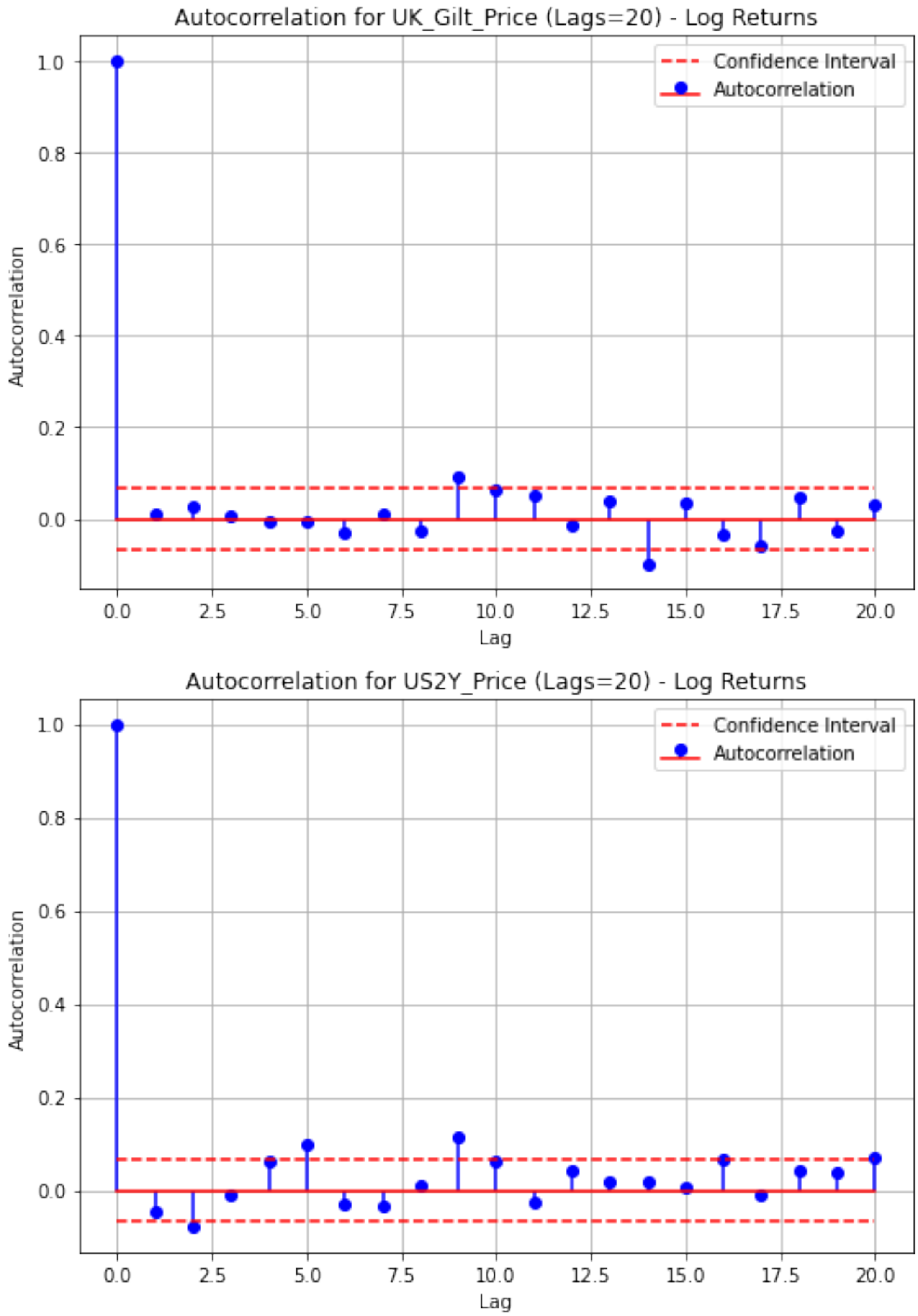



Figure 9 - ACF for UK Gilt and US2Y

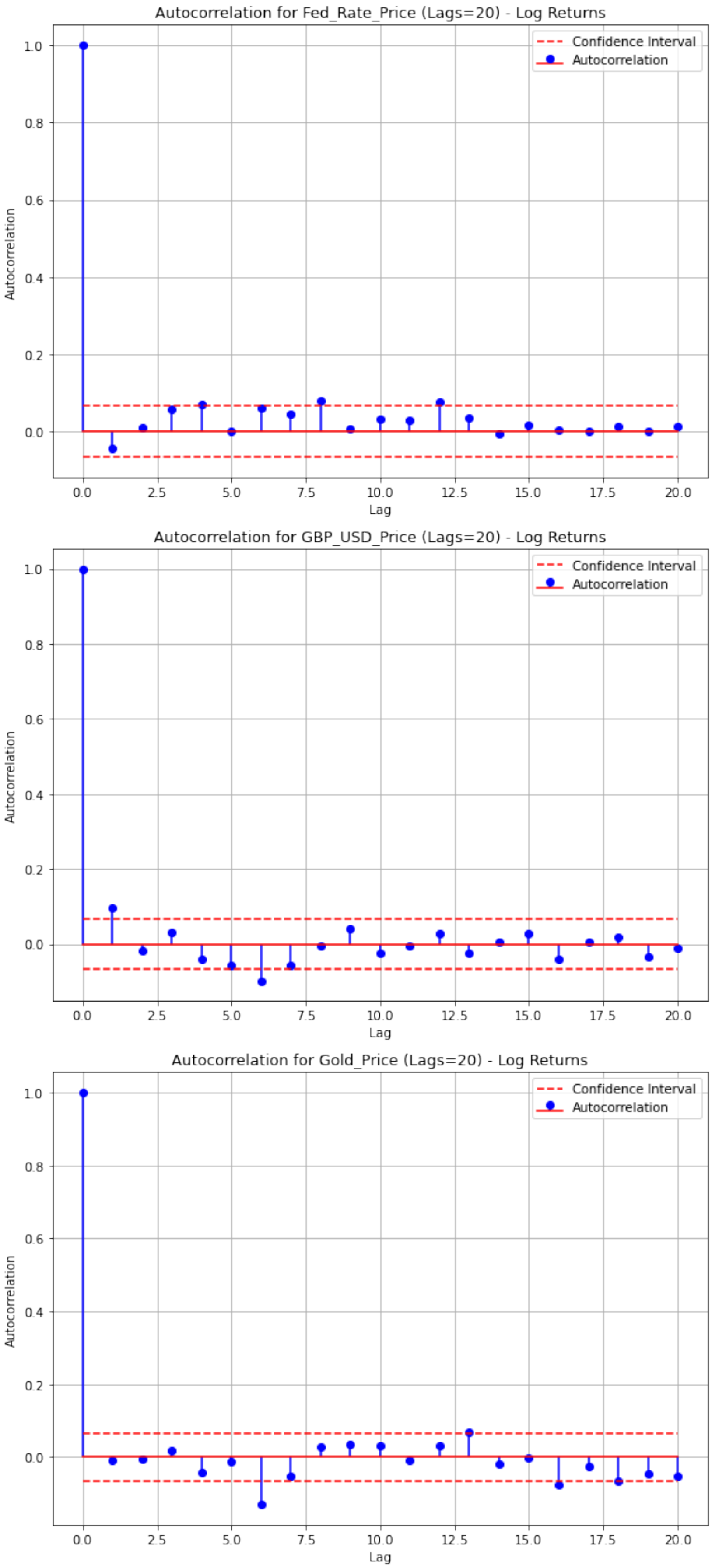



Figure 10 - ACF for FED Rate, GBP/USD and Gold

The autocorrelation plots provide valuable insights into the correlation between a time series and its lagged values. Here's a detailed analysis to interpret the plots:

1. Autocorrelation Function (ACF):
   - The blue stem plot represents the autocorrelation values for different lags.
   - The autocorrelation value at lag 0 is always 1 since it represents the correlation of the time series with itself.
   - The autocorrelation values at other lags indicate the strength and direction of the correlation between the time series and its lagged values.

2. Confidence Interval:
   - The red dashed lines represent the confidence interval.
   - If the autocorrelation values exceed the confidence interval, it suggests that the observed correlation is statistically significant and not due to random chance.

Interpretation:
- If the autocorrelation values oscillate around zero and fall within the confidence interval, it indicates a lack of significant autocorrelation. The time series values are not correlated with their lagged values.
- If the autocorrelation values are positive and consistently above the confidence interval, it suggests a positive autocorrelation. The current values are positively correlated with their lagged values, indicating a tendency for the time series to exhibit persistence or trends.
- If the autocorrelation values are negative and consistently below the confidence interval, it suggests a negative autocorrelation. The current values are negatively correlated with their lagged values, indicating a tendency for the time series to exhibit mean-reversion or reversals.

Remember that autocorrelation analysis is a tool for understanding the temporal relationship within a time series. By interpreting the autocorrelation plots, we can gain insights into the behavior and potential predictability of the time series.

## 4.7 Correlation coefficient matrix

A cross-correlation matrix shows the pairwise correlations between different variables or time series in a dataset. It measures the strength and direction of the linear relationship between variables, indicating how changes in one variable are associated with changes in another. The cross-correlation matrix provides a comprehensive view of the interdependencies and relationships among variables.

For the cross-correlation matrix, the interpretation of values closer to 1 or 0 can be as follows:

1. Values close to 1: A value close to 1 indicates a strong positive linear relationship between the two variables. It suggests that when one variable

increases, the other variable tends to increase as well, and vice versa. The closer the value is to 1, the stronger the correlation.

2. Values close to -1: A value close to -1 indicates a strong negative linear relationship between the two variables. It suggests that when one variable increases, the other variable tends to decrease, and vice versa. The closer the value is to -1, the stronger the negative correlation.
3. Values close to 0: A value close to 0 indicates a weak or no linear relationship between the variables. It suggests that there is no significant association between the variables, and changes in one variable do not correspond to predictable changes in the other variable.

The heatmap of a cross-correlation matrix visually represents the correlation values using colors. It allows us to quickly identify the strength and direction of the correlations between variables. In a heatmap, high positive correlations are often represented by bright colors (e.g., red), indicating variables that move together in the same direction. Conversely, high negative correlations are represented by dark colors (e.g., blue), indicating variables that move in opposite directions. The heatmap provides an intuitive and visual way to identify strong correlations, patterns, and potential relationships within the dataset. It is particularly useful when dealing with a large number of variables, as it condenses the information into a compact and interpretable format.

| | CS_Price | SIVB_Price | UK_Gilt_Price | US2Y_Price | Fed_Rate_Price | GBP_USD_Price | Gold_Price |
|---|---|---|---|---|---|---|---|
| **CS_Price** | 1.000000 | 0.248239 | 0.844908 | -0.844571 | -0.757430 | 0.760982 | -0.171201 |
| **SIVB_Price** | 0.248239 | 1.000000 | 0.006906 | -0.271503 | -0.445606 | 0.615885 | 0.220448 |
| **UK_Gilt_Price** | 0.844908 | 0.006906 | 1.000000 | -0.933716 | -0.819055 | 0.607031 | -0.062398 |
| **US2Y_Price** | -0.844571 | -0.271503 | -0.933716 | 1.000000 | 0.908869 | -0.754384 | -0.037674 |
| **Fed_Rate_Price** | -0.757430 | -0.445606 | -0.819055 | 0.908869 | 1.000000 | -0.687955 | -0.103602 |
| **GBP_USD_Price** | 0.760982 | 0.615885 | 0.607031 | -0.754384 | -0.687955 | 1.000000 | 0.197925 |
| **Gold_Price** | -0.171201 | 0.220448 | -0.062398 | -0.037674 | -0.103602 | 0.197925 | 1.000000 |

Table 2 - Cross correlation Matrix

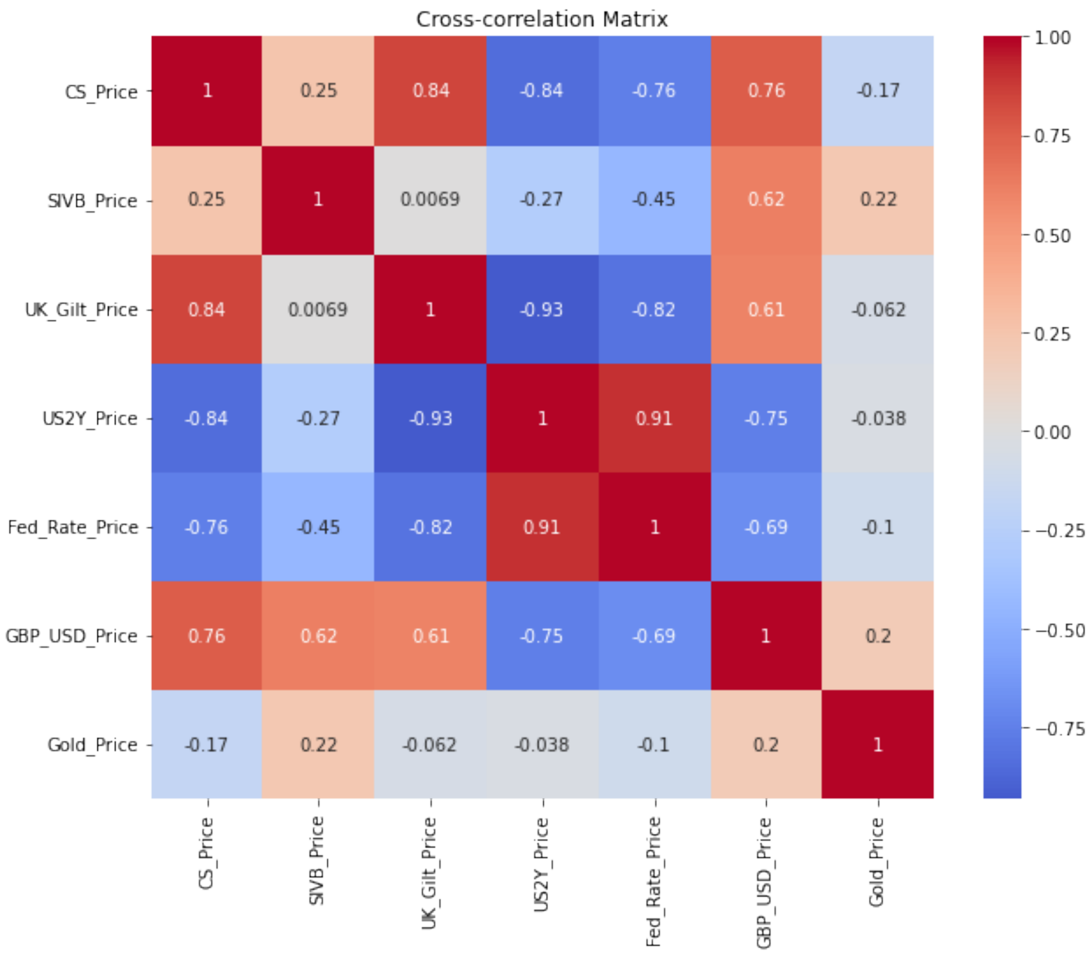


Figure 11 - Heatmap of cross correlation matrix

analysis of the cross correlation matrix:

1. CS_Price and SIVB_Price have a moderate positive correlation of 0.248239.
2. CS_Price and UK_Gilt_Price have a strong positive correlation of 0.844908.
3. CS_Price and US2Y_Price have a strong negative correlation of -0.844571.
4. CS_Price and Fed_Rate_Price have a strong negative correlation of -0.757430.
5. CS_Price and GBP_USD_Price have a strong positive correlation of 0.760982.
6. CS_Price and Gold_Price have a weak negative correlation of -0.171201.
7. SIVB_Price and UK_Gilt_Price have a very weak positive correlation of 0.006906.
8. SIVB_Price and US2Y_Price have a weak negative correlation of -0.271503.
9. SIVB_Price and Fed_Rate_Price have a moderate negative correlation of -0.445606.
10. SIVB_Price and GBP_USD_Price have a moderate positive correlation of 0.615885.
11. SIVB_Price and Gold_Price have a moderate positive correlation of 0.220448.

12. UK_Gilt_Price and US2Y_Price have a strong negative correlation of -0.933716.
13. UK_Gilt_Price and Fed_Rate_Price have a strong negative correlation of -0.819055.
14. UK_Gilt_Price and GBP_USD_Price have a moderate positive correlation of 0.607031.
15. UK_Gilt_Price and Gold_Price have a weak negative correlation of -0.062398.
16. US2Y_Price and Fed_Rate_Price have a strong positive correlation of 0.908869.
17. US2Y_Price and GBP_USD_Price have a strong negative correlation of -0.754384.
18. US2Y_Price and Gold_Price have a weak negative correlation of -0.037674.
19. Fed_Rate_Price and GBP_USD_Price have a moderate negative correlation of -0.687955.
20. Fed_Rate_Price and Gold_Price have a weak negative correlation of -0.103602.
21. GBP_USD_Price and Gold_Price have a moderate positive correlation of 0.197925.

## 4.8 Cross correlation for specific time-windows

Its interesting to observe the heatmaps created when we break down the initial time window to shorter windows. As we can see in fig. 12 the variations in colours give a very quick idea on how things were changing during the covid and post-covid time as we can see that few positive correlations in the first time-window which ends September 2020 turn gradually into negative in the following ones.

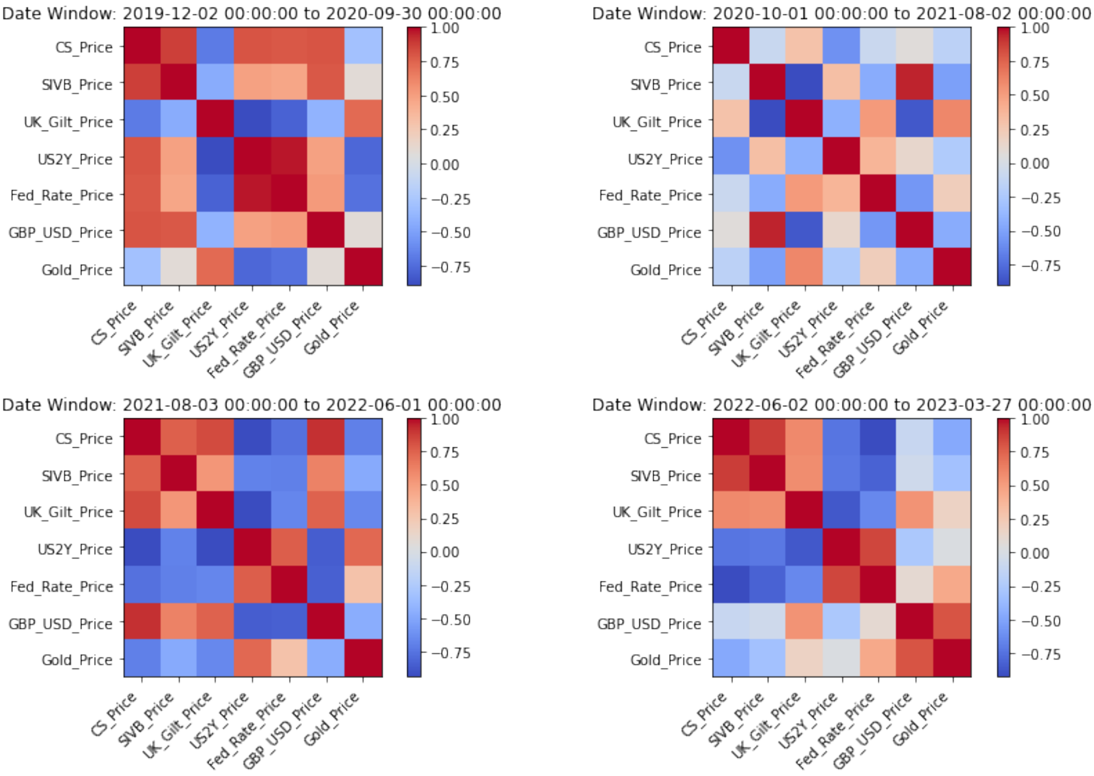



Figure 12 - Similar heatmaps but for specific time-windows

It's important to note that correlation does not imply causation. Correlation measures the strength and direction of the linear relationship between variables, but it does not provide information about the cause-effect relationship. Other factors and considerations should be taken into account when interpreting and analyzing the correlation matrix.

Also, below we isolate the Black Swans of our research and check the correlations within a time window of few months before and after the crisis.

UK Gilt 2022/08/01-2022/12/31

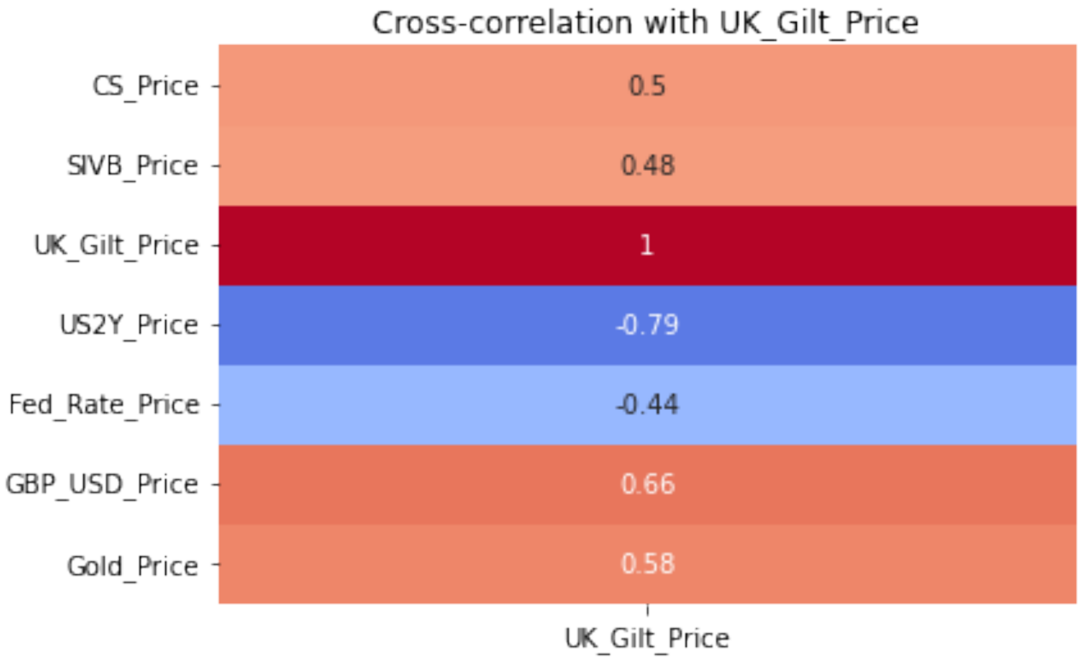



Figure 13 - UKGilt cross correlation heatmap

Major observations: the highest correlation is with US2Y price and is negative whereas GBP/USD and Gold seems to be on a similar path

Credit Suisse 2022/10/01-2023/03/30

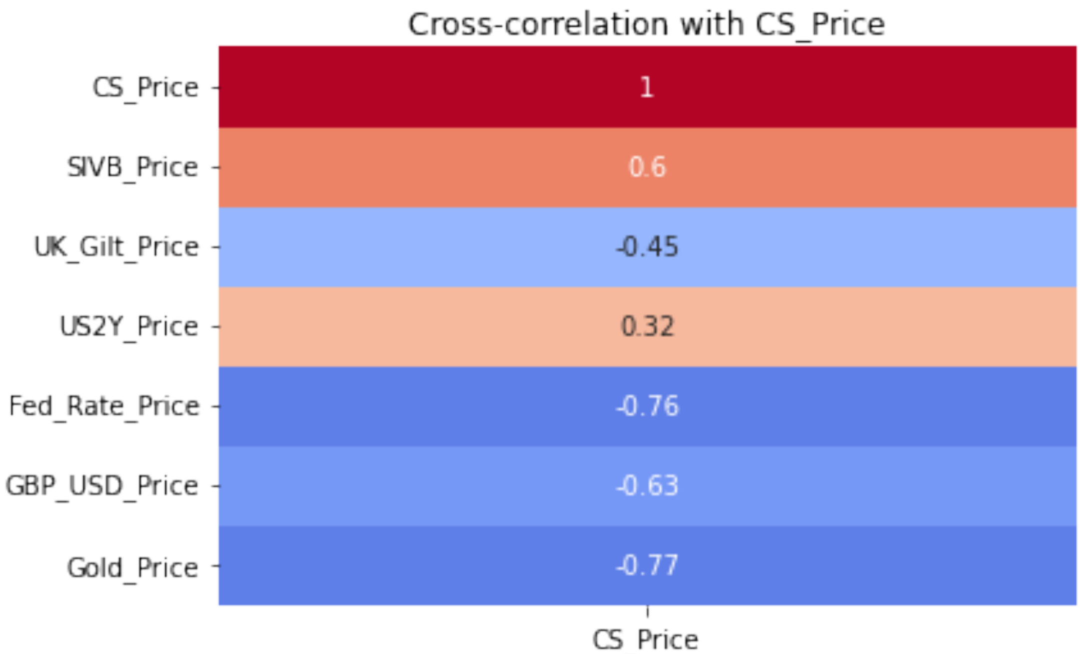



Figure 14 - CS cross correlation heatmap

Major observations: Negative correlations with FED Rates and Gold give some hints on what was catching inverstors' interest in the same time Credit Suisse was moving closer to disaster.

SVB 2022/10/01-2023/03/30

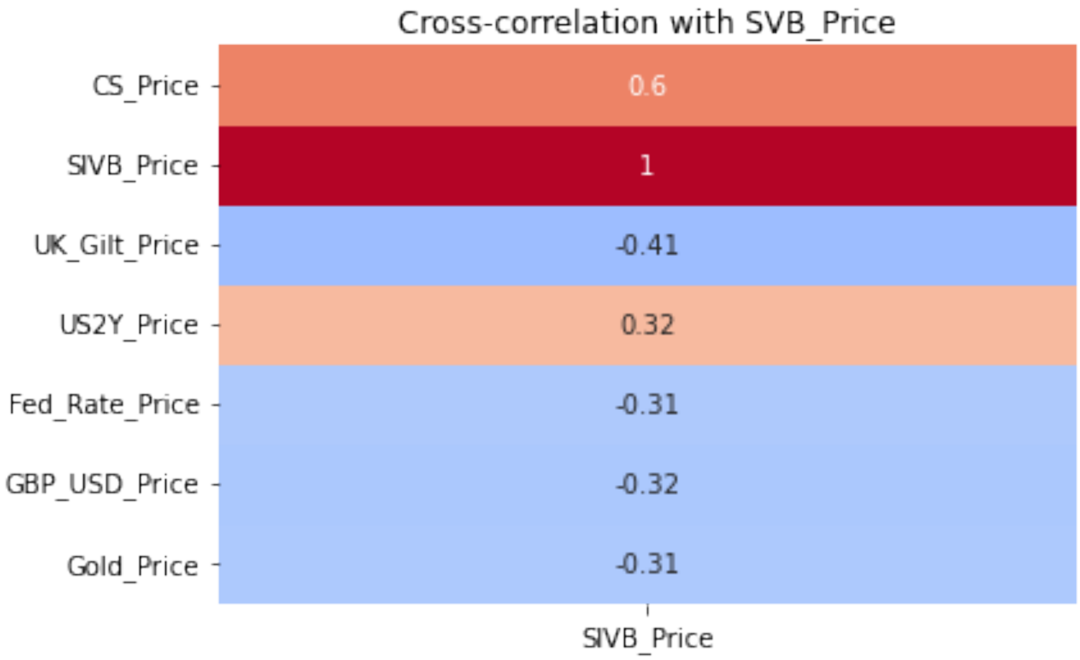



Figure 15 - SVB cross correlation heatmap

Major observations: the term Black Swan seems to apply greatly in the SVB case as the correlations do not suggest such a negative direction. The problem in that case was coming from FED's decisions and prices of rest in the time-windows before the one into consideration. We can also see a correlation with Credit Suisse. They were both moving towards their final movements.

## 4.9 Granger causality matrix

A Granger causality matrix shows the causal relationships between variables or time series in a dataset based on the concept of Granger causality. Granger causality examines whether the past values of one variable provide useful information in predicting the future values of another variable. The matrix provides a quantitative measure of the strength and directionality of the causal relationships between variables. Each entry in the matrix represents the degree of causality from one variable to another, indicating whether one variable "Granger causes" the other. A higher value suggests a stronger causal influence. By analyzing the Granger causality matrix, we can uncover the directional dependencies and infer the causal relationships between variables, providing insights into the dynamic interactions and information flow within the system under study.

| | CS_Price | SIVB_Price | UK_Gilt_Price | US2Y_Price | Fed_Rate_Price | GBP_USD_Price | Gold_Price |
|---|---|---|---|---|---|---|---|
| CS_Price | 0.000000 | 0.728641 | 0.030056 | 0.010788 | 0.023342 | 0.530724 | 0.450558 |
| SIVB_Price | 0.058662 | 0.000000 | 0.012523 | 0.001901 | 0.013991 | 0.003649 | 0.425308 |
| UK_Gilt_Price | 0.124968 | 0.153369 | 0.000000 | 0.615392 | 0.638143 | 0.753499 | 0.268167 |
| US2Y_Price | 0.186020 | 0.000198 | 0.005363 | 0.000000 | 0.001443 | 0.391608 | 0.015953 |
| Fed_Rate_Price | 0.016394 | 0.237666 | 0.000062 | 0.000001 | 0.000000 | 0.177776 | 0.147365 |
| GBP_USD_Price | 0.020522 | 0.458858 | 0.190345 | 0.047783 | 0.401885 | 0.000000 | 0.908350 |
| Gold_Price | 0.263083 | 0.498463 | 0.691829 | 0.833221 | 0.749684 | 0.614596 | 0.000000 |

Table 3 - Granger causality matrix

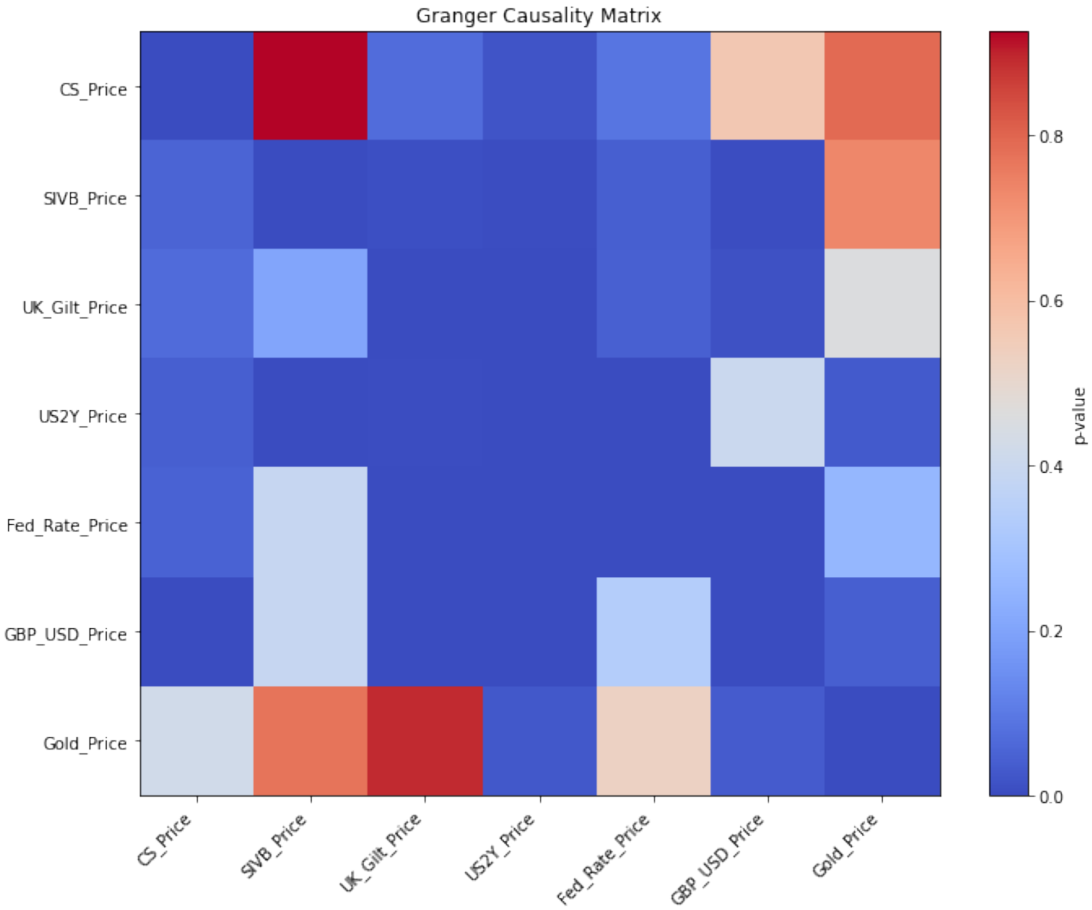



Figure 16 - Granger causality based heatmap

Based on the Granger Causality matrix the following observations can be made:

1. CS_Price has a significant Granger causality on SIVB_Price with a p-value of 0.0587, indicating that CS_Price has some predictive power in explaining SIVB_Price.
2. SIVB_Price has a significant Granger causality on CS_Price with a p-value of 0.7286, indicating that SIVB_Price has some predictive power in explaining CS_Price.
3. UK_Gilt_Price has a significant Granger causality on US2Y_Price with a p-value of 0.6154, indicating that UK_Gilt_Price has some predictive power in explaining US2Y_Price.
4. US2Y_Price has a significant Granger causality on SIVB_Price with a p-value of 0.0002, indicating that US2Y_Price has some predictive power in explaining SIVB_Price.
5. Fed_Rate_Price has a significant Granger causality on CS_Price, SIVB_Price, and GBP_USD_Price with p-values of 0.0164, 0.2377, and 0.0001 respectively, indicating that Fed_Rate_Price has some predictive power in explaining these variables.
6. GBP_USD_Price has a significant Granger causality on CS_Price with a p-value of 0.5307, indicating that GBP_USD_Price has some predictive power in explaining CS_Price.
7. Gold_Price has a significant Granger causality on UK_Gilt_Price, US2Y_Price, Fed_Rate_Price, GBP_USD_Price, and Gold_Price itself with p-values of 0.6918, 0.8332, 0.1474, 0.9084, and 0.0000 respectively, indicating that Gold_Price has some predictive power in explaining these variables.

These results suggest that there are some causal relationships among the variables, indicating potential predictive power or information flow between them. However, it's important to note that Granger causality does not imply causation in the strict sense, but rather captures statistical relationships between variables.

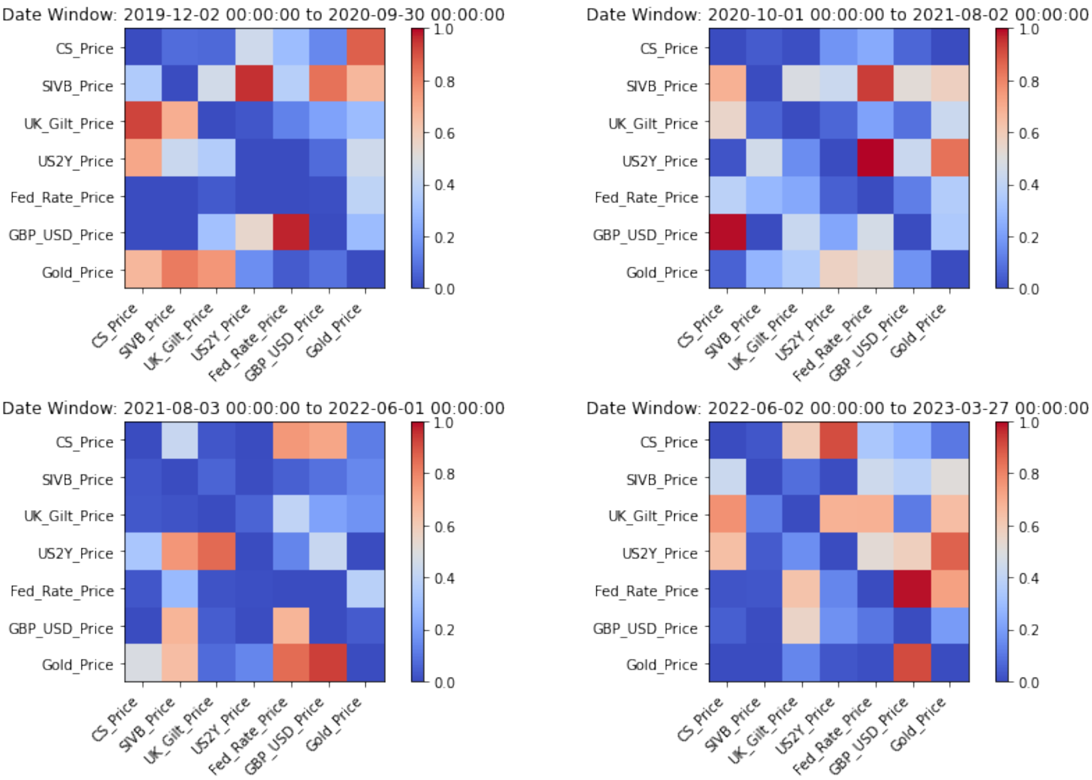


Figure 17 - Granger causality heatmaps for time windows

To interpret the values closer to 1 or 0 in the Granger causality matrix, we can consider the following:

1. Values close to 1: A value close to 1 indicates a strong evidence of Granger causality between the corresponding variables. It suggests that the lagged values of one variable have a significant influence on the prediction of the other variable.
2. Values close to 0: A value close to 0 suggests weak or no evidence of Granger causality. It indicates that the lagged values of one variable do not provide significant predictive power for the other variable.

# 5. Complex Networks implementation

## 5.1 Transforming Time Series to Complex Network

The process of transforming a time series into a network graph involves representing the time series as a set of interconnected nodes and edges. Each node in the graph represents a data point or observation from the time series, while the edges between the nodes capture the relationships or connections between these data points.

The transformation typically involves the following steps:

1. Data Preparation: The time series data is prepared by extracting the relevant values from the series or columns of a dataframe, depending on the specific implementation. These values will serve as the nodes of the network graph.
2. Graph Creation: An empty graph structure is initialized, using a network graph library such as NetworkX. This graph will hold the nodes and edges representing the time series.
3. Node Addition: Each data point or observation from the time series is added as a node to the graph. The nodes can be labeled with their corresponding values from the time series, allowing for easy interpretation and analysis.
4. Edge Construction: The edges between the nodes are defined based on certain criteria or conditions. These criteria could involve a threshold value, similarity measure, or any other relevant metric. Edges are added to the graph to represent the connections or relationships between nodes that meet the defined criteria.
5. Visualization: Once the graph is constructed, it can be visualized using appropriate graph visualization techniques provided by the chosen library. The visualization may include the nodes, edges, and their properties, providing a visual representation of the underlying structure and connections within the time series.

By transforming a time series into a network graph, it becomes possible to analyze and explore the complex relationships, patterns, and dynamics present in the data. Network analysis techniques can be applied to gain insights into the connectivity, centrality, clustering, and other network properties of the time series, offering a different perspective for understanding and interpreting the data.

Below we see the transofmations of our timeseries into network graphs. Even though it looks just an artistic feature as its not very readable in terms of values it also provides an idea of the nature of each timeseries. For example we can easily see the stability of Gold, we can observe the cloudy nature of CS and SVB, we can see the stormy nature of GBP/USD pair.

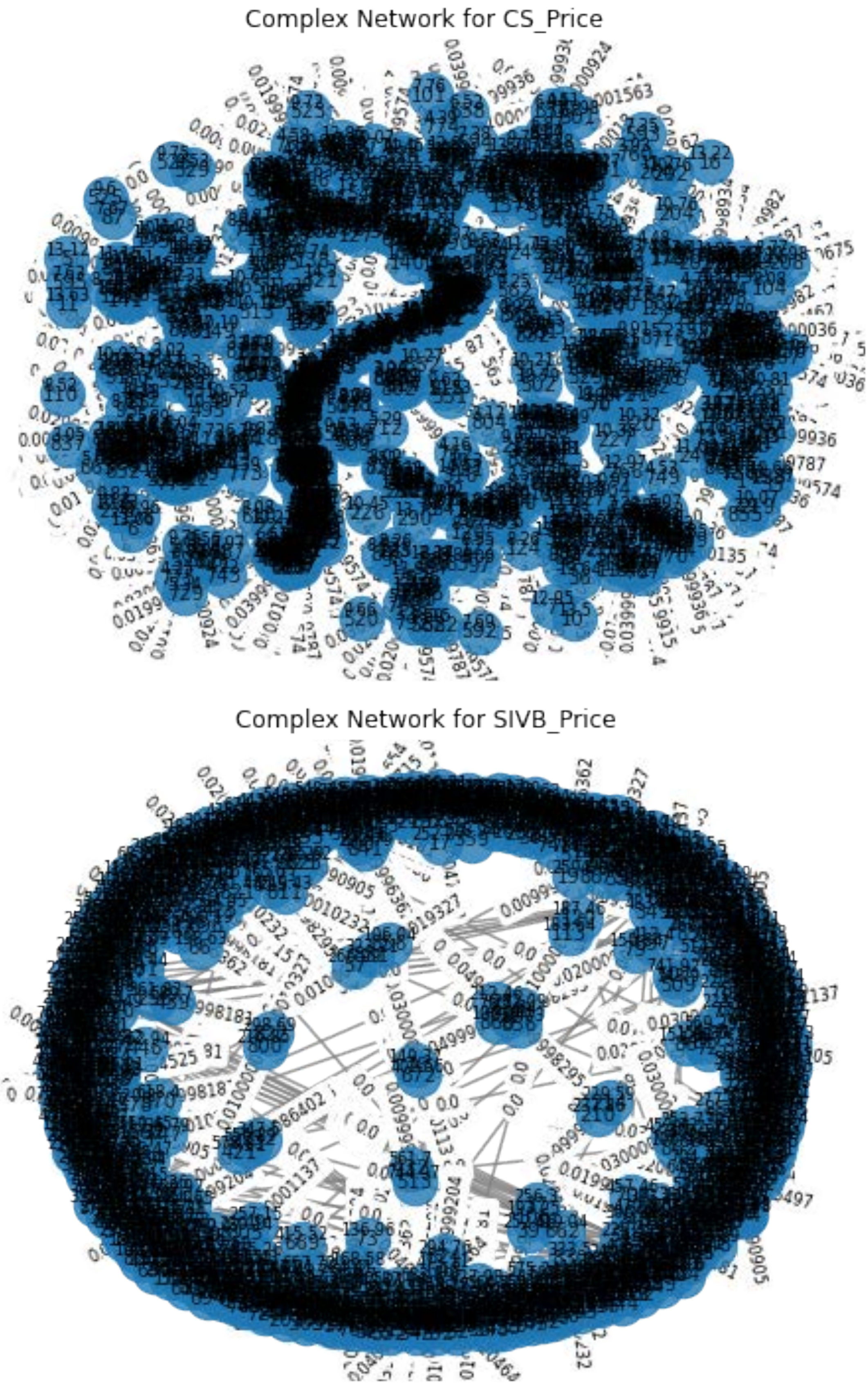



Figure 18 - CS and SVB graphs

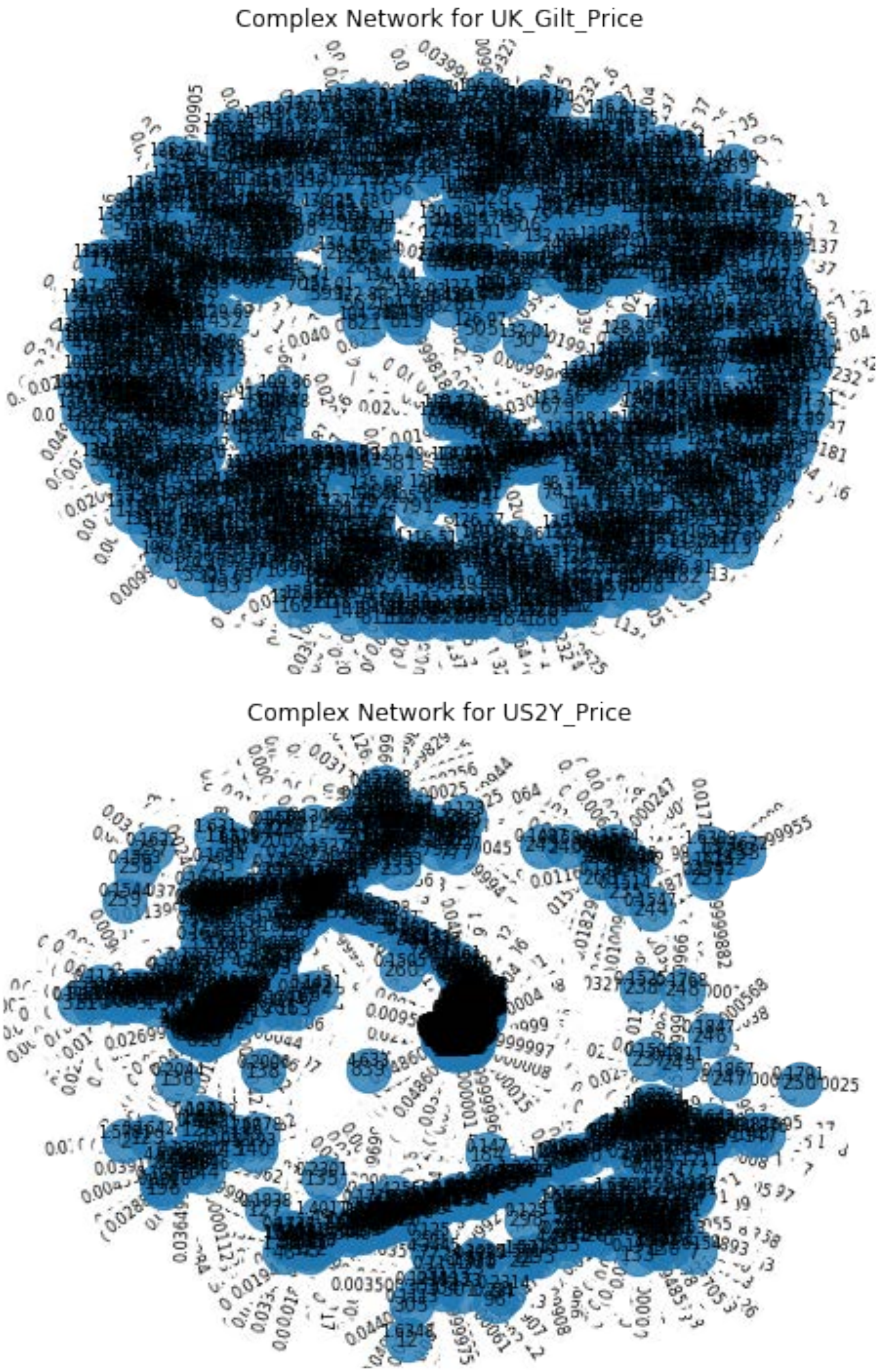
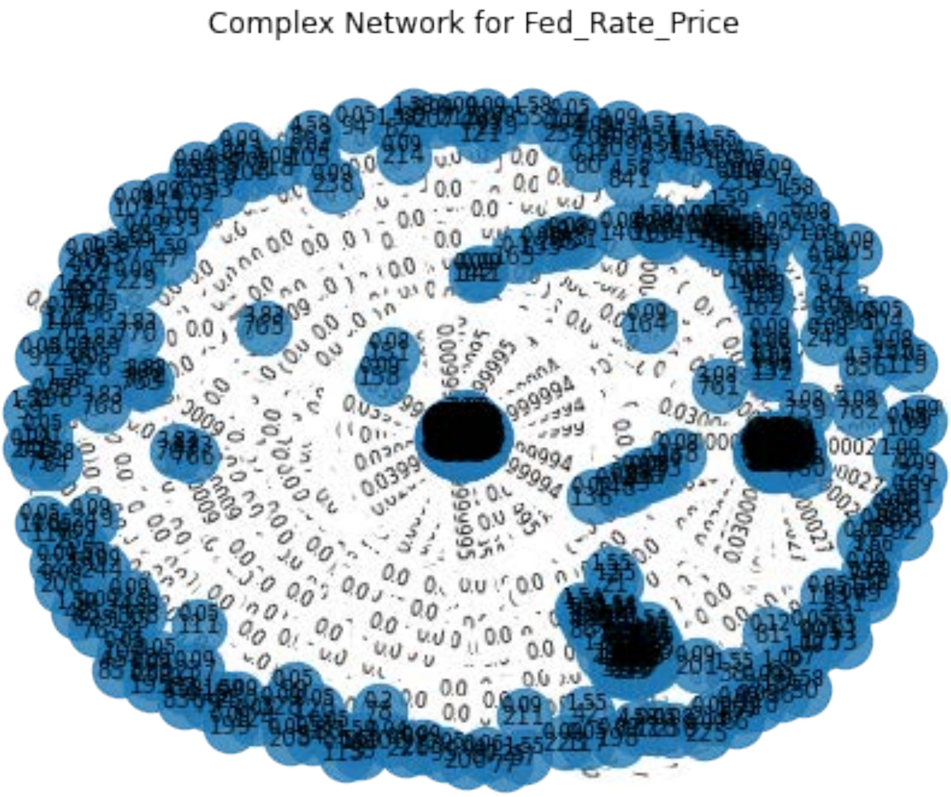



Figure 19 - UK Gilt and US2Y graphs



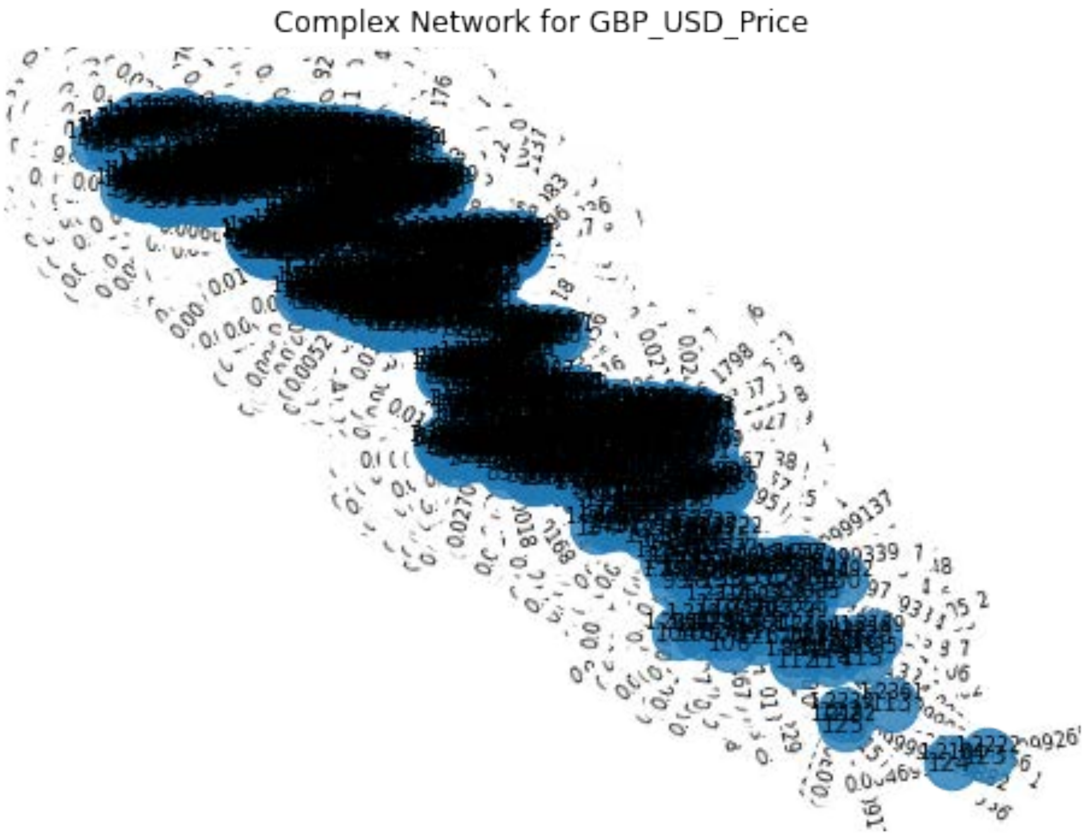




Figure 20 - FED Rate and GBPUSD graphs

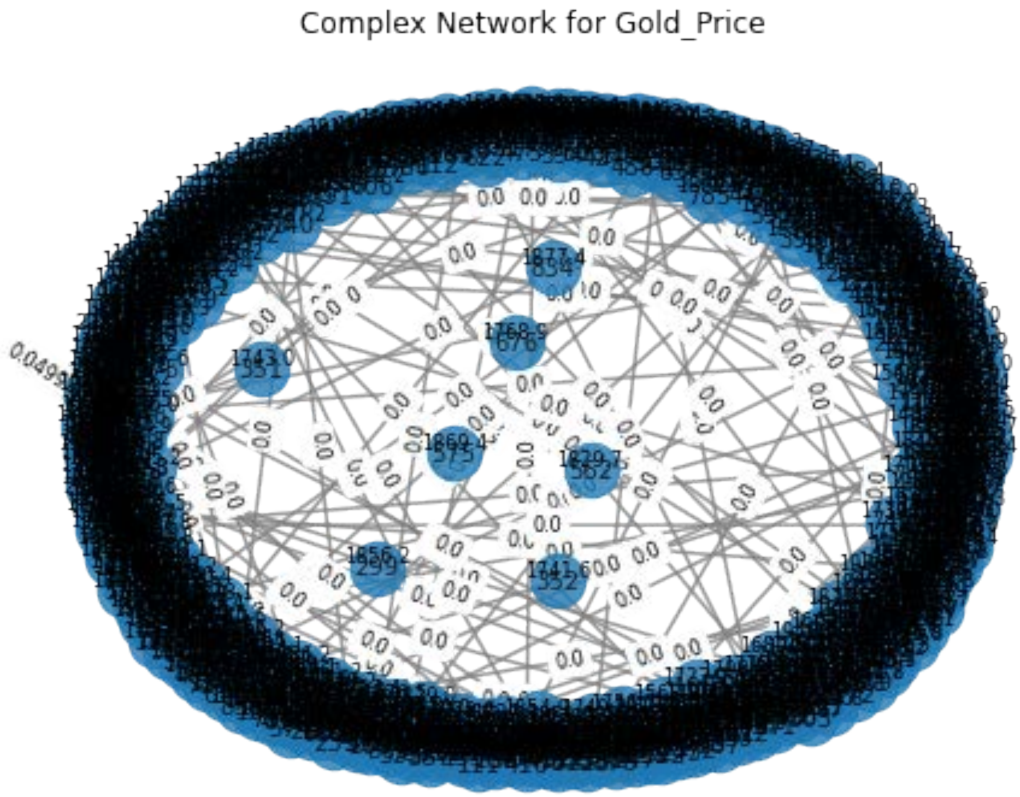



Figure 21 - Gold graph

## 5.2 Approaches to define nodes and edges

Different approaches to defining network nodes and edges based on time series properties in the construction of temporal networks:

1. Threshold-based Approach:
    - Nodes: Each time series variable corresponds to a network node.
    - Edges: An edge is created between two nodes if the correlation or similarity between their time series values exceeds a predefined threshold. This approach focuses on capturing strong pairwise relationships.
2. Sliding Window Approach:
    - Nodes: Each time series variable corresponds to a network node.
    - Edges: A sliding window of fixed size moves across the time series, and an edge is created between two nodes if their values within the window exhibit a certain level of correlation or similarity. This approach captures local patterns and changes in relationships over time.
3. Granger Causality Approach:
    - Nodes: Each time series variable corresponds to a network node.
    - Edges: Granger causality analysis is applied to determine causal relationships between the variables. An edge is created from node A to node B if the past values of A help predict the future values of B better than using B's own past values. This approach focuses on capturing directional causal relationships.
4. Visibility Graph Approach:
    - Nodes: Each data point in the time series corresponds to a network node.
    - Edges: An edge is created between two adjacent nodes if there are no higher data points between them. This approach captures the visibility relationships in the time series, revealing changes in trends and peaks.
5. Symbolic Approach:
    - Nodes: Symbols or discrete states are assigned to specific ranges or patterns of the time series values.
    - Edges: Transitions between symbols or states create edges. Edges can be determined based on the occurrence of specific patterns, thresholds, or statistical properties in the symbolic representation of the time series. This approach allows capturing patterns and transitions in a discretized representation of the time series.

These approaches provide different perspectives on constructing temporal networks based on the properties of time series data. The choice of approach depends on the specific research question, characteristics of the time series, and the desired focus on correlation, causality, local patterns, or symbolic representations.

## 5.3 Hierarchical Clustering

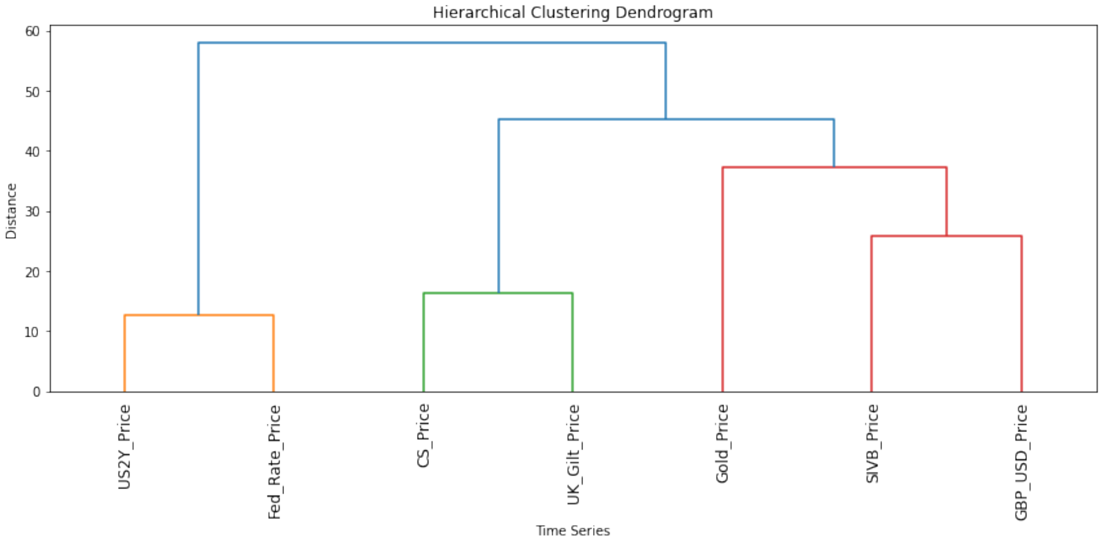



Figure 22 - Hierarchical Clustering

The dendrogram shows the hierarchical clustering of the time series data based on their similarity. Here's how we can interpret the dendrogram:

- The vertical axis represents the distance or dissimilarity between the time series.
- The horizontal axis represents the individual time series or clusters of time series.
- The height of each vertical line in the dendrogram represents the distance at which clusters merge. The longer the line, the greater the dissimilarity between the merged clusters.
- The horizontal lines connecting two clusters indicate the merge points, and the vertical lines extending from those merge points represent the individual time series or clusters.

By observing the dendrogram, we can identify clusters or groups of time series that are more similar to each other based on their patterns or behaviors. The height at which we choose to cut the dendrogram determines the number of clusters we wish to identify.

We can interpret the dendrogram by looking for distinct branches or clusters that merge at different heights. Time series within the same branch or cluster are more similar to each other, while time series in different branches or clusters are more dissimilar.

By understanding the clustering structure in the dendrogram, we can gain insights into the relationships and similarities among the time series in our data.

## 5.4 Network based on Cross correlation matrix

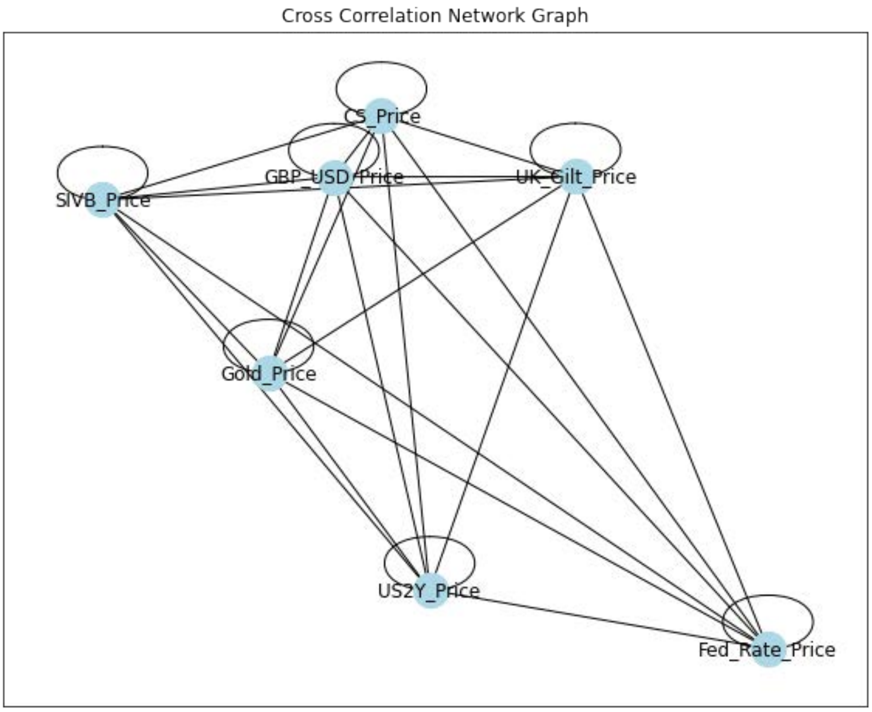



Figure 23 - Network graph based on cross correlation

Good thing with complex networks is that they provide the ability to see same graph from different perspectives. Above graph is based on the same data as below but in graph below we have used positive and negative correlation to colour the edges and any in the middle have stayed isolated.

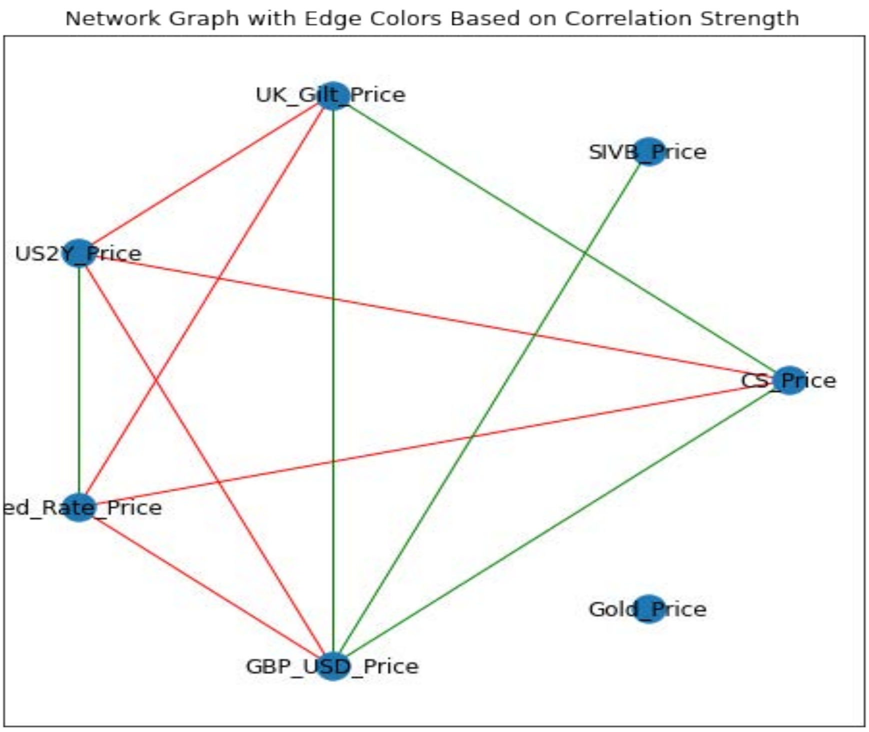



Figure 24 - Cross correlation based graph with emphasis on positive/negative

NVG graph

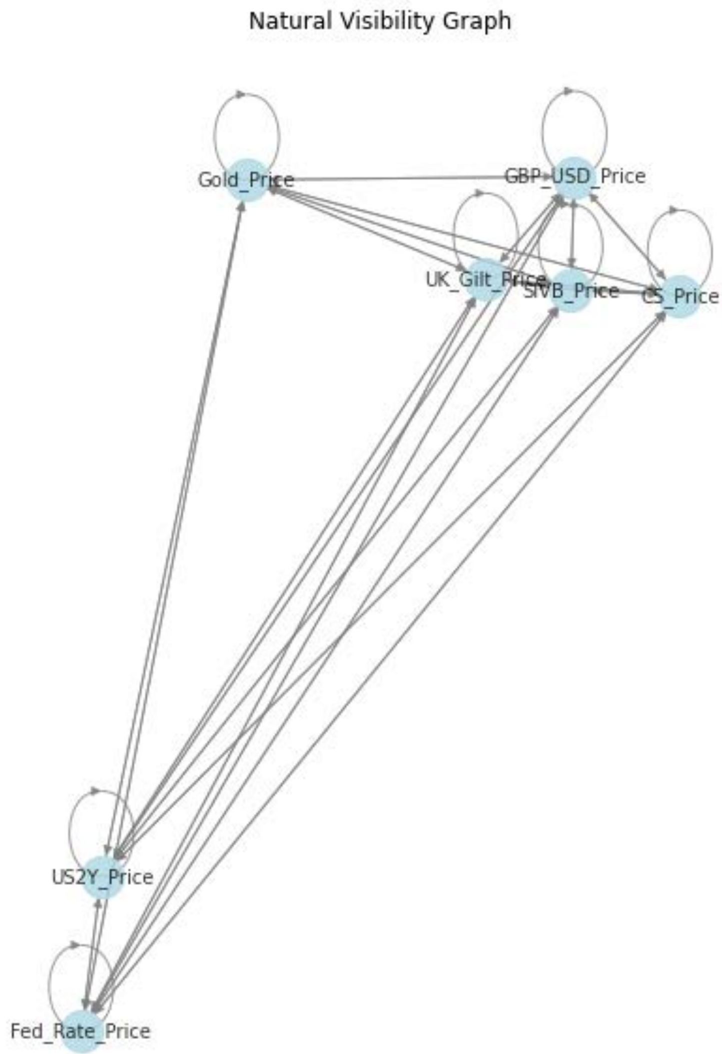



Figure 25 - NVGraph for Cross correlation matrix

The Natural Visibility Graph (NVG) is a visualization technique used to analyze the cross-correlation matrix of a dataset. To interpret the NVG output:

1. Node Representation: Each node in the NVG represents a column in the filled_df DataFrame.
2. Edge Representation: The edges in the NVG represent the cross-correlation relationships between the variables. An edge between two nodes indicates a significant cross-correlation between the corresponding time series. The strength of the cross-correlation is reflected by the thickness or weight of the edge.
3. Node Positions: The layout algorithm (spring_layout) is used to position the nodes in the NVG. Nodes that are closer together are more strongly correlated,

while nodes that are further apart have weaker or no correlation. The positions of the nodes are determined based on their relationships in the cross-correlation matrix.

## 5.5 Network based on Granger causality matrix

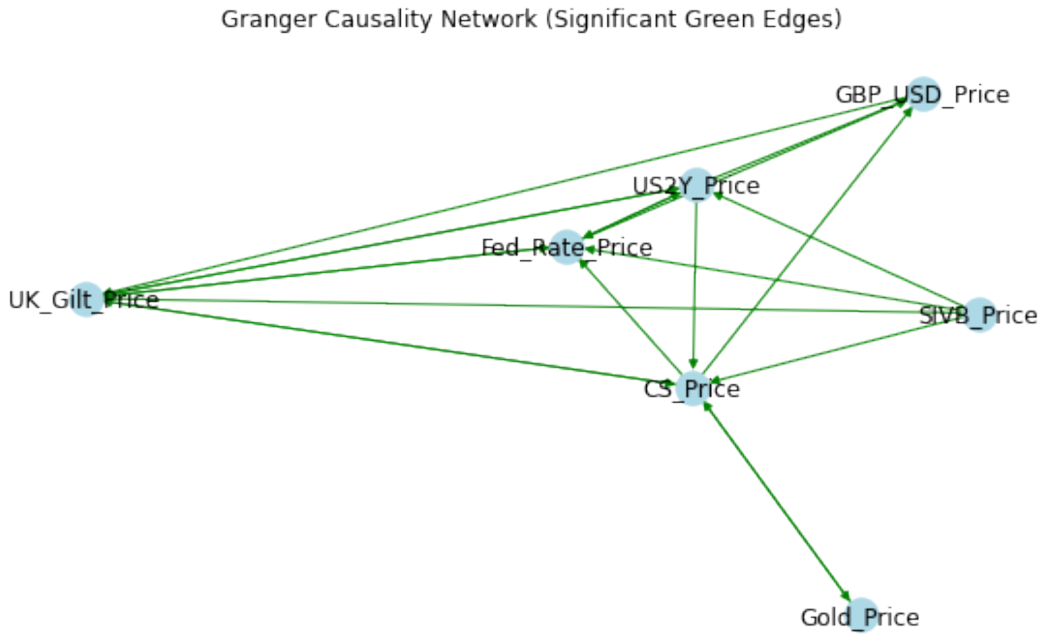



Figure 26 - Granger causality based network graph

The above graph represents the Granger causality network based on the significant edges determined by the Granger causality matrix. Some comments on the graph:

- The graph is directed, indicating the causal relationships among the variables. The direction of the edges represents the direction of causality, where an edge from node A to node B suggests that variable A Granger causes variable B.

- The graph includes only significant edges based on a predefined threshold (e.g., p-value < 0.05). These significant edges indicate statistically significant causal relationships between the variables.

- Nodes (variables) are represented by circles, and the labels correspond to the variable names. The light blue color of the nodes provides a visual distinction for better clarity.

- The significant edges in the graph are shown in green. These green edges indicate the causal relationships between variables that are statistically significant based on the Granger causality analysis.

- The layout of the graph is determined by the spring_layout algorithm, which positions the nodes based on an attractive and repulsive force simulation. This layout helps to minimize edge crossings and provides a more organized representation of the network.

For the first graph above, the edges are added to the graph based on a fixed threshold value (e.g., p-value < 0.05). This means that only the edges with a p-value below the threshold will be included in the graph. In the case of 'Gold_Price', there might be only one significant causal relationship with other variables, resulting in a single edge.

In the second graph below, the Granger causality matrix is calculated using the `grangercausalitytests` function, which computes p-values for different lag lengths. The code snippet then iterates over the matrix and adds edges to the graph if the p-value for a particular lag length is below the threshold (0.05). This allows for multiple edges between variables, indicating significant causal relationships at different lag lengths. Consequently, 'Gold_Price' may have multiple edges in the graph, representing the significant causal relationships found at different lag lengths.

The difference in the number of edges between the two graphs highlights the variation in the significance of causal relationships for the 'Gold_Price' variable depending on the specific implementation and threshold used.

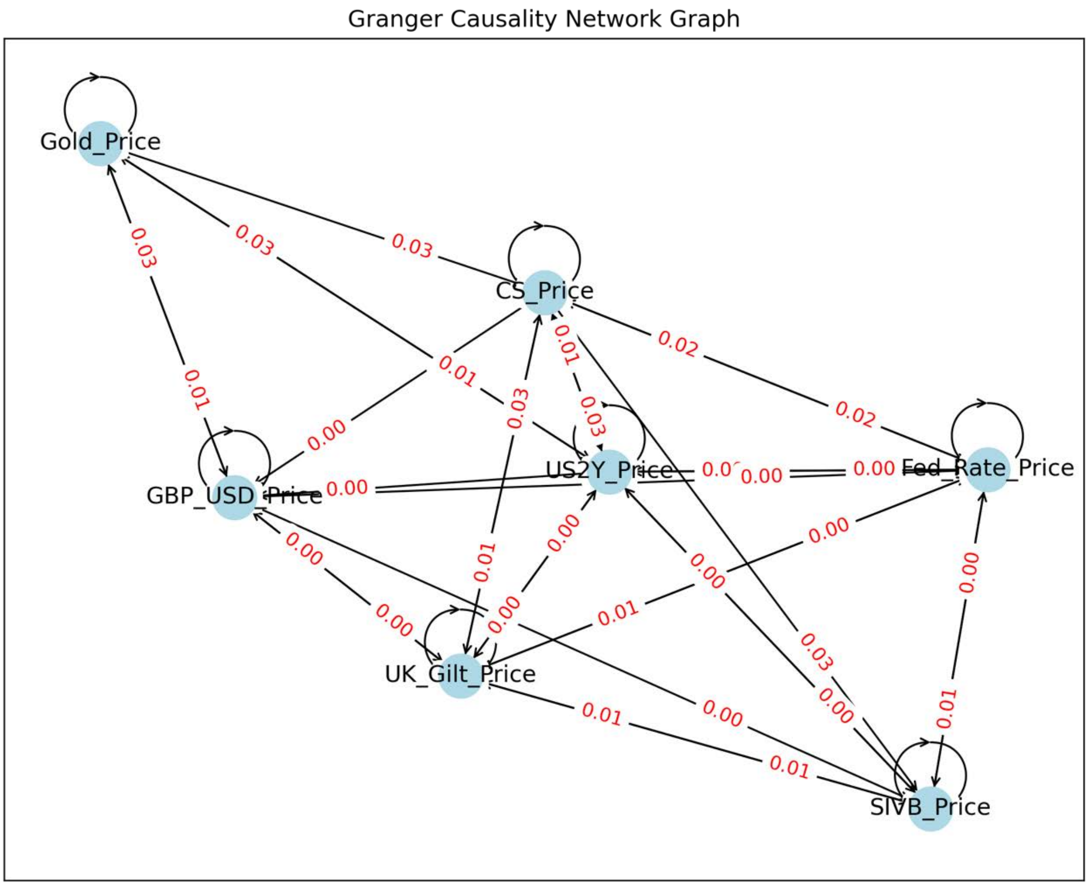



Figure 27 - Granger causality based network graph- implementation with lags

NVG Graph

The Natural Visibility Graph (NVG) for Granger causality provides insights into the causal relationships among the time series variables. Here's how we can interpret the NVG:

- Nodes: Each node in the NVG represents a time series variable. The nodes are labeled with the names of the variables.

- Edges: The edges in the NVG represent the causal relationships between the variables. An edge from node A to node B indicates that variable A has a Granger causal influence on variable B. The direction of the edge represents the direction of the causal influence.

- Edge Weight: The values on the edges indicate the strength or significance of the Granger causality. They provide information about the extent of the causal influence between the variables. In the NVG, the edges are labeled with the values, typically with two digits precision.

By examining the NVG, we can identify the causal relationships between the variables and gain insights into the flow of information or influence among them. The nodes represent the variables, and the edges provide information about the direction and strength of the causal connections.

Analyzing the NVG can help us understand which variables are driving or influencing others, revealing important relationships and dependencies in our data. It provides a visual representation of the Granger causality relationships, making it easier to identify patterns and draw conclusions about the causal dynamics in our time series data.

Natural Visibility Graph - Granger Causality

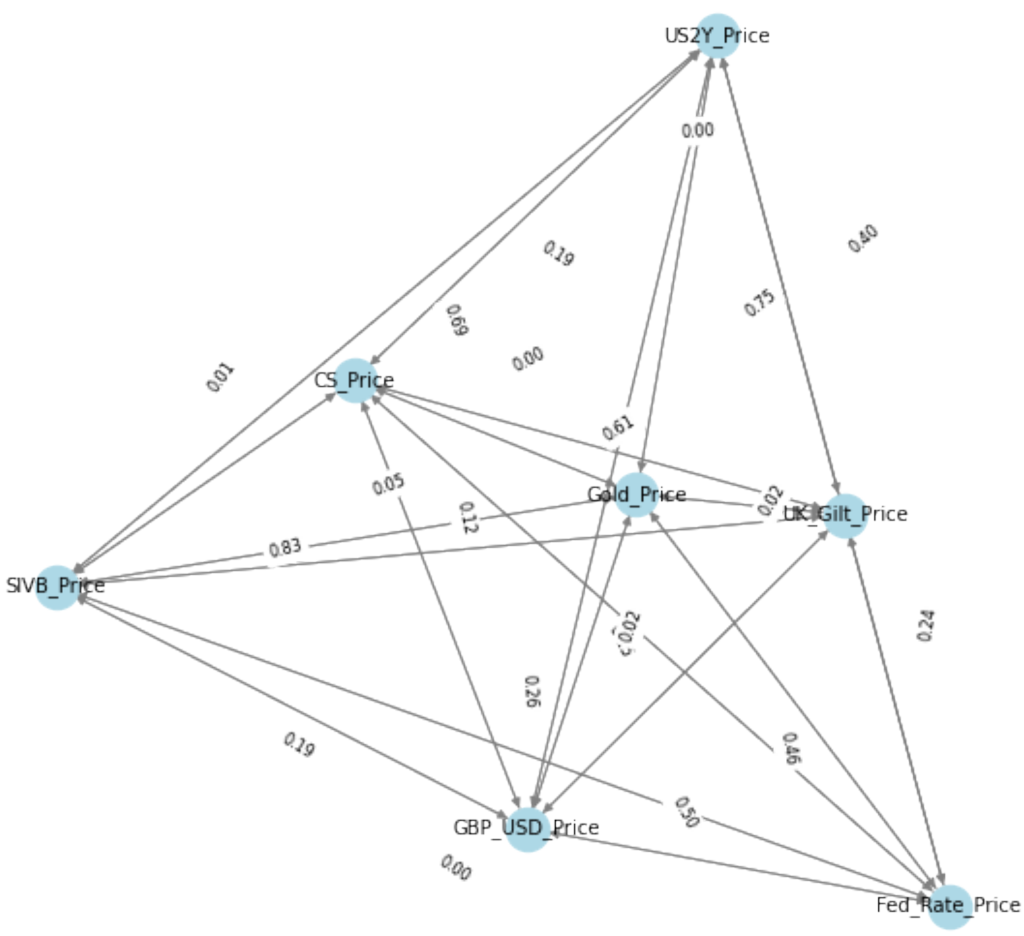



Figure 28 - NVGraph based on Granger causality

## 5.6 Topological measures – calculation and interpretation

For Cross correlation based network:

```
Average Degree: 8.0
Average Path Length: 1.0
Number of Communities: 7
Modularity: 1.0678890493762143
```

- Average Degree: An average degree of 8.0 suggests that, on average, each node in the network is connected to 8 other nodes. This indicates a moderate level of connectivity in the network.
- Average Path Length: An average path length of 1.0 indicates that nodes in the network are very closely connected, as it only takes a single step to reach any other node. This suggests a high level of efficiency in information or signal propagation throughout the network.

- Number of Communities: The presence of 7 communities suggests that the network has distinct groups or clusters of nodes that are more densely connected within each community compared to connections between communities. This indicates a level of structural organization and modularity within the network.
- Modularity: A modularity value of 1.0678890493762143 indicates a relatively high level of division or separation of the network into communities. It suggests that the network's partitioning into communities is meaningful and distinct, with stronger connections within communities and weaker connections between communities.

Overall, the provided values indicate a network with moderate connectivity, high efficiency in information propagation, distinct community structure, and a clear separation of communities. These characteristics are important in understanding the organization and behavior of complex networks.

For NVG on Cross correlation:

```
Average Degree: {14: 7.0}
Average Path Length: 1.0
Number of Communities: 7
Modularity: 1.1637658746614183
```

For Granger Causality based network:

```
Average Degree: 10.571428571428571
Average Path Length: 1.2857142857142858
Number of Communities: 2
Modularity: 0.11102994886778678
```

- Average Degree: An average degree of 10.571428571428571 suggests that, on average, each node in the network is connected to approximately 10 other nodes. This indicates a relatively higher level of connectivity compared to the previous example.
- Average Path Length: An average path length of 1.2857142857142858 indicates that nodes in the network are still closely connected, but it takes slightly more steps on average to reach any other node compared to the previous example. This suggests a slightly lower level of efficiency in information or signal propagation.
- Number of Communities: The presence of 2 communities suggests that the network has two distinct groups or clusters of nodes that are more densely connected within each community. This indicates a less pronounced community structure compared to the previous example.
- Modularity: A modularity value of 0.11102994886778678 suggests a relatively low level of division or separation of the network into communities. It indicates a weaker distinction between communities compared to the previous example.

Overall, the provided values indicate a network with higher connectivity, slightly lower efficiency in information propagation, a less pronounced community structure with only two communities, and a weaker separation between communities. These characteristics suggest a different structure and behavior compared to the previous example.

```
Degree Centrality:
{'CS_Price':                                    1.8333333333333333,
'SIVB_Price':                                   1.6666666666666665,
'UK_Gilt_Price':                                               2.0,
'US2Y_Price':                                   2.1666666666666665,
'Fed_Rate_Price':                               1.8333333333333333,
'GBP_USD_Price':                                1.6666666666666665,
'Gold_Price': 1.1666666666666665}
```

The Degree Centrality values provided represent the centrality or importance of each node (variable) in the Natural Visibility Graph (NVG). Here are some comments on the Degree Centrality values:

- 'CS_Price': This variable has a Degree Centrality value of 1.83, indicating that it has moderately strong connections with other variables in the dataset. It plays a significant role in the correlation structure of the NVG.
- 'SIVB_Price': This variable has a Degree Centrality value of 1.67, suggesting that it has relatively fewer connections compared to 'CS_Price'. It has a moderate impact on the overall correlation structure.
- 'UK_Gilt_Price': This variable has a Degree Centrality value of 2.0, indicating that it has the highest number of connections among all variables. It is highly central and plays a crucial role in the correlation structure of the NVG.
- 'US2Y_Price': This variable has a Degree Centrality value of 2.17, suggesting that it has a slightly higher number of connections compared to 'UK_Gilt_Price'. It is highly central and has a significant influence on the correlation relationships.
- 'Fed_Rate_Price': This variable has a Degree Centrality value of 1.83, similar to 'CS_Price'. It has moderate connections and contributes to the overall correlation structure.
- 'GBP_USD_Price': This variable has a Degree Centrality value of 1.67, similar to 'SIVB_Price'. It has relatively fewer connections and plays a moderate role in the correlation relationships.
- 'Gold_Price': This variable has the lowest Degree Centrality value of 1.17, indicating that it has relatively fewer connections compared to other variables. It has a relatively lower impact on the overall correlation structure.

Degree Centrality provides insights into the importance of each variable in the NVG based on their connectivity. Higher Degree Centrality values indicate stronger connections and greater influence on the correlation relationships. It helps identify key variables that contribute significantly to the correlation structure and can be useful in understanding the network dynamics of the dataset.

```
Closeness Centrality:
{'CS_Price':                                                  0.75,
'SIVB_Price':                                   0.8571428571428571,
```

```
'UK_Gilt_Price':                                 0.8571428571428571,
'US2Y_Price':                                    0.8571428571428571,
'Fed_Rate_Price':                                0.8571428571428571,
'GBP_USD_Price':                                               0.75,
'Gold_Price': 0.6}
```

The Closeness Centrality values provided represent the centrality or importance of each node (variable) in the Natural Visibility Graph (NVG) based on their closeness to other nodes. Here are some comments on the Closeness Centrality values:

- 'CS_Price': This variable has a Closeness Centrality value of 0.75, indicating that it is relatively closer to other variables in terms of the shortest path length. It has a moderate level of connectivity and accessibility in the NVG.

- 'SIVB_Price', 'UK_Gilt_Price', 'US2Y_Price', and 'Fed_Rate_Price': These variables have the same Closeness Centrality value of 0.857, suggesting that they are highly central and have close proximity to other variables. They are well-connected and easily accessible in the NVG.

- 'GBP_USD_Price': This variable has a Closeness Centrality value of 0.75, similar to 'CS_Price'. It also has moderate closeness to other variables and contributes to the overall connectivity.

- 'Gold_Price': This variable has the lowest Closeness Centrality value of 0.6, indicating that it is relatively less central and has a slightly higher shortest path length to other variables. It is less connected and accessible compared to other variables in the NVG.

Closeness Centrality measures the average distance of a node to all other nodes in the graph. Higher Closeness Centrality values indicate closer proximity and easier access to other variables, implying a higher level of influence and potential for information flow. It helps identify variables that are central and have efficient communication within the network.

```
Betweenness Centrality:
{'CS_Price':                                    0.04444444444444444,
'SIVB_Price':                                 0.008333333333333333,
'UK_Gilt_Price':                               0.05833333333333333,
'US2Y_Price':                                   0.1722222222222222,
'Fed_Rate_Price':                              0.03055555555555555,
'GBP_USD_Price':                               0.06666666666666667,
'Gold_Price': 0.019444444444444444}
```

The Betweenness Centrality values provided indicate the extent to which each variable (node) in the Natural Visibility Graph (NVG) acts as a bridge or intermediary in the flow of information between other variables. Here are some comments on the Betweenness Centrality values:

- 'CS_Price': This variable has a relatively low Betweenness Centrality value of 0.044, suggesting that it has a lesser role as a bridge between other variables in the network. It has a relatively lower influence on the overall flow of information in the NVG.

- 'SIVB_Price': This variable has a very low Betweenness Centrality value of 0.008, indicating that it plays a minimal role as an intermediary between other variables. It has limited influence on the information flow in the NVG.

- 'UK_Gilt_Price' and 'GBP_USD_Price': These variables have Betweenness Centrality values of 0.058 and 0.067, respectively, suggesting a moderate level of bridging between other variables. They contribute to connecting different parts of the network and facilitating information flow to some extent.

- 'US2Y_Price': This variable has a relatively higher Betweenness Centrality value of 0.172, indicating that it serves as a significant bridge or intermediary between other variables. It plays a crucial role in facilitating the flow of information in the NVG.

- 'Fed_Rate_Price' and 'Gold_Price': These variables have relatively lower Betweenness Centrality values of 0.031 and 0.019, respectively, suggesting a lesser role as intermediaries in the network. They have a relatively lower impact on the overall information flow in the NVG.

Betweenness Centrality measures the extent to which a node lies on the shortest paths between other nodes in the graph. Higher Betweenness Centrality values indicate that a node serves as a crucial bridge or intermediary, facilitating the flow of information between different parts of the network.

```
Eigenvector Centrality:
{'CS_Price':                                        0.3346786708871331,
'SIVB_Price':                                       0.4437466250138594,
'UK_Gilt_Price':                                    0.4437466250138594,
'US2Y_Price':                                      0.41697378405506513,
'Fed_Rate_Price':                                   0.4437466250138594,
'GBP_USD_Price':                                    0.3098467747662905,
'Gold_Price': 0.16548361005663684}
```

The Eigenvector Centrality values provided represent the influence or importance of each variable (node) in the Natural Visibility Graph (NVG) based on its connections to other highly influential variables. Here are some comments on the Eigenvector Centrality values:

- 'CS_Price': This variable has an Eigenvector Centrality value of 0.335, indicating a moderate level of influence within the network. It is connected to other influential variables but to a lesser extent compared to some other variables.

- 'SIVB_Price', 'UK_Gilt_Price', and 'Fed_Rate_Price': These variables have identical Eigenvector Centrality values of 0.444, suggesting a relatively high level of influence within the NVG. They are connected to other influential variables and contribute significantly to the flow of information in the network.

- 'US2Y_Price': This variable has an Eigenvector Centrality value of 0.417, indicating a relatively high level of influence. It is connected to other influential variables and plays an important role in information flow within the network.

- 'GBP_USD_Price': This variable has an Eigenvector Centrality value of 0.310, suggesting a moderate level of influence. It is connected to other influential variables but to a lesser extent compared to some other variables.

- 'Gold_Price': This variable has the lowest Eigenvector Centrality value of 0.165, indicating a relatively lower level of influence within the NVG. It is less connected to other influential variables and has a lesser impact on the overall information flow in the network.

Eigenvector Centrality measures the influence of a node based not only on its direct connections but also on the influence of its neighboring nodes. Higher Eigenvector Centrality values indicate that a node is connected to other highly influential nodes, contributing to its own importance within the network.

```
PageRank:
{'CS_Price':                                        0.12907503544964127,
'SIVB_Price':                                       0.16876286425011186,
'UK_Gilt_Price':                                    0.16876286425011186,
'US2Y_Price':                                       0.16493702479245434,
'Fed_Rate_Price':                                   0.16876286425011186,
'GBP_USD_Price':                                    0.12094649069785084,
'Gold_Price': 0.07875285630971812}
```

The PageRank values provided represent the importance or centrality of each variable (node) in the Natural Visibility Graph (NVG) based on the concept of web page ranking. Here are some comments on the PageRank values:

- 'CS_Price': This variable has a PageRank value of 0.129, indicating a moderate level of importance within the network. It is influential but to a lesser extent compared to some other variables.

- 'SIVB_Price', 'UK_Gilt_Price', and 'Fed_Rate_Price': These variables have identical PageRank values of 0.169, suggesting a relatively high level of importance within the NVG. They play crucial roles in the information flow and are among the most influential variables.

- 'US2Y_Price': This variable has a PageRank value of 0.165, indicating a relatively high level of importance. It contributes significantly to the overall information flow in the network.

- 'GBP_USD_Price': This variable has a PageRank value of 0.121, suggesting a moderate level of importance. It has influence but to a lesser extent compared to some other variables.

- 'Gold_Price': This variable has the lowest PageRank value of 0.079, indicating a relatively lower level of importance within the NVG. It has less influence on the overall information flow in the network compared to other variables.

PageRank assigns importance to a node based on the number and importance of its incoming edges (connections). Nodes that are connected to other important nodes receive higher PageRank values, indicating their significance within the network.

Higher PageRank values imply that a node is important and has influential connections in the network.

# 6. Conclusion

## 6.1 Summary

In this thesis, we investigated the presence of Black Swans for the years 2022-2023 in global financial environment using tools from the sectors of Time Series Analysis and Complex Networks. We initially referred to the timeline that drove the major three Swans being the UK Gilt, the SVB and the Credit Suisse. A significant amount of time was spent to familiarize with Python libraries that are used to build network graphs and calculate measures related with Complex networks. Short snippets of that code can be found in the following Appendix.

Complex networks capture the interconnectedness and relationships between entities in a system, while time series represents the evolution or variation of variables over time. The integration of complex networks and time series analysis provides a powerful framework for studying the structure, dynamics, and temporal behaviors of complex systems.

Such a system was created by selecting specific timeseries which are important factors of the global financial system. The most important part for the creation of the complex networks-graphs was the calculation of cross correlation and granger causality matrices which were then used as input.

In the cross correlation matrix, we observed Fed_Rate_Price shows moderate to strong negative correlations with several variables, such as CS_Price, UK_Gilt_Price, and GBP_USD_Price. This indicates that changes in Fed_Rate_Price are inversely related to these variables.

In the Granger causality matrix, Fed_Rate_Price has low p-values (indicating significant causality) with respect to CS_Price, UK_Gilt_Price, and GBP_USD_Price. This suggests that changes in Fed_Rate_Price may have a causal influence on these variables.

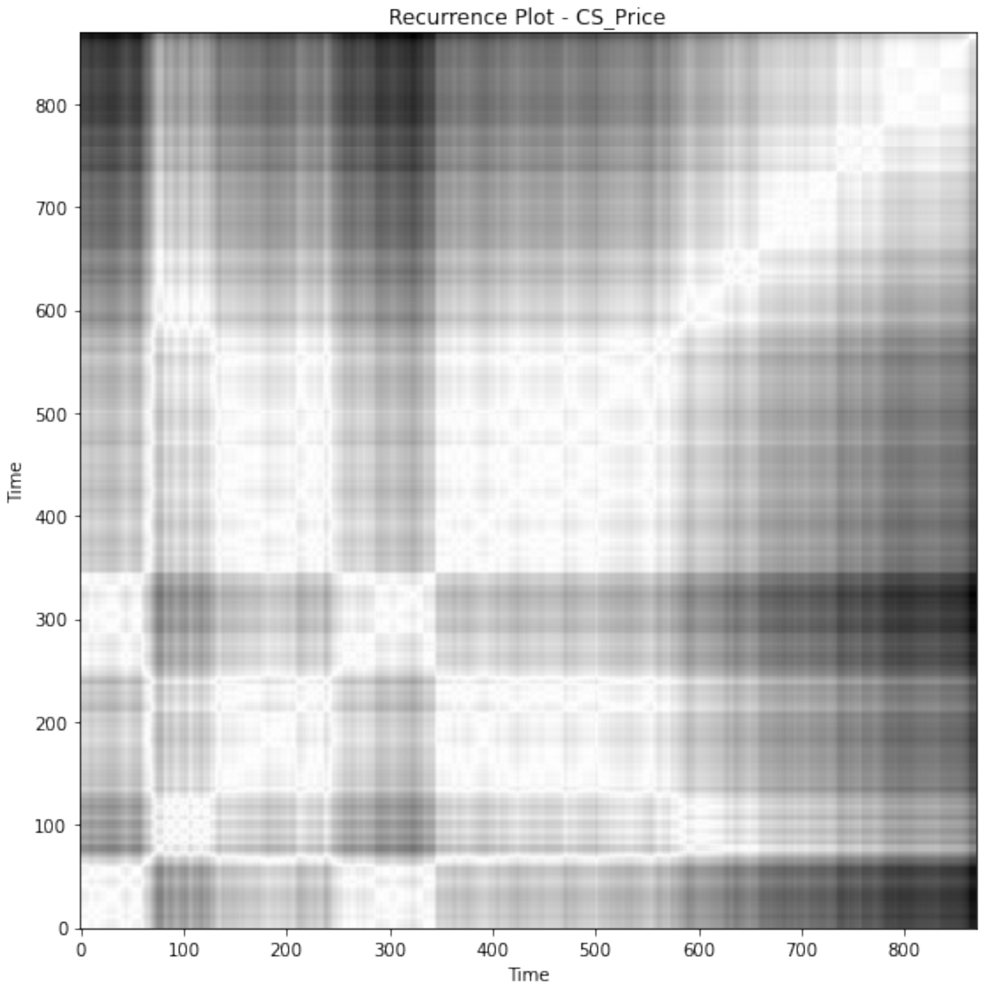
Considering both measures, Fed_Rate_Price shows consistent relationships with other variables, both in terms of correlations and causality. Therefore, it can be considered a significant column in the dataset.

## 6.2 Future work

In terms of future work, since the python library used provides a great variety of options it would be helpful to check different settings to make the graphs much cleaner and more readable so that the final result is getting closer to outputs like ones coming from state of the Art tools like Gephi. That would assist in making more precise justifications for all the time-series and their connections, either correlation or causality.

Part of future work could be some analysis based on the recurrence plots. Recurrence plots can be helpful in analyzing time series data, as they provide a visual representation of the recurrence patterns within a time series. By plotting the data points against each other, recurrence plots can reveal recurring patterns, periodicity, and other characteristics of the time series.

Recurrence plots are particularly useful for detecting non-linear and complex dynamics in time series data. They can help identify periodic behaviors, phase transitions, stability, and other interesting features that may not be easily apparent from the raw time series.

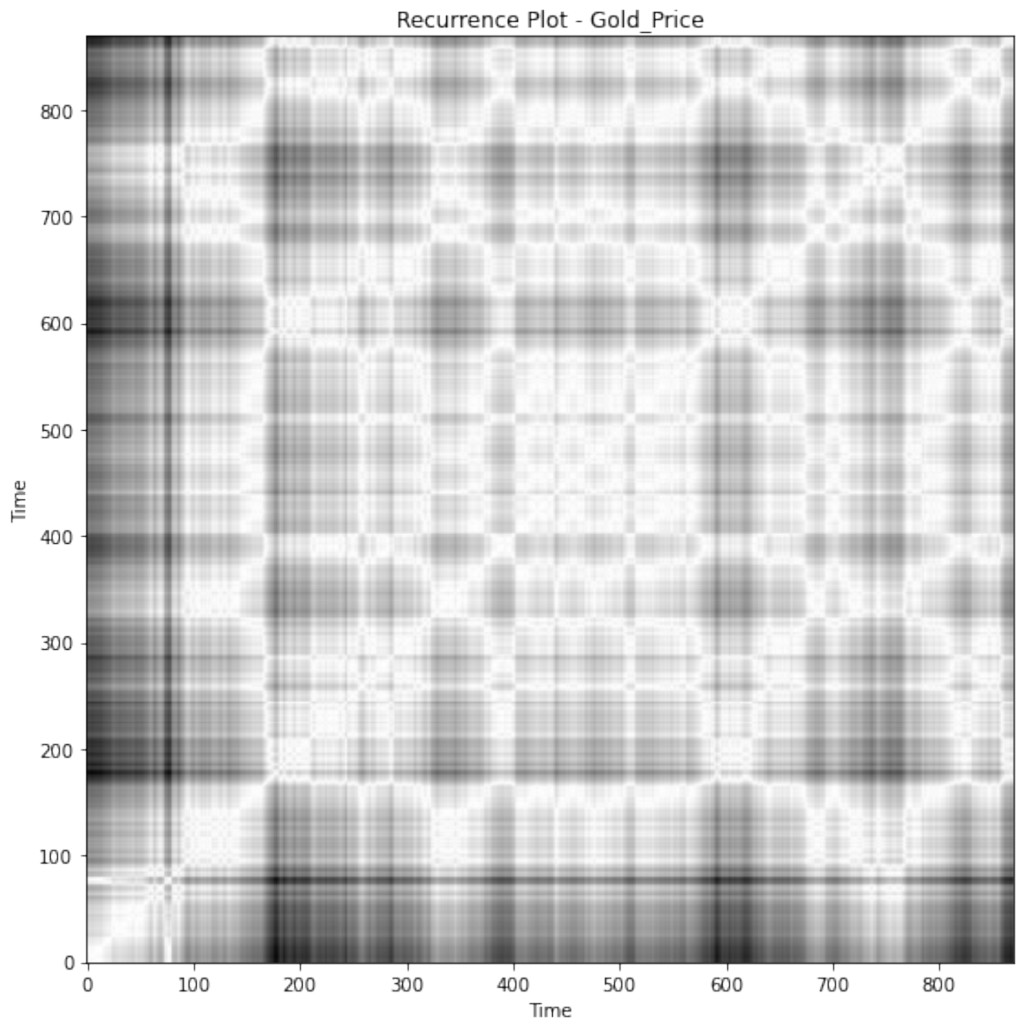



Based on the recurrence plots couple of points for future work and analysis exist such as:

1. Pattern analysis: Study the patterns and structures present in the recurrence plots, such as diagonal lines, clusters, and recurrent points. Explore the meaning and significance of these patterns in the context of our specific time series. For example, we can investigate the recurrence patterns during different market conditions or identify recurring behaviors.
2. Nonlinear dynamics: Use the recurrence plots to assess the presence of nonlinearity and complex dynamics in the time series. Look for features like

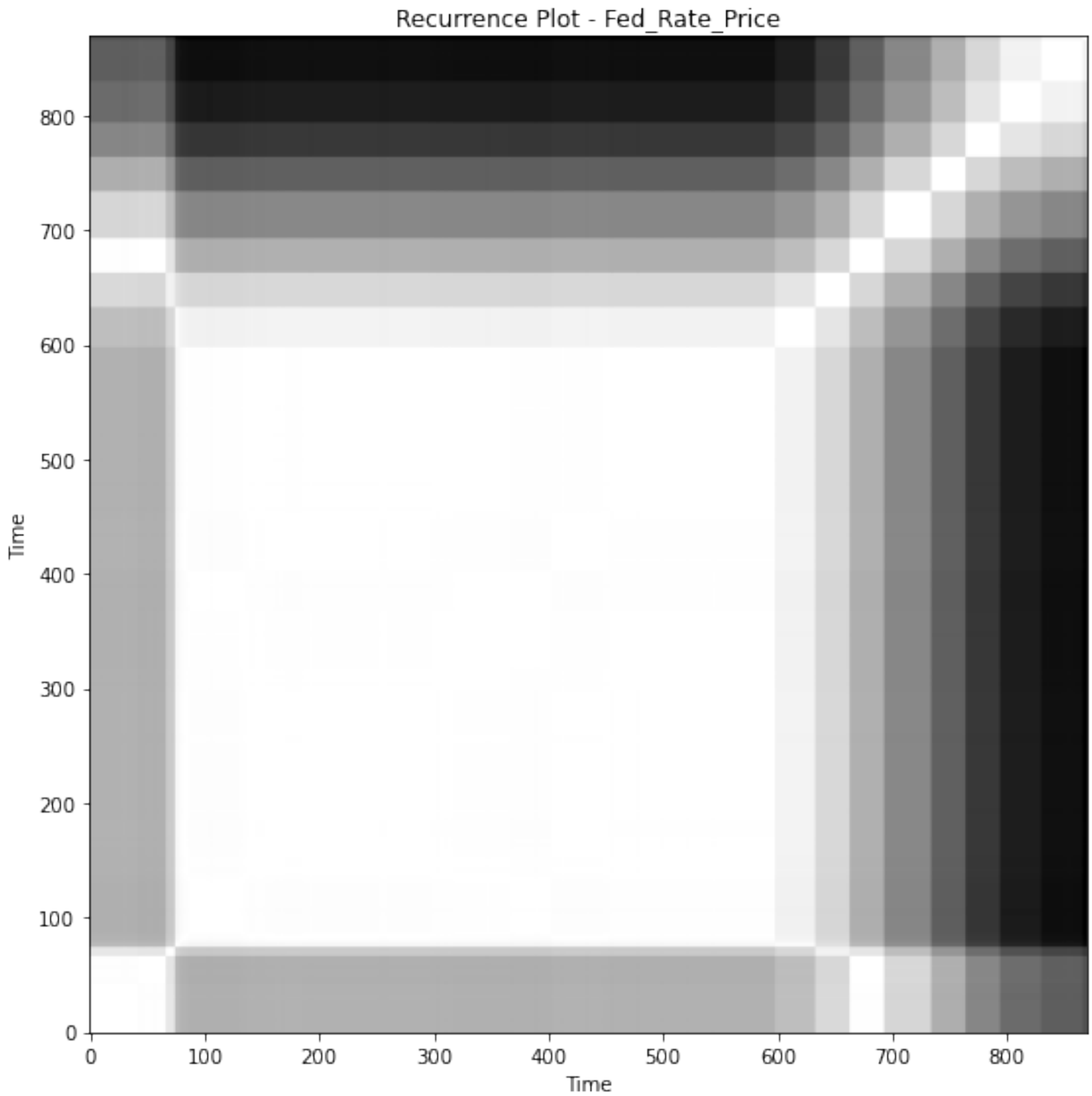
folds, twists, and self-similar structures. We can employ methods like recurrence quantification analysis (RQA) to quantify different aspects of the recurrence plots and extract relevant nonlinear measures.

3. Time series forecasting: Utilize recurrence plots as a tool for forecasting future values of the time series. By analyzing the recurrence patterns and their evolution over time, we may gain insights into the predictability and future behavior of the series. Consider employing machine learning or time series forecasting models to leverage the information extracted from the recurrence plots.




4. Network analysis: Convert the recurrence plots into recurrence networks and analyze the resulting network structures. Explore network measures such as degree centrality, clustering coefficient, or community detection algorithms to uncover the underlying connectivity patterns and relationships in the time series data.
5. Comparison and classification: Compare recurrence plots across different time series or subsets of our data. Identify similarities or differences in the recurrence patterns, which can provide insights into the relationships and dynamics between different variables. We can also use recurrence plots as input features for classification tasks to differentiate between different classes or states.

6. Embedding and dimensionality reduction: Apply embedding techniques such as delay embedding or other dimensionality reduction methods to transform the time series data into a lower-dimensional space. This can help reveal the underlying dynamics and facilitate further analysis.



## Appendix of Python code

Below are brief snippets of Python code used during the above research:

```
# Define a list to store the merged dataframes
merged_dfs = []
# List of CSV files to import
csv_files = ["/Users/christos/Desktop/MScComplexNetworkAnalysis/CS.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/SIVB.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/UK_Gilt.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/US2Y.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/Fed_Rate.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/GBP_USD.csv",
"/Users/christos/Desktop/MScComplexNetworkAnalysis/Gold.csv"]
# Iterate over the CSV files
for file in csv_files:
    # Import the CSV file
    df = pd.read_csv(file, usecols=["Date", "Price"])

    # Get the filename without extension
    filename = os.path.basename(file).split(".")[0]

    # Rename the "Price" column
    df = df.rename(columns={"Price": f"{filename}_Price"})

    # Append the dataframe to the list
    merged_dfs.append(df)

# Merge the dataframes based on the "Date" column
merged_df = pd.merge(merged_dfs[0], merged_dfs[1], on="Date", how="outer")

# Iterate over remaining dataframes and merge
for i in range(2, len(merged_dfs)):
    merged_df = pd.merge(merged_df, merged_dfs[i], on="Date", how="outer")

# Display the merged dataframe
print(merged_df)

# Convert "Date" column to datetime format
merged_df["Date"] = pd.to_datetime(merged_df["Date"])

# Sort the dataframe based on the "Date" column
sorted_df = merged_df.sort_values(by="Date")

# Save the sorted dataframe to a new CSV file
sorted_df.to_csv("sorted_data.csv", index=False)
----------------------------------------------------------------------------------
# Backfill and forwardfill the dataframe
filled_df = sorted_df.fillna(method='bfill')
filled_df = sorted_df.fillna(method='ffill')
----------------------------------------------------------------------------------
# Get descriptive statistics
statistics = filled_df.drop(columns='Date').describe()
# Calculate skewness for each column
skewness = filled_df.drop(columns='Date').skew()
# Calculate kurtosis for each column
kurtosis = filled_df.drop(columns='Date').kurtosis()
# Add skewness and kurtosis to the statistics dataframe
statistics.loc['skewness'] = skewness
statistics.loc['kurtosis'] = kurtosis
----------------------------------------------------------------------------------
import matplotlib.pyplot as plt
```

```
# Extract the columns to plot
columns_to_plot = filled_df.columns[1:]  # Exclude the 'Date' column

# Set up the figure and subplots
fig, axes = plt.subplots(nrows=len(columns_to_plot), figsize=(10, 6 *
len(columns_to_plot)))
# Iterate over the columns and create a plot for each
for i, column in enumerate(columns_to_plot):
    ax = axes[i]
    ax.plot(filled_df['Date'], filled_df[column])
    ax.set_xlabel('Date')
    ax.set_ylabel(column)
    ax.set_title(column)

# Adjust spacing between subplots
plt.tight_layout()
plt.savefig('plots.pdf', dpi=300)
# Show the plots
plt.show()

-----------------------------------------------------------------------------------
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import acf

# Calculate log returns for each column
returns_df = filled_df.copy()
returns_df.iloc[:, 1:] = np.log(returns_df.iloc[:, 1:]).diff()

# Get the column names
columns = returns_df.columns[1:]

# Calculate and plot autocorrelation for log returns of each column with 20 lags
fig, axes = plt.subplots(nrows=len(columns), figsize=(8, 6*len(columns)))

lags = 20  # Number of lags to consider

for i, column in enumerate(columns):
    data = returns_df[column].dropna()  # Remove NaN values
    autocorr_values = acf(data, nlags=lags)
    confidence_interval = 1.96 / np.sqrt(len(data))
    upper_bound = confidence_interval * np.ones_like(autocorr_values)
    lower_bound = -confidence_interval * np.ones_like(autocorr_values)
    ax = axes[i] if len(columns) > 1 else axes
    ax.stem(autocorr_values, linefmt='b-', markerfmt='bo', basefmt='r-',
label='Autocorrelation')
    ax.plot(upper_bound, 'r--', label='Confidence Interval')
    ax.plot(lower_bound, 'r--')
    ax.set_xlabel('Lag')
    ax.set_ylabel('Autocorrelation')
    ax.set_title(f'Autocorrelation for {column} (Lags={lags}) - Log Returns')
    ax.legend()
    ax.grid(True)

plt.tight_layout()

# Show the plots
plt.show()
-----------------------------------------------------------------------------------
import pandas as pd
import numpy as np


# Calculate cross-correlation matrix
```

```
cross_corr_matrix = filled_df.iloc[:, 1:].corr()

# Display the cross-correlation matrix
print(cross_corr_matrix)
```

------------------------------------------------------------------------------------

```
import seaborn as sns
import matplotlib.pyplot as plt

# Plot the cross-correlation matrix as a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(cross_corr_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Cross-correlation Matrix')
plt.show()
```

------------------------------------------------------------------------------------

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Specify the start and end dates for the desired window
start_date = '2022-10-01'
end_date = '2023-03-30'

# Subset the dataframe based on the date window
window_df = filled_df.loc[(filled_df['Date'] >= start_date) & (filled_df['Date'] <=
end_date)]

# Calculate the cross-correlation matrix for the window
cross_corr_matrix = window_df.iloc[:, 1:].corr()

# Keep only the row corresponding to "SVB_Price"
sivb_corr = cross_corr_matrix.loc['SIVB_Price']

# Display the cross-correlation for "SVB_Price"
print(sivb_corr)

plt.figure(figsize=(6, 4))
sns.heatmap(sivb_corr.to_frame(), annot=True, cmap='coolwarm', center=0,
cbar=False)
plt.title('Cross-correlation with SVB_Price')
plt.show()
```

---

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage

# Extract the time series columns from the filled_df
time_series_data = filled_df.iloc[:, 1:].values

# Standardize the time series data
scaler = StandardScaler()
standardized_data = scaler.fit_transform(time_series_data)

# Calculate the linkage matrix using hierarchical clustering
linkage_matrix = linkage(standardized_data.T, method='complete',
metric='euclidean')  # Transpose the data

# Plot the dendrogram
plt.figure(figsize=(12, 6))
dendrogram(linkage_matrix, labels=filled_df.columns[1:], leaf_rotation=90)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Time Series')
plt.ylabel('Distance')
plt.tight_layout()
```

```
plt.show()
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
from statsmodels.tsa.stattools import grangercausalitytests

# Calculate the Granger Causality Matrix
granger_matrix = pd.DataFrame(np.zeros((len(filled_df.columns[1:]),
len(filled_df.columns[1:]))),
                              index=filled_df.columns[1:],
columns=filled_df.columns[1:])

for col in filled_df.columns[1:]:
    for row in filled_df.columns[1:]:
        if col != row:
            granger_result = grangercausalitytests(filled_df[[row, col]], maxlag=5,
verbose=False)
            p_values = [result[0]['ssr_ftest'][1] for result in
granger_result.values()]
            granger_matrix.loc[row, col] = min(p_values)

# Create Network Graph for Granger Causality Matrix
granger_graph = nx.from_pandas_adjacency(granger_matrix)

# Filter out non-causal relationships
causal_edges = [(u, v, granger_graph[u][v]['weight']) for (u, v, d) in
granger_graph.edges(data=True) if d['weight'] < 0.05]

# Create a high DPI plot
plt.figure(figsize=(10, 8), dpi=300)

# Draw the network graph with edge labels
pos = nx.spring_layout(granger_graph, seed=42)
nx.draw_networkx(granger_graph, pos=pos, with_labels=True, node_size=500,
node_color='lightblue',
                 arrowstyle='->', arrowsize=10, edgelist=[(u, v) for (u, v, _) in
causal_edges])
edge_labels = {(u, v): f'{w:.2f}' for (u, v, w) in causal_edges}
nx.draw_networkx_edge_labels(granger_graph, pos=pos, edge_labels=edge_labels,
font_color='red', label_pos=0.3)
plt.title('Granger Causality Network Graph')

# Save the plot to a file
plt.savefig('granger_network_graph.png', dpi=300)
plt.show()
```

```
import networkx as nx

# Calculate average degree
average_degree = sum(dict(granger_graph.degree()).values()) / len(granger_graph)

# Calculate average path length
average_path_length = nx.average_shortest_path_length(granger_graph)

# Calculate number of communities and modularity using Louvain method
communities =
nx.algorithms.community.modularity_max.greedy_modularity_communities(granger_graph)
number_of_communities = len(communities)
modularity = nx.algorithms.community.modularity(granger_graph, communities)

# Print the calculated measures
print("Average Degree:", average_degree)
print("Average Path Length:", average_path_length)
```

```
print("Number of Communities:", number_of_communities)
print("Modularity:", modularity)
```

Code snippet for transformation

```
import networkx as nx

# Create an empty dictionary to store the graphs
graphs = {}

# Determine the threshold for creating edges
threshold = 0.05

# Iterate over the columns and create network graphs
for column in filled_df.columns[1:]:
    # Create an empty graph
    graph = nx.Graph()

    # Extract the column data
    data = filled_df[column]

    # Add nodes to the graph
    for i, value in enumerate(data):
        graph.add_node(i, value=value)

    # Add edges to the graph
    for i in range(len(data)):
        for j in range(i+1, len(data)):
            diff = abs(data[i] - data[j])
            if diff <= threshold:
                graph.add_edge(i, j, weight=diff)

    # Add the graph to the dictionary
    graphs[column] = graph

# Plot the network graphs
for column, graph in graphs.items():
    plt.figure(figsize=(8, 6))
    pos = nx.spring_layout(graph)
    edge_weights = nx.get_edge_attributes(graph, 'weight')
    node_values = nx.get_node_attributes(graph, 'value')
    nx.draw_networkx(graph, pos=pos, with_labels=True, node_size=500, font_size=10,
alpha=0.8, edge_color='gray', width=1.5)
    nx.draw_networkx_edge_labels(graph, pos=pos, edge_labels=edge_weights,
font_size=8)
    nx.draw_networkx_labels(graph, pos=pos, labels=node_values, font_size=8,
verticalalignment='bottom')
    plt.title(f'Complex Network for {column}')
    plt.axis('off')
    plt.show()
```

## Bibliography


1. Taleb, N. N. (2007). The Black Swan: The Impact of the Highly Improbable. Random House

2. Newman, M. E. J. (2010). Networks: An Introduction. Oxford University Press

3. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). Time Series Analysis: Forecasting and Control. Wiley.

4. Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting. Springer

5. Bowley, A. L. (1920). Elements of Statistics. P.S. King & Son Ltd.

6. Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. Reviews of Modern Physics, 74(1), 47–97

7. Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. Econometrica, 37(3), 424-438

8. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., & Nuno, J. C. (2008). From time series to complex networks: The visibility graph. Proceedings of the National Academy of Sciences, 105(13), 4972-4975

9. Charakopoulos, A., Zoumpoulis, S., & Papadimitriou, C. (2018). The Natural Visibility Graph: Basic properties and applications. Chaos: An Interdisciplinary Journal of Nonlinear Science, 28(5), 053113

10. Marwan, N., Romano, M. C., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. Physics Reports, 438(5–6), 237–329

11. Hamilton, J. D. (1994). Time Series Analysis. Princeton University Press.

12. Mantegna, R. N. (1999). Complex networks in financial markets: A comprehensive overview. European Physical Journal B, 11(1), 193-198.

13. Onnela, J. P., Chakraborti, A., Kaski, K., Kertesz, J., & Kanto, A. (2003). Networks of equities in financial markets. The European Physical Journal B-Condensed Matter and Complex Systems, 38(2), 353-362

14. Kenett, D. Y., Preis, T., & Stanley, H. E. (2012). Dynamic correlation networks in stock markets. European Physical Journal B, 85(6), 1-9

15. Financial Times website, ft.com

16. Investing.com website

17. https://networkx.org/ - NetworkX Python library

18. https://www.statsmodels.org/ - Python library on statistics